VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi-590018, Karnataka

Internship report

ON

“KEYLOGGER”

BACHELOR OF ENGINEERING IN
CSE-AIML and ISE

Submitted by

NAME : H R BINDU MAHALAKSHMI
D N CHANDAN GOWDA
USN : 1AM21CI015
1AM21IS029

Conducted at **INCERD**

AMC ENGINEERING COLLEGE

Department of branch CSE-AIML and ISE

Accredited by NAAC&NBA, New Delhi

AMC CAMPUS, BANNERGHATTA MAIN ROAD, BENGLURU, KARNATAKA
560083

AMC ENGINEERING COLLEGE

Department of branch CSE-AIML

Accredited by NAAC&NBA, New Delhi

AMC CAMPUS, BANNERGHATTA MAIN ROAD, BENGLURU
KARNATAKA-560083

CERTIFICATE

This is to certify that the Internship titled **“**CYBER SECURITY**”** carried out by **Ms H R BINDU MAHALAKSHMI and Mr D N CHANDAN GOWDA,** a bonafide students of AMC Engineering College, in partial fulfillment for the award of **Bachelor of Engineering**, in **CSE AIML and ISE** under Visvesvaraya Technological University, Belagavi, during the year 2022-2023. It is certified that all corrections/suggestions indicated have been incorporated in the report.

The project report has been approved as it satisfies the academic requirements in respect of Internship prescribed for the course Internship / Professional Practice.

Signature of Guide **Signature of HOD** **Signature of Principal**

____

____

DECLARATION

We, **H R Bindu Mahalakshmi and D N Chandan Gowda**, third year student of CSE AIML, AMC Engineering College - 560 083, declare that the Internship has been successfully completed, in **INCERD**. This report is submitted in partial fulfillment of the requirements for award of Bachelor Degree in CSE AIML and ISE , during the academic year 2022-2023.

Date : 10/12/2023

Place : Bangalore

USN

:1AM21CI015

:1AM21IS029

NAME : H R BINDU MAHALAKSHMI

D N CHANDAN GOWDA

ACKNOWLEDGEMENT

This Internship is a result of accumulated guidance, direction and support of several important persons. We take this opportunity to express our gratitude to all who have helped us to complete the Internship.

We would like to thank INCERD, for providing us an opportunity to carry out Internship and for their valuable guidance and support.

We express our deep and profound gratitude to our guide, Tarun Balaji K S, for his keen interest and encouragement at every step in completing the Internship.

We would like to thank all the coordinators for the support extended during the course of Internship.

Last but not the least, we would like to thank our parents and friends without whose constant help, the completion of Internship would have not been possible.

ABSTRACT

In today's digitally interconnected world, the significance of cybersecurity cannot be overstated. As cyber threats continue to evolve in complexity and scale, there is a growing need for skilled professionals to safeguard digital ecosystems. This abstract delves into the experiential learning gained through a cybersecurity internship, examining the bridge between theoretical knowledge and practical application in the realm of cybersecurity.

The internship provided an immersive experience within a dynamic cybersecurity environment, offering a hands-on opportunity to apply theoretical concepts learned in academic settings. The internship covered a broad spectrum of cybersecurity domains, including but not limited to network security, penetration testing, incident response, and security policy enforcement.

Throughout the internship, emphasis was placed on the application of theoretical knowledge to real-world scenarios, enabling the development of practical skills such as vulnerability assessment, threat detection, and mitigation strategies. The intern had the opportunity to work alongside seasoned professionals, gaining insights into industry best practices and the latest advancements in cybersecurity technology.

The internship also facilitated exposure to diverse cybersecurity tools and platforms, allowing the intern to navigate through simulated cyber incidents, assess system vulnerabilities, and implement proactive security measures. The experience contributed to a deeper understanding of the challenges faced by cybersecurity professionals and the critical role they play in safeguarding sensitive information.

In essence, this abstract provides a panoramic view of cybersecurity, encapsulating its foundational principles, contemporary challenges, and the dynamic strategies employed to protect digital assets and privacy in an interconnected world.

Table of Contents

CHAPTER 1

INTRODUCTION

1.INTRODUCTION

Socket programming is a foundational concept in computer networking, providing a mechanism for communication between software applications across a network. In the context of cybersecurity, the utilization of sockets becomes particularly noteworthy when developing backdoors—malicious tools designed to clandestinely establish unauthorized access to a target system.

At its core, a socket serves as an endpoint for sending or receiving data across a computer network. In the realm of backdoor development, a backdoor typically opens a network socket that facilitates remote communication between the attacker and the compromised system. This communication channel becomes the conduit for executing commands, transferring files, and maintaining persistent control over the infiltrated system.

Socket programming in backdoors involves the implementation of both server and client components. The server side resides on the compromised system, waiting for incoming connections, while the client side operates on the attacker's system, initiating connections to the compromised hosts. Through these connected sockets, data flows bidirectionally, enabling the transmission of instructions and exfiltration of sensitive information.

Understanding socket programming is essential for developers, network administrators, and cybersecurity professionals. It forms the backbone of numerous networking protocols and lays the groundwork for building scalable, efficient, and responsive networked applications.

Two primary protocols commonly used in socket programming are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP provides a reliable, connection-oriented communication channel, ensuring that data is delivered accurately and in the correct order. UDP, on the other hand, offers a connectionless, lightweight alternative suitable for scenarios where speed is prioritized over reliability.

Socket programming emerges as a powerful paradigm for enabling this communication, allowing processes on different devices to exchange data seamlessly over a network. This introduction provides an overview of socket programming, delving into its basic principles and applications.

CHAPTER 2
SOFTWARE REQUIREMENTS AND SPECIFICATIONS

2.SOFTWARE REQUIREMENTS AND SPECIFICATIONS

SOFTWARE REQUIREMENTS:

1. Programming Language: Choose a suitable programming language for backdoor development. Common choices include Python, C, C++, or even scripting languages like PowerShell.

2. Operating System Compatibility: Specify the target operating systems for the backdoor. Ensure compatibility with Windows, Linux, or macOS, depending on the intended deployment environment.

3. Network Library or Module: Select a network library or module for socket programming. For Python, the pynput module is commonly used. In other languages, choose libraries that facilitate socket communication.

CHAPTER 3

PROJECT PLAN

3.PROJECT PLAN

A keylogger is a tool collecting and recording all keystrokes from the target device. Some keyloggers record the data in a hidden mode and transfer it to the spy via an online account. There are also keyloggers that require physical access to the target computer but they are not widely used.

Keyloggers are one of the most common tools in a Hacker’s toolbox. They are infact one of the most basic tools and are quiet easy to make.

Keyloggers aren’t always used for illegal purposes. Consider the following examples of legal uses for keylogging software:

- Parents might use a keylogger to monitor a child’s screen time.
- Companies often use keylogger software as part of employee monitoring software to help track employee productivity.
- Information technology departments can use keylogger software to troubleshoot issues on a device.

When keyloggers run, they track every keystroke entered and save the data in a file. Hackers can access this file later, or the keylogger software can automatically email the file to the hacker. Some keyloggers, which are called screen recorders, can capture your full screen at random intervals as well.

Keyloggers can recognize patterns in keystrokes to make it easier to identify sensitive information. If a hacker is looking for password information, they can program the keylogger to monitor for a particular keystroke, such as the at sign (@). Then, the software only notifies them when you are likely entering password credentials alongside an email username. This technique helps malicious users quickly identify sensitive information without needing to sift through all your keystroke data.

CHAPTER 4
DESIGN STARTEGY

4.Design Strategy

A keylogger is a program that records the keystrokes typed on a keyboard, usually without the user’s knowledge. Keyloggers can be used for various purposes, such as troubleshooting, monitoring, or stealing sensitive information.

There are different ways to design a keylogger, depending on the operating system and the programming language.

Installing the required modules:

We are going to use the keyboard module for the keylogger. To install it, open your terminal type in pip install keyboard. If no error occured, you are good to go.

Then we defined our logger(event) function, it takes event as parameter and then it print's it's .name out. Then to the next line, we assigned
the logger function as a parameter to the keyboard.on_press method.
The keyboard.on_press method takes a function as parameter and passes an event to it.
Keylogger tools can either be hardware or software meant to automate the process of keystroke logging. These tools record the data sent by every keystroke into a text file to be retrieved at a later time.

CHAPTER 5
IMPLEMENTATION

5.**Implementation**

Required Module: pynput
This module simply uses a backend engine, depending on your Operating System to monitor your keyboard.
For example, if you're using Linux, you might have an xorg server which you'd use as the backend.
This module interacts with the backend engine, to fetch input from the keyboard.

As a result, this module will work across different Operating Systems, since it does all the work of taking care of the backend calls

- We create a main loop which simply waits for a key to be pressed.
- As soon as the listener detects a key-press, we'll print it on the console.

Now, we'll to listen to a keyboard, we'll monitor two kinds of events:

- Key Presses – Whenever a key is pressed
- Key Releases – Whenever a key is released

We simply need to define these functions, and call pass them as arguments to our keyboard Listener, using pynput.

It's just two lines of code! Here, there are two callback functions called on_press() and on_release() that will get called accordingly.
The second line simply waits for the listener thread to finish executing, using the join() method.

CHAPTER 6
FUTURE WORK

6.**Future Work**

A keylogger is a tool that records the keystrokes made by a user on a device. Keyloggers can be used for legitimate purposes, such as monitoring employee productivity or troubleshooting issues, but they can also be used for malicious purposes, such as stealing personal information or spying on user activity.

Some possible future work of keylogger projects in cyber security are:

- Developing a keylogger from scratch using coding skills. This would require knowledge of programming languages, such as Python, C++, or Java, and how to access the keyboard input of a device.

- Creating a process for detecting and removing keyloggers from a system. This would involve using tools, such as antivirus software, firewall, or malware scanners, to identify and eliminate any suspicious programs or files that may be logging keystrokes.

- Making a keylogger for virtual keyboards. This would be a challenging project, as virtual keyboards are designed to prevent keylogging by using random layouts, encryption, or other methods. A keylogger for virtual keyboards would need to bypass these security features and capture the input.

CHAPTER 7
CONCLUSION

7.**CONCLUSION**

A keylogger is a tool that records the keystrokes made by a user on a computer. It can be used for ethical hacking purposes such as parental control, employee monitoring, or personal information and password retrieval. However, it can also be used for malicious purposes such as stealing sensitive information, spying, or identity theft. Therefore, the conclusion of keylogger in ethical hacking is that it is a powerful tool that should be used responsibly and legally, and with the consent of the user or the owner of the computer. Keyloggers can be detected and removed by using antivirus software, secure coding practices, and encryption.

Keyloggers are software or hardware tools that record every keystroke made by a user on a computer.
They are often used by hackers to steal sensitive information such as passwords, credit card numbers, and other personal data.
However, keyloggers can also be used for ethical hacking purposes such as parental control, employee monitoring, or personal information and password retrieval

In conclusion, keyloggers are powerful tools that can be used for both good and bad purposes. It is important to use them ethically and legally, and to ensure that they are not used to harm others or violate their privacy.