USB Device Control & Monitoring Framework
(Project Documentation)

1. Project Overview / Description

This project focuses on developing a USB Device Control & Monitoring Framework designed to detect, monitor, and restrict unauthorized USB activity on a system.

USB devices pose significant security risks such as data exfiltration, malware introduction, unauthorized access, and device spoofing.

This framework provides:

- Unauthorized USB detection

- Auto-blocking of suspicious USB devices

- File movement and data transfer auditing

2. Practical Motivation

USB-based attacks remain one of the most common insider and physical access threats. Examples include:

- USB malware (e.g., BadUSB, USB-based worms)

- Unauthorized data copying onto USB drives

- Plug-and-play backdoors

- Insider threats leveraging portable storage

Monitoring USB activity helps detect:

- Data theft attempts

- Malicious device connections

- Rogue USB implants

- Unauthorized hardware usage

This project provides real-world exposure to system monitoring and endpoint security engineering.

3. Project Objectives

1. Detect USB devices when they are plugged/unplugged.

2. Maintain an allowlist/blocklist of approved USB devices.

3. Automatically block suspicious or unauthorized USB devices.

4. Track file transfers between system drives and USB devices.

5. Generate detailed USB activity and audit reports.

4. Practical Scope of the Project

A. Unauthorized USB Detection:

- Identify newly connected USB storage devices.

- Compare device IDs against allowlist/blocklist.

- Detect USB device spoofing attempts.

B. Auto-Block Suspicious Devices:

- Deny access to unapproved USB devices.

- Log attempted unauthorized connections.

- Optionally disable USB storage temporarily.

C. File Movement Auditing:

- Monitor copy, delete, and modification operations on USB drives.

- Track which files were transferred, by whom, and when.

- Detect large or abnormal data movements.

D. Reporting & Alerting Module:

- Log all USB events with timestamps.

- Highlight suspicious or unauthorized access attempts.

- Generate an audit log or final security report.

5. Tools & Technologies Used

Programming Languages:

- Python (recommended)

- PowerShell (optional)

Modules/Tools:

- pyudev / wmi (USB event monitoring)

- win32api / win32file (Windows-based USB monitoring)

- PowerShell Get-PnpDevice, Get-WMIObject

- hashlib (file hashing for auditing)

Documentation Tools:

- Word / Google Docs

- Draw.io for diagrams

6. Practical Techniques Implemented

Security Techniques:

- Hardware event monitoring

- Device fingerprinting (Vendor ID, Product ID, Serial)

- File auditing & logging

- Allowlist/Blocklist policy enforcement

Blue Team Techniques:

- Tracking insider activity

- Detecting unauthorized hardware

- Monitoring suspicious data movements

- Strengthening endpoint protection

7. Workflow / Architecture (Practical Explanation)

STEP 1: Monitor USB Ports

- Detect real-time connect/disconnect events.

STEP 2: Identify USB Device

- Extract vendor ID, product ID, serial number.

STEP 3: Authorization Check

- Compare against allowlist/blocklist.

STEP 4: Enforcement

- Allow or block the device automatically.

STEP 5: File Auditing

- Monitor transfers between USB and system.

STEP 6: Reporting

- Generate security logs and final audit report.

8. Flowchart (Text Version)

START

↓

Monitor USB Events

↓

Extract Device Information

↓

Check Allowlist / Blocklist

↓

Is Device Authorized? → Yes → Allow

↓ No

Block & Log Attempt

↓

Audit File Movements

↓

Generate Final USB Security Report

↓

END

9. Expected Practical Output

The framework should output:

- Unauthorized USB device alerts
- Block/allow decisions with timestamps
- File transfer logs
- Violations and suspicious activities
- Complete USB audit report

Examples:

- Unauthorized USB detected: VendorID 0x1234, blocked automatically
- File movement detected: 45 files copied to USB drive
- Suspicious device with spoofed serial detected

10. Learning Outcomes

This project teaches:

- How USB hardware communicates with OS
- How attackers misuse USB for malware & data theft

- How USB device monitoring enhances endpoint security

- File auditing techniques

- Implementing allowlist/blocklist policies

11. Project Deliverables

1. Project documentation (Word/PDF)

2. USB monitoring & control toolkit

3. Logs/screenshots of activity detection

4. Flowcharts & architecture diagrams

5. Final presentation (PPT)