



# Python Encrypted Keylogger — Setup & Execution Report



## Introduction

This document outlines the setup, execution, and testing of an advanced encrypted Python keylogger with GUI support. All steps were performed in a **Kali Linux environment** using a virtual Python environment (**venv**).

---

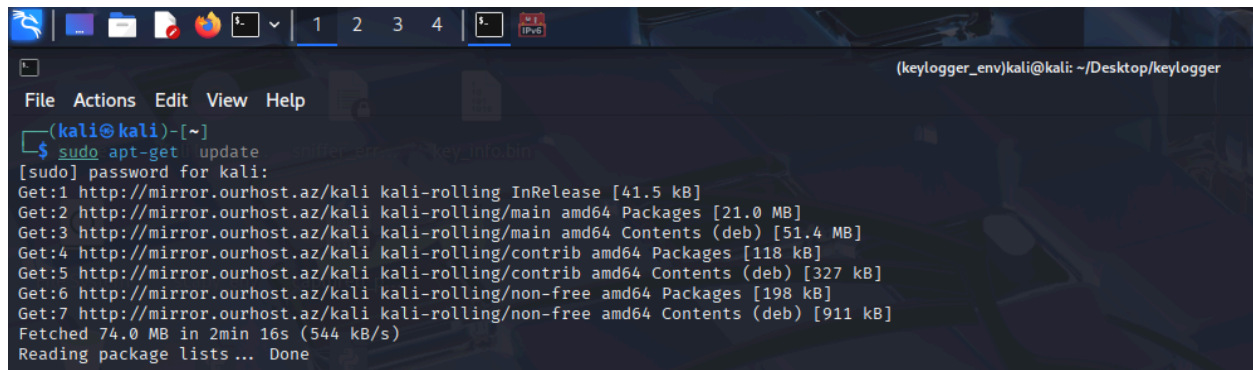


## Environment Setup

### 1. System Update

bash

`sudo apt-get update`



```
(keylogger_env)kali@kali: ~/Desktop/keylogger
File Actions Edit View Help
(kali@kali)-[~]
└─$ sudo apt-get update
[sudo] password for kali:
Get:1 http://mirror.ourhost.az/kali kali-rolling InRelease [41.5 kB]
Get:2 http://mirror.ourhost.az/kali kali-rolling/main amd64 Packages [21.0 MB]
Get:3 http://mirror.ourhost.az/kali kali-rolling/main amd64 Contents (deb) [51.4 MB]
Get:4 http://mirror.ourhost.az/kali kali-rolling/contrib amd64 Packages [118 kB]
Get:5 http://mirror.ourhost.az/kali kali-rolling/contrib amd64 Contents (deb) [327 kB]
Get:6 http://mirror.ourhost.az/kali kali-rolling/non-free amd64 Packages [198 kB]
Get:7 http://mirror.ourhost.az/kali kali-rolling/non-free amd64 Contents (deb) [911 kB]
Fetched 74.0 MB in 2min 16s (544 kB/s)
Reading package lists ... Done
```

### 2. Navigate to Desktop & Create Project Folder

bash

`cd ~/Desktop`

`mkdir keylogger`

`cd keylogger`

```

(kali㉿kali)-[~]
$ cd Desktop
(kali㉿kali)-[~/Desktop]
$ mkdir keylogger
(kali㉿kali)-[~/Desktop]
$ cd keylogger

```

### 3. Create Files

bash

nano encrypted\_keylogger.py

nano decrypt\_logs.py

nano keylog\_viewer\_gui.py

```

(kali㉿kali)-[~/Desktop/keylogger]
$ nano encrypted_keylogger.py
(kali㉿kali)-[~/Desktop/keylogger]
$ nano decrypt_logs.py
(kali㉿kali)-[~/Desktop/keylogger]
$ ls
decrypt_logs.py  encrypted_keylogger.py
(kali㉿kali)-[~/Desktop/keylogger]
$ nano keylog_viewer_gui.py

```

### 4. Create Virtual Environment

bash

python3 -m venv keylogger\_env

source keylogger\_env/bin/activate

```

(kali㉿kali)-[~/Desktop/keylogger]
$ python3 -m venv keylogger_env
source keylogger_env/bin/activate

```

---

## Dependency Installation

Inside the virtual environment:

bash

pip install pynput cryptography

```
(keylogger_env)-(kali@kali)-[~/Desktop/keylogger]
$ pip install pynput cryptography

Collecting pynput
  Using cached pynput-1.8.1-py2.py3-none-any.whl.metadata (32 kB)
Collecting cryptography
  Using cached cryptography-45.0.5-cp311-abi3-manylinux_2_34_x86_64.whl.metadata (5.7 kB)
Collecting six (from pynput)
  Using cached six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Collecting evdev>=1.3 (from pynput)
  Using cached evdev-1.9.2-cp313-cp313-linux_x86_64.whl
Collecting python-xlib>=0.17 (from pynput)
  Using cached python_xlib-0.33-py2.py3-none-any.whl.metadata (6.2 kB)
Collecting cffi>=1.14 (from cryptography)
  Using cached cffi-1.17.1-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting pycparser (from cffi>=1.14->cryptography)
  Using cached pycparser-2.22-py3-none-any.whl.metadata (943 bytes)
Using cached pynput-1.8.1-py2.py3-none-any.whl (91 kB)
Using cached cryptography-45.0.5-cp311-abi3-manylinux_2_34_x86_64.whl (4.5 MB)
Using cached cffi-1.17.1-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (479 kB)
Using cached python_xlib-0.33-py2.py3-none-any.whl (182 kB)
Using cached six-1.17.0-py2.py3-none-any.whl (11 kB)
Using cached pycparser-2.22-py3-none-any.whl (117 kB)
Installing collected packages: six, pycparser, evdev, python-xlib, cffi, pynput, cryptography
Successfully installed cffi-1.17.1 cryptography-45.0.5 evdev-1.9.2 pycparser-2.22 pynput-1.8.1 python-xlib-0.33 six-1.17.0
```

Modules Installed:

- **pynput** – for listening to keyboard inputs.
- **cryptography** – for AES encryption.
- **python-xlib**, **evdev**, **six**, **cffi**, **pycparser** (auto-handled dependencies)

---

## Script Execution & Testing

### Run Encrypted Keylogger

bash

python encrypted\_keylogger.py

```
(keylogger_env)-(kali@kali)-[~/Desktop/keylogger]
$ ls
decrypt_logs.py  encrypted_keylogger.py  keylogger_env  keylog_viewer_gui.py

(keylogger_env)-(kali@kali)-[~/Desktop/keylogger]
$ python encrypted_keylogger.py

Encrypted Keylogger started. Press 'Esc' to stop.
jaghsjdkasjfhla Helloo, H0w are you!Keylogger stopped.
^[
```

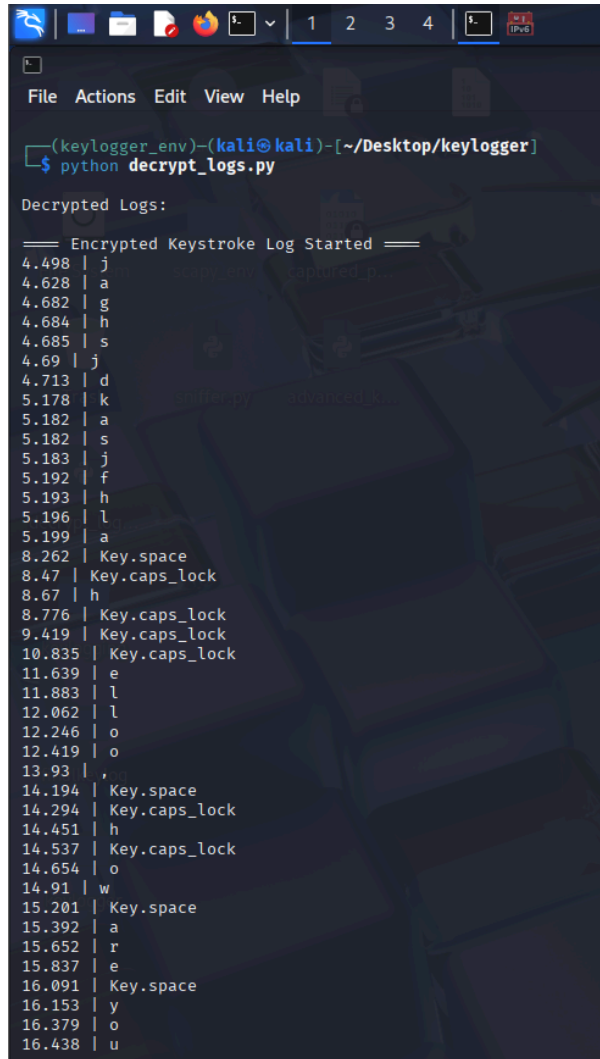
📌 Keystrokes captured in encrypted format and saved as `keystrokes.enc` & key metadata in `key_info.bin`.

---

## ✅ Run Decryptor

bash

`python decrypt_logs.py`



```
(keylogger_env)-(kali@kali)-[~/Desktop/keylogger]
$ python decrypt_logs.py

Decrypted Logs:

=== Encrypted Keystroke Log Started ===
4.498 | j
4.628 | a
4.682 | g
4.684 | h
4.685 | s
4.69 | j
4.713 | d
5.178 | k
5.182 | a
5.182 | s
5.183 | j
5.192 | f
5.193 | h
5.196 | l
5.199 | a
8.262 | Key.space
8.47 | Key.caps_lock
8.67 | h
8.776 | Key.caps_lock
9.419 | Key.caps_lock
10.835 | Key.caps_lock
11.639 | e
11.883 | l
12.062 | l
12.246 | o
12.419 | o
13.93 | ,
14.194 | Key.space
14.294 | Key.caps_lock
14.451 | h
14.537 | Key.caps_lock
14.654 | o
14.91 | w
15.201 | Key.space
15.392 | a
15.652 | r
15.837 | e
16.091 | Key.space
16.153 | y
16.379 | o
16.438 | u
```

```
16.438 | u
17.738 | Key.shift
17.97 | !
20.236 | Key.esc
=== Logging Ended ===

Saved to decrypted_log.log
```

## ✓ Run GUI Viewer

bash

```
python keylog_viewer_gui.py
```

```
(keylogger_env)-(kali㉿kali)-[~/Desktop/keylogger]  
$ python keylog_viewer_gui.py
```

- Opens a GUI window.
- Click on “**Open Log File**” → Select `decrypted_log.log` → View output in clean GUI format.



The screenshot shows a window titled "Keylogger Log Viewer" with a table of keylog data. The table has three columns: "Timestamp", "Action", and "Details". The data is as follows:

| Timestamp | Action        | Details |
|-----------|---------------|---------|
| 4.498     | j             |         |
| 4.628     | a             |         |
| 4.682     | g             |         |
| 4.684     | h             |         |
| 4.685     | s             |         |
| 4.69      | j             |         |
| 4.713     | d             |         |
| 5.178     | k             |         |
| 5.182     | a             |         |
| 5.182     | s             |         |
| 5.183     | j             |         |
| 5.192     | f             |         |
| 5.193     | h             |         |
| 5.196     | l             |         |
| 5.199     | a             |         |
| 8.262     | Key.space     |         |
| 8.47      | Key.caps_lock |         |
| 8.67      | h             |         |
| 8.776     | Key.caps_lock |         |
| 9.419     | Key.caps_lock |         |
| 10.835    | Key.caps_lock |         |
| 11.639    | e             |         |
| 11.883    | l             |         |
| 12.062    | l             |         |
| 12.246    | o             |         |
| 12.419    | o             |         |
| 13.93     | ,             |         |
| 14.194    | Key.space     |         |
| 14.294    | Key.caps_lock |         |
| 14.451    | h             |         |
| 14.537    | Key.caps_lock |         |
| 14.654    | o             |         |
| 14.91     | w             |         |
| 15.201    | Key.space     |         |
| 15.392    | a             |         |
| 15.652    | r             |         |
| 15.837    | e             |         |

At the bottom of the window, there is a button labeled "Open Log File".

| Timestamp | Action        | Details |
|-----------|---------------|---------|
| 5.182     | a             |         |
| 5.182     | s             |         |
| 5.183     | j             |         |
| 5.192     | f             |         |
| 5.193     | h             |         |
| 5.196     | l             |         |
| 5.199     | a             |         |
| 8.262     | Key.space     |         |
| 8.47      | Key.caps_lock |         |
| 8.67      | h             |         |
| 8.776     | Key.caps_lock |         |
| 9.419     | Key.caps_lock |         |
| 10.835    | Key.caps_lock |         |
| 11.639    | e             |         |
| 11.883    | l             |         |
| 12.062    | l             |         |
| 12.246    | o             |         |
| 12.419    | o             |         |
| 13.93     | .             |         |
| 14.194    | Key.space     |         |
| 14.294    | Key.caps_lock |         |
| 14.451    | h             |         |
| 14.537    | Key.caps_lock |         |
| 14.654    | o             |         |
| 14.91     | w             |         |
| 15.201    | Key.space     |         |
| 15.392    | a             |         |
| 15.652    | r             |         |
| 15.837    | e             |         |
| 16.091    | Key.space     |         |
| 16.153    | y             |         |
| 16.379    | o             |         |
| 16.438    | u             |         |
| 17.738    | Key.shift     |         |
| 17.97     | !             |         |
| 20.236    | Key.esc       |         |

## File Structure

keylogger/

|                          |                                    |
|--------------------------|------------------------------------|
| — encrypted_keylogger.py |                                    |
| — decrypt_logs.py        |                                    |
| — keylog_viewer_gui.py   |                                    |
| — keylogger_env/         | # Virtual environment folder       |
| — keystrokes.enc         | # Encrypted log (auto-created)     |
| — key_info.bin           | # Encrypted AES key (auto-created) |
| — decrypted_log.log      | # Output of decryption             |

## Observations

- All scripts work seamlessly in **Kali Linux on X11**.
- Encrypted logs are secure and accurate.
- GUI viewer simplifies post-analysis of keystrokes.
- Virtual environment isolates dependencies and keeps the host clean.

---

## Conclusion

The keylogger project successfully demonstrates:

- Secure keystroke logging using AES encryption.
- Modular script separation for better maintainability.
- GUI-based visualization for improved usability.
- Compatibility with Linux and portable to Windows/macOS with minor changes.