# What Game Hackers Teach us About Offensive Security and Red Teaming

Joe “Juno” Aurelio
@JunoBytez

*I do not condone cheating in online or competitive games. Game hacking is incredibly interesting and deep subject with overlap in multiple areas of security.*

*Also, cheaters are usually bad at the game they're playing (skill issue)*

# About

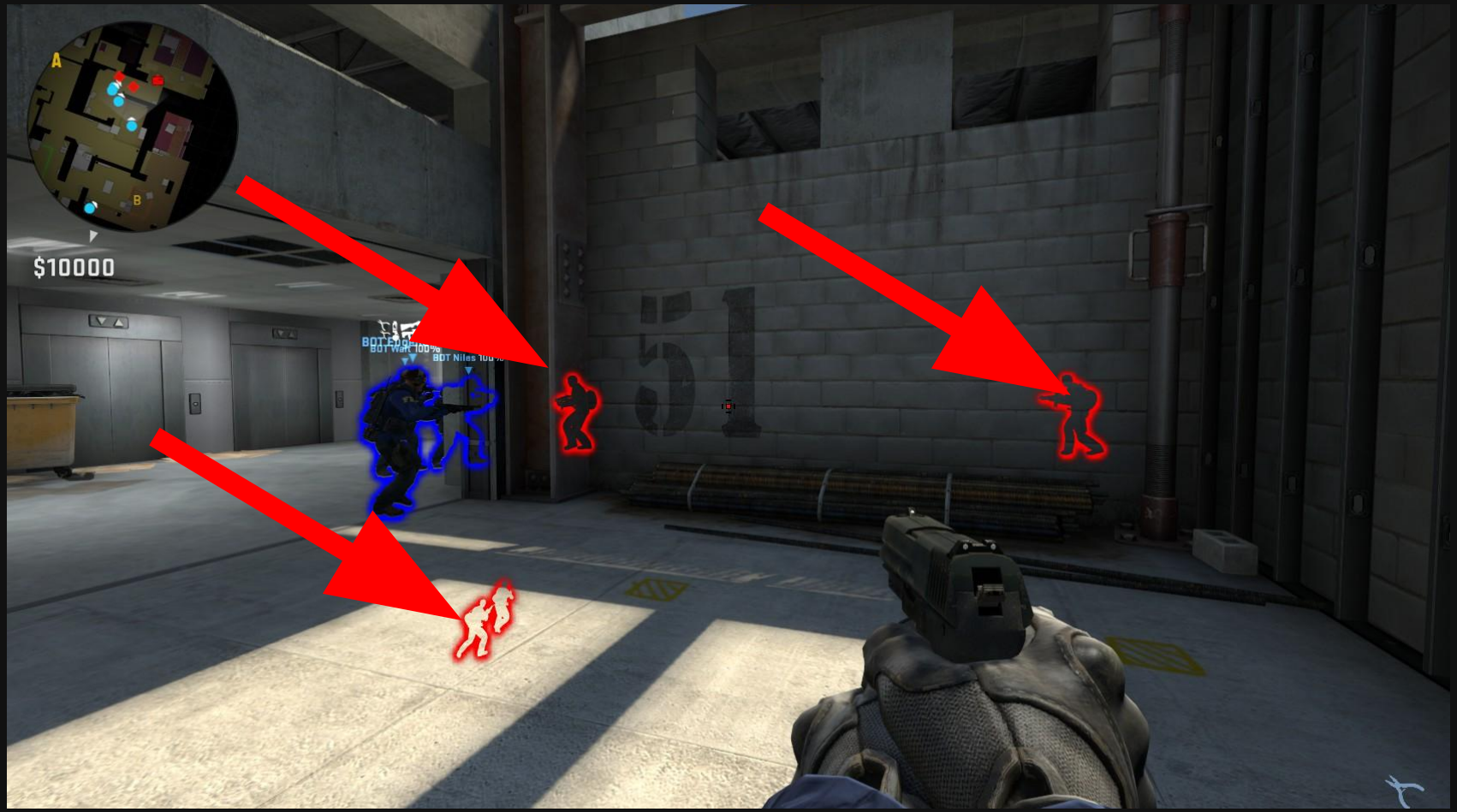
- ➔ Day Job: Security researcher finding new vulnerabilities in various systems, primarily in mobile applications
- ➔ Previously worked in a malware research lab researching methods for signature detection
- ➔ Really enjoy playing competitive game (legitimately!), but always found myself wondering why it was so difficult to prevent cheating

# Overview


- ➔ What is Game Hacking?
- ➔ Internal vs. External Cheats
- ➔ Hardware Cheats
- ➔ Cheat Loaders and Process Injection
- ➔ Process Injection Example
- ➔ AntiCheat Detections
- ➔ Parallel between Anticheat and EDRs

# Game Hacking

Using software to create an advantage beyond normal gameplay



A
B
$10000

# Game Hacking

- This is open source!
- 30,000 lines of code
- Features: Aimbot, Triggerbot, ESP, Cosmetic Changes
- Configs and a robust UI

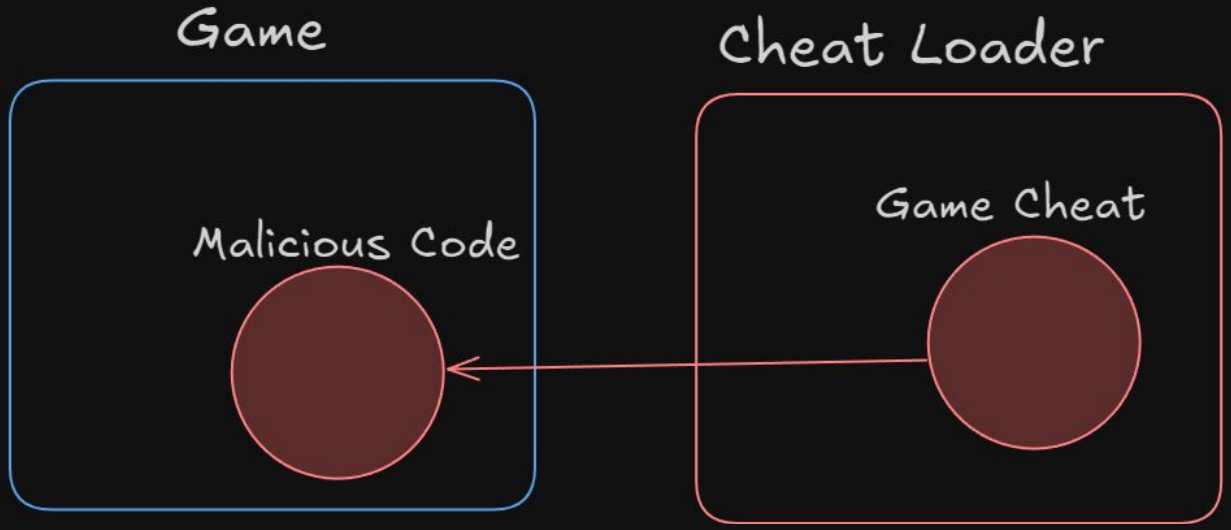
Source: https://github.com/degeneratehyperbola/NEPS
(Disclaimer: CS:GO has been replaced by a newer version, this game is not actively competitive)

# Internal and External Cheats

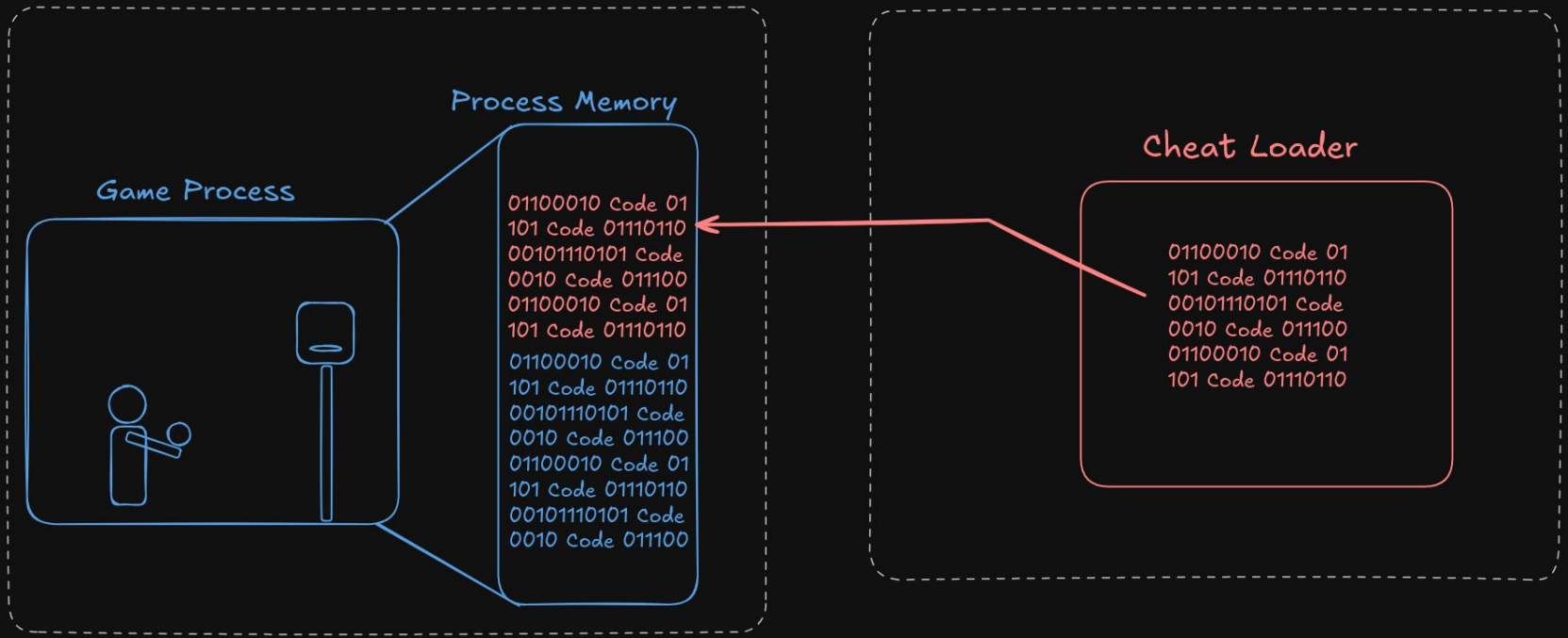
Internal Cheat:
Runs inside the game process. Can directly access game memory and functions. Requires a cheat loader or a technique.

External Cheat:
Runs outside the game process. Reads/writes memory from another process using OS APIs. Less access, but harder to detect.

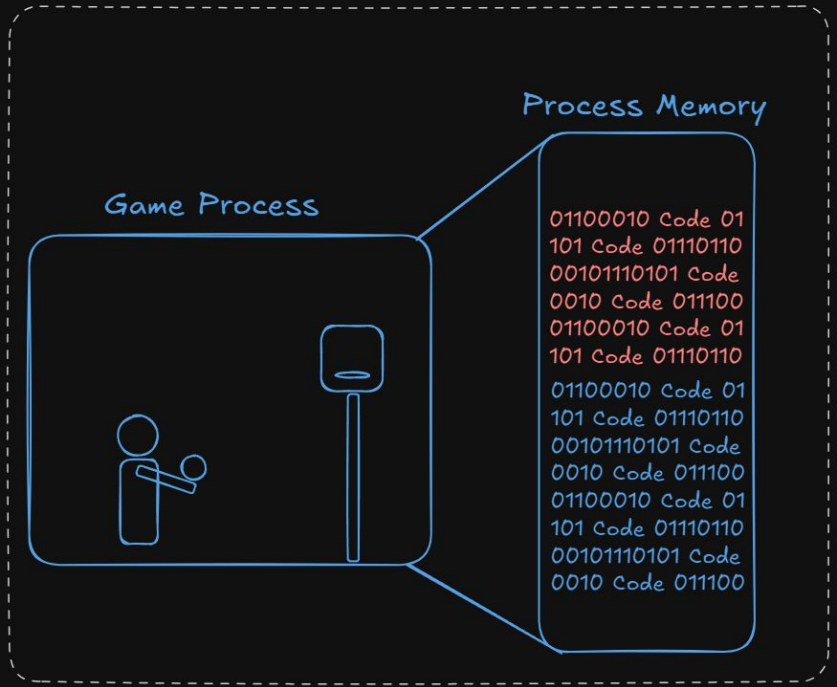
# Internal Cheat High Level



# Internal Cheat - Technical

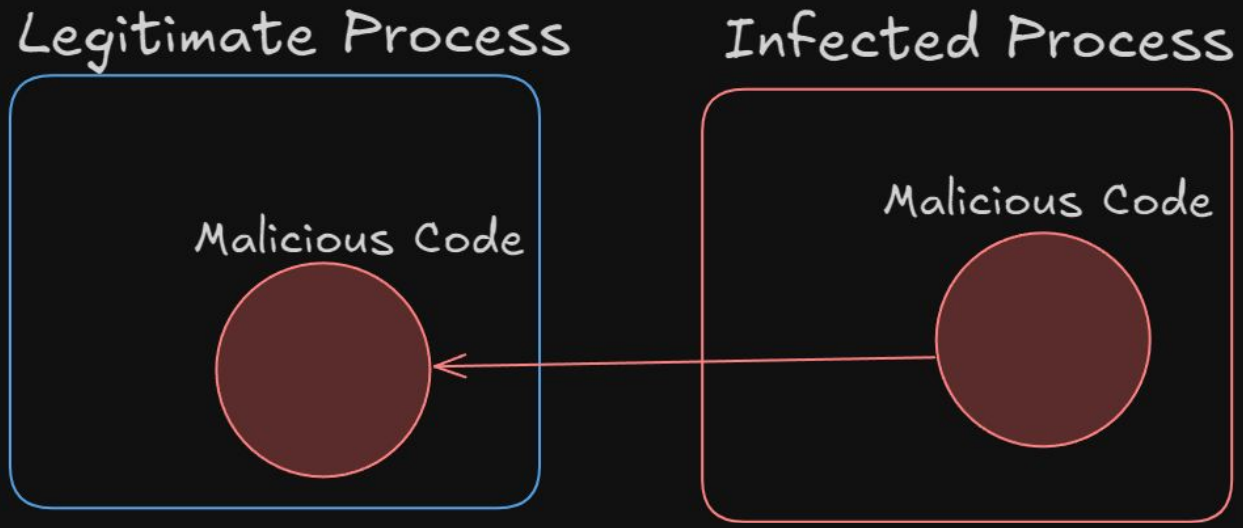


# Internal Cheat Example (Continued)



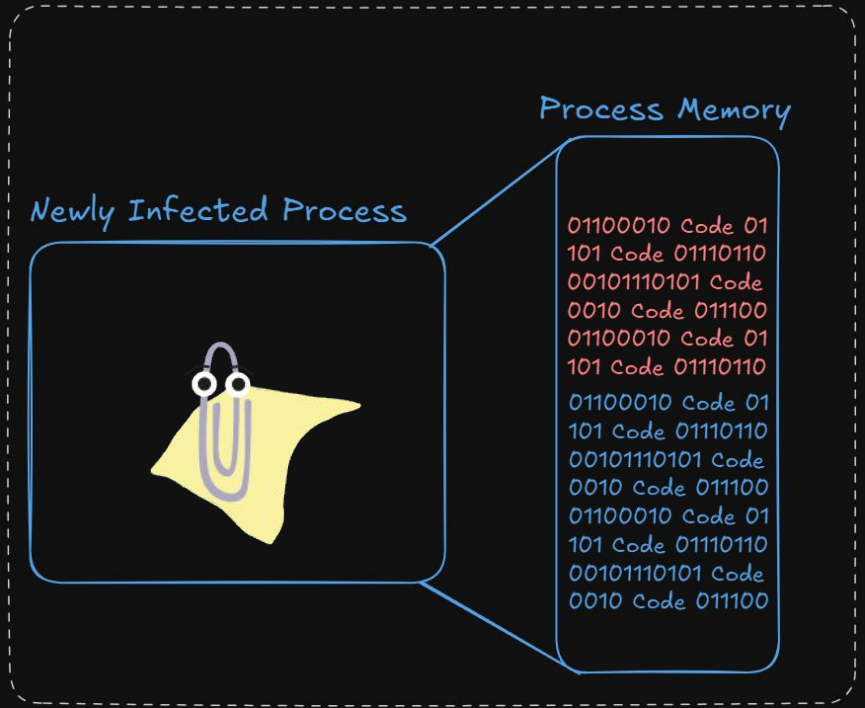

- The cheat now exists as part of the game

# Isn’t this just like Process Migration… Yes.



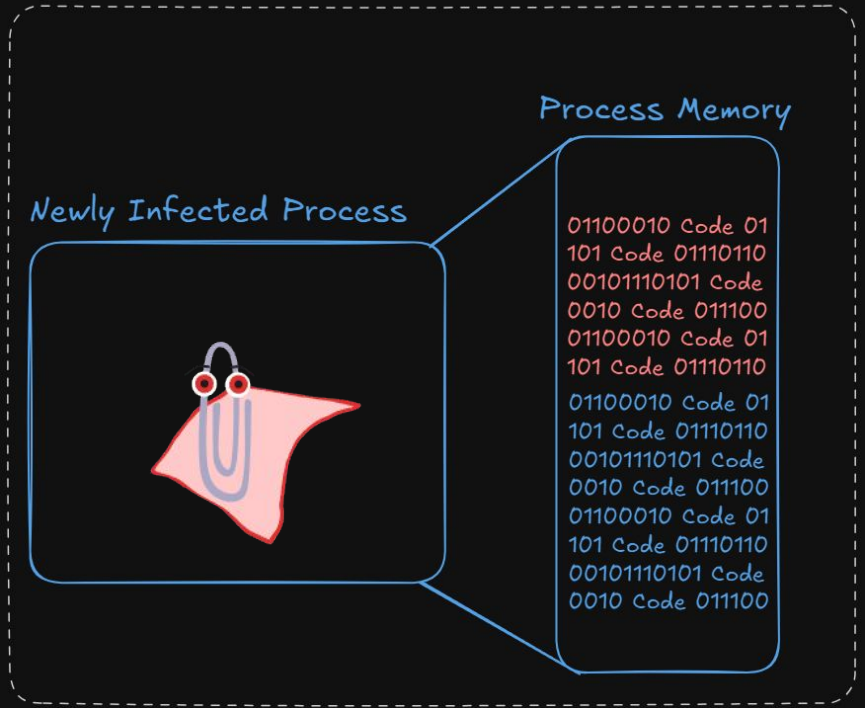
# Process Migration Example

- The malware now exists as part of clippy :(



# Process Migration Example

- The malware now exists as part of clippy :(



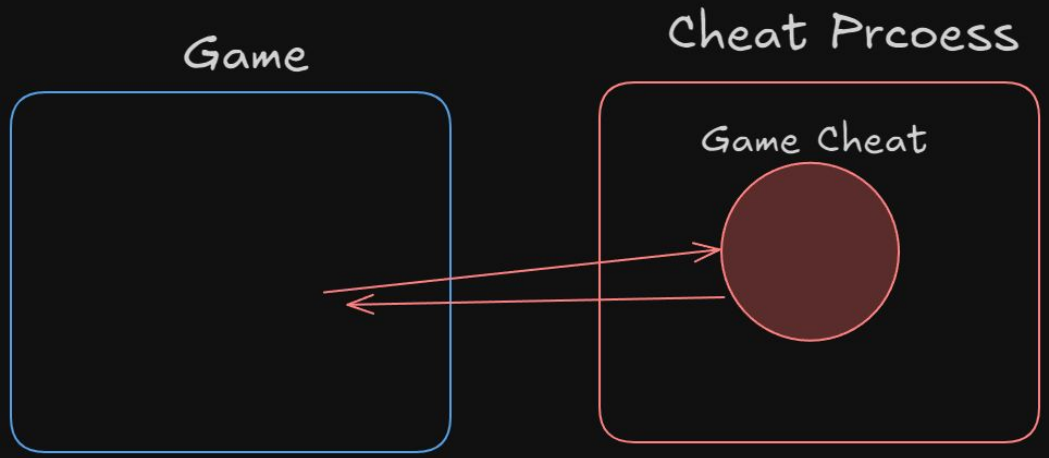
# Cheats use the same techniques as malware!

…and have the same characteristics too

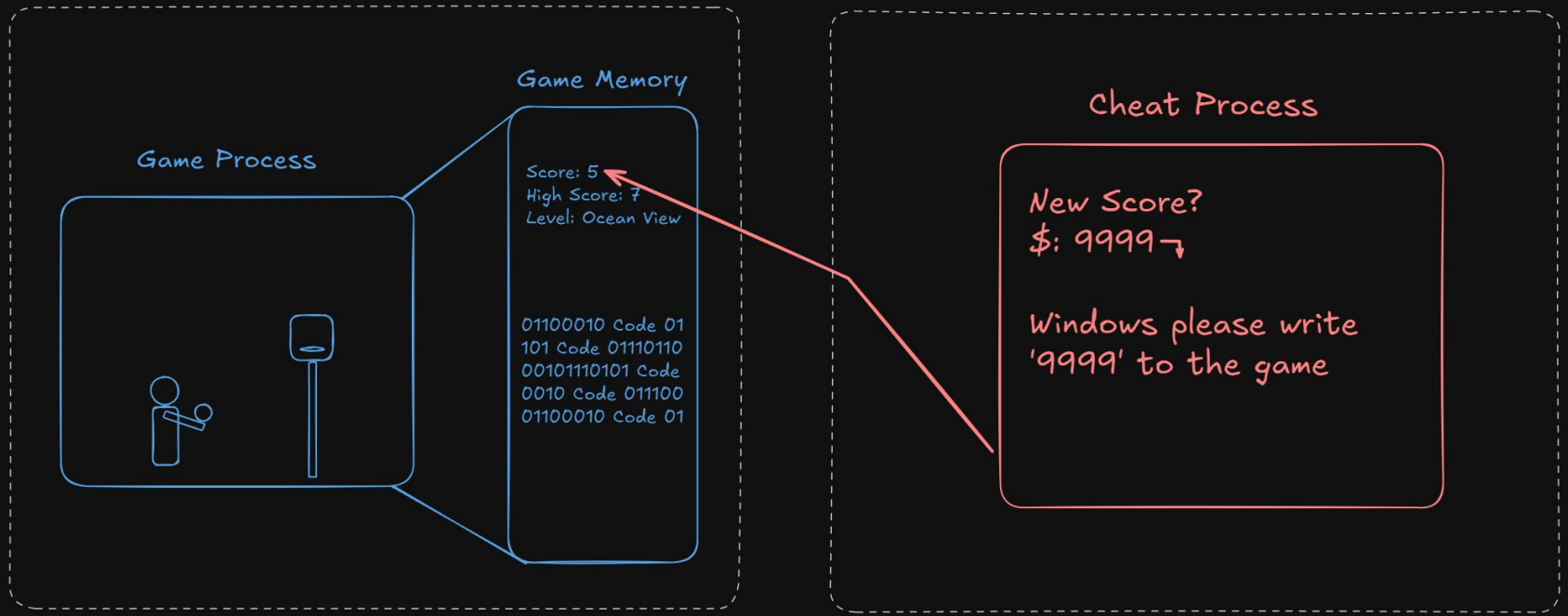
- Stealth is a goal
- Technically Sophisticated
- Cat & Mouse Game
- Often Obfuscated
- Anti-Analysis and Anti-Debug

# External Cheat - High Level

- The cheat operates independently of the game



# External Cheat - Technical

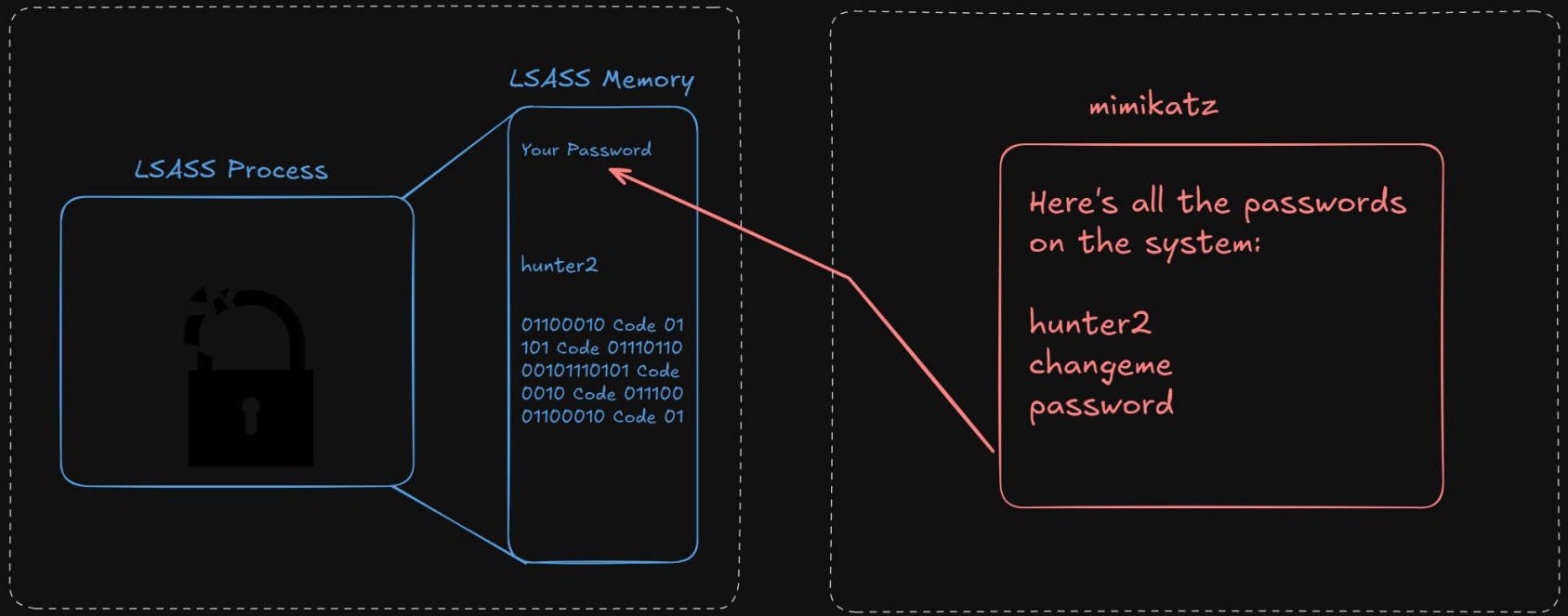


# For Red-Teamers: What is Mimikatz?

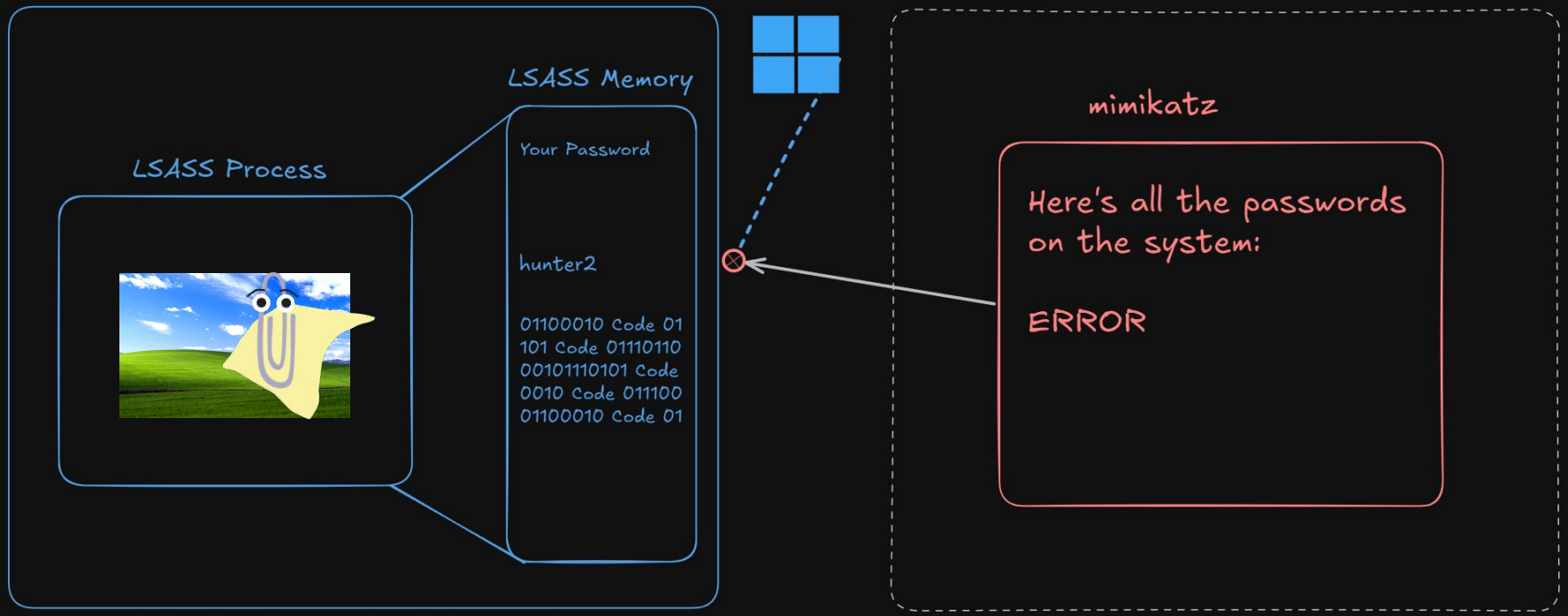
“Mimikatz is a well-known hacktool used to extract Windows passwords in plain-text from memory, [...] and more. This tool is used by red teams and real threat actors alike [...] Mimikatz is often delivered and executed without writing to disk (fileless) in an attempt to avoid detection. “

https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=HackTool:Win32/Mimikatz

# Mimikatz Example



# Mimikatz Example



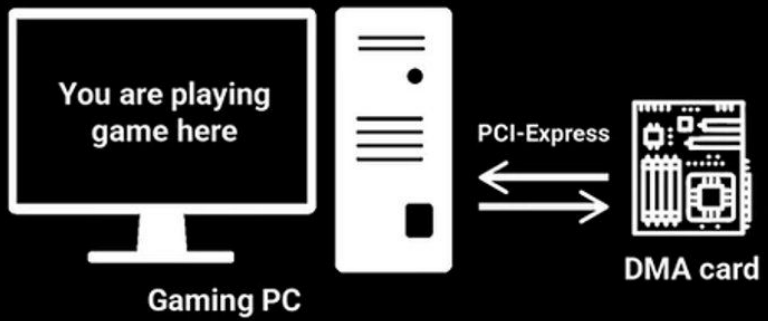
# They’re not so different after all

- ➔ Cheats leverage the same Windows API calls to perform ‘malicious’ actions
- ➔ All provided by Windows, they are legitimate functions
- ➔ There are products created to prevent abuse of the Windows API

# Why is it difficult to detect cheaters?

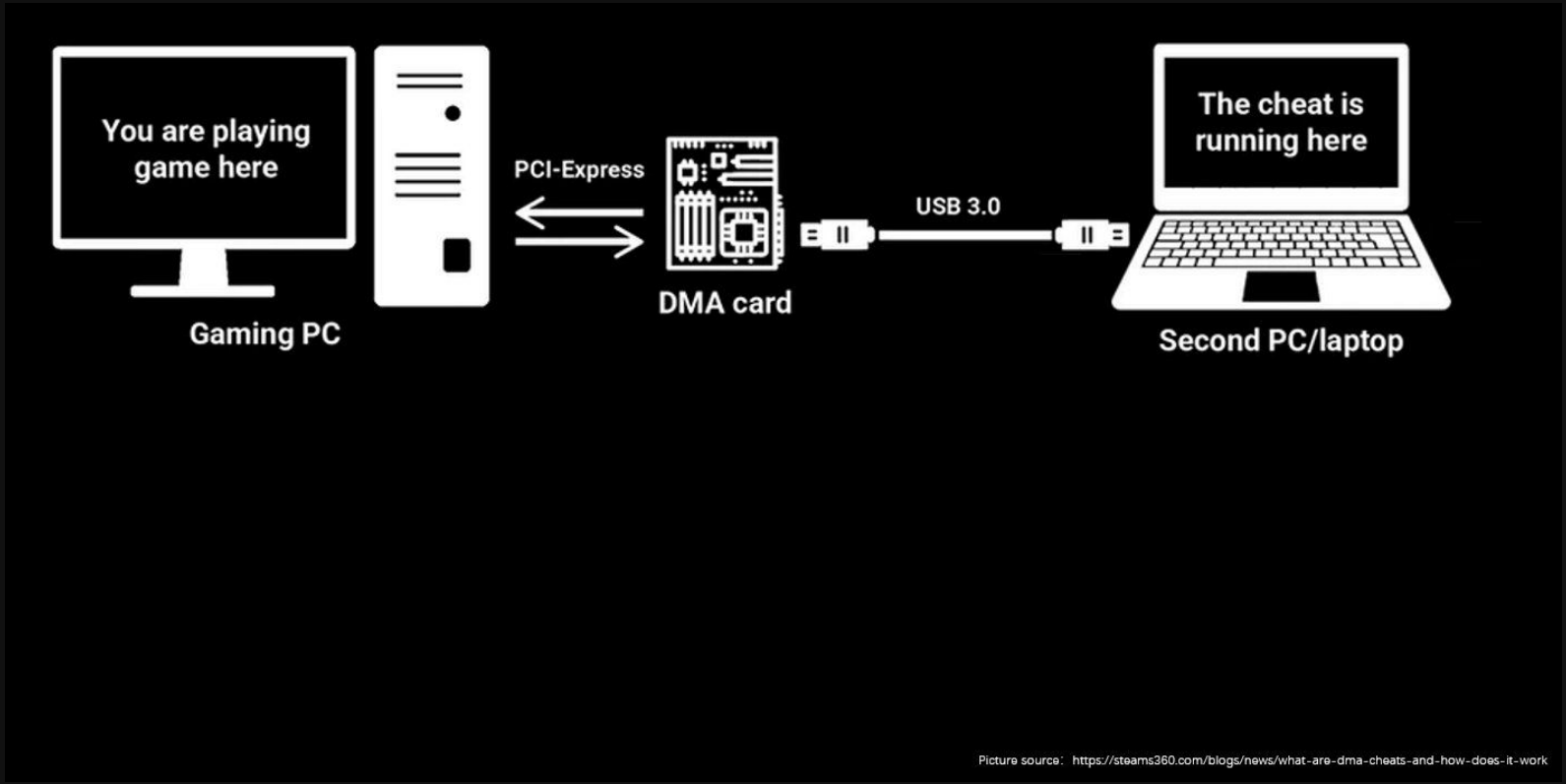

Picture source: https://steams360.com/blogs/news/what-are-dma-cheats-and-how-does-it-work

# Why is it difficult to detect cheaters?

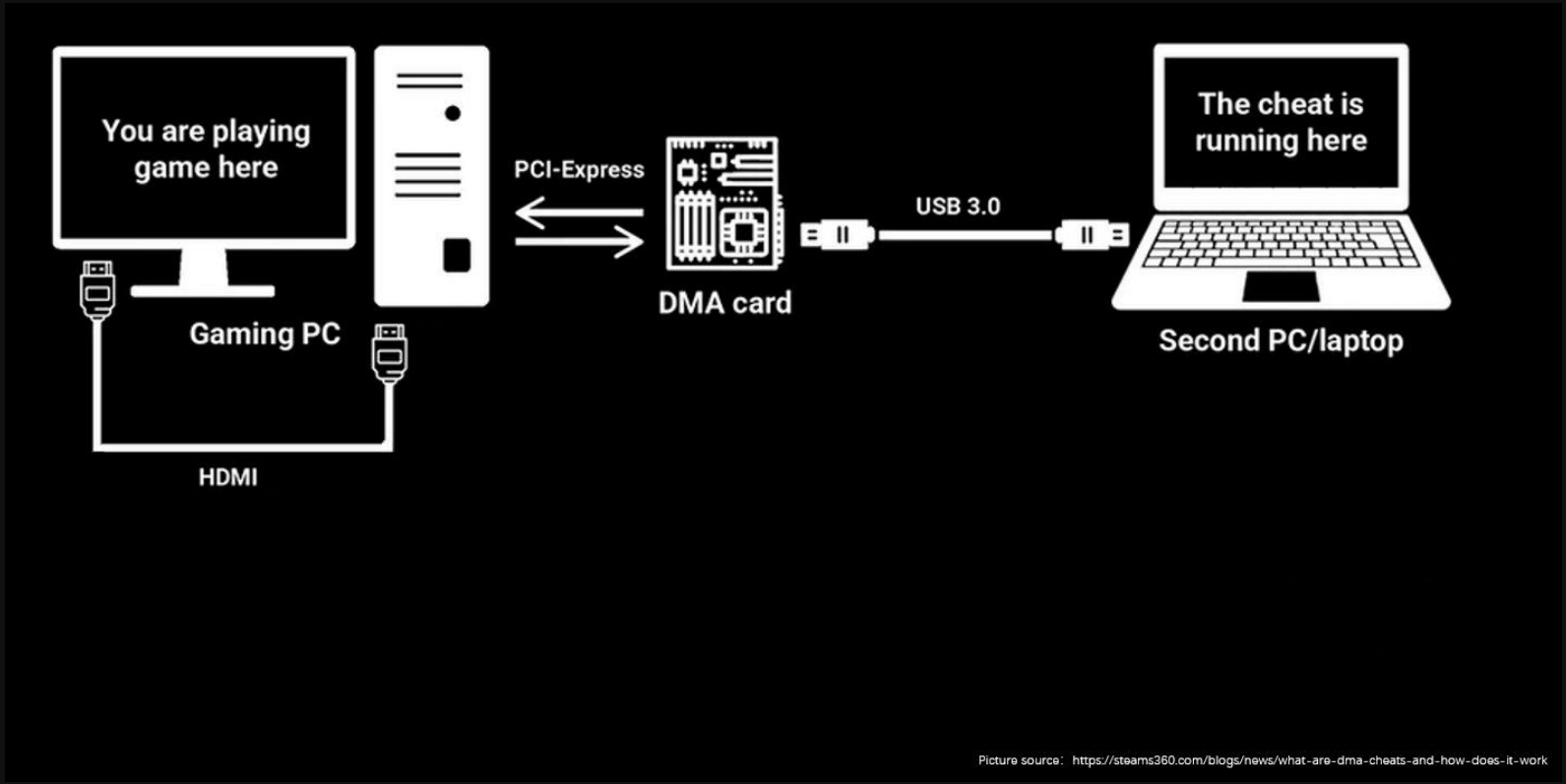



Picture source: https://steams360.com/blogs/news/what-are-dma-cheats-and-how-does-it-work

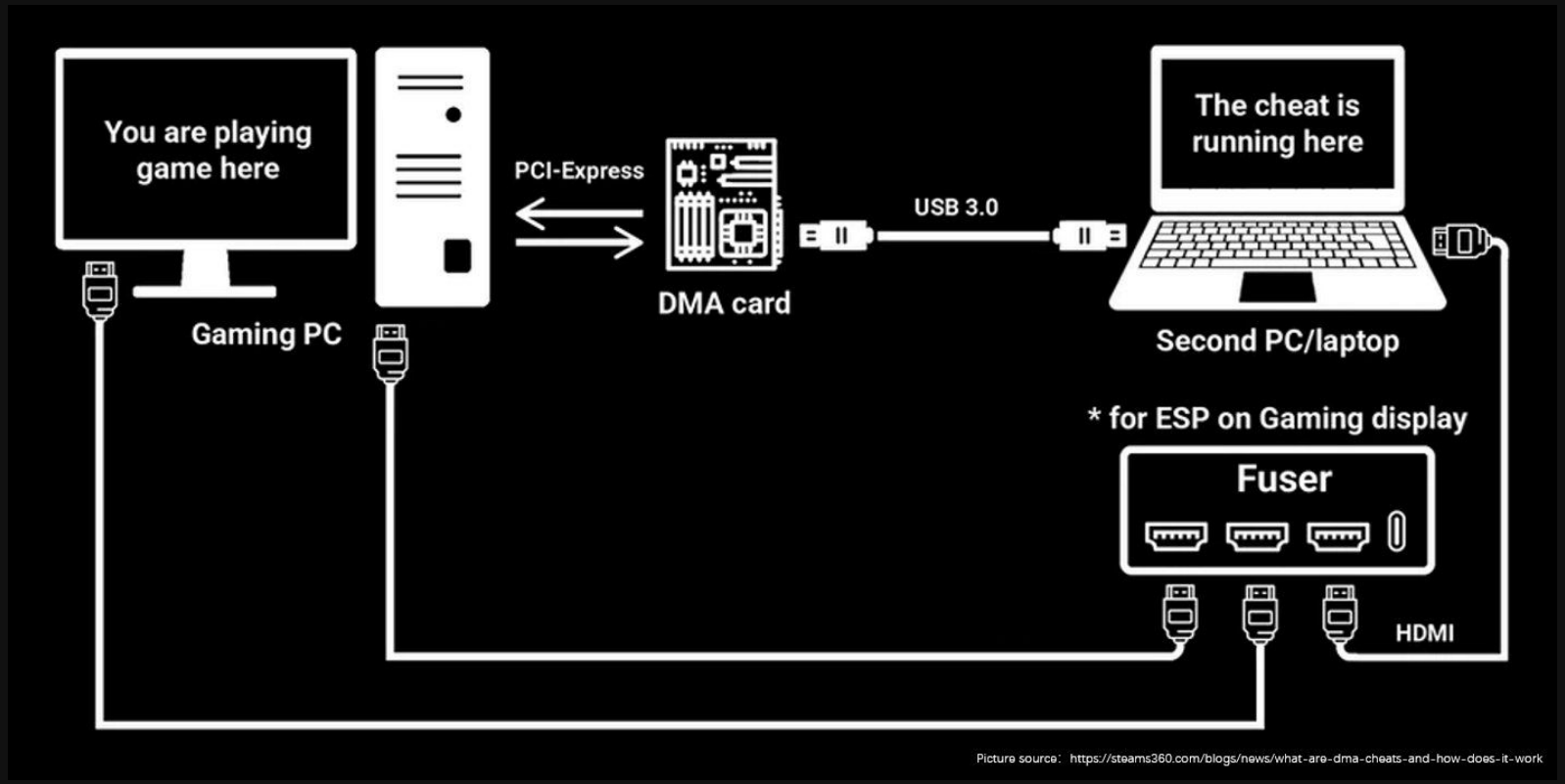
# Why is it difficult to detect cheaters?



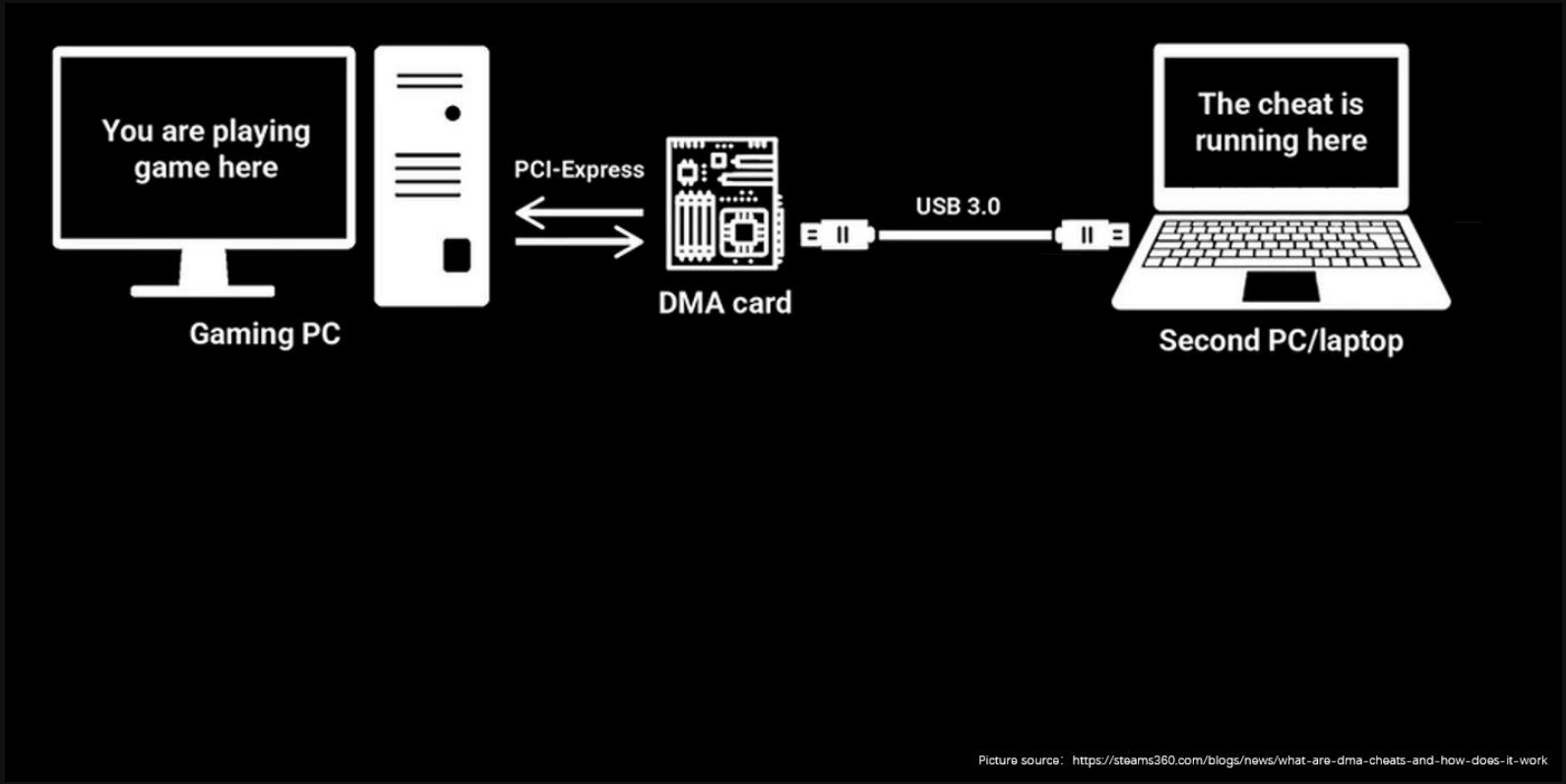
# Why is it difficult to detect cheaters?



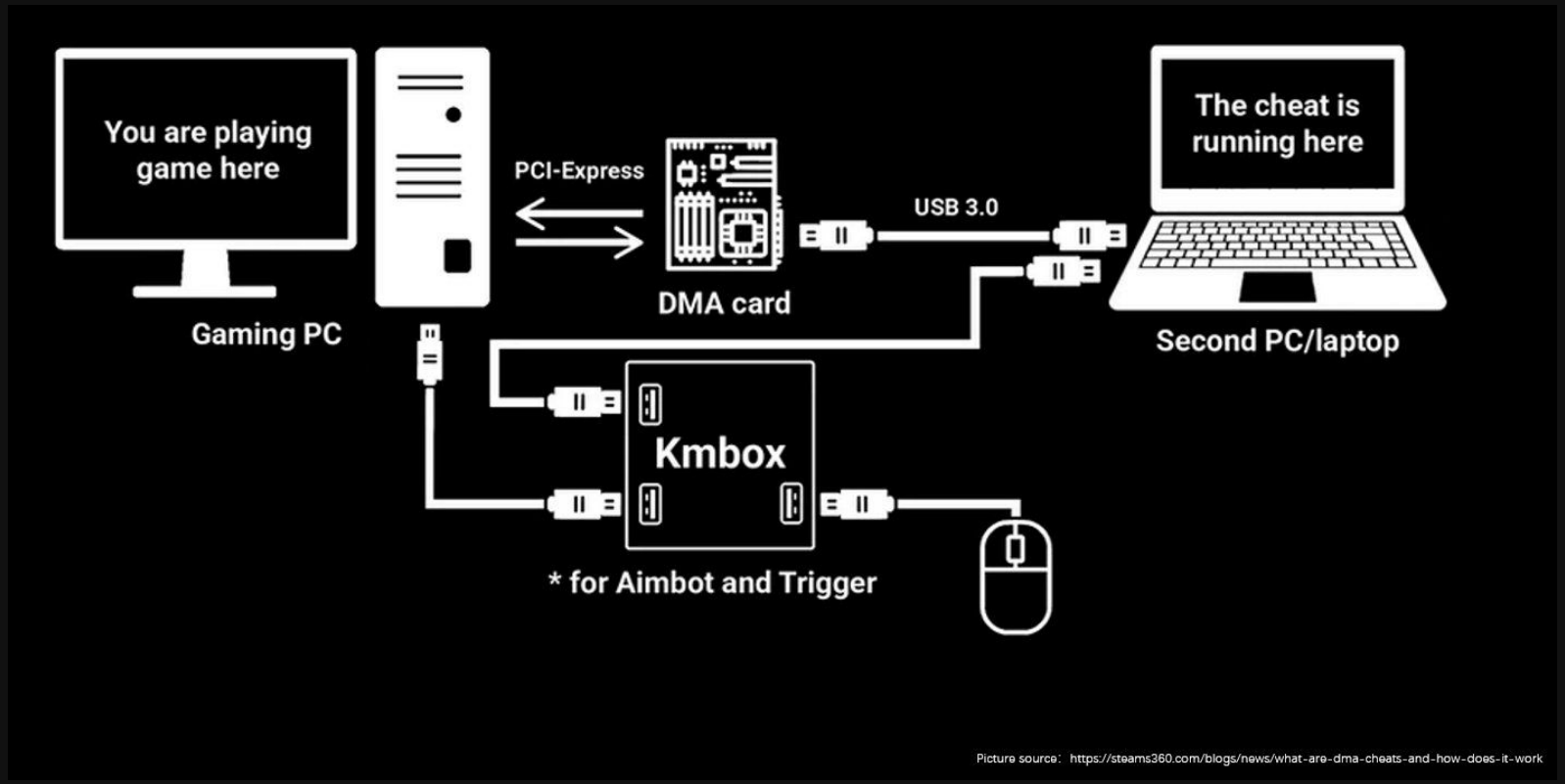
# Why is it difficult to detect cheaters?



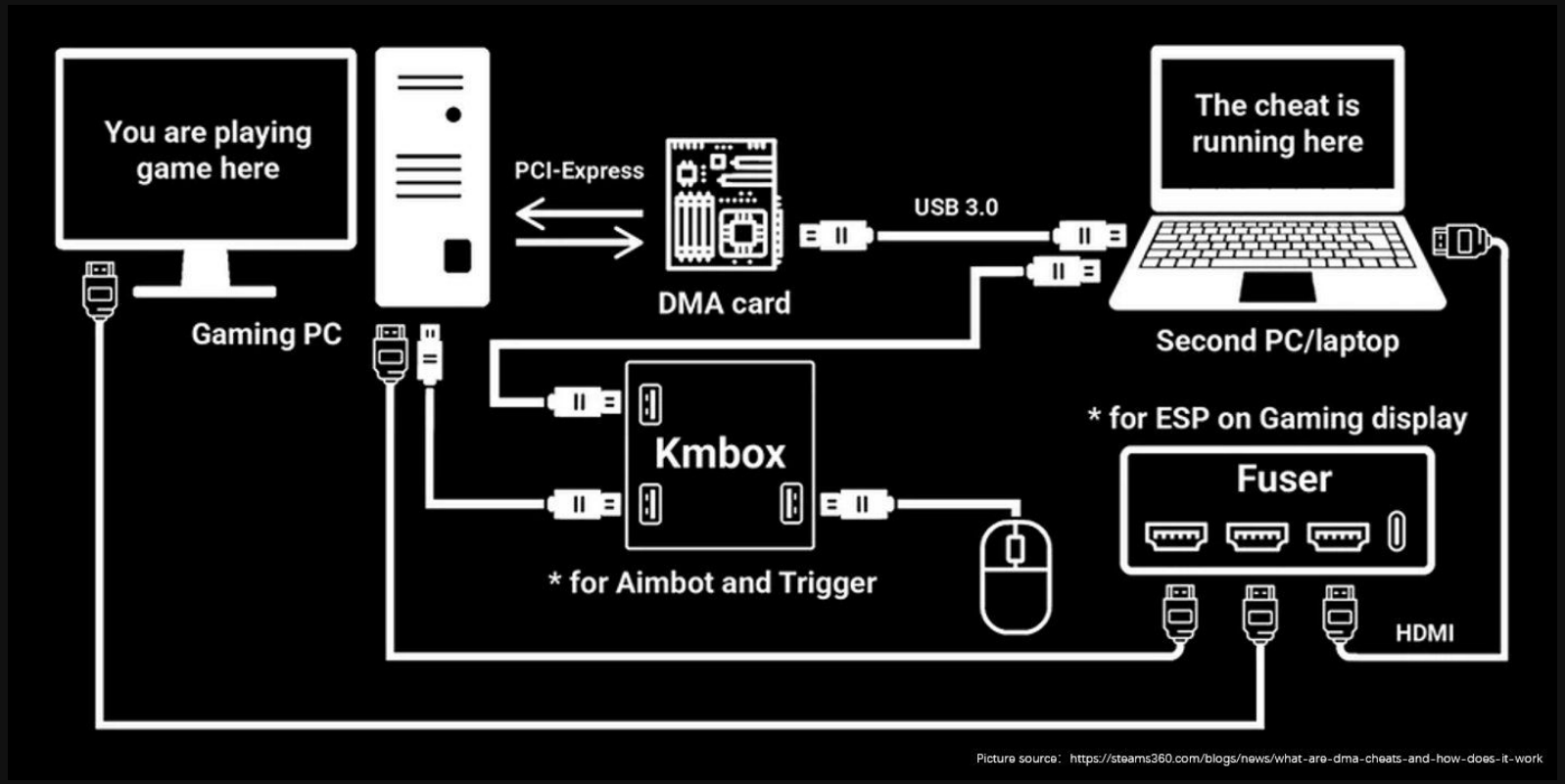
# Why is it difficult to detect cheaters?



# Why is it difficult to detect cheaters?

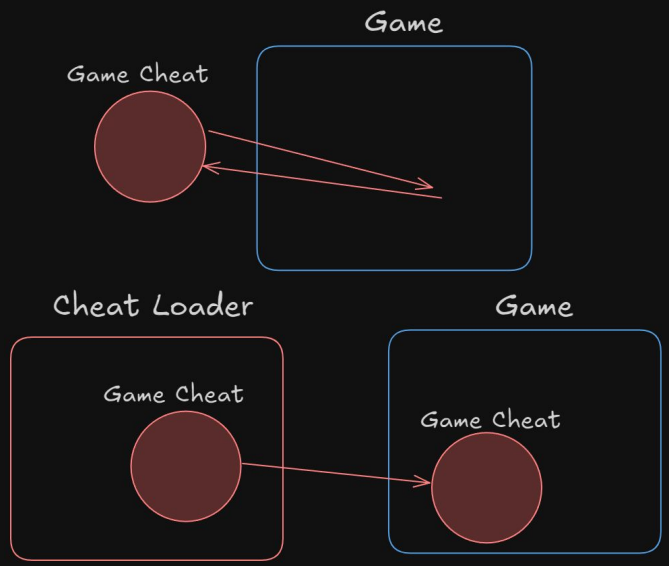


# Why is it difficult to detect cheaters?



# What we've covered

- ➔ Cheat Loaders
- ➔ Cheat Software
- ➔ Internal Cheats
- ➔ External Cheats
- ➔ Malware uses the same techniques
- ➔ Hardware Cheats



# Process Injection

“A method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process′s memory, [and] system/network resources.”

https://attack.mitre.org/techniques/T1055/

# Steps to Perform Process Injection

1. Allocate Memory
2. Write Memory
3. Execute Code
4. ??
5. Profit

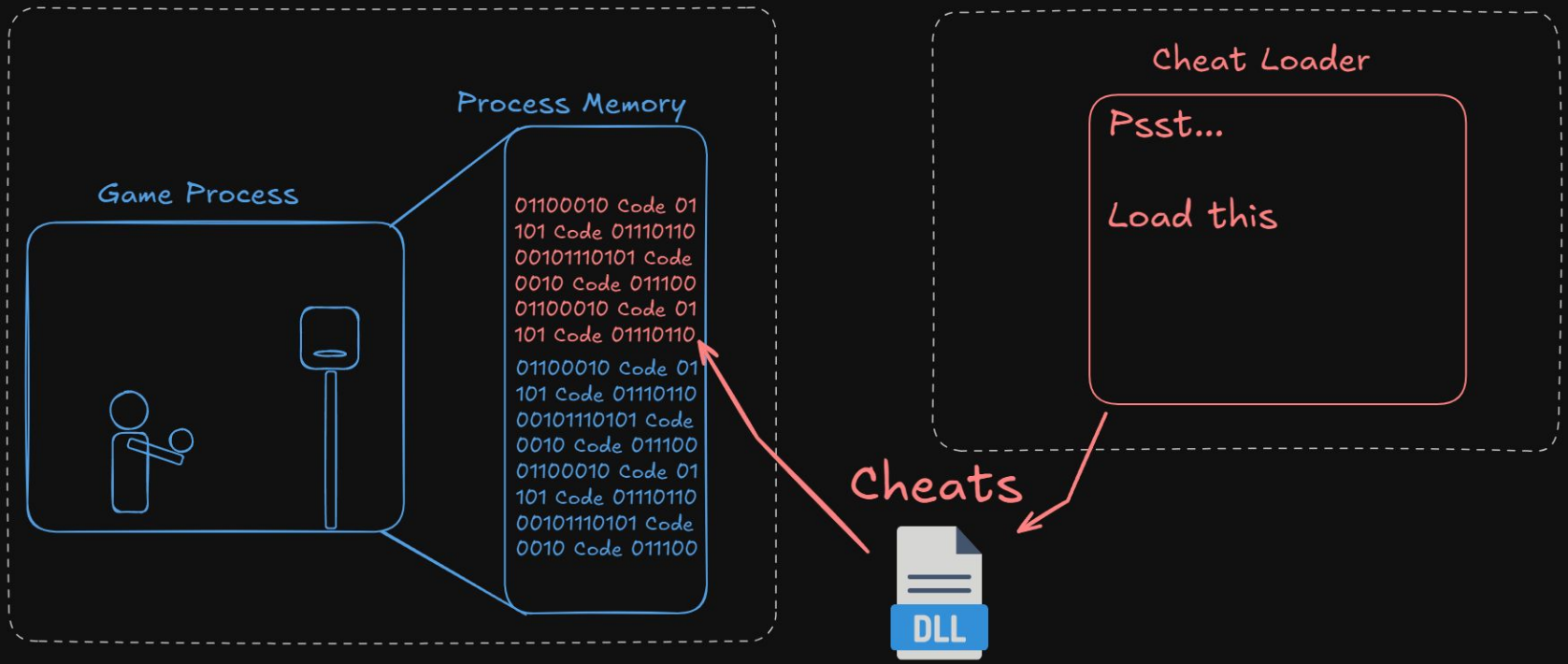
# Endless Techniques

Some common techniques for writing and executing

- DLL Injection
- APC (Asynchronous Procedure Call) Injection
- Thread Execution Hijacking
- Ghost writing
- Stack Bombing
- To name a few


https://i.blackhat.com/USA-19/Thursday/us-19-Kotler-Process-Injection-Techniques-Gotta-Catch-Them-All.pdf

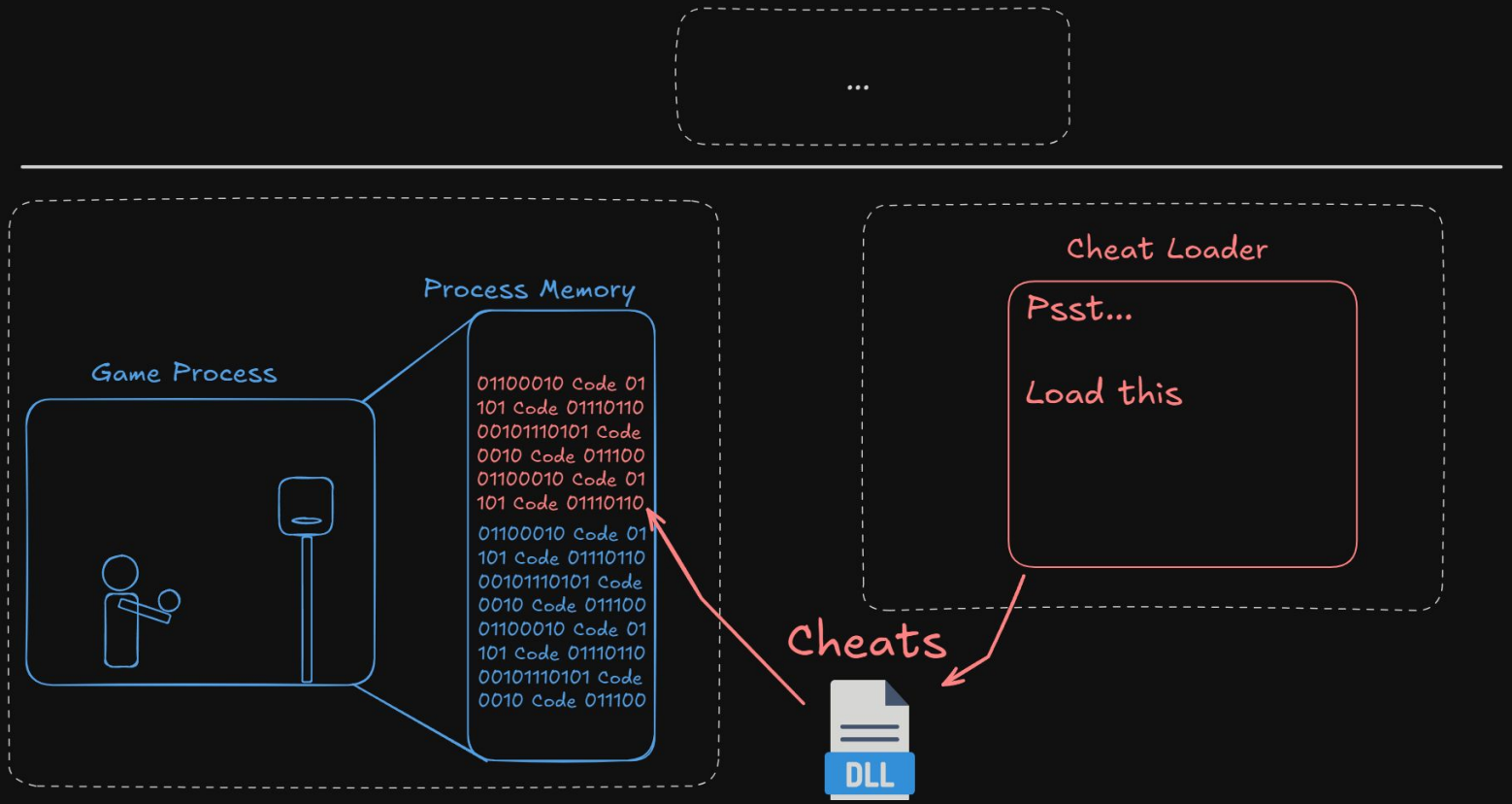
# Process Injection - DLL Injection



# Process Injection - DLL Injection

```
// Open a handle to the target process
HANDLE ph = OpenProcess(PROCESS_ALL_ACCESS, 0, target_pid);

// Allocate new memory in the target process for the DLL path
LPVOID remote = VirtualAllocEx(ph, NULL, MAX_PATH + 1, MEM_COMMIT, PAGE_READWRITE);

// Write `dll_path` to remote inside the target process
WriteProcessMemory(hProcess, remote, dll_path, MAX_PATH + 1, NULL);

// Get the address of LoadLibraryA. This is a function pointer.
LPVOID LoadLibraryA = GetProcAddress(GetModuleHandleA("kernel32.dll"), "LoadLibraryA");

// Create a new thread in the target process
// We are executing LoadLibraryA(dll_path) in the target process in a new Thread
CreateRemoteThread(ph, nullptr, 0, LoadLibraryA, remote , NULL, NULL);
```

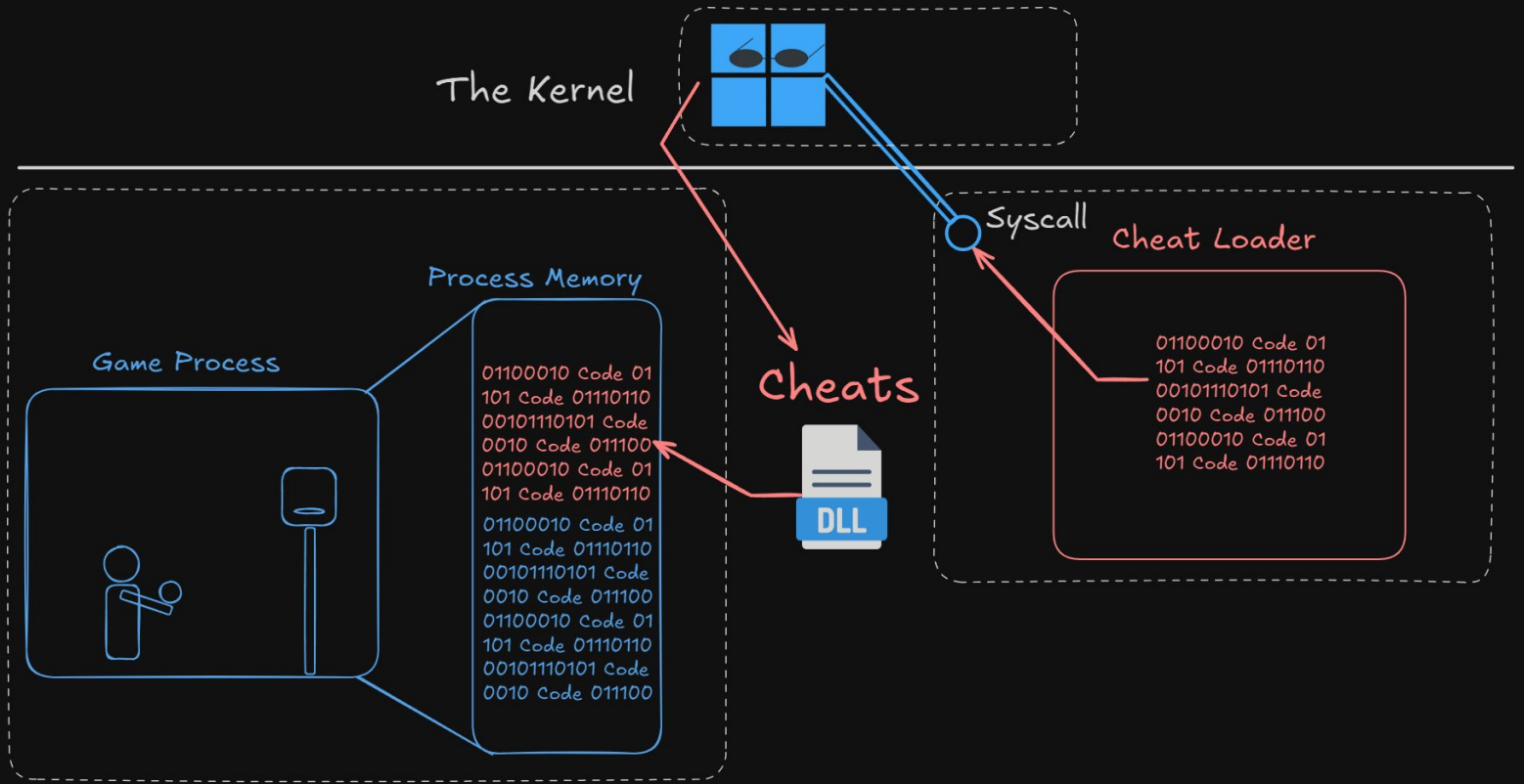
# DLL Injection



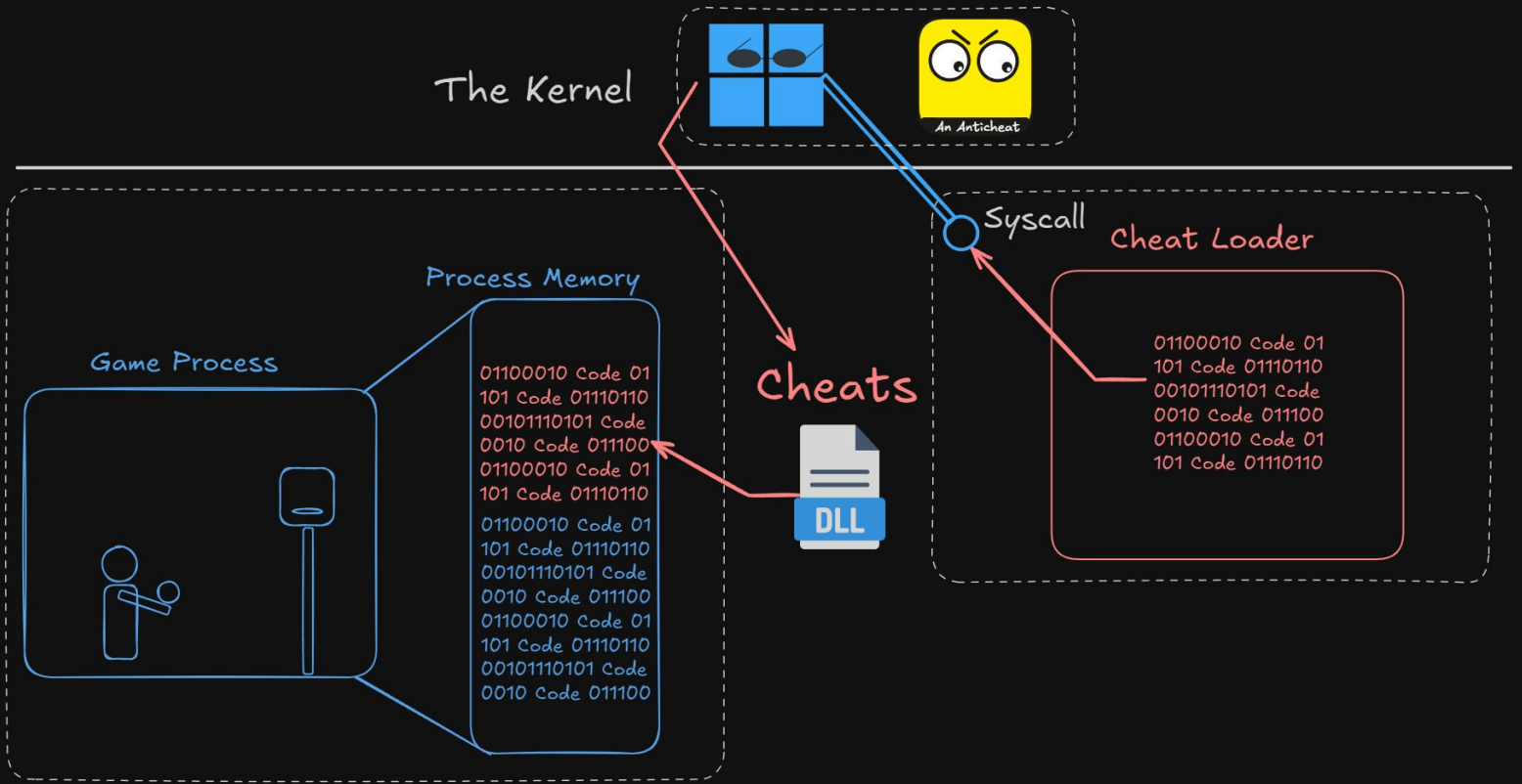
# System Calls

- A CPU instruction that allows the transition from userspace to kernelspace
- Modifies a register and changes the address currently being executed
- Enables access to privileged instructions and memory
- Essentially an API for the operating system

# DLL Injection



# The Anti-Cheat is Watching



# Anti-cheats and EDRs


What is an AntiCheat?

An Anti-Cheat detects any modifications made to a computer that can be used for cheating. If something is detected by the Anti-Cheat's scan, the modification is sent for review, and prevention mechanisms are employed, which may result in the user receiving a ban from the game.

https://pk.ign.com/counter-strike-global-offensive/241497/news/what-is-an-anti-cheat-all-you-need-to-know

# Anti-cheats and EDRs

What is an EDR?

Endpoint Detection and Response (EDR) is a security solution that continuously monitors devices to detect and respond to cyber threats...

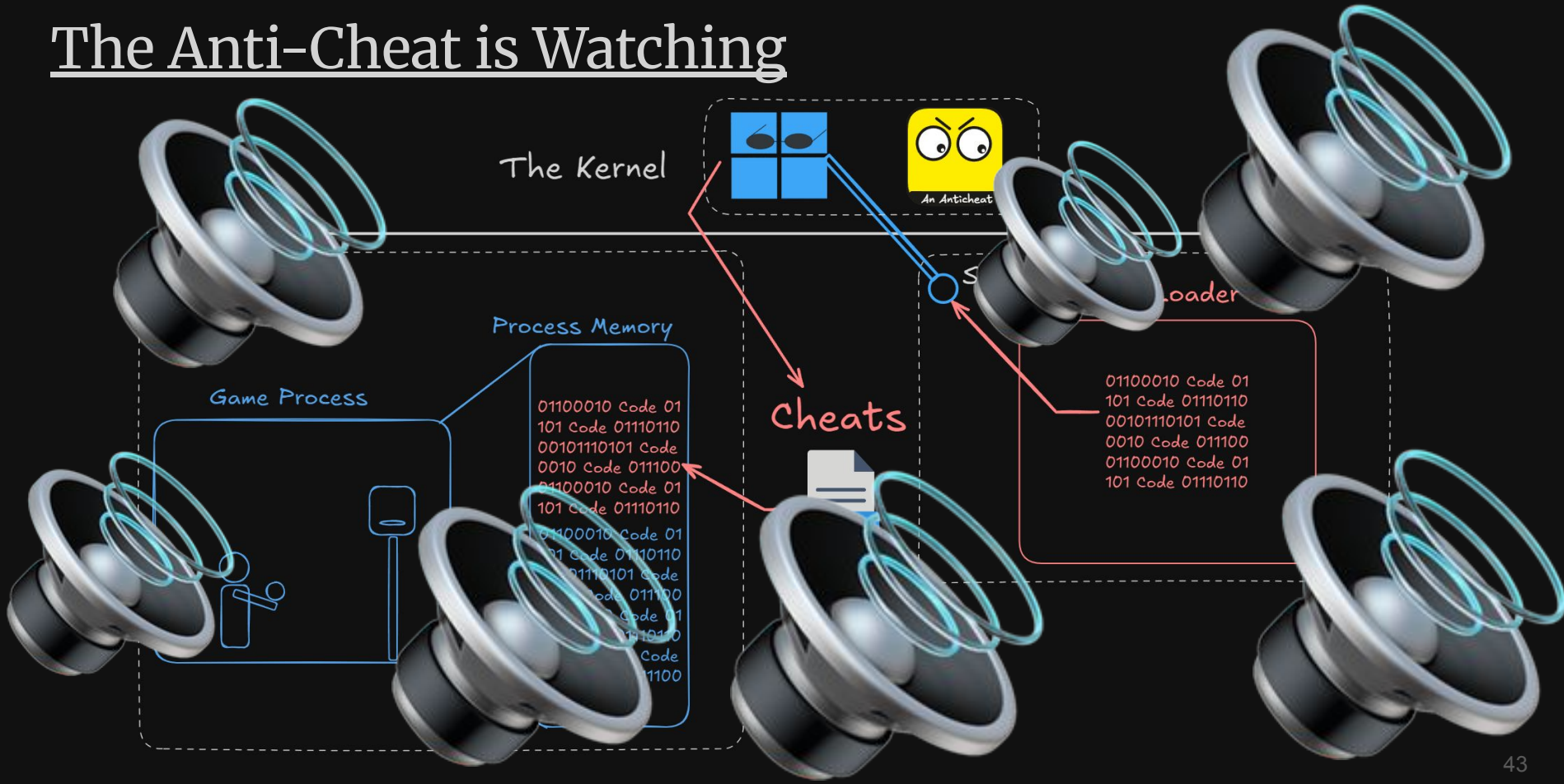
EDRs record the activities and events taking place [...], providing security teams with visibility to uncover incidents that would otherwise remain invisible.

https://www.crowdstrike.com/en-us/cybersecurity-101/endpoint-security/endpoint-detection-and-response-edr/

# Anti-Cheat Detection Mechanics

- ➔ API Monitoring (Kernel and/or Userspace)
- ➔ Signature Detection
  - ◆ In-memory
  - ◆ Filesystem
- ➔ Heuristics
- ➔ Memory Scanning -> New or unknown modules (DLLs)
- ➔ New or unknown threads

# The Anti-Cheat is Watching



# Detection of Process Injection – DLL Injection

```
HANDLE ph = OpenProcess(PROCESS_ALL_ACCESS, 0, target_pid);
LPVOID remote = VirtualAllocEx(ph, NULL, MAX_PATH + 1, MEM_COMMIT, PAGE_READWRITE);
WriteProcessMemory(hProcess, remote, dll_path, MAX_PATH + 1, NULL);
LPVOID LoadLibraryA = GetProcAddress(GetModuleHandleA("kernel32.dll"), "LoadLibraryA");
CreateRemoteThread(ph, nullptr, 0, LoadLibraryA, remote , NULL, NULL);
```

- ➔ Suspicious call to OpenProcess
- ➔ Suspicious call to WriteProcessMemory
- ➔ Common heuristic sequence of OpenProcess -> VirtualAllocEx -> WriteProcessMemory -> CreateRemoteThread
- ➔ Unexpected new thread
- ➔ Suspicious entry point for thread
- ➔ DLL may be missing on disk when scanning memory later

# What we've covered

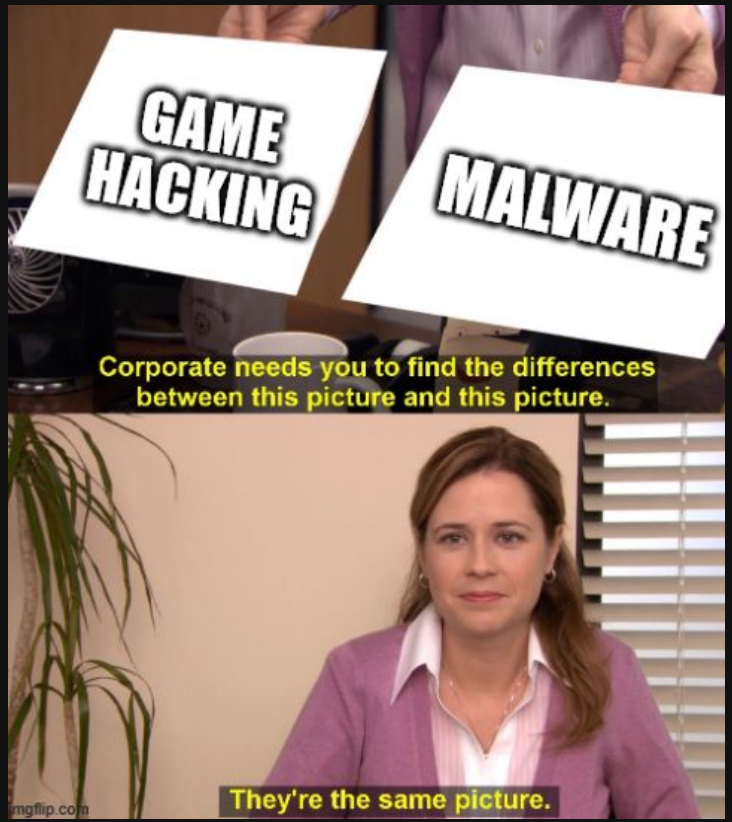
- ➔ Earlier: types of cheats
- ➔ Process Injection
- ➔ Anti-Cheats
- ➔ Endpoint Detection and Response

# Red Teamers Takeaways

- Use game hacking techniques for Red Team operations
- Use classic game hacking tools like Cheat Engine to find offsets and ReClass for memory structures
- Incredible amount of knowledge on game hacking forums
  - Well-Documented Code
  - Signature Bypassing

Game hacking teaches us that it’s just a different side of the same coin.

Thank you


GAME HACKING
MALWARE
Corporate needs you to find the differences between this picture and this picture.
They're the same picture.
imgflip.com


Questions?
GAMEHACKING
VILLAGE
V 1.0
DEFCON

# Learn More

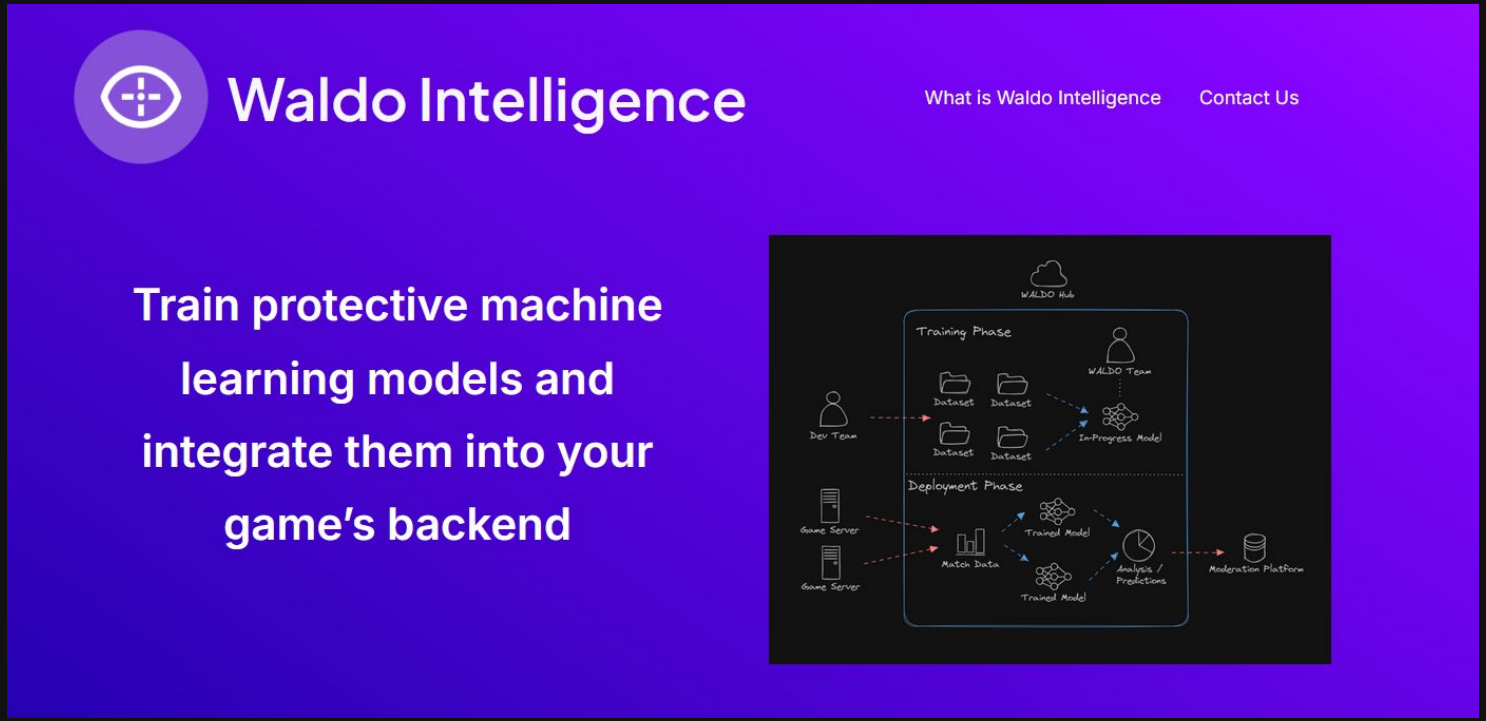

- Game Hacking Village @ DEF CON

  Website: Gamehacking.gg
- Unknown Cheats (https://www.unknowncheats.me/forum/index.php)
- Guided Hacking (https://guidedhacking.com/)
  (Paid sponsor of the Game Hacking Village)

# AI Anticheat

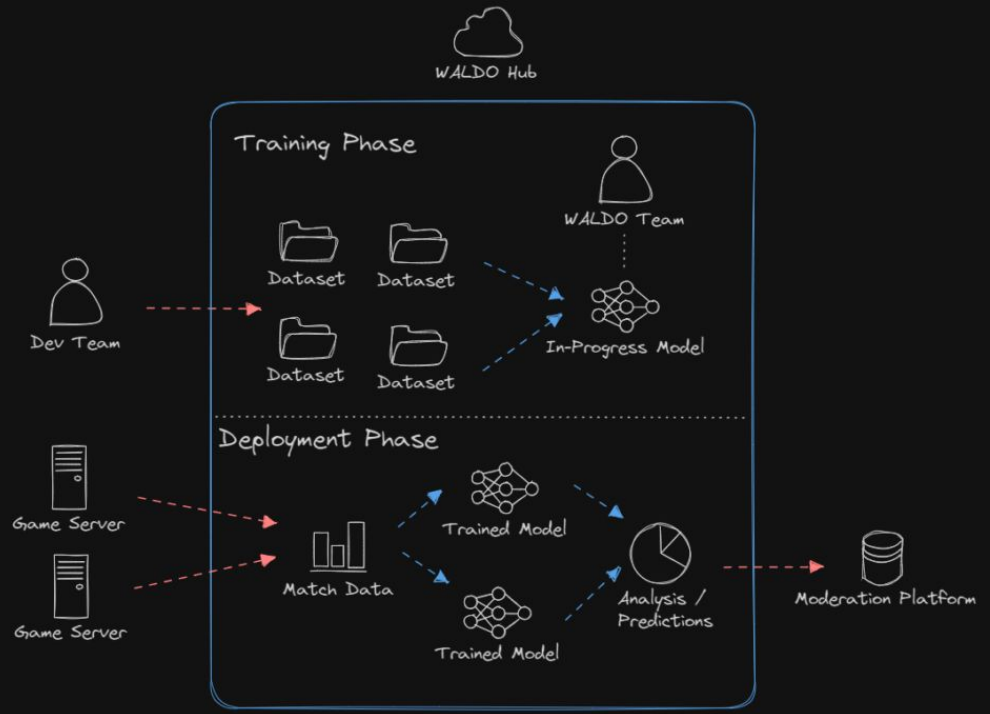

Source: Game Developers Conference (GDC) 2018
Image from https://www.youtube.com/watch?v=SnRgW54EWwA covered by 3kliksphilip

# AI Anticheat




Source: https://waldointelligence.com/ (Not affiliated)

# AI Anticheat




Source: https://waldointelligence.com/ (Not affiliated)

# Kernel Mode Anticheats

- ➔ Ongoing debate for EDRs and Anticheats
- ➔ I think they are necessary as it's the only way there's a chance of catching cheaters
- ➔ However, I also use a separate computer for just gaming
- ➔ User-space anti-cheats will still run as Administrator or SYSTEM and have access to everything you'd consider important

## 2024 CrowdStrike-related IT outages

33 languages

Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

(Redirected from 2024 CrowdStrike incident)

On 19 July 2024, the American cybersecurity company CrowdStrike distributed a faulty update to its Falcon Sensor security software that caused widespread problems with Microsoft Windows computers running the software. As a result, roughly 8.5 million systems crashed and were unable to properly restart[1] in what has been called the largest outage in the history of information technology[2] and "historic in scale".[3]

The outage disrupted daily life, businesses, and governments around the world. Many industries were affected—airlines, airports, banks, hotels, hospitals, manufacturing, stock markets, broadcasting, gas stations, retail stores, and governmental services, such as emergency services and websites.[4][5] The worldwide financial damage has been estimated to be at least US$10 billion.[6]

Within hours, the error was discovered and a fix was released,[7] but because many affected computers had to be fixed manually,[8] outages continued to linger on many services.[9][10]

### Background [edit]

CrowdStrike produces a suite of security software products for businesses, designed to protect computers from cyberattacks. Falcon, CrowdStrike's endpoint detection and response agent, works at the operating system kernel level on individual computers to detect and prevent threats.[11] Patches are routinely distributed by CrowdStrike to its clients to enable their computers to address new threats.[12]

**2024 CrowdStrike-related IT outages**

Multiple blue screens of death caused by a faulty software update on baggage carousels at LaGuardia Airport, New York City

| | |
|---|---|
| **Date** | 19 July 2024; 12 months ago |
| **Location** | Worldwide |
| **Type** | IT outage, computer crash |
| **Cause** | Faulty CrowdStrike software update |
| **Outcome** | ~8.5 million Microsoft Windows operating systems crashed worldwide, causing global disruption of critical services |