Windows Registry Change Monitoring System
(Project Documentation)

1. Project Overview / Description

This project focuses on designing a Windows Registry Monitoring System that tracks unauthorized or suspicious changes made to registry keys.

The Windows Registry stores critical configuration data, startup locations, system settings, and application behavior. Malware often targets the registry

to maintain persistence, modify system policies, or disable security tools.

This project provides a complete monitoring toolkit that detects changes in sensitive registry paths, identifies malware-like modifications,

and verifies registry integrity using baseline comparisons.

2. Practical Motivation

The registry is a high-value target for:

- Persistent malware

- Ransomware configuration changes

- Unauthorized startup entries

- Privilege escalation modifications

- Security tool tampering

Unauthorized registry changes may indicate:

- Compromise by malware

- Privilege misuse

- Policy violations

This project enables hands-on defensive monitoring experience used in SOC, IR, and digital forensics.

3. Project Objectives

1. Monitor autorun registry keys for persistence mechanisms.

2. Detect malware-like registry changes (e.g., disabling security tools).

3. Create a registry integrity checker using baseline comparison.

4. Provide real-time or scheduled alerts on modifications.

5. Generate a detailed registry change report for analysis.

4. Practical Scope of the Project

A. Autorun Key Monitoring:

- Monitor Run and RunOnce keys in HKCU and HKLM.

- Detect new startup entries added without authorization.

- Identify executables launched automatically upon login.

B. Malware-Like Change Detection:

- Monitor keys often modified by malware:

  * Security policy modifications

  * Firewall/Defender disable keys

  * Shell replacements

  * UAC bypass-related keys

C. Registry Integrity Checker:

- Capture a baseline registry snapshot.

- Compare current registry state to the baseline.

- Detect additions, deletions, and value modifications.

D. Alerting & Reporting Module:

- Log changes with timestamp, key path, old value, new value.

- Optional on-screen or file-based alerts.

- Generate a consolidated change-analysis report.

5. Tools & Technologies Used

Programming Languages:

- Python (recommended)

- PowerShell (optional)

Modules/Tools:

- winreg (Python registry module)

- PowerShell Get-ItemProperty / Set-ItemProperty

- Task Scheduler (optional monitoring automation)

- hashlib (for integrity checksums)

Documentation Tools:

- Word / Google Docs

- Draw.io for diagrams

6. Practical Techniques Implemented

Detection Techniques:

- Polling-based registry monitoring

- Baseline comparison for integrity checking

- Autorun persistence detection

- Malware behavior pattern recognition

Blue Team Techniques:

- Monitoring startup persistence

- Detecting configuration tampering

- Registry forensics analysis

- Logging suspicious system modifications

7. Workflow / Architecture (Practical Explanation)

STEP 1: Define Sensitive Registry Keys

- Autorun paths

- Security configurations

- System behavior keys

STEP 2: Baseline Creation

- Capture initial registry values for comparison.

STEP 3: Continuous Monitoring

- Poll registry keys at defined intervals.

- Track additions, deletions, or modifications.

STEP 4: Change Analysis

- Compare changes with known malware behaviors.

STEP 5: Alert/Log Generation

- Output alerts for unauthorized modifications.

STEP 6: Final Reporting

- Consolidated log of all detected changes.

8. Flowchart (Text Version)

START

↓

Load Monitoring Configuration

↓

Capture or Load Baseline Registry Snapshot

↓

Monitor Registry Keys at Regular Intervals

↓

Detect Changes (Add/Modify/Delete)

↓

Compare with Malware Behavior Patterns

↓

Generate Alerts + Logs

↓

Export Final Report

↓

END

9. Expected Practical Output

The toolkit should output:

- Registry change logs (timestamped)

- Summary of unauthorized or suspicious modifications

- Autorun program detections

- Integrity check results

- Malware-pattern alerts

Examples:

- New autorun entry detected: C:\Users\User\AppData\malware.exe

- Registry key altered: Windows Defender disabled

- Integrity failure: Value mismatch from baseline

10. Learning Outcomes

This project teaches:

- Windows registry structure and key categories

- How malware persists via registry modifications

- How defenders monitor and analyze unauthorized changes

- Practical scripting for registry auditing

- Building a basic endpoint-monitoring system

11. Project Deliverables

1. Project documentation (Word/PDF)

2. Registry monitoring toolkit

3. Screenshots of monitoring results

4. Baseline and integrity check logs

5. Flowcharts and architecture diagrams

6. Final presentation (PPT)