Forensic Analysis Computer Tool For Video-Game Cheat Detection In-Game

Jack Papaioannou – jp1g19@soton.ac.uk

Goals

The purpose of this project is split into two parts.

- The first part involves research on existing anti cheat systems and methodologies as well as analyse their various advantages and disadvantages
- The second part is occupied with the development of an anti-cheat tool that proposes a method using memory forensics that is built on the elements that other anti-cheat systems lack

Motivation

- Revenue: the video game industry is very lucrative with a high market value that contributes to the global economy
- Attackers: attracts adversaries that aim to disrupt that market and even capitalise on it
- Workforce: jobs get disrupted and ruined affecting many employees and employers within a workforce in the video game sector
- Fairness: game experiences are sabotaged, and an unfair playing field is created
- Player satisfaction: players are more enticed to keep playing a game if they have fewer negative experiences from cheaters

Video-Game cheats

What are video game cheats?

- A wide collection of tools, software, hardware, methods and techniques that aim to modify a game in a way that alters the experience for the player, usually in an advantage gaining way
- Those advantages vary from game to game and genre to genre but usually lead to results that increase game values, provide inhuman input to game controls or provide information that other players don’t have access to (wall hacking).

Anti-Cheat

What is an anti-cheat?

- An anti-cheat consists of software in the form of a tool or a system with the primary purpose of maintaining a game's fair playing field
- It detects various types of cheating that can exist inside a game (DLL injection, memory address changes) and notifies the game in order to produce mitigative actions (bans).
- They are usually provided to a game either from the game's developer for increased accuracy in that specific game or from a third-party company that specializes in anti-cheat systems for specific genres of games.

Existing Methods

Existing commonly used methods for anti-cheats include:

- Machine learning based methods (gameplay frame analysis)
- Artificial intelligence methods (gameplay behavior analysis and classification)
- Memory forensic approach to anti-cheats (memory monitoring)
- Mixture of the above

Memory Forensics Approach

Positives

- Can accurately scan for malicious files belonging to cheats or code injections
- Live changes to memory sections can be detected
- Real time analysis of the game memory
- Wide range of cheating methods can be detected
- No need for re-training a model like the AI and ML approach

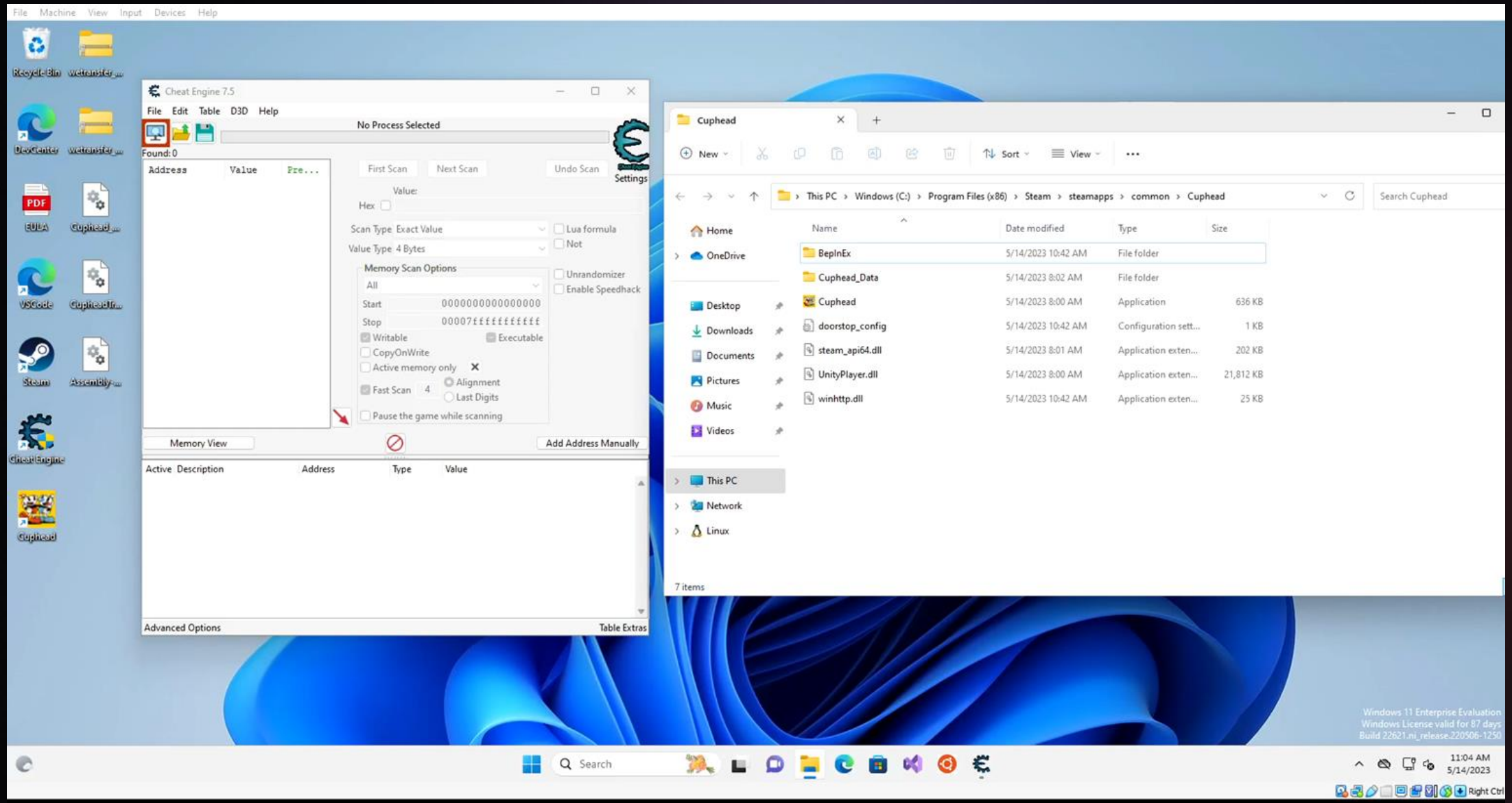
Negatives

- False positives can occur just like other methods
- Privacy concerns for kernel level access to systems
- Game performance concerns due to the computational power of using memory forensics in real time
- Approach can be outdated due to new cheats released if not kept up to date consistently

Cheat Detection Tool

- The tool consists of a python script that aims to automate the use of the Volatility python library along with threading and a weight system
- It is applied to a memory dump of a game process to detect any cheats applied to it, more specifically DLL injection
- The plugins used from the Volatility library are `windows.pslist`, `windows.dlllist` `windows.malfind` and `windows.vadyarascan`
- It aims to efficiently detect cheats based on those Volatility plugins without being updated using a YARA rule file that can be updated based on community input

Demonstration



Testing Results

Testing was conducted on three games. Cuphead and Valheim consist of newly developed games using the Unity game engine and Half Life 2 is an old game, but its game engine is still used today.

- Results yielded many indications of cheats being present 80% of the time
- DLL injection was detected as well as modification to original game files
- Half Life 2 had less indications due to the fact the game is very old and outdated including the cheating files that were found were mostly outdated and scarce to find

Future Work

Future work planned for this tool involves:

- Open-source database for community input on common cheat file signatures
- Added functions to the tool to detect more cheating methods for specific games
- Productization of the tool for a general user market with the development of a user interface
- More in depth research for more samples on a bigger variety of games to test the performance of the tool more accurately

Any Questions?