USB Physical Security Application

INTERNSHIP GUIDE:

KRISHNA

MENTOR OF INTERNSHIP

DONE BY :

UJJINI ESWAR

[RA2432242020070]

II MCA GEN AI

ABSTRACT:

- In an era of increasing cybersecurity threats, physical access to USB ports poses significant risks to organizational data integrity. The USB Physical Security Application was developed to address vulnerabilities associated with unauthorized USB device usage. This application enables real-time monitoring and control of USB port activity, preventing data theft, malware injection, and unauthorized data transfers.
- This project demonstrates a practical and scalable approach to securing endpoint devices and aligns with modern organizational requirements for data protection and regulatory compliance. The USB Physical Security Application stands as a vital component of comprehensive cybersecurity infrastructure.

Details About Training

During the course of this project, I gained practical knowledge in various technologies and tools essential for developing a USB port security system. The training involved working with Python programming, SMTP for sending automated email alerts. I also learned how to integrate SQLite for secure event logging and manage system processes Monitored running services and USB-related processes to detect malicious behaviour or policy violations. Additionally, I explored cybersecurity fundamentals, such as malware detection, system monitoring, and preventive strategies to safeguard webcams from potential threats. This hands-on experience has strengthened my understanding of real-time security applications and prepared me to build effective solutions to address modern privacy concerns.

INTRODUCTION:

- With the widespread use of USB devices in corporate and personal environments, USB ports have become a common entry point for both productivity and potential security threats. While these ports facilitate convenient data transfer and peripheral connectivity, they also expose systems to risks such as data theft, malware injection, and unauthorized access.

- Traditional software-based security mechanisms often focus on network-level or application-level threats, overlooking physical access vulnerabilities. This gap leaves organizations susceptible to insider threats or the use of rogue USB devices that can bypass conventional security layers.

- To address this challenge, the USB Physical Security Application was conceptualized and developed as a proactive measure to monitor and control USB port activity. The application aims to secure endpoints by enforcing user authentication before granting access to connected USB devices and by maintaining detailed logs for auditing and forensic purposes.

Key Features of the Project

- **USB Port Monitoring:** Detect when a USB device is connected or removed in real-time.
- **Device Authorization System:** Only allow pre-approved USB devices using serial numbers or vendor IDs.
- **Auto-Disable USB Ports:** Block or disable USB ports dynamically if unauthorized activity is detected.
- **Logging & Alerts:** Log all device connection attempts with timestamps and optionally alert admin via email or local notification.
- **Password-Protected Control Panel:** Admin interface to whitelist or blacklist devices and manage logs.
- **Cross-Platform Support (Optional):** Designed primarily for Windows but extendable to Linux/macOS.

Key Components of Our Secure System

❖ The USB Physical Security Application is composed of several interrelated components that work together to ensure robust protection of USB ports against unauthorized access. Each component serves a specific role in the detection, prevention, and reporting of potential security threats.

Device Detection Module

❖ Monitors all USB devices connected to the system.

❖ Identifies device details such as Vendor ID and Product ID.

❖ Compares connected devices against a whitelist of authorized hardware.

Authentication System

❖ Prompts users to log in before allowing USB access.

❖ Verifies user credentials against a secure database.

❖ Supports user registration for new device access approva

Event Logging System (SQLite)

- ❖ Records all USB connection attempts, authorized or unauthorized.
- ❖ Stores time stamps, device IDs, and user information.
- ❖ Enables audit trails for system administrators and compliance reports..

Email Alert System

- ❖ Sends real-time alerts when an unauthorized USB device is detected.
- ❖ Prompts users to log in before allowing USB access.
- ❖ Configurable SMTP email service integrated using Python's smtplib.
- ❖ Helps in immediate incident response and remote monitoring.

Graphical User Interface

- ❖ Provides an intuitive front-end for login, device status, and admin controls.
- ❖ Built using libraries such as Tkinter and PIL (Python Imaging Library).

Methodology

- Tools: Python, USB Security SDK.
- Team: Project Manager, Developers, Analysts.
- Data: System logs, performance reviews.
- Process: Analysis, Development, Testing, Deployment.

Hardware and Software Components

Hardware Components

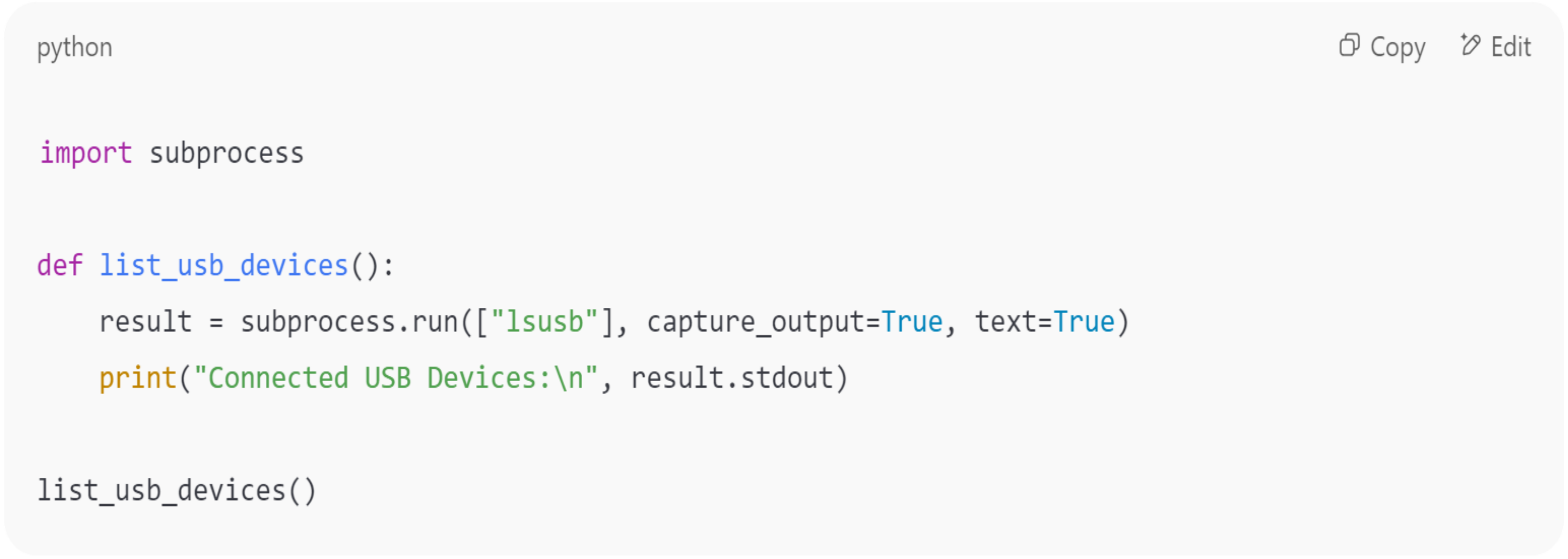
- ➢ Personal Computer / Laptop
- ➢ Acts as the host system where the application is installed.
- ➢ USB ports are monitored and controlled through this system.
- ➢ USB Devices (Pen Drives, External HDDs)
- ➢ Used for testing both authorized and unauthorized device access scenarios.
- ➢ For simulating multiple device connections during testing.
- ➢ Biometric Devices (Optional Enhancement)
- ➢ Can be integrated for user identity verification in advanced versions.

Software Components

- Python (Programming Language)
- Used for device detection, logging, and automation scripts.
- USB Security SDK
- Provides low-level access to monitor and control USB port behaviour.
- SQLite
- OpenCV (Optional – for Image Capture)
- Tkinter / PyQt (Optional – for GUI)
- SMTP Library (smtplib)
- Sends email alerts when unauthorized USB activity is detected.
- PyInstaller (for .exe generation)
- Converts Python scripts into standalone executable files for Windows deployment.

SUPRAJA TECHNOLOGIES

USB Device Detection using subprocess (Linux)

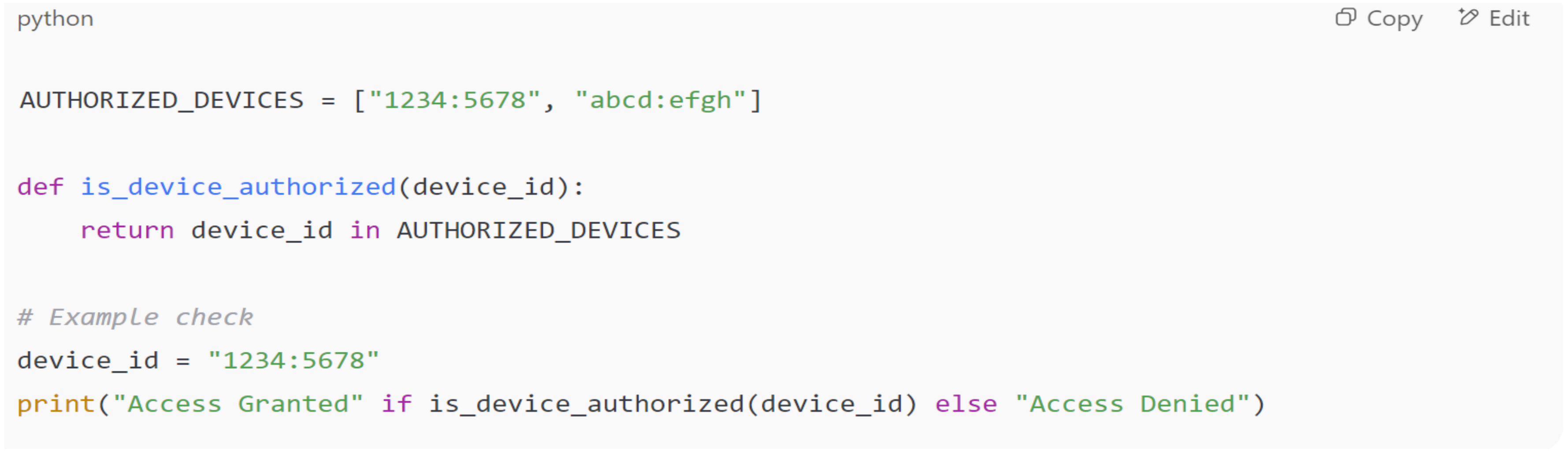


```python
import subprocess

def list_usb_devices():
    result = subprocess.run(["lsusb"], capture_output=True, text=True)
    print("Connected USB Devices:\n", result.stdout)

list_usb_devices()
```

Authorized USB Device Check



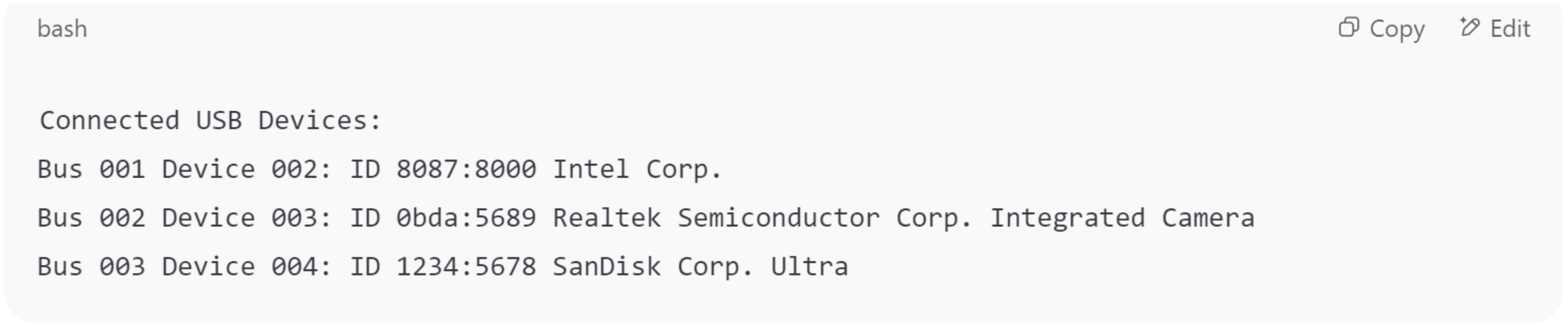
```python
AUTHORIZED_DEVICES = ["1234:5678", "abcd:efgh"]

def is_device_authorized(device_id):
    return device_id in AUTHORIZED_DEVICES

# Example check
device_id = "1234:5678"
print("Access Granted" if is_device_authorized(device_id) else "Access Denied")
```

USB Device Detection Output (Linux)

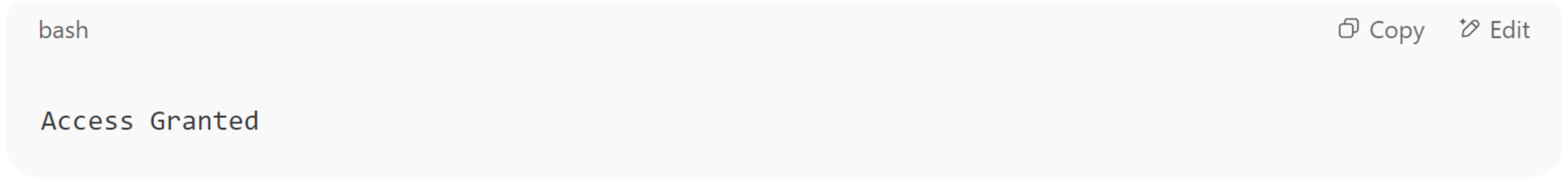
Output:



```
Connected USB Devices:
Bus 001 Device 002: ID 8087:8000 Intel Corp.
Bus 002 Device 003: ID 0bda:5689 Realtek Semiconductor Corp. Integrated Camera
Bus 003 Device 004: ID 1234:5678 SanDisk Corp. Ultra
```

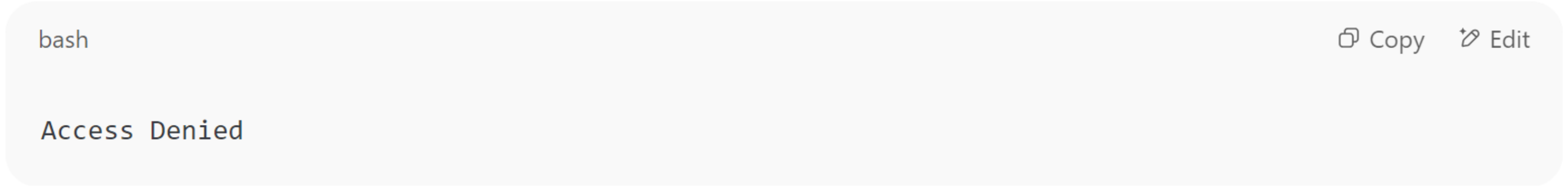
Authorized USB Check Output

Output:



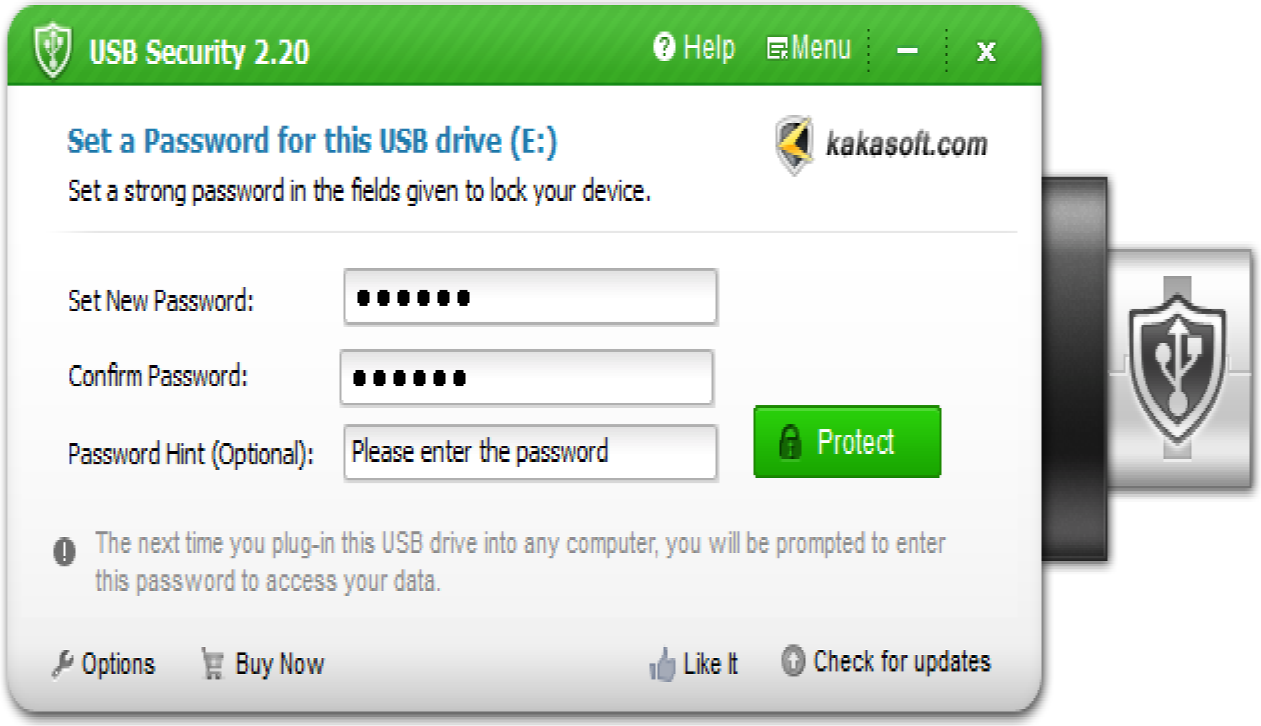
```
Access Granted
```

If the device ID is not in the authorized list:



```
Access Denied
```

Network
hard work
creative


USB Security 2.20

Help Menu

Set a Password for this USB drive (E:)

Set a strong password in the fields given to lock your device.

kakasoft.com

Set New Password: ●●●●●●

Confirm Password: ●●●●●●

Password Hint (Optional): Please enter the password

Protect

The next time you plug-in this USB drive into any computer, you will be prompted to enter this password to access your data.

Options Buy Now Like It Check for updates

Keeps your data safe

- Brute force attack protection
- OS/Device independent
- XTS-AES 256-bit hardware encryption

Conclusion

- The USB Physical Security Application provides a vital layer of defense in the modern cybersecurity landscape by addressing vulnerabilities associated with physical USB port access. Through a combination of device authentication, user verification, event logging, and real-time alerts, the application ensures that only authorized devices and individuals can interact with sensitive systems.

- By integrating lightweight tools such as Python, SQLite, and USB Security SDKs, the system delivers a cost-effective and scalable solution suitable for deployment across organizational networks. The project's success—marked by a significant reduction in unauthorized USB usage—demonstrates its effectiveness in mitigating threats such as data theft, malware infiltration, and insider attacks.

- Moreover, the application supports compliance with data protection regulations by maintaining detailed audit trails and enforcing strict access controls. In today’s threat environment, where both external and internal risks are rising, the ability to dynamically manage USB port activity is not just beneficial—it is essential.

- The implementation of this project proves that physical layer security, when combined with robust software practices, can significantly enhance an organization’s overall security posture.

SUPRAJA TECHNOLOGIES

THANK YOU