Embarking on a Quest: The Art of Video Game Hacking

Aaron Hnatiw (.com)

What am I talking about?

- Game hacking: Modifying video games to gain an advantage or alter gameplay.
- Core Skills:
  - Reverse engineering
  - Binary exploitation
  - Assembly language

What am I *not* talking about?

- **Cheating**: Game hacking is not about ruining the experience for others.
- **Online Play**: Hacking online games can lead to bans and legal repercussions.

Why hack video games?

- **Foundation**: Game hacking cultivates skills in binary exploitation and reverse engineering.
- **Career Boost**: These core skills have vast applications, from cybersecurity to software development.
- It's **fun**!

Beginner Game Hacking

How: Beginner

- Find and modify game values in memory.
- Main tool: **Cheat Engine**.
- Example: Infinite money.

Infinite Money Example

An example demonstrating infinite money using Cheat Engine.

Beginner Resources

- Guided Hacking's intro to Cheat Engine video series:
  - Watch here
- Stephen Chapman YouTube channel:
  - Visit channel
- FearLess Cheat Engine forum for Cheat Engine tables:
  - Join discussions

Intermediate Game Hacking

How: Intermediate

- Code injection: alter game mechanics by modifying game code (not just memory values).
- Example: Gwent in The Witcher 3.

Gwent Hacking Example

Alter game mechanics for an "auto-win" in Gwent.

Reverse Engineering & Debugging

- Find what you're looking for through reverse engineering (static analysis) and debugging (dynamic analysis).
- Main static analysis tool:
  - Ghidra, IDA Pro (expensive; alternatively use IDA Home), Binary Ninja.
- Main dynamic analysis tool: x64dbg.
- Learn to read assembly instructions.

Common Assembly Instructions

- `mov`: move data from one location to another.
- `lea`: load effective address.
- `jmp/jxx`: jump to another location in the code.
- `cmp`: compare two values and set flags.
- `test`: bitwise AND operation.
- `add` / `sub` / `mul`: arithmetic operations.
- `push` / `pop`: push a value onto the stack or pop it off.
- `call`: call a function.
- `ret`: return from a function.

Understanding Calling Conventions

- `cdecl`: Arguments are pushed onto the stack in reverse order, caller cleans the stack.
- `stdcall`: Similar to cdecl, but the callee cleans the stack.
- `fastcall`: First two arguments are passed in registers, the rest on the stack.
- `thiscall`: The `this` pointer is passed in ECX, other arguments on the stack.
- `x64`: Uses registers for the first four arguments, additional arguments on the stack.

Combatting Anti-Debug Measures

- Many games include mechanisms to block or detect debuggers.
  - A comprehensive list of techniques is found in The Ultimate Anti-Debugging Reference.
- Bypass these measures with tools like ScyllaHide.

Cheat Engine's Auto Assembler

- Inject code into the process using Cheat Engine's powerful "auto assembler" feature.
- Cheat Engine: Auto Assembler Guide

Cheat Engine Auto Assembler Demo

Example fly hack in Bioshock Infinite showcases its capability:

Lua Scripting with Cheat Engine

Cheat Engine supports Lua scripting for even more complex and dynamic hacks.

- Extensive documentation and undocumented functions.
- Example: Cheat Table demonstrates Lua's power within Cheat Engine.

Advanced Game Hacking

Going Beyond Cheat Engine with C++

- Use the Windows API directly for more control and flexibility:
  - `OpenProcess()`, `ReadProcessMemory()`, `WriteProcessMemory()` for interacting with game processes.
  - `CreateRemoteThread()`, `VirtualAllocEx()` for code injection.
  - Debugging functions like `DebugActiveProcess()` and `WaitForDebugEvent()` for custom debugging solutions.

Explore more: Windows API reference

Advanced Techniques: Bypassing Anti-Cheat

Anti-cheat measures are sophisticated, but not insurmountable.

Anti-cheat Techniques # 1: Kernel Mode Drivers

Kernel-mode drivers: These run in the kernel and have the highest level of access to the system. They can monitor and block unauthorized modifications to game memory.

Example: BattlEye, Easy Anti-Cheat.

Kernel Mode Drivers Bypass

- Develop a kernel-mode driver to intercept and modify system calls made by the anti-cheat software.
  - Here is some starter code for a kernel-mode driver: https://thehackerdev.com/windows-kernel-driver-starter-code/
- There are many vulnerable kernel mode drivers that can be exploited to gain kernel mode access (using a valid signed certificate): https://www.loldrivers.io/

Anti-cheat Techniques # 2: Code Signing

Code signing: Ensures that the software has not been tampered with since it was signed. Anti-cheat software can detect unsigned or modified code.

Bypass: Patch the game's executable to remove the code signing checks.

Common Windows API function used to check code signing:

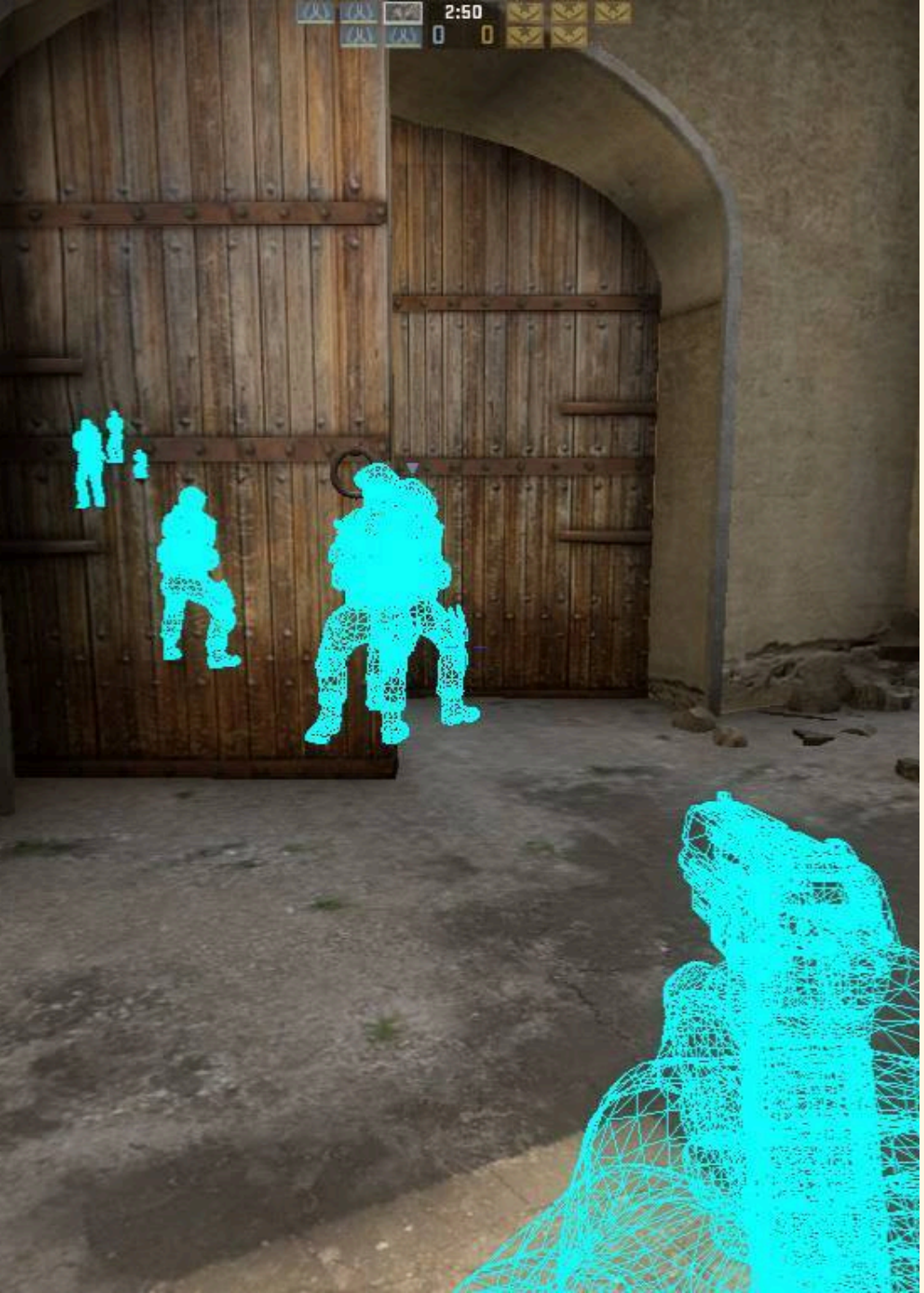
`WinVerifyTrust`

Anti-cheat Techniques # 3: Memory Scanning

Memory scanning: Anti-cheat software scans the game's memory for known cheat signatures.

Bypass: Encrypt or obfuscate your cheat code to avoid detection. Use polymorphic code to generate unique signatures each time the cheat is run.

Malware is generally *excellent* at hiding itself from memory scans. You can use similar techniques to hide your cheat code.



Visual Overlays

- Create visual overlays to display information on top of the game.
- Use this for wall hacks (showing player locations, health, etc.).
- Use DirectX or OpenGL hooks to render graphics on top of the game window.

Network Interception

Modify network packets to hack games that have an online component. Most games (not all) use certificate pinning to prevent this. Some solutions below...

- Overwrite the game's SSL certificate pinning function with a no-op.
- Find and replace the game's pinned certificate with your own (either in the binary or in memory at runtime).
- At some point, the traffic needs to be decrypted in memory, so you can intercept it there (if you're quick enough).

Network Interception (Cont'd)

This is my particular specialty outside of game hacking. I've written an open-source tool called Cartograph to do HTTP network interception and injection.

Other Resources I Haven't Mentioned

- **Guided Hacking Forum:** A treasure trove of knowledge covering what I've discussed here, with examples and actual code.
  - Their Game Hacking Bible is particularly noteworthy.
- **ReversingHub YouTube Channel:** Focused on reverse engineering content.
- **OALabs YouTube Channel:** Specializes in malware analysis and reverse engineering.

Other Resources (Cont'd)

- **CasualGamer YouTube Channel:** Offers insights into game hacking with C++.
- **Tuts 4 You Forum:** Discussions and resources on reverse engineering and debugging.
- **Unknown Cheats Forum:** A community for game hacking enthusiasts.
- **Twitter List of Game Hackers:** Stay up-to-date with game hacking-related content.

Thank You!

- **Name**: Aaron Hnatiw
- **Contact**: https://aaronhnatiw.com
- Slides will be available on my website.

Thank you for joining this journey into game hacking. Keep exploring, learning, and hacking responsibly!