**USB Drop Attack Detection & Analysis - PoC Report:-**

**Intern Name**: Ishan Chowdhury  
**Intern ID**: 159  
**Project Title**: USB Drop Attack Detection and Log Analysis Tool  
---------------------------------------------------------------------

### ****Objective:-****

To develop a lightweight Python-based tool that can detect unauthorized USB insertions (potential USB drop attacks) and analyze historical logs to identify patterns or threats.

### ****Tools Used:-****

1. Python 3
2. pywin32 (for WMI support)
3. os, re, datetime, collections

**Components:-**

#### usb\_monitor.py — Real-time USB Drop Attack Detection

import win32com.client

import pythoncom

import datetime

import os

TRUSTED\_DEVICES = ["Kingston", "SanDisk", "Logitech", "HP"]

log\_file = "usb\_attack\_log.txt"

def log\_event(event\_type, device\_name, status):

    timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    log\_entry = (

        f"[{timestamp}] EVENT: {event\_type}\n"

        f"→ Device Name : {device\_name}\n"

        f"→ Status      : {status}\n"

        f"{'-'\*50}"

    )

    print(log\_entry)

    with open(log\_file, "a", encoding="utf-8") as f:

        f.write(log\_entry + "\n")

    os.system(f'msg \* "USB {status}! Device: {device\_name}"')

import win32com.client

import pythoncom

import datetime

import os

log\_file = "usb\_alert\_log.txt"

def log\_event(event\_type, device\_name, status):

    timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    log\_entry = (

        f"[{timestamp}] EVENT: {event\_type}\n"

        f"→ Device Name : {device\_name}\n"

        f"→ Status      : {status}\n"

        f"{'-'\*50}"

    )

    print(log\_entry)

    with open(log\_file, "a", encoding="utf-8") as f:

        f.write(log\_entry + "\n")

    os.system(f'msg \* "USB {status}! Device: {device\_name}"')

def is\_suspicious(device\_name):

    return not any(kw in device\_name.lower() for kw in ['sandisk', 'kingston', 'seagate', 'wd'])

def monitor\_usb():

    print("🔍 USB Drop Attack Detection Tool Started (Windows)\n")

    wmi = win32com.client.GetObject("winmgmts:")

    watcher = wmi.ExecNotificationQuery(

        "SELECT \* FROM \_\_InstanceCreationEvent WITHIN 2 WHERE TargetInstance ISA 'Win32\_USBHub'"

    )

    try:

        while True:

            pythoncom.PumpWaitingMessages()

            event = watcher.NextEvent()

            usb\_device = event.TargetInstance

            drives = wmi.InstancesOf("Win32\_DiskDrive")

            for d in drives:

                if "USB" in d.InterfaceType:

                    device\_name = f"{d.Model} ({d.DeviceID})"

                    status = "⚠️ Suspicious (Possible Attack)" if is\_suspicious(d.Model) else "✅ Safe"

                    log\_event("USB Device Inserted", device\_name, status)

    except KeyboardInterrupt:

        print("\n🛑 Detection tool stopped by user.")

        log\_event("Detection Tool Stopped", "User interrupted", "🛑")

if \_\_name\_\_ == "\_\_main\_\_":

    monitor\_usb()

1. Detects USB device insertions in real-time
2. Compares against a list of trusted device names
3. Logs suspicious insertions with timestamps and notifies the user
4. **log\_analyzer.py — Log Analysis Tool**

import re

from collections import Counter

log\_file = "usb\_attack\_log.txt"

def parse\_log():

    try:

**with open(log\_file, 'r') as f:**

**content = f.read()**

**except FileNotFoundError:**

**print("❌ Log file not found. Run the detector first.")**

**return**

**entries = content.strip().split("-" \* 50)**

**total = len(entries)**

**attack\_count = 0**

**safe\_count = 0**

**devices = []**

**for entry in entries:**

**if "Device Name" in entry:**

**device\_name = re.search(r'Device Name\s+:\s(.+)', entry)**

**status = re.search(r'Status\s+:\s(.+)', entry)**

**if device\_name and status:**

**name = device\_name.group(1).strip()**

**state = status.group(1).strip()**

**devices.append(name)**

**if "Suspicious" in state:**

**attack\_count += 1**

**else:**

**safe\_count += 1**

**unique\_devices = set(devices)**

**device\_counts = Counter(devices)**

**print("\n🔎 USB Log Analysis Summary\n" + "="\*40)**

**print(f"🗂️ Total Events         : {total}")**

**print(f"🛡️ Safe Devices        : {safe\_count}")**

**print(f"⚠️ Suspicious Devices  : {attack\_count}")**

**print(f"🔌 Unique Devices      : {len(unique\_devices)}")**

**print(f"📊 Top Devices         :")**

**for dev, count in device\_counts.most\_common(5):**

**print(f"  - {dev} : {count} times")**

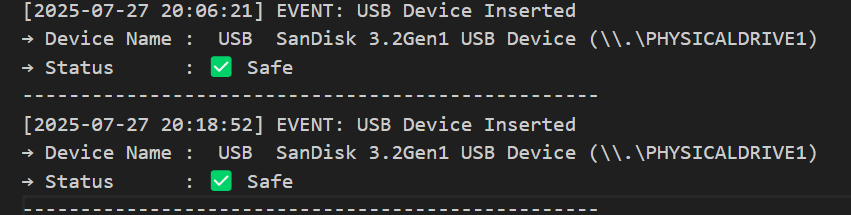
**print("="\*40)**

**if \_\_name\_\_ == "\_\_main\_\_":**

**parse\_log()**

1. **Parses usb\_attack\_log.txt**
2. **Summarizes suspicious vs. safe devices**
3. **Displays top devices and usage frequency**

**Sample Output:-**

****

### ****Conclusion:-****

### This tool provides a practical defense mechanism against USB drop attacks using real-time monitoring and intelligent log analysis, helping IT/security teams stay alert to physical-level threats.