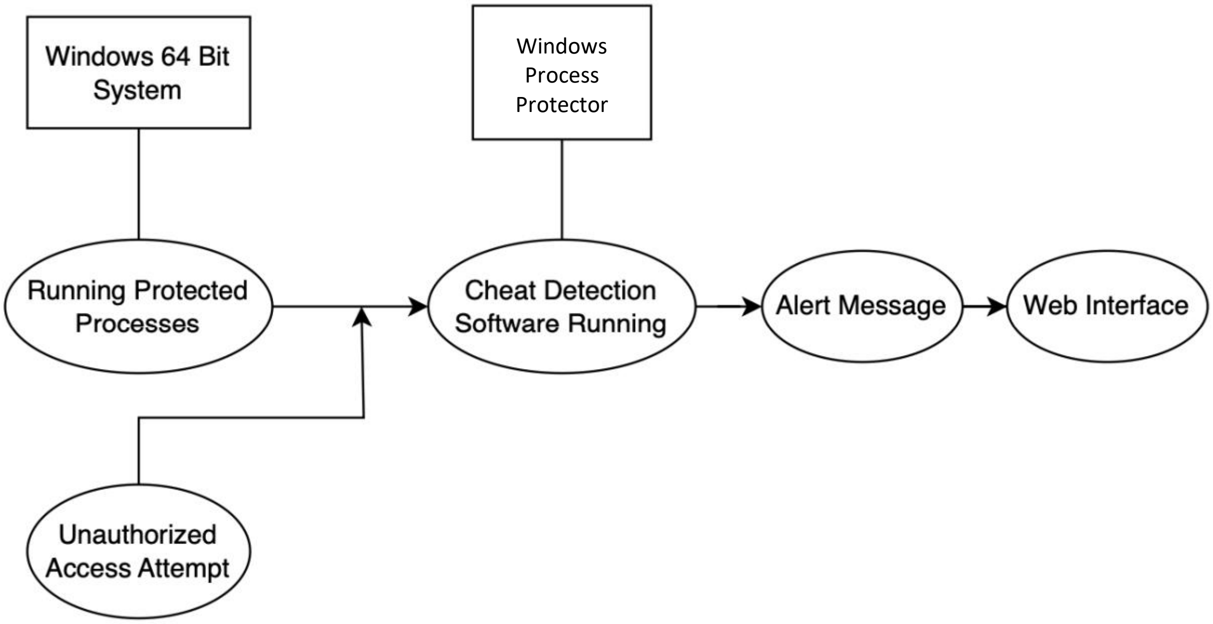
Assignment 4

Harshil Patel, Bartosz Kawalkowski

Project: Windows Process Protector

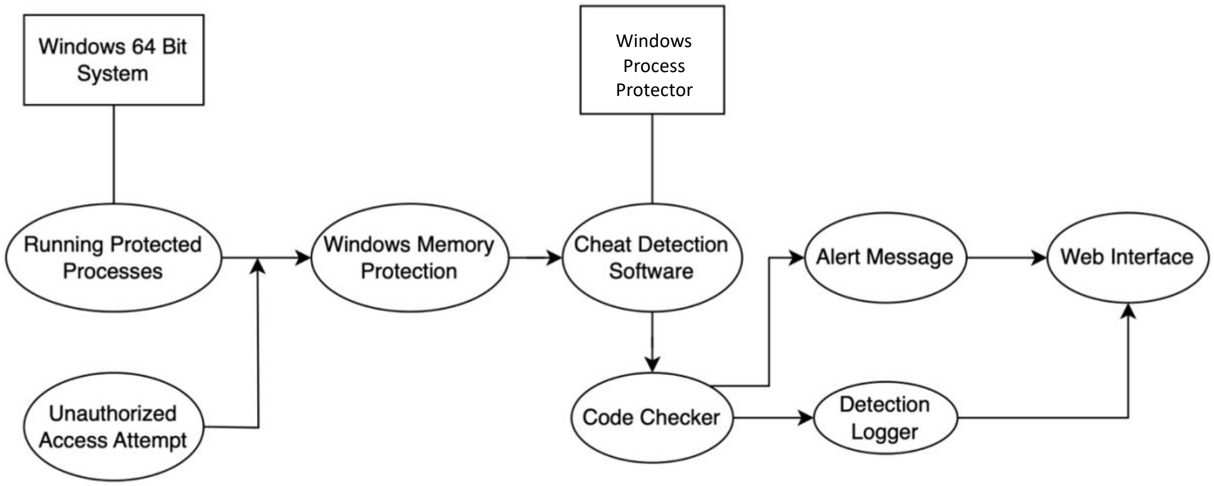
Goal: Develop software the detects and alerts users about any unauthorized access (cheat detection) that are being made in the current running processes.

D0:


Windows 64 Bit System
Windows Process Protector
Running Protected Processes
Cheat Detection Software Running
Alert Message
Web Interface
Unauthorized Access Attempt

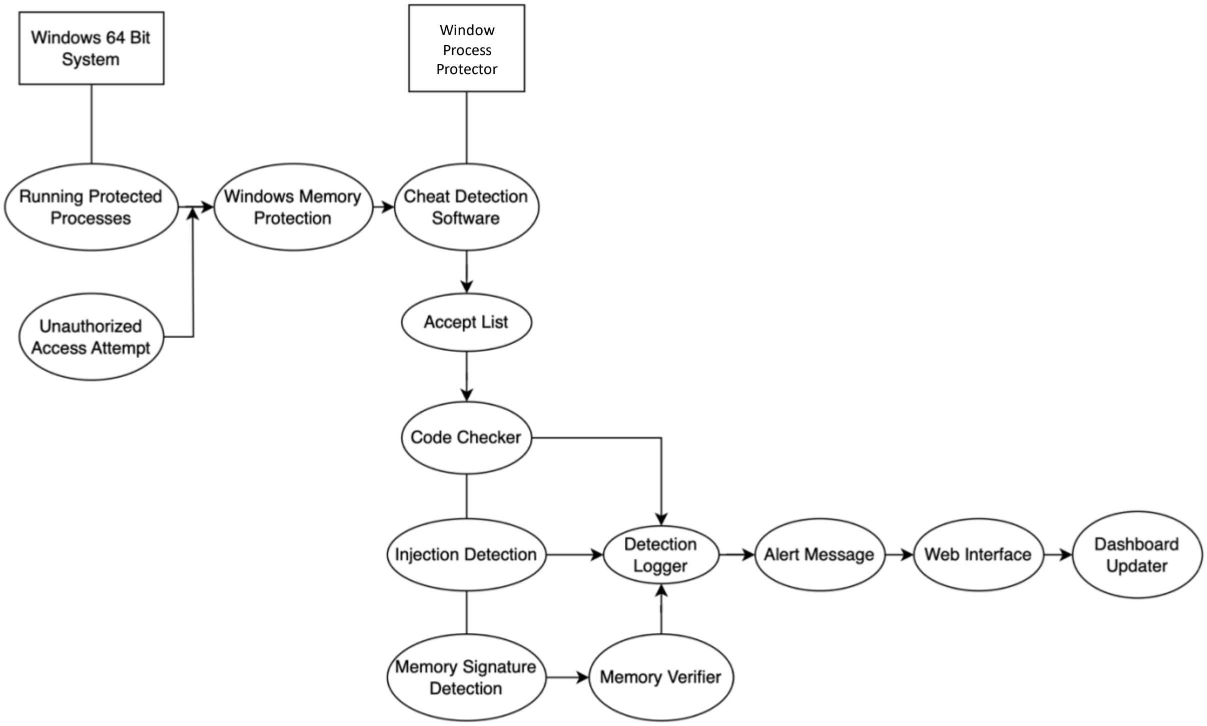
- There are two systems in the top boxes: Windows 64 bit and the Windows Process Protector (WPP).
- We see the processes running on Windows and any unauthorized access attempts being made. These are both the inputs used in the WPP.
- WPP then outputs a message that alerts the user about unauthorized access attempt and displays it on the web interface.
- The boxes show running systems, the ovals show what the software is doing.
- The lines connect the System to its running process, the arrows show the flow of WPP.

D1:


Windows 64 Bit System
Windows Process Protector
Running Protected Processes
Windows Memory Protection
Cheat Detection Software
Alert Message
Web Interface
Unauthorized Access Attempt
Code Checker
Detection Logger

- After further expansion, we have a windows memory protection stage that is an expansion of the input being made in WPP. This module detects any suspicious activity that is being detected in the memory and inputs it into WPP.
- We also show a Code Checker component in WPP to check the integrity of the code. This is to detect any unauthorized alterations being made in the code.
- Any unauthorized alterations get forwarded to the detection logger.
- The outputs we see going to the user web interface is the alert message and a logged message.

D2:



- Within WPP, we first go through the Trust List. This checks to see if any detected processes were previously detected or if they are already trusted and marked as safe.
- Further expanding on WPP:
  - Injection Detection: Detects any cheating happening through memory injection.
  - Memory Signature Detection: Collects the memory signature for further analysis.
    - Memory Verifier: Identifies if the suspicious activity happening in memory has been detected before.
- After these components have been executed, they output the unauthorized access being made in any running processes and forwards them to the Detection Logger.
- Then the alert message is created and outputted to the Web Interface.
- A Dashboard Updater component updated the list of detected process on the Web Interface, as well as adding the alert message.