International Journal of Innovative Research in Computer Science and Technology (IJIRCST)
ISSN (Online): 2347-5552, Volume-14, Issue-5, Sep-Oct 2026
DOI: https://doi.org/10.55524/ijircst.2026.14.5.2
Article ID IRP-1786, Pages 9-18
www.ijircst.org

Keyloggers: Modern Malware Implementations, Techniques, and Security Threats

Arijit Das

Department of Computer Applications, Lovely Professional University, Phagwara, Punjab, India

Correspondence should be addressed to Arijit Das; dasarijit604@gmail.com

Received: 13 August 2026; **1st Revision:** 26 August 2026; **2nd Revision:** 1 Sep 2026; **3rd Revision:** 7 Sep 2026; **Accepted:** 14 Sep 2026; **Published Online:** 23 Sep 2026

ABSTRACT- This paper surveys keyloggers and provides a real-time coding demonstration to illustrate the working behind keylogging. The paper traces the evolution of keylogging technology and classifies the tools into four broad categories – hardware, software, remote and electromagnetic – with attack vectors and operational risks. Hardware keyloggers operate in a physical connection or via a wireless link, while software keyloggers operate at the application or OS level typically through phishing campaigns and malicious downloads. Remote and electromagnetic variants push this further still, either exfiltrating stolen data across a network or picking up ambient signals from a distance.

A working Python implementation is included to illustrate how keyboard input is recorded. The paper also looks at the current threat landscape wherein keyloggers are increasingly folded into sophisticated malware suites, and discusses their dual-use nature in both cybercrime and digital forensics. In the end, the study argues for early detection, user awareness and solid defensive practices that are able to help protect individuals and organizations from this kind of covert threat.

KEYWORDS- Credential Theft, Cybersecurity, Forensic Analysis, Malware Keyloggers, Remote Monitoring, Software Security, Trojans.

I. INTRODUCTION

Malware is the key to both offensive and defensive cybersecurity work today. Some threats remain dormant until triggered, like traditional viruses, while others self-propagate through a system, the way worms do. Apart from these common categories, the malware ecosystem also includes trojans, spyware, ransomware and keyloggers. Discussions tend to focus on worms, viruses or headline-grabbing ransomware, rendering keyloggers somewhat overlooked. This neglect is partly misplaced since they quietly harvest sensitive data from a machine without the user ever noticing. A practical Python implementation is provided to show exactly how keyboard input is captured and logged. This study argues that user awareness, stronger detection mechanisms and proactive prevention reduce the risk keyloggers pose to individuals and organizations alike.

II. LITERATURE REVIEW

Several researchers have evaluated and analyzed the mechanisms and procedures used in the construction and detection of keyloggers. These works contain diverse approaches from the perspective of hardware implants and electromagnetic surveillance. Table 1 below shows the main features of the works by the authors cited above. Each article focuses on particular keylogger types and provides significant data on detection and classification mechanisms.

Table 1: Literature review: Author/Year, type of keylogger, method, and major finding

Author	Keylogger Type	Method	Major Finding	Limitation
Creutzburg [2]	Hardware/Software (overview)	Literature survey of keylogger types	Classified keyloggers by mechanism and provided detection taxonomy	Survey-based; no experimental validation
Vuagnoux & Pasini [9]	Electromagnetic	EM side-channel signal analysis	Showed keystroke recovery from EM emanations at a distance	Requires specialized equipment; limited range
Singh et al. [11]	Software	Detection & prevention framework	Proposed layered detection that combines traffic analysis and memory forensics	The focus is on Windows – and the generalizability to other systems was not evaluated
Dadkhah [13]	Software	Heuristic approach	Proposed a method of keylogger detection that used heuristic rules	Not updated to modern malware – the solution has not been validated
Moonlock	Mixed (family of	Threat intelligence	Conducted an analysis of the	This is a blog post – the quality of

Security [12]	malware)	analysis	distribution of the family of malware around the globe to identify the countries most affected – the United States appeared to be the most affected country	evidence may not be high
Reddy et al. [5]	Hardware (USB Rubber Ducky)	Hardware payload analysis	Shows how USB devices may be used to compromise a computer and perform various tasks, including keylogging	The findings are limited to USB devices – wireless devices were not considered
Kumbhani & Maniar [4]	Hardware (USB attacks)	Case study review of hardware attacks	Demonstrates the risks posed by USB devices	The analysis did not consider software
Tahir [7]	Software	Malware detection review	Reviewed the methods used to detect malware	keyloggers were mentioned as a type of malware – the focus of the article was not keyloggers – the findings were limited to malware in general
Muñoz Martín et al.[1]	Software	Cognitive translation study	Explored keylogging in the context of cognitive translation research	This is not a security-focused article – keylogging is not discussed in depth
Airhopper Project [10]	Electromagnetic	Van Eck Phreaking demonstration	how keystrokes may be captured and processed	a proof-of-concept – not widely applicable
Pansare et al. [6]	Software	Keylogger implementation and monitoring	Presented a keylogger-oriented monitoring approach for computer activity	Limited scope; requires stronger modern validation

III. KEYLOGGER CLASSIFICATION AND CHARACTERISTICS

A keylogger is a type of malware that threat actors deploy to record keystrokes and extract the sensitive information such as usernames and passwords. In addition to keystrokes, more advanced keyloggers also log metadata – exact timestamps, dates, and how often certain keys repeat. Interestingly, keylogging technology is not purely malicious, with organizations also using it legitimately for employee monitoring, tracking productivity and daily work activity. Keyloggers have been around since the 1970s and 1980s, initially built for system maintenance and debugging. Several have since been repurposed as malware, quietly recording and transmitting information to hackers and other malicious actors, such as attackers Keyloggers can also analyze Inter-Keystroke Intervals (IKIs), measuring typing metrics such as delays, pauses, speed, and rhythm alongside standard timestamps. The rest of this paper walks through keylogging mechanisms, how they work, and how to defend against them. The following categories are discussed according to their implementation, operational characteristics and attack vectors: Keyloggers fall under different categories based on their design, functionality, installation and attack vectors. These are discussed in detail below:

A. Hardware Based Keyloggers

Hardware based keyloggers do not require any sorts of software installations as they work at the hardware level of a computer system [2]. They can capture data from direct physical connection or from indirect wireless methods, usually through the use of tools such as USB devices, PS/2 connectors or wireless sniffers.

i) Implementation mechanism: A hardware keylogger is essentially surveillance hardware that records every keystroke to a log file, typically encrypted [3]. A few common implementations showcase how this works:

- **PS/2 connector:** The PS/2 connector is an older six-pin interface for keyboards and mice, and allows a hardware keylogger to intercept raw keystrokes before reaching the operating system. Figure 1 shows the traditional PS/2 plug used for hardware-based keylogger insertion.

Figure 1: Legacy PS/2 Keyboard Connector

- **USB dongles:** USB is the most common interface for connecting peripherals to a computer [4]. The USB Rubber Ducky looks like an ordinary flash drive but functions as a keyboard, typing commands automatically when plugged in [5]. Figure 2 Illustrates exposed USB circuit board and consumer-ready USB flash drive form factor

Figure 2: USB Hardware Keylogger Devices

- **Wireless Sniffers:** These are also known as protocol analyzers and are used to capture and analyze wireless data packets that are transmitted between two or more devices. They've become increasingly important since research shows that wireless communication issues – primarily interference and misconfigured protocols – are responsible for approximately 60% of IoT device failures. These failures are not merely inconvenient, but can compromise home automation systems and security networks altogether. By capturing packets in real time, sniffers allow developers to identify protocol-level issues, find timing problems, detect interference, and confirm that wireless standards are properly implemented [6]. On a wireless keyboard, every keystroke is transmitted as a radio-frequency (RF) signal. Figure 3 depicts a compact wireless transceiver module used for remote/electromagnetic keylogging.

Figure 3: USB RF Transceiver with External Antenna

- **Microcontroller Implants:** This technique involves using a microcontroller to tap into the keyboard data streams. This method requires the attacker to decipher the codes received from the targeted computer and use them to extract the desired information. Figure 4 shows a board with IC, oscillator, and antenna trace – used as a microcontroller receiver for EM keylogger signals.)

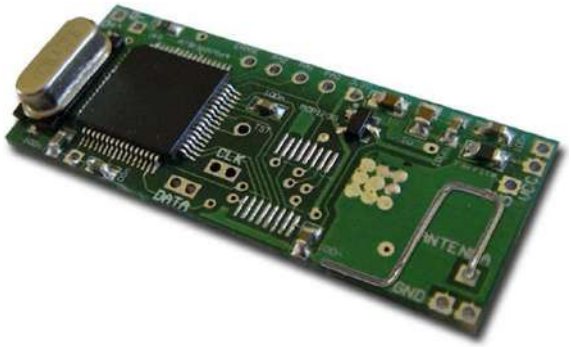

Figure 4: Microcontroller-Based Receiver Module

ii) Deployment Method

- **Inline Keyloggers (PS/2 and USB):** This type of keylogger is connected inline between the keyboard and computer. Being wired, they are relatively cheap: and easy to deploy.
- **Peripheral Replacement:** Here, threat actors swap the target's legitimate keyboard or input device for one with built-in logging circuitry. As the victim types normally, the substituted hardware silently records every keystroke without alerting.
- **Compromised USB hubs and Docking Station:** This captures every keystroke passing through a compromised hub or docking station. It is notoriously hard to detect since the tampered hardware looks identical to the real one.
- **Wireless Sniffers:** These pick up the radio-frequency signals broadcast by wireless keyboards. The same approach is also successful against the optical transmissions used by laser-projected virtual keyboards.

iii) Attack Vector

- **Physical Access:** The attacker takes advantage of an unguarded computer to install the logging hardware manually.
- **Peripheral Compromise:** The adversaries take advantage of an opportunity to replace a legitimate USB hub or docking station with a modified one, which contains a keylogger.
- **Insider Access:** The insider is given access to the target organization to install the keylogger hardware illegally.

B. Software Based Keyloggers

Unlike hardware versions, software keyloggers use malicious computer codes to save keystrokes and their timestamps. Running at either the application or kernel level, this type of spyware is intended to record user input and steal passwords, credit card numbers and other sensitive data. It typically reaches a machine when malicious software is installed or when a user visits a compromised site [7]. Before diving into keylogger implementations, it's helpful to understand APIs:

i) **API Definition:** APIs or Application Programming Interfaces are sets of programming tools that enable different software applications to exchange data with

one another. APIs provide the programmer with a convenient way to communicate with the target software without understanding the complex internal mechanisms of the underlying application. In other words, an API allows a programmer to make a request and receive the necessary response without having the needed expertise in the particular software. The APIs are available in many forms including web APIs that facilitate internet communication and library APIs that are used to add functions to a software library. There are several types of APIs, such as RESTful, based on HTTP, and SOAP, which is built on XML. The choice between the two depends on the requirements of the project [8].

ii) Implementation Mechanism

- **API Hooking:** Keyloggers use API hooking to manipulate the keyboard drivers in an operating system.
- **Form Grabbing:** This is a method of capturing data before it is encrypted.
- **Stealth Execution:** Keyloggers disguise as normal computer processes to avoid being detected. However, it can be identified by checking the list of processes in the Task Manager of the OS.

iii) Deployment Method

- **Malware:** This is installed when the user's system runs a link or an attachment that contains malware.
- **Trojan:** Key loggers can be disguised as trojan horse software. Thus, when the user installs the software, the keylogger is deployed alongside the intended program.
- **Malicious Browser Add-ons:** These are mainly acquired from third-party software developers. These add-ons run on the browser and record all the keystrokes made by the user on the computer.
- **Weaponized legitimate software:** This is achieved when legitimate software is compromised and weaponized to track passwords.

Real-World Example: Insider Threat

A New York hospital employee illegally accessed his colleagues' computers using a keylogger to capture their passwords. This resulted in him facing a 30-month jail term after being convicted of computer fraud.

iv) Attack Vector

- **Phishing:** This involves tricking the user into clicking a link or visiting a website that contains malicious programs that install keyloggers on the computer.
- **Trojaned Applications:** This is when the user downloads applications from untrusted sources. These applications contain malware that scans and captures passwords from the infected system.
- **Malicious Scripts:** This is common in Bash and Power-shell scripts. The scripts give the attacker administrative access to a system hence installing keyloggers to capture the keystrokes of the user.

C. Remote Keyloggers

Remote keyloggers are deployed using phishing and are mostly software keyloggers. However, they differ from the ones discussed above because remote keyloggers capture the keystrokes and transmit them to a remote server where the attacker accesses the information. The presence of remote keyloggers compromises the security of the infected system as the attacker can access sensitive information such as passwords hence full access to the computer.

i) The rationale for dividing remote keyloggers from software keyloggers

The remote keyloggers are considered to be a different category from the other software keyloggers because they operate differently from the other keyloggers. For instance, the other software keyloggers save the captured information and delete them when the attacker accesses them. On the other hand, remote keyloggers capture the information and transmit it to a remote server. Thus, the constant connection of remote keyloggers to a remote server is a primary concern as it can be easily detected by the SOC analyst.

- **Deployment method and Attack vector:** Software-based and remote keyloggers have many common techniques and attack vectors. However, there are some differences, such as installation, storage, and data exfiltration. The remote keylogger exfiltrates data on its own, so it can be more easily discovered during a network scan, as shown in Table 2.

Table 2: Comparison of Installation, Implementation, Detection, and Handling

Aspect	Software Keylogger (Local)	Remote Keylogger
Installation Process	Should be installed locally on the target system, where it can be deployed as a standalone module or attached to other software via Trojans.	Can be deployed using Remote Access Trojan (RAT), which utilizes network modules.
Data Storage and Transmission	The data is saved in a hidden folder on the computer where the keylogger was installed and manually retrieved by the intruder.	The data is sent to a remote server via the internet in real-time.
Implementation Complexity	Requires direct attacker intervention to manually access or extract the stored log files.	Operates autonomously; no direct interaction is required to harvest the data once deployed.
Handling & Exfiltration	Managed locally on the host, data remains stagnant until a second stage of exfiltration takes place.	Managed by remote command protocols; Email, FTP, or HTTP uploads to exfiltrate data

D. Electromagnetic Keyloggers

This technique makes use of the electromagnetic (EM) side-channel analysis to passively pick up a victim's keystrokes. In order to extract confidential data, it manipulates the operating frequency of a processor, and uses the resulting EM emission pattern from the power supply which is influenced by the keyboard input. The signals can be picked up from several metres away using a standard RF receiver, or a smartphone with a basic antenna. This section also reviews related literature that examines the current keylogging techniques and introduces a multi key modulation technique and evaluates its ability to withstand different typing speeds and on-screen keyboard layout. Since keyboards are made up of electronic components, they are naturally inclined to emit electromagnetic waves when operating, which can leak out sensitive information such as keystrokes. Detecting such compromising emanation usually requires a wide-band receiver which is tuned to a specific frequency [9].

i) Deployment method

- **Distance Hardware Placement:** Small, low-power RF receivers are placed close to target hardware; keyboards, CPUs, monitors, to pick up compromising electromagnetic emissions.
- **Portable Device Utilization:** Portable hardware such as laptops, smartphones and single-board computers can be used to capture EM signals, often analyzed in real-time through terminal environments like Termux. This type of interception hardware is also starting to appear in everyday looking objects; smart pens, smart glasses and ID badges.
- **Power line coupling:** This is a covert way for attackers to inject adversarial EM signals into to local power lines. This technique's main advantage is that it completely bypasses an organization's data network, which makes it effective against even air-gapped systems. In below figure 5 explaining how keystrokes can be covertly transmitted through electrical wiring.

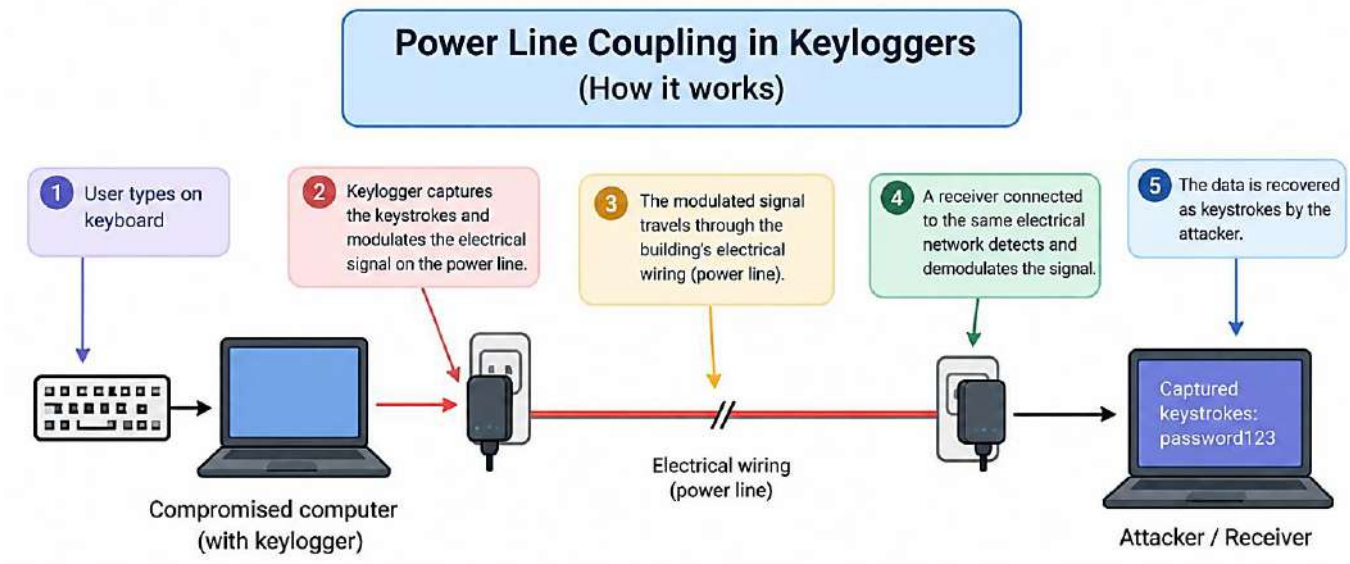

Figure 5: Power Line Coupling in Keyloggers

- **Drone Based Deployment:** Drones are increasingly used to capture EM emissions from mechanical or capacitive keystrokes, enabling mobile, short-term surveillance without the need for physical entry into the target facility.
- **Satellite or Relay Integration:** This approach found in nation-state espionage and advanced persistent threat campaigns, which involves the use of long-range satellite or terrestrial relay systems to capture EM emanations from great distances. This allows complete removal of the requirement for any physical proximity.

ii) Attack Vector

- **Electromagnetic Emanation Capture:** This attack captures the unintentional EM radiation given by ordinary hardware – keyboards, mice, CPUs – and analyzes the variations in that spectrum to deduce the user input or data being processed on the device.
- **Radio Frequency (RF) Interception:** This approach targets the wireless peripherals by tuning into the frequencies that they use to transmit data and decoding the intercepted signal to extract the keystrokes in real-time.
- **Hybrid Side-Channel Attacks:** Here the attacker places hidden EM sniffers throughout smart building systems or local power grids. By combining multiple interception methods, they are able to execute a large-scale monitoring of multiple targets simultaneously.

Real-World Example

In January 2015, the Airhopper project from Georgia Institute of Technology, United States demonstrated (at Ben Gurion University, Israel) the use of Van Eck Phreaking to enable a keylogger to communicate, via video signal manipulation, keys pressed on the keyboard of a standard PC, to a program running on an Android cellphone with an earbud radio antenna. Such attacks could be used to exfiltrate data from air-gapped systems [10].

IV. CURRENT TRENDS IN KEYLOGGING THREATS AND CAPTURE MECHANISMS

The threat landscape for keyloggers has evolved notably, encompassing a wide range of hardware and environmental vectors beyond traditional software payloads. Modern keylogging methods capture user input in two main ways: direct injection and indirect interception.

Direct mechanisms require access to the target system – for instance, when an attacker physically attaches a malicious hardware adapter to a peripheral port, or when a victim unknowingly runs malware by opening a trojanized attachment.

Software keyloggers, unlike hardware counterparts, can be introduced onto a victim's system relatively easily, which is a part of why they are so widespread. They do not damage hardware, but pose a serious threat to both businesses and individuals [11]. The example below shows this.

Overview of the environment:

A. Testing Environment

- OS: Kali Linux 2026.3 (Kali Rolling)
- Python: 3.14.6
- Kernel: Linux 7.0.12+kali-amd64
- Processor: 12th Gen Intel® Core™ i3-1215U
- Virtual CPUs: 4
- Platform: VMware
- Environment: Kali Linux 2026.3 on VMware

B. Test Results

The experimental setup had a 100% accuracy of key capture, out of the 27 keys pressed none of them were captured wrongly as shown in the table 3 below, which proves that our system can capture keystrokes in a controlled environment.

Table 3: Test results

Parameter	Result
Test Sessions	3
Expected Keystrokes	27
Correctly Detected	27
Missed Events	0
False Positives	0
Keystroke Capture Accuracy	**100%**

Test Sessions

1. Random keystrokes: k, m, o, d, m, s, o, w, m, d, j, e, l, s
2. kali, followed by Shift and Shift + S

Detection Accuracy

Accuracy = (Correctly Detected / Expected Keystrokes) × 100 = (27/27) × 100 = 100%

Implementation of the Code

```
import keyboard
import time
from datetime import datetime

file_path=r'/home/kali/Desktop/output.txt'
buffer=[]

def key(event):
    date=datetime.now().strftime("%Y-%m-%d %H-%M-%S")
    buffer.append(f"\n[{date}] the target entered \n{event.name}")
    print(f"\n[{date}] the target entered \n{event.name}")

def flush():
    if not buffer:
        return
    with open(file_path,'a', encoding='utf-8') as f:
        f.write("\n"+"\n".join(buffer)+'\n')
    buffer.clear()

keyboard.on_press(key)
keyboard.add_hotkey('Shift', flush)
keyboard.add_hotkey('Shift+s', exit)

while True:
    time.sleep(30)
    flush()
```

The program was run under the experimental conditions, within a native Linux environment. The keyboard input was simulated by the program, to imitate the normal user's behavior. Upon pressing a key, the key was displayed in the terminal for the confirmation purposes, and further stored in the memory buffer.

In order to reduce the memory overhead, the program would flush the memory buffer into a local text file every 30 seconds.

In figure 6, it illustrates execution of a Python script within the Kali Linux environment, demonstrating the logging of keystrokes with timestamps in real-time

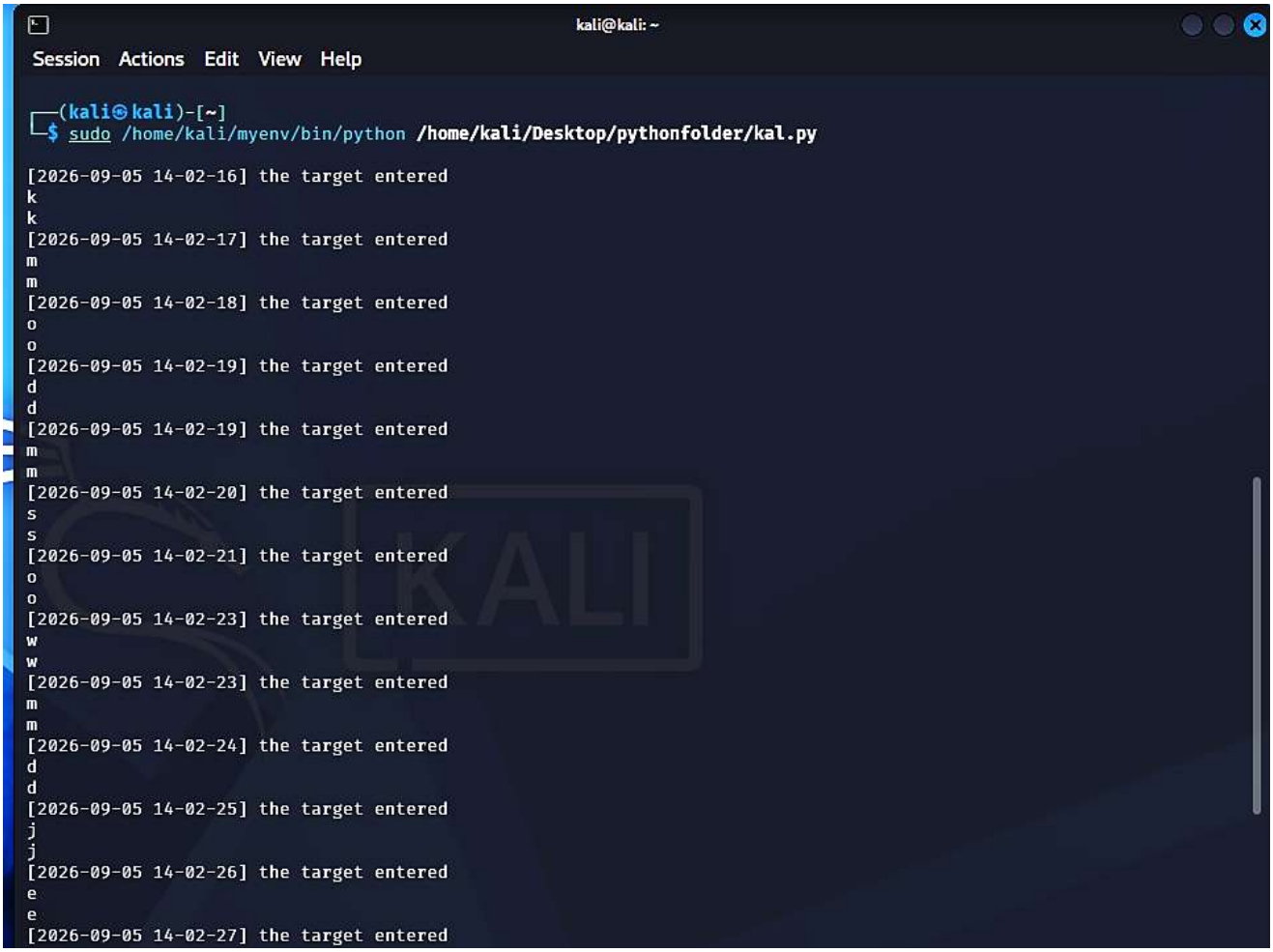

Figure 6: Keystroke Logging Output in Kali Linux Terminal

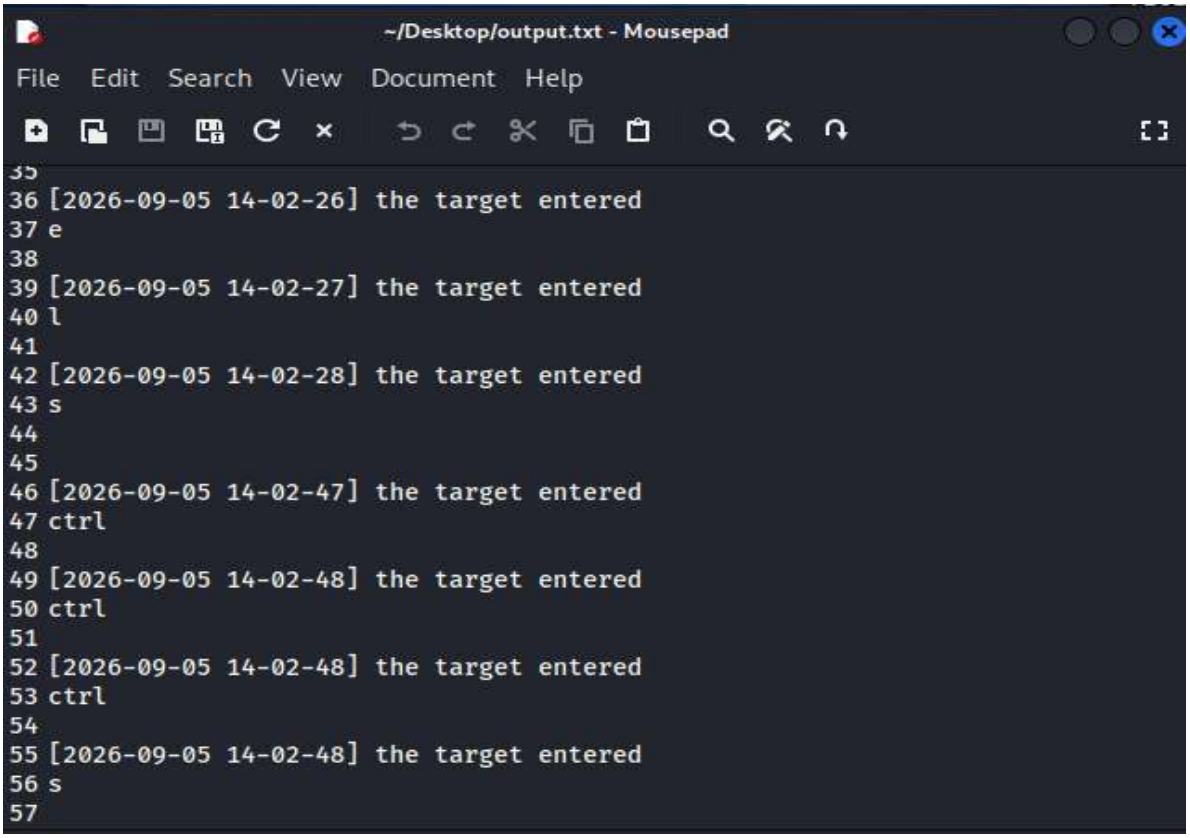

Figure 7: Output Log File in Text Editor (Mousepad)

In figure 7, it showcases the recorded keystrokes with timestamps saved in the output.txt file, indicating the manner in which they are represented for analysis.
Essentially, the recorded keystrokes would originate directly from the operating system's input layer preceding application-level processing
Preventive strategies are dependent on the category of keylogger as stated in Table 4 below. Thus, using tamper-proof devices can help prevent hardware-based keyloggers, whereas the preventive measure for software-based keyloggers is having updated systems and avoiding risky browsing behaviors. On the other hand, remote loggers rely heavily on drivers to exfiltrate data, making driver updates and phishing prevention techniques the best option to avoid them. Electromagnetic eavesdropping requires shielding sensitive hardware, which is also a preventive measure stated in Table 4.

Table 4: Keylogger, Detection and Prevention methods

Keylogger Category	General method of capture	Prevention strategies
Hardware-based	Microcontroller implants, USB hubs, or inline keyboard microchips	Prevent unauthorized physical devices from being connected whenever possible and also use tamper-proof hardware
Software-based	Downloaded and installed via malicious software, trojans, or malware	Avoid downloading or clicking on suspicious applications and also keep the system updated
Remote	Uses API hooking, polling, or Raw Input APIs to ex-filtrate data remotely.	Uses driver updates, phishing-email detection and user awareness.
Electromagnetic	Captures signals through Radio Frequency (RF) sniffing or electromagnetic spectrum analysis.	Shielded hardware and avoid unwanted access of the sensitive hardware

V. DETECTION TECHNIQUES

A. Signature-Based Detection

Anti-virus software and other endpoint security solutions scan systems in search of known code signatures, hashes, or behavior patterns typical of thousands of known keyloggers. It is a very effective method for known software but less efficient against zero-day and obfuscated attacks.

C. Traffic Analysis

This technique monitors network traffic for signs of exfiltration attempts by analyzing the software's traffic for uncommon patterns or behaviors. For instance, it would detect any unusual HTTP, HTTPS, or FTP activity on the machine, making it an effective way to spot a remote keylogger.

C. Static Analysis and YARA Rule Matching

This method scans the file or binary code of a given program against a database of known codes using specific YARA rules. It is an efficient way of detecting known backdoors and malware since it requires no execution of the code being examined. Thus, it can be used to scan entire directories for dormant malicious programs.

D. Memory Forensics

This technique examines the memory of a given system or target process to detect signs of tampering, such as unusual process injection, or unauthorized API hooks. It is a valuable technique because it can reveal the presence of a fileless keylogger at the moment of execution.
Figure 8 reports the percentage of keyloggers detected in each country; the United States leads at 21.2%, followed by France at 15.1%, Ukraine at 6.6%, and the United Kingdom at 6.4%.

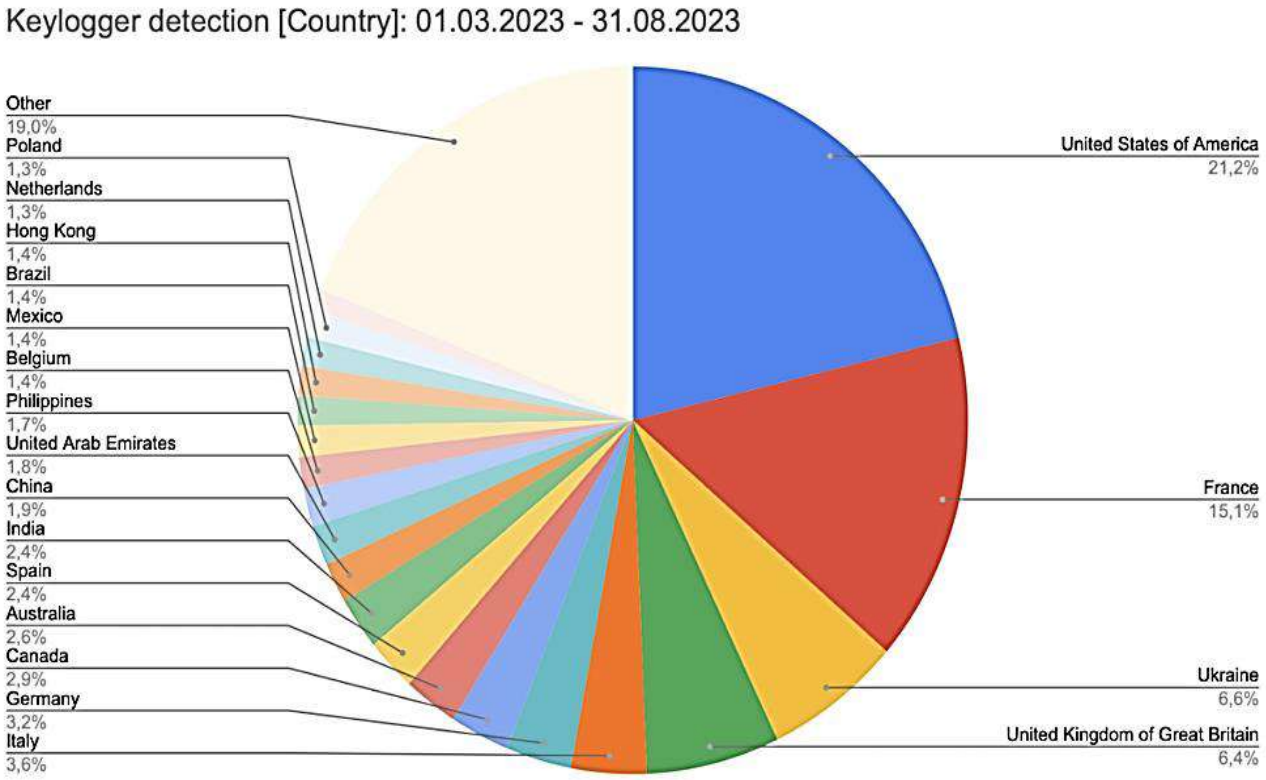

Figure 8: Distribution of Keyloggers around the World: March – August [12]

VI. CONCLUSION

The spread of keyloggers remains one of the most dangerous aspects of cybersecurity as these devices and programs continue to evolve and improve, bypassing even the most sophisticated forms of protection and gaining access to valuable data. This article discussed keyloggers and their variants, including hardware implants, software payloads, remote exfiltration, and electromagnetic surveillance, presenting each of their unique challenges and dangers to businesses and individuals.

Keyloggers are malicious programs that track keyboard inputs of the victim and are often used to gather and steal valuable data [13]. As shown in the video, this type of attack is rather simple to carry out, especially in a physical environment where the target's computer is available, and can lead to high success rates due to the ease of key extraction with precise results. However, more often than not, attackers use combinations of the described methods to get access to data, which is why the use of multiple protective measures, such as host monitoring, traffic inspection, and memory scanning, is needed.

Modern keyloggers are rather dangerous as they continue to evolve and improve existing technology. As such, it is vital to employ updated protective measures within an organization and provide employees with knowledge about the latest cyber threats. At the same time, keyloggers are dual-use devices or programs, which means that they are often used not only by criminals but by law enforcement and government agencies. As such, it is necessary to analyze both sides of the technology to ensure that the most effective and reliable security measures are taken.

ACKNOWLEDGMENT

I would like to extend my sincere thanks and appreciation to the faculty members of the department of Computer Applications, Lovely Professional University for their kind support and motivation in preparing this study. Their suggestions and academic contributions helped the author explore the subject deeper and enhance the quality of this work.

The author would also like to extend his appreciation to the researchers, authors, and cybersecurity professionals who shared their research data and knowledge in the form of published work for this paper. Their work regarding keyloggers, malware, detection, system security, and prevention provided an in-depth knowledge of keyloggers and related malware's different aspects.

The author would like to thank the peers and reviewers that participated by providing suggestions and insights while preparing and concluding this manuscript. The suggestions contributed additional information on the subject and assisted in finding areas of research and explanations that need further attention.

The author would also like to acknowledge the broader cybersecurity research community for their time and effort in researching keyloggers, other malware, and related prevention measures. The research papers, technical documents, and other materials helped the author investigate the subjects from different angles and conclude with the findings presented in this paper. Finally, the author would like to express his gratitude to all those directly and indirectly involved in the preparation of this study for their support, guidance, and contributions toward this work's completion. Their effort and knowledge have helped the author realize the significance of the study and conclude successfully.

REFERENCES

[1] R. Muñoz Martín, S. Sun, Z. Du, and S. Puerini, "Keylogging," in *Research Methods in Cognitive Translation and Interpreting Studies*, A. M. Rojo López and R. Muñoz Martín, Eds. Amsterdam: John Benjamins, 2025, pp. 157–182. Available from: https://doi.org/10.1075/rmal.10.07mar.

[2] R. Creutzburg, "The strange world of keyloggers – an overview, Part I," *Electronic Imaging*, vol. 2017, no. 6, pp. 139–148, 2017. Available from: https://doi.org/10.2352/ISSN.2470-1173.2017.6.MOBMU-313.

[3] *Techno Security's Guide to Securing SCADA: A Comprehensive Handbook on Protecting Critical Infrastructure*, ch. 5, "Working with Law Enforcement on SCADA Incidents." Burlington, MA: Syngress, 2016, pp. 85–102.

[4] K. Kumbhani and H. Maniar, "Understanding USB-Based Hardware Attack Vectors: Security Risks and Defensive Mechanisms Against USB Rubber Ducky Attacks," *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, vol. 15, no. 4, pp. 5419–5423, Apr. 2026. Available from: https://doi.org/10.15680/IJIRSET.2026.1504043.

[5] R. V. Reddy, R. Suhasini, V. Rohith, B. Arun Kumar, and T. Vutkoor, "Unveiling the Power of USB Rubber Ducky: An Analysis of Its Hardware Capabilities," *International Journal of Engineering Research & Technology (IJERT)*, vol. 13, no. 3, Mar. 2024, paper ID IJERTV13IS030163.

[6] S. Pansare, M. Adate, P. Bhalerao, P. Vadhwani, and A. Wagaskar, "Keylogger: An Advanced Method for Computer Monitoring," *International Journal for Multidisciplinary Research (IJFMR)*, vol. 6, no. 6, pp. 112–120, Nov.–Dec. 2024. Available from: https://doi.org/10.36948/ijfmr.2024.v06i06.51938.

[7] R. Tahir, "A Study on Malware and Malware Detection Techniques," *International Journal of Education and Management Engineering (IJEME)*, vol. 8, no. 2, pp. 20–30, Mar. 2018. Available from: https://doi.org/10.5815/ijeme.2018.02.03.

[8] Patel and J. Singh, "The Role of APIs in Modern Software Development," *World Journal of Advanced Engineering Technology and Sciences*, vol. 13, no. 1, pp. 12–20, 2024. Available from: https://doi.org/10.30574/wjaets.2024.13.1.0515.

[9] M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards," in *Proc. 18th USENIX Security Symposium*, Montreal, Canada, Aug. 2009, pp. 1–18.

[10] Airhopper Project, "Van Eck Phreaking Demonstration," Georgia Institute of Technology & Ben Gurion University, 2015. Available from: https://en.wikipedia.org/wiki/Van_Eck_phreaking.

[11] Singh, P. Choudhary, A. K. Singh, and D. K. Tyagi, "Keylogger Detection and Prevention," *Journal of Physics: Conference Series*, vol. 2007, no. 1, Art. no. 012005, 2021. Available from: https://doi.org/10.1088/1742-6596/2007/1/012005.

[12] Moonlock Security, "An in-depth look at the keylogger malware family," *Moonlock Threat Intelligence Blog*, Sep. 2023. Available from: https://moonlock.com/keylogger-malware-family.

[13] M. Dadkhah, "A Novel Approach to Deal with Keyloggers," *Oriental Journal of Computer Science & Technology*, vol. 7, no. 1, pp. 25–28, Apr. 2014.