Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/5/20230525

A never ending warfare: Fighting cheaters in online video game

Alexander Zhou

Sentinel Secondary School, West Vancouver, V7S 2R2, Canada

zhoualexander9@gmail.com

Abstract. As the gaming industry gradually expands, more and more people begin taking notice of the industry. Though with blossom, problems emerge. Many people don't want to take the time to master a video game, so some will then resort to cheating. This is terrible news for developers as it ruins their reputation and player base. This paper will lay the groundwork for anyone interested in the industry on what has already been done to fight cheaters, as little research was conducted. This paper, will introduce the modern online connection architecture, categorize the most common cheats, and most importantly, introduce modern-day anti-cheat methods. The anti-cheat methods will be analyzed on their effectiveness on different online connection architecture and the type of cheat it works against. Lastly, the paper will also introduce the idea of kernel-level and its impact on anti-cheat.

Keywords: Cyber security, online connection, anti-cheat, kernel-level, video games.

1. Introduction

Ever since the existence of video games, video games have started to become an integral part of many people's lives. It is a source of joy and happiness, for the most part - some games are designed to be extremely challenging, to make the player feel frustrated, to give them the urge to throw their computer out of the window. Consequently, some people will take the easy path - cheating - they will either try to find unintentional exploits in the game, or just outright use third party software to gain an unfair advantage. This is mostly fine in single player games - some games even have cheat codes to allow everyone a chance to enjoy the game, or to give players further control to the game. though this is not fine in multiplayer games. It is quite evident: everyone starts at the same point, but you use cheats to gain an unfair advantage. For example getting a really good item in a MMORPG (Massively Multiplayer Online Role-Playing Game), aimbotting in shooter games(cheat that makes the player's aim to be at an inhumanly possible level), etc. This can potentially cause massive frustration amongst the community, then players will start quitting, consequently hurting the game.

Moreover, with the ever expanding competitive esports scene, it is important to develop a robust anti-cheat system to ensure the competitive integrity. From a practical perspective, the constant appearance of cheaters will hurt the name of the game, whether the players were compensated eventually or not; and the name of the game is often the make or break factor for a player. A terrible anti-cheat can put the entire game at stake. A great example showing the effects of a successful anti-cheat system is the game VALORANT, it is a first person shooter game developed by game publisher "Riot Games". They utilise a custom-made game security software that uses kernel mode driver, we

DOI: 10.54254/2755-2721/5/20230525

will get into the technical terms later [1]. Although it is impossible to get a grasp of how many total cheaters are in the game and how many of them are banned, there are some statistics and player feedback that can be used to see how effective it is. First of all, only about 0.3% of the player base got banned for cheating, and out of the 0.3%, 58% were detected as cheaters before they even got reported - the machines are doing more than the player [2]. The community feedback was also quite positive - it is a rare sight to see complaints of cheater problems on the social media about VALORANT, compared to similar games like Call of Duty. There aren't many statistics that are publicly released, so this mostly has to be based on player experiences.

How does Riot Vanguard fight against cheaters? What kind of ingenious methods does it take to put these rule-breakers in their place? To truly understand, we will need to get a bit more technical, which is the purpose of this literary review: outline different types of common cheating methods and anti-cheat methods. Although for security and privacy reasons, anti-cheat developers and companies rarely disclose much information about the functionality of their anti-cheat; thus, this paper will be based on public research.

2. Online connection architecture

Modern online multiplayer consists of three types of connection architecture: Client to Server, Peer to Peer, and Cloud. This matters because anti-cheat (and cheat) methods are divided into client side and server side. These three architectures are essentially the three levels of utilisation of client and server side solutions. More specifically, due to the nature of how these connection architectures differ, some anti-cheat methods may not work on others; it's more so that the server-side anti-cheat methods wouldn't work on P2P(peer to peer) architecture, or that client-side anti-cheat isn't needed in cloud gaming. So what do these words actually mean?

2.1. Client to server

Client to Server is when the user downloads the game file onto their personal device, then they connect to the server of the game they are playing. (CSGO, Call of Duty, League of Legends, etc) This gives the server the control for the game and improves security. One downside is that it is relatively inflexible - player experience depends on server performance and server locations. Oftentimes multiple servers for different regions are needed to decrease network travel time and provide a more competitive environment [3]. Though this may increase matchmaking time if the game does not have enough audience in one server region. In addition, even the current latency(30 ms to 60ms on average, if not intentionally connecting to an unintended server. For example, a US West player connecting to a London server) is not enough at the highest level of competition. One of the reasons that esport has LAN events is to connect elite players around the globe to one local network and ensure the highest level of competition.

2.2. P2P(Peer to Peer)

Peer to peer is when all devices become a part of a network and send updates to each other. This is also called a serverless architecture. The bandwidth of this network increases with the amount and devices there are, and how the algorithm is implemented. This was the standard up until the 2010s, where criticisms about its security flaws began appearing. Nowadays P2P is only used for smaller, more casual games that are not so intensive in the quality of connection and the security of connection (Co-op games, etc) [3].

2.3. Cloud gaming service

Cloud service is a rising new technology that makes video games much more affordable and available, more so than gaming consoles. They are essentially removing most hardware limitations to a video game; of course, you still need a device. Similar to video on demand service, the user only needs a device that is capable of displaying graphics [4]. This is kind of like the opposite of P2P gaming, the service provider will run the game that the user desires remotely from their server and then streams it

DOI: 10.54254/2755-2721/5/20230525

directly to the user's device. This technology is only available because of the advancement in connections speeds and data centres, etc. Similar to Client to Server, there are certainly connection latency concerns; it can massively benefit from the wide implementation of technologies like 5G. There is no doubt that this will greatly expand the audience of video games and may even be the future of gaming.

3. Cheating methods

To prepare yourself for a fight, first you need to know who you are fighting. To fully understand anti-cheat, there needs to be an understanding of current cheats. There are two general categories of cheats: soft cheats and hard cheats [5]. Soft cheats are usually unintentional bugs and exploits that were caused by a mistake during the development of the game. For example, performing certain actions at certain places in order to gain certain items or get to places that were not intended. These bugs can give players unwanted advantages, but they exist inside the game itself; so it's kind of cheating, but not really. Thus called soft cheats. These cheats are not based on external programs to mess with game mechanics. Oftentimes they cannot be detected by anti-cheat software and require developers to keep close connection with the player base, allowing them to find and fix these bugs as soon as possible.

3.1. Hard cheats

Anti-cheat software is more focused on the detection of hard cheats, as they are often more disruptive and harder to prevent: you have to clean them up one by one. They are usually third party programs that help users perform actions/gather information that is not normally attainable. There are hundreds of different types of cheats dedicated to different types of games. Below are some of the most common types of hard cheats.

3.1.1. Automated tools. These are programs that are programmed to perform repeated actions. These are mostly applicable in clickers games or RPG(role-playing games) where you have to farm tons of resources. This reduces the labour and allows players to gain advantages as they technically just machines working tirelessly 24/7. They can also be bots that detect a player on the user's screen and automatically lock the user's cursor to the player, giving the user superhuman reflexes [6].

3.1.2. Utilising hidden game data. These programs get data from game memory that is hidden to the player but necessary to construct locations and statuses of other players. For example, the so-called wallhacks, where players can see the positions of all enemy players, gain a tactical advantage.

3.1.3. DDOS (distributed denial of service) or similar network/server related attack. DDOS is an attack that is not only used in video games, but all over the internet. It is a network of inter-connected machines with malware infected bots, allowing them to be controlled by a single user. Then on a single command, all of these machines send out requests to connect to the server IP address, potentially overwhelming the server and resulting in a denial of service to normal traffic [7].

There are thousands of ways to cheat in a video game, these are just three categories that summarise most common cheats. On a side note, with the advances in Artificial Intelligence and Image recognition, it is evident that future cheats will be much harder and may fundamentally change the way anti-cheats are developed.

DOI: 10.54254/2755-2721/5/20230525

4. The "width" and the "depth" of cheats

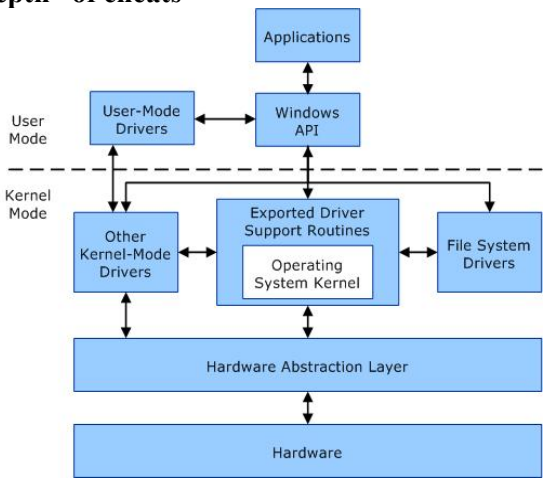

Figure 1. Courtesy: Microsoft.

Most of the cheat methods discussed above rely on third party software to be running. These softwares can give the user all kinds of "superpowers" in game. This is what the author describes as the "width" of cheats, it's a wide variety of attacks from different angles that's trying to find the weak spots in the game's defence against cheaters. Though there is also a level of depth to the cheats. This describes the privilege level system within an operating system. For example, the Windows operating system has a user mode and a kernel mode (Figure 1). User mode is somewhat like the frontend of your computer. This is where all your video games, editing software, and other applications that you normally download are run [8]. Kernel mode accesses the computer's hardware and manages different processes that should be running, sort of like the bridge between software and hardware [8]. One of main differences between user mode and kernel mode is that when a user mode application is running, Windows gives the application a private virtual address space, meaning each application is independent; the crashing of one application would not affect other applications. Kernel mode is the opposite, all codes share one single virtual space address, [9] (Virtual space address is an assigned physical memory location before the read or write process that allocates the application into the right physical memory.) if one crashes, the entire system crashes. The individual virtual space in user mode not only isolates each application, it also protects the data of the operating systems. Although there are still ways to get around it (viruses), it is a relatively secure system [10]. If you can't wrap your head around this, an analogy is that the video game anti-cheat is a x-ray scanner that scans for pests (cheats) that's 5 metres into the ground (operating system privilege level), but the cheat dug itself in 10 metres underground; there is just no way for the scanner to detect that cheat.

4.1. Why does it matter?

All those user modes and kernel levels seem fine, right? The problem is: since some cheaters are willing to buy cheats that run on kernel mode, which essentially gives the developer of the cheat a botnet without the need to gather hardware. Traditional anti-cheats that run on user-mode will not do the job anymore, which is where kernel mode anti-cheat comes in [11]. We will discuss different anti-cheats later.

Though there are privacy concerns around kernel mode anti-cheat. Using the words from the game developer Riot Game's blog post, "This isn't giving us any surveillance capability we didn't already have. If we cared about grandma's secret recipe for the perfect Christmas casserole, we'd find no issue in obtaining it strictly from user-mode and then selling it to The Food Network. The purpose of this upgrade is to monitor system state for integrity (so we can trust our data) and to make it harder for cheaters to tamper with our games (so you can't blame aimbots for personal failure)." [11]. Users can rest assured that there is no real privacy on the internet, it's just a matter of it is in the hands of people that do not care or the people that can use it to harm others.

DOI: 10.54254/2755-2721/5/20230525

5. Anti-cheat methods

The "depth" of anti-cheats are relatively straightforward and easy for the developers to implement. The author thinks we should get back on countering the wide variety of cheats. Depending on the type of connection architecture the video games uses, anti-cheat mostly relies on two departments: server, and client [12].

5.1. Server: utilise statistics

On the server-side, the developer, who is in control of the server, can use statistics and anomaly detection to protect the game. Specifically, as the cheater community grows, many cheats will be distributed a number of times. Game developers can see which cheat software has been appearing repeatedly; in the future, automatically banning players who have this software running. They can also look at player statistics and find the ones with abnormal statistics, although this may require manual examination as some players are really just insane at video games [5]. What can be automatically done is to examine game data sent from clients. For example, if a player appears at places that are virtually unattainable with their current speed, then it's quite obvious they are teleporting. Or if they are shooting players that aren't even on their screen.

5.2. Server: server protection

This is more focused on cloud gaming as it is more server-reliant. Server protection can include anti-DDOS attack solutions, and firewall that prevents hackers from modifying the data, etc. This will be information from another department of Web Security.

5.3. Client: data encryption:

Oftentimes online game datas are sent to the client as they are essential for the construction of scenes on the client's devices, these data sent through various packets. If it is not encrypted, players can get information from the packet that was not supposed to be available to the player. Developers will need to encrypt the data so that it cannot be easily obtained. There should be as little interaction done on the client side as possible. For example, if a fight result is being calculated locally, then the user has a chance to tamper the result and then send it to the server. Whereas if the user only sends actions to the server, then the server will compute the result. If the user tampers actions, a scan from the server side can be done to ensure integrity of actions [5]. One thing to note is that cloud gaming does not suffer from this issue as all game data is inside the server, this can pose to be one of the positives of cloud gaming. These memory protection and data encryption ties to the user mode access level that was introduced earlier, as higher permission levels can often view the data with ease. It is important for the existence of kernel level anti-cheat so that data also requires the highest level of permission, preventing them from being accessed by unwanted programs.

5.4. Client: file encryption:

If the files of the game are not encrypted, clients can easily modify the way their game is constructed, or add in codes that give the user client-side advantage. For example, the user may be able to go in and modify the texture of walls so that they become transparent, essentially giving user wall hacks. The adversary may also reverse-engineer the program if it is not encrypted, these can help people develop cheats faster and find security flaws much easier. This is also very useful in offline games, as they can prevent unauthorised distribution, or pirating.

6. Conclusion

Similar to web security, video game anti-cheat is a non-stop warfare against the adversaries that are trying to disrupt the order and gain unfair advantages. As cheating methods evolve, anti-cheat methods are also evolving. Cloud gaming is a great new technology that solves many client-sided anti-cheat issues. It is clear that as gaming becomes more popular, more eyes will turn towards this realm; with recent advancements in Artificial Intelligence, both cheaters and anti-cheat developers can use it to

DOI: 10.54254/2755-2721/5/20230525

their advantage [6]. Many things discussed in this paper originated from personal experience, and despite the lack of information during the research of the paper, it is fascinating to see the amount of work developers have put in to provide their players a safe and fair place to compete.

References

[1] "What Is Vanguard? – Valorant Support." VALORANT Support, Riot Game, Inc, https://support-valorant.riotgames.com/hc/en-us/articles/360046160933-What-is-Vanguard-.
[2] Chamberlain, Paul. "Valorant Anti-Cheat: Cheater, Reported!" VALORANT, Riot Games, Inc., 31 Aug. 2020, https://playvalorant.com/en-gb/news/dev/valorant-anti-cheat-cheater-reported/.
[3] Ronkainen, Waranyoo. "Prevention vs Detection in Online Game Cheating." (2021).
[4] "What Is Nvidia GeForce Now?" NVIDIA, https://www.nvidia.com/en-us/geforce-now/faq/.
[5] Lehtonen, Samuli Johannes. "Comparative Study of Anti-cheat Methods in Video Games." (2020).
[6] Spijkerman, Ruan, and Elizabeth Marie Ehlers. "Cheat Detection in a Multiplayer First-Person Shooter Using Artificial Intelligence Tools." 2020 The 3rd International Conference on Computational Intelligence and Intelligent Systems. 2020.
[7] "What Is a Distributed Denial-of-Service (Ddos) Attack? - Cloudflare." Cloudflare, https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/.
[8] Aviviano. "User Mode and Kernel Mode - Windows Drivers." Windows Drivers | Microsoft Docs, 14 Dec. 2021, https://docs.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/user-mode-and-kernel-mode.
[9] Aviviano. "Virtual Address Spaces - Windows Drivers." Windows Drivers | Microsoft Docs, 14 Dec. 2021, https://docs.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/virtual-address-spaces.
[10] "Difference between User Mode and Kernel Mode." GeeksforGeeks, 4 July 2022, https://www.geeksforgeeks.org/difference-between-user-mode-and-kernel-mode/.
[11] Koskinas, Phil. "/Dev/Null: Anti-Cheat Kernel Driver - League of Legends." /Dev/Null: Anti-Cheat Kernel Driver - League of Legends, 3 Feb. 2020, https://www.leagueoflegends.com/en-us/news/dev/dev-null-anti-cheat-kernel-driver/.
[12] Auriemma, Luigi, and Donato Ferrante. MULTIPLAYER ONLINE GAMES INSECURITY. p. 16.