WINDOWS REGISTRY CHANGE MONITORING SYSTEM

(Detection Only)

Project Report submitted in partial fulfillment of the requirements for the completion of

UNIFIED INTERNSHIP PROGRAM

(Cyber Security / Ethical Hacking)

Submitted By

Name : Aquib Ahmad Khan

Intern ID : UMID27112571911

Domain : Cyber Security

Submitted To

Unified Internship

Operating System Used

Microsoft Windows 11

Project Duration

January 2026 – February 2026

1. Project Overview

The Windows Registry is a hierarchical database used by Microsoft Windows to store configuration settings for the operating system and installed applications. Malware frequently abuses autorun registry keys to maintain persistence. This project implements a Windows Registry Change Monitoring System that detects unauthorized registry modifications using Python. The project strictly follows a detection-only approach without exploitation.

2. Objectives

- To understand Windows Registry structure
- To monitor autorun registry keys
- To detect unauthorized registry changes
- To generate alerts for suspicious activity
- To follow ethical cybersecurity practices

3. Tools and Technologies Used

- Operating System: Microsoft Windows 11
- Programming Language: Python 3.14.2
- Modules: winreg, json, time
- Utilities: PowerShell, Registry Editor

4. Methodology

Step 1: Create a baseline snapshot of autorun registry keys.
Step 2: Continuously monitor registry keys for changes.
Step 3: Compare current values with baseline.
Step 4: Generate alerts for new or removed entries.
Step 5: Analyze and document findings.

5. Automated Monitoring Script

```
#!/usr/bin/env python

import winreg
import json
import time

BASELINE_FILE = "baseline.json"

KEYS = [
(winreg.HKEY_CURRENT_USER,
r"Software\\Microsoft\\Windows\\CurrentVersion\\Run"),
]

def read_key(root, path):
  data = {}
  try:
    key = winreg.OpenKey(root, path)
    i = 0
    while True:
      name, value, _ = winreg.EnumValue(key, i)
      data[name] = value
      i += 1
  except:
    pass
  return data

with open(BASELINE_FILE, "r") as f:
  baseline = json.load(f)

print("[*] Registry monitoring started...")
print("[*] Press CTRL + C to stop")

while True:
  for root, path in KEYS:
    current = read_key(root, path)
    old = baseline.get(path, {})

    for k in current:
      if k not in old:
        print(f"[ALERT] New autorun entry detected: {k} ->
{current[k]}")

    for k in old:
      if k not in current:
        print(f"[ALERT] Autorun entry removed: {k}")
```

```
time.sleep(10)
```

6. Findings

During execution, a new autorun registry entry named 'TestEntry' was detected. This demonstrates a common persistence technique used by malware. The monitoring system successfully generated alerts in real time.

7. Mitigation and Recommendations

- Regularly monitor autorun registry keys
- Restrict unauthorized registry modifications
- Apply least privilege access control
- Use endpoint protection solutions
- Perform periodic system audits

8. Screenshots

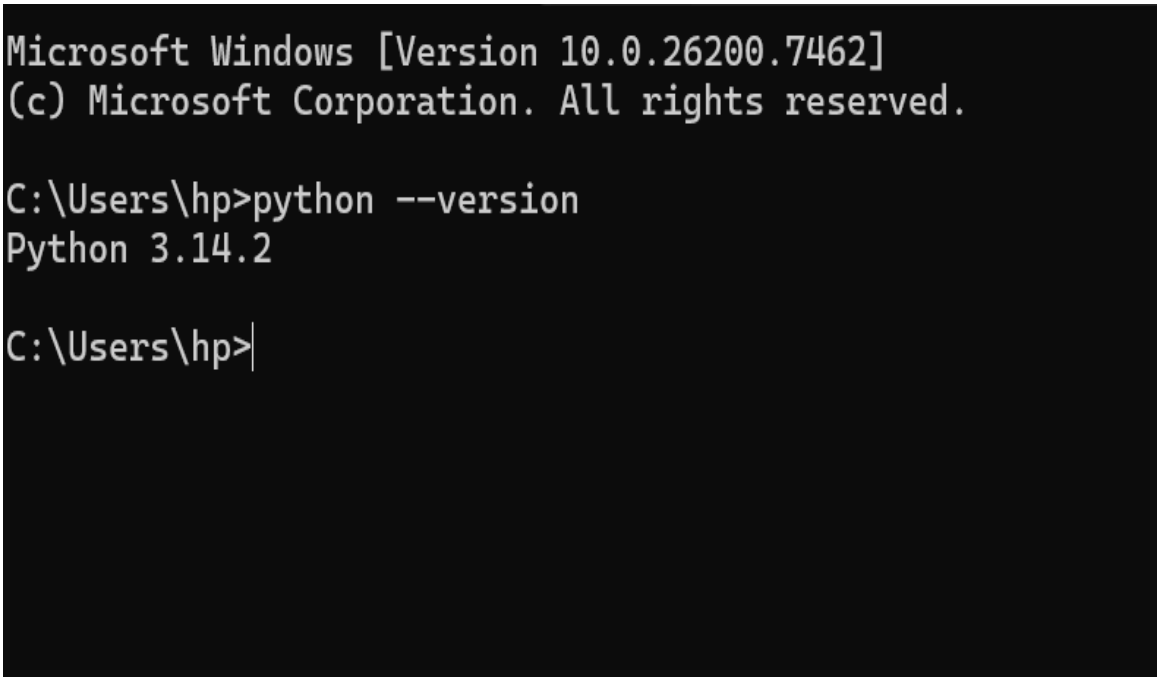

Figure 1: Registry monitoring output

Figure 2: Registry monitoring output

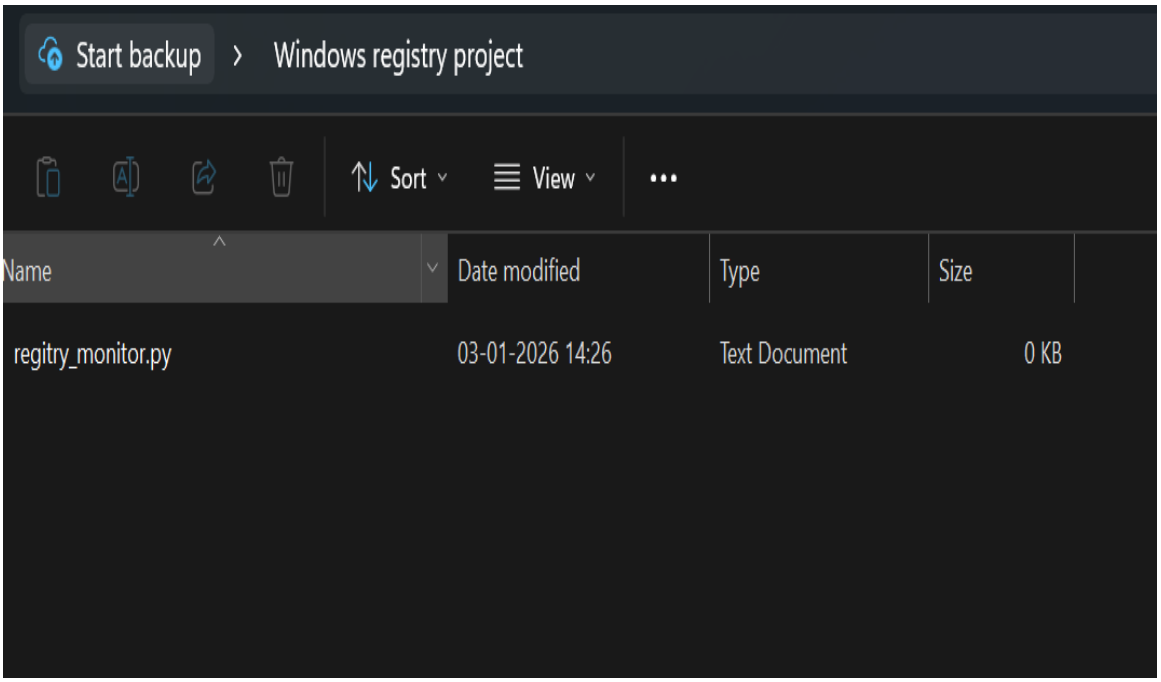

Figure 3: Registry monitoring output

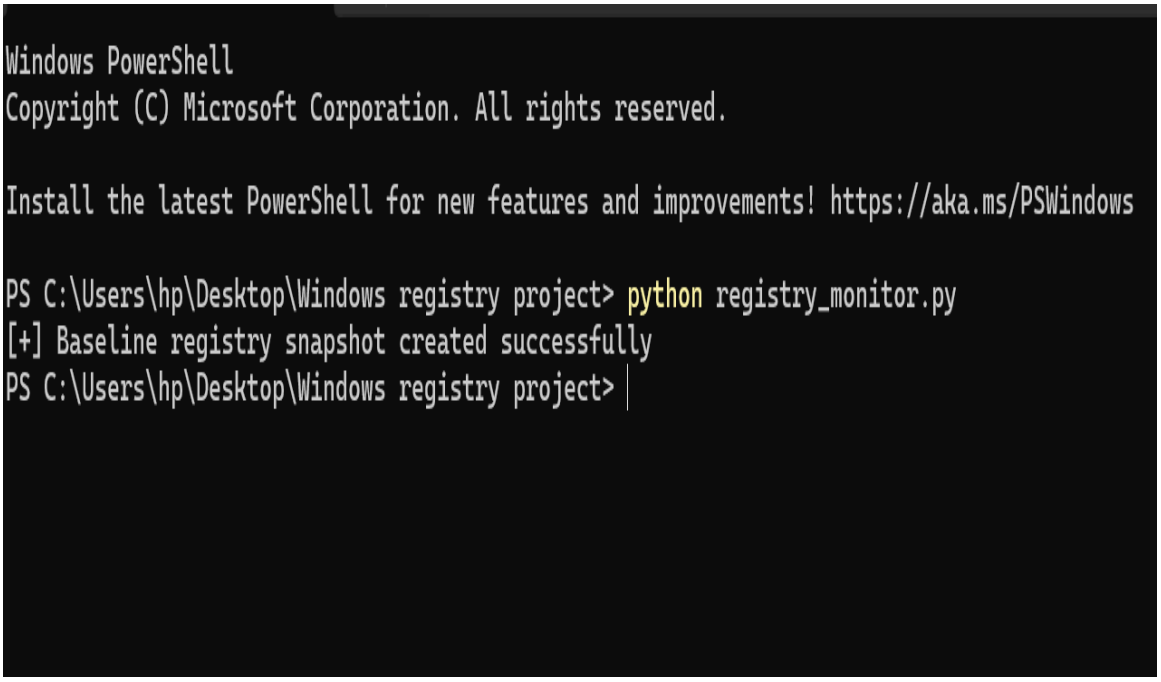

Figure 4: Registry monitoring output

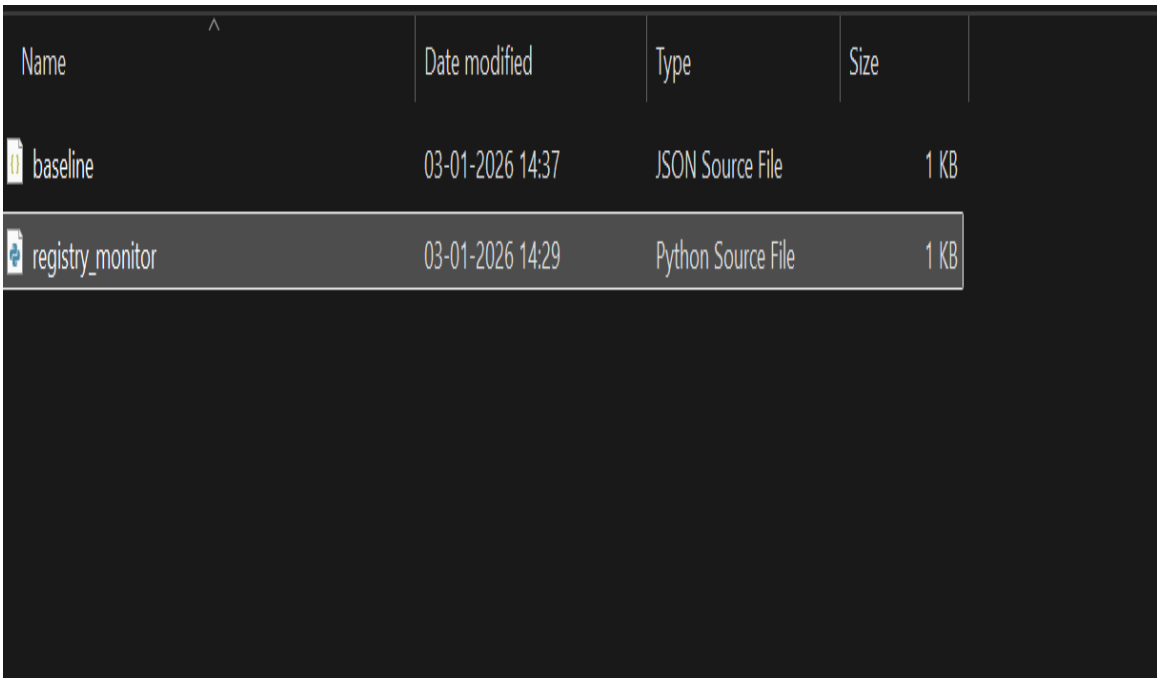

Figure 5: Registry monitoring output

Figure 6: Registry monitoring output

Figure 7: Registry monitoring output

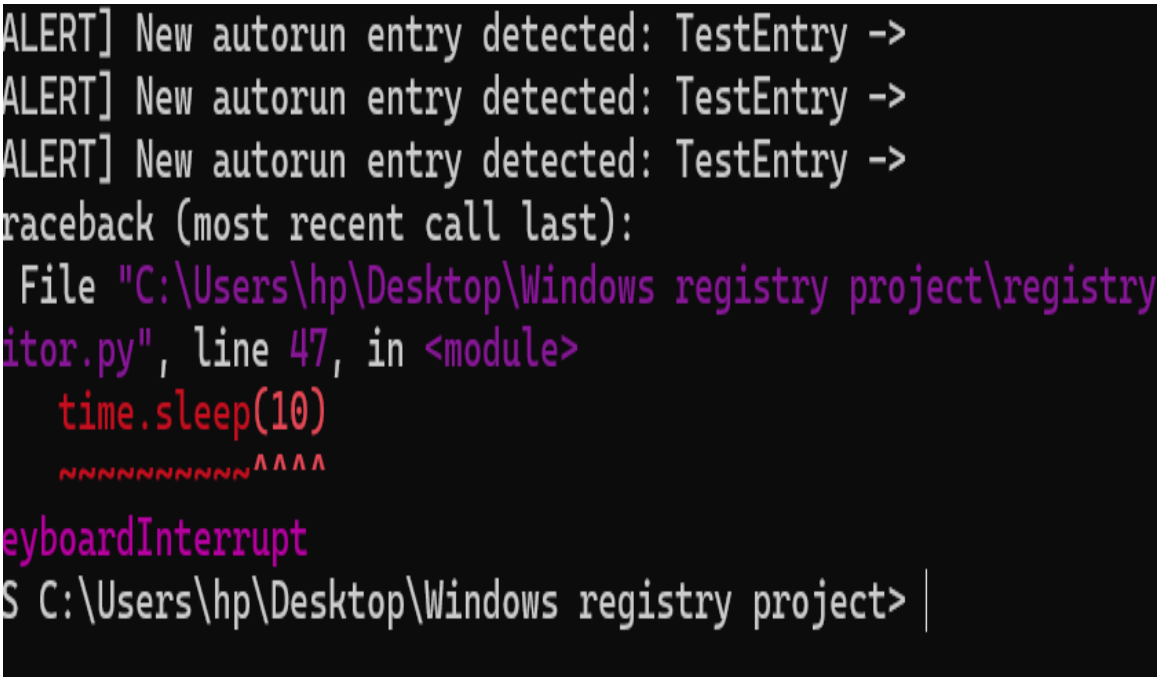

Figure 8: Registry monitoring output

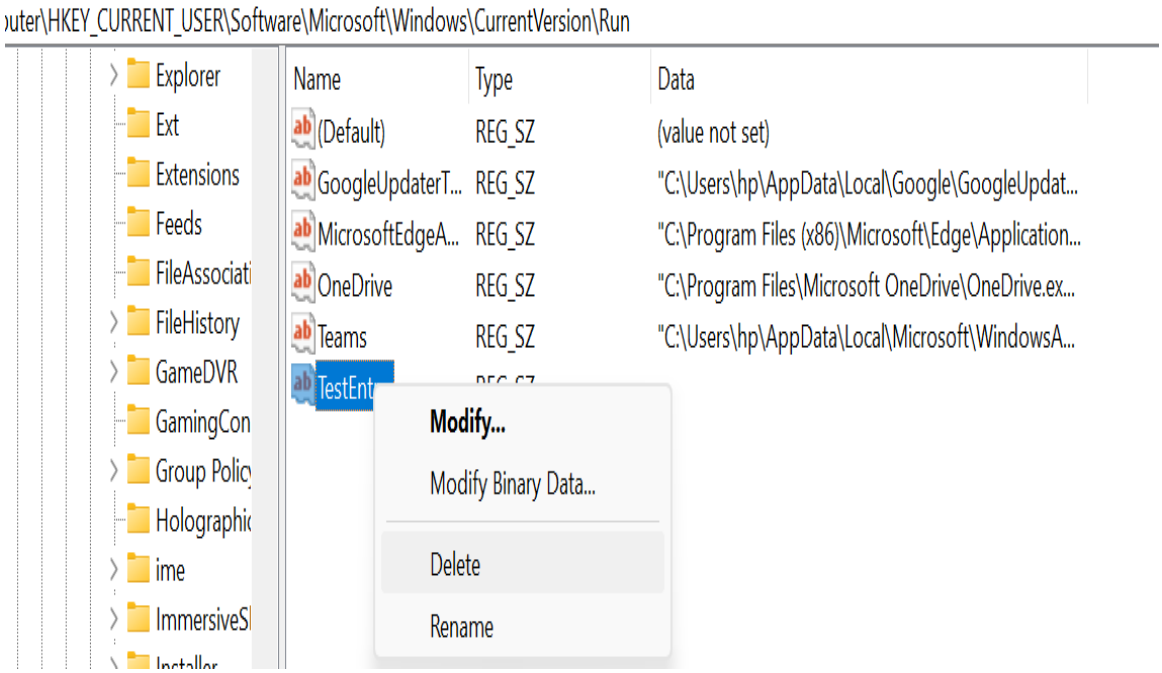

Figure 9: Registry monitoring output

9. Conclusion

The Windows Registry Change Monitoring System successfully demonstrates an effective and ethical approach to detecting persistence-related registry modifications using Python.

10. Declaration

I hereby declare that this project is my original work completed as part of the Unified Internship Program and was performed strictly for educational purposes.