Windows Process Protector

Problem:

Unauthorized process access and **memory tampering** are common attack vectors used by malicious software, including cheats in online games and unauthorized access attempts in enterprise applications.

Existing security solutions either impose high system overhead, slowing down performance, or lack real-time monitoring, making them ineffective in dynamically changing environments.

Game developers and cybersecurity analysts need a lightweight, proactive security tool that can detect and prevent unauthorized modifications to running processes without affecting user experience.

Our Solution:

Windows Process Protector ensures real-time monitoring and security for Windows applications with:

- **Kernel-level process protection** against unauthorized access.
- **Memory access prevention** to block malicious modifications.
- **Code integrity verification** to prevent tampering.
- **DLL injection and remote thread blocking** to stop external attacks.

Web-based dashboard that enables real-time security alerts and process management.

This solution is optimized for **low system impact**, making it effective for both **gaming and enterprise security**.

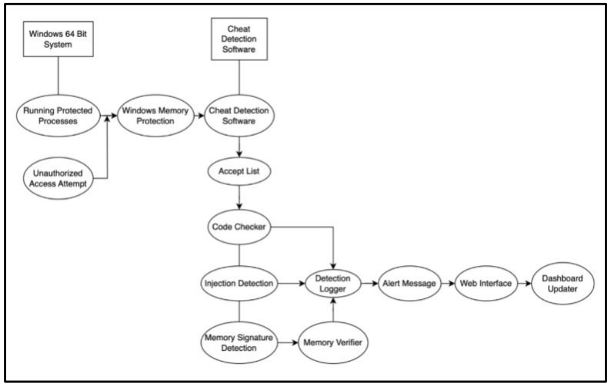
Meet The Team

Bartosz Kawalkowski

Harshil Patel

Advisor: Nitin Nitin

Our Design:



Frontend: Using HTML5, CSS3, and Material Design for a modern, responsive UI.
Backend: Developed in C++17 for high-performance programming.
Windows API: Leverages NT API for advanced system operations and Windows Security Framework for privilege management and access control
Database: MySQL stores user data, security logs, and protection settings.

System Components:

Kernel Driver: Monitors and blocks unauthorized access attempts.
Threat Logger: Records security events for tracking and analysis.
Web Dashboard: Displays security alerts and process management options.

Challenges:

Ensuring system stability while integrating kernel-level monitoring.

Optimizing performance to avoid slowing down protected applications.

Navigating Windows security policies to allow real-time monitoring.

Developing real-time UI updates for an intuitive user experience.

Future Experience:

Live demonstration of memory protection and real-time threat detection.

Showcase of the web dashboard with interactive security alerts.

Performance benchmarking to demonstrate minimal system impact.

Hands-on interaction, allowing users to simulate threats and observe system response.

Contact Us:

Bartosz Kawalkowski: kawalkba@mail.uc.edu

Harshil Patel: patel3hs@mail.uc.edu