GAME SECURITY

CENG 3544 - COMPUTER NETWORK SECURITY

Sultan Güvenbaş
sultanguvenbas@posta.mu.edu.tr
Waseem Wasaya
180709715@posta.mu.edu.tr

Wednesday 16th June, 2021

Abstract

Gaming industry is a multi-billion-dollar industry that has been on the rise for the past few decades, all including its own esports scene and tournaments. Online and competitive gaming is one of the biggest genres of the gaming industry, and it is also one of the most vulnerable for cheating and hacking operations. In this paper we will discuss the why and the how of the vulnerability of online gaming and what factors go into mitigating the issue or reducing it. Cheating exists in some games more than the others, and that depends mainly on data being accepted by the client and verified server-side versus data relying entirely on the server-side of each operation.

Key words: online games, game security, game cheating, game hacking, game vulnerability

1 Introduction

Since the game industry is getting more and more important every passing day, cheating has become a very important problem in gaming, sometimes costing millions or even billions of dollars to mitigate. Cheating cannot be removed entirely using our current technology, but there are ways to reduce it and to discourage further attempts from bad actors. We will explain the methodology that is used by the games we play today to reduce the problem, and what can be done about it in the near, as well as the far future. Game hacking is separate from network security problems. Even if the network is, in theory, completely secure and unhackable, games can still be hacked, and people can still cheat. The reason is that there are operations that are completely done on the client-side and they cannot practically be verified by the server in our current technologies. Operations that are very primitive such as moving a mouse or clicking a button. In some games, a delay is acceptable and these operations can be verified server side, but in others this is simply impossible without a delay that would make the game uncompetitive or plain unplayable..

2 Fundamentals

This section will explain the basics of how game clients and game servers interact with each other, where cheating happens, and how it is undetectable by the server without having a software made specifically for the purpose of detecting cheats running on the client computer.

2.1 Types of Actions

When an action is made on the game client in an online game, there are 2 types of how the process will be like so it can appear on the screens of other players:

2.1.1 Client-Side Server-Authoritative

Action will happen on the screen while the data is sending to the server, and then the server will verify the data while or before sending it to the other players

2.1.2 Fully Server-Side

Action will be sent to the server, server calculates the difference and makes the required operation if it was a possible action depending on the data on the server, then sends a response to all the players including the action doer, and then the action will happen on the screen for everyone..

However, some game actions are entirely client-side, like recoil compensation in shooting games as it depends on mouse movement (also known as aiming), and a mouse can be moved at any speed to any point so it is not practical to send mouse movements to the server without a throttling process in order to not overload the network. A game cheat (or 'hack') can move the mouse pointer perfectly in order to have all shots precisely at one place. In this case, the server would have no way to verify that it was actually a cheat program and not a regular player movement. In some cases, it can be obvious as depending on the game as no player can be 100cheats compensate for that by adding randomness levels and other settings or modifications so it makes it even more difficult or almost impossible to detect this kind of cheat server-side. And because of this example and many other similar scenarios, it's a must to have anti-cheat software running on the client computers.

2.2 Lag Compensation

specific genres of games. For example, there are games that depend on server-side interaction, and the client acts only as a guest in the game. While there are game servers that allow the clients to provide some of the information for it, and that gives a bigger opportunity for clients to cheat. Some games, specifically shooting games, require minimal delay or lag in interactions, so much so that there is no time to depend on the server response for the action to appear on your screen, and that doesn't allow for server-side interaction for many of the functionalities in those games, the server would just act as a verifier for the information it receives from the client after the fact of it appearing on everyone's screens.

In games, generally, when a target makes an action in game, you would see the action after a delay, depending on your ping, which is basically the delay in milliseconds between your computer and the server you are connected to. So in shooting games for example, when a target moves, and your ping is 200ms, you would see him move 200ms later on your screen. Games usually set an acceptable amount of delay depending on the achievable network speeds by most people in the technology most popular in our time, and the general affordable speeds of the networks world-wide. So if your network delay is within that range, and you hit a target that appears in your screen differently than what it appears in that person's screen due to the unavoidable delay, the hit would still count by the server as it would compensate for the lag and calculate that the difference is less than the specified range so it is acceptable.

The following Figure 1 demonstrates an example of where you would see a remote moving avatar if you have a delay of 100ms.

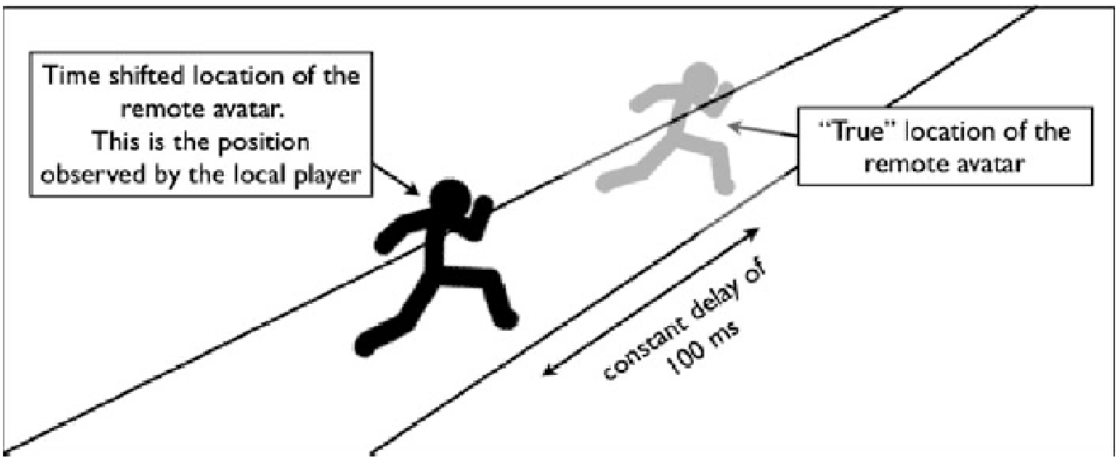
Time shifted location of the remote avatar. This is the position observed by the local player
"True" location of the remote avatar
constant delay of 100 ms

Figure 1: Delay Demonstration of a Moving Avatar

True location is where the target is at on their own screen, while the time shifted location is where the target is at on the screen of a client that has a network delay of 100ms. Since 100ms is acceptable in most games these days, hitting that target 100ms from where he is right now on his screen would count as an actual hit by the server.

However, if your delay is very high such as 300ms for example, and you take a shot at where the target was 300ms ago, your shot wouldn't count, even if it shows the shooting effects on your screen, the server would simply consider your delay too high and cancel any actions you have taken. So in conclusion, lag compensation is necessary to make games playable in the current age and the current state of technology, however, lag compensation also gives an easier time for cheat programs to be able to work and provide an unfair advantage to bad actors.

2.3 Information Exchange

Online games require information about the state of each player and any object a player can interact with on the server and the client. Some information is not critical and thus is shared by everyone on the server, such as player equipment and other things depending on the game. However, some information, even though are shared by everyone on the server, shouldn't be visible to the client, like for example, a player behind a wall, the game client knows that a player is behind a wall because the position is sent to the server, in order to deal damage even when shooting from behind a wall, however, that position is not revealed on the screen as the player should not be able to know the position of other players behind walls, but since this information is sent to the client, some cheats are able to intersect and read this information and present them on the screen, so the player will have an unfair advantage in that they are able to know the positions of other players in a situation where they should not be able to. In this case, the player exists behind the wall but you are not able to see them because the wall is blocking your view, and not because your client does not render that player.

Such cheat programs are called wallhacks, because they are able to reveal positions of players behind walls, either by showing dots of where the players are at, or simply rendering the walls as transparent allowing you to see what is behind them. The following Figure 2 shows how a player can cheat by seeing the positions of other players behind walls.

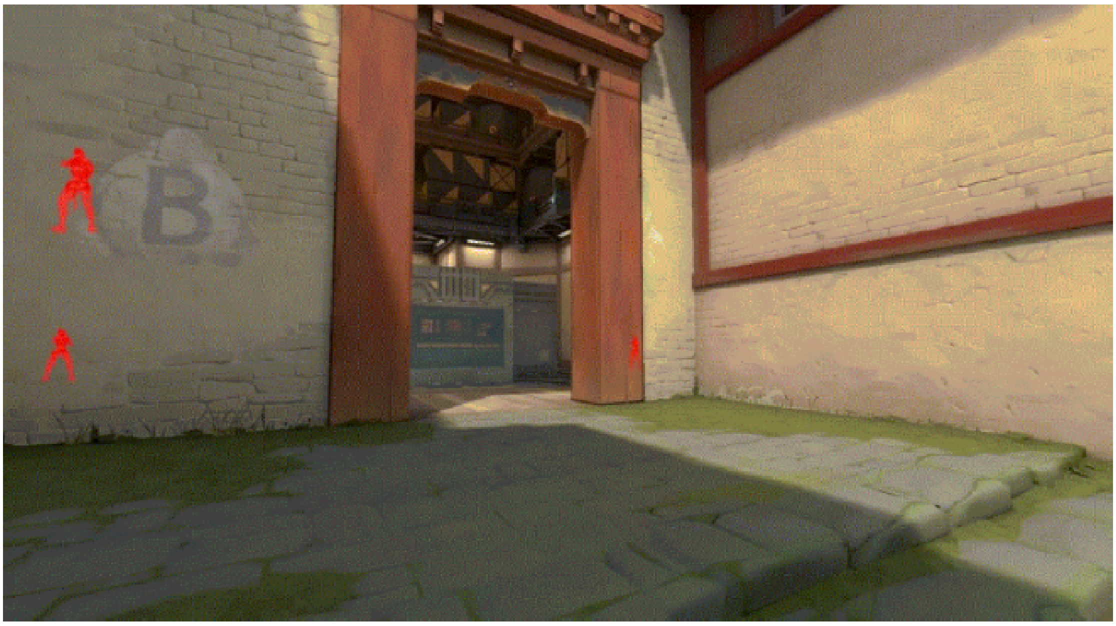

Figure 2: Positions of Players Behind Walls

The reason why the server sends the positions of players even behind walls, is because when you pass by a wall you should be able to instantaneously see the other players, without any delay whatsoever, otherwise you would be well past that wall and then players would just appear.

Games are able to mitigate this issue by doing heavy calculations server-side on each player so they can only provide the positions of players behind walls only when that player is close to passing a wall that would allow them to see those players. That works by the server sending the positions of other players only to the players who are close enough to a position that should allow them to see other players. However this has its own issue, in that some games depend on physics simulation that works client-side such as glass breaking or other types of objects interfering with each other that would not be uniform each time it happens, and in order to render this kind of interaction even when it happens from behind a wall, it is necessary to provide the positions of all players and their actions at all time to everyone on the server. This type of problem can be mitigated in the future when we have the technology that can do all of this calculations server-side, but as of right now, doing so would slow down the server so much so that it is unable to keep up with the players; and because of this reason, many competitive games opt to stay away from such physics engine reliance even at the cost of having effects being unrealistic in order to preserve competitive integrity, especially in conditions such as huge tournaments that have a huge sum of money as a reward.

3 Future State Discussion

Many solutions are proposed to solve the cheating problems of gaming and especially competitive gaming in the future; however, these solutions are entirely dependent on server and network infrastructure that is way beyond the reach of our current technology at this time. This section will discuss some of the proposed solutions and the possible outcomes of each.

3.1 Cloud Gaming

Cloud gaming is a term used for services that run the game clients on their own servers and sending the player a video stream of that game; this has many benefits in that it allows you to play any game that is available at that service without requiring any of the graphical power required to play the game, since your computer will not be running the game client but will only be running a video stream of the game. This works in single player games or online games where having a delay especially in mouse movements or aiming does not affect the gaming experience, but the delay hugely affects the competitive integrity in largely competitive games such as shooters, and thus while this is not a viable solution right now, having close to zero delay in the far future in our network and internet infrastructure would make this a close to perfect solution to prevent cheating.

3.2 Server-Side Only Actions

Server-side only actions are actions that are only done on the server, which means the client can only send the input to the server such as mouse movement and keyboard buttons, instead of what it currently sends, which is the actions it makes or the shots it takes. In this case, sending only the input means. So far this is close to cloud gaming but the most important difference is that it does not rely on the cloud and a video stream for the game, which means people will still have the games running on their own computers, while similarly to cloud gaming, only providing the input of their devices (mouse, keyboard, controller) and sending them to the server, the server then decides the actions depending on the inputs. It also does not send any extra information such as the positions of other players unless the player is in a position where they are supposed to see the other players; unlike the current method of providing this info when the players are close enough, because close enough is not a solution as you can still have a huge advantage by taking actions before the other players have a chance to react. This solution also requires a close to perfect internet infrastructure that allows close to zero ping, and it would also allow cheats that would act on input devices rather than sending actions to the server.

4 CONCLUSION

In conclusion, we can see that currently there are multiple ways of cheating, with varying cheating intensity depending on the genre and most importantly the nature of the game; the more competitive a game is, the more people that would be willing to cheat to gain an advantage on it. One of the only ways to mitigate this problem right now is having anti-cheat software running in the client computers, which can still be modified and fiddled with by bad actors, and so this becomes a race of updating it to make it harder to fiddle with, and bad actors using more advanced methods of fiddling with it, until a perfect solution is made to fix the cheating problem which can be one of the proposed solutions that have the potential to be used in the far future since they are dependant on internet and general network infrastructure that is not possible to achieve with the current technology we have.

References

[1] Savery, C., Graham, T. C. N. (2012). Timelines: simplifying the programming of lag compensation for the next generation of networked games. Multimedia Systems, 19(3), 271–287. doi:10.1007/s00530-012-0271-3 https://sci-hub.se/https://link.springer.com/article/10.1007/s00530-012-0271-3

[2] Yan, J., Randell, B. (2009). An Investigation of Cheating in Online Games. IEEE Security Privacy Magazine, 7(3), 37–44. doi:10.1109/msp.2009.60 https://sci-hub.se/https://ieeexplore.ieee.org/abstract/document/5054908

[3] Paul "Arkem" Chamberlain (2020). Demolishing Wallhacks with VALORANT's Fog of War https://technology.riotgames.com/news/demolishing-wallhacks-valorants-fog-war

[4] Ki, J., Hee Cheon, J., Kang, J., Kim, D. (2004). Taxonomy of online game security. The Electronic Library, 22(1), 65–73. doi:10.1108/02640470410520122 https://sci-hub.se/https://www.emerald.com/insight/content/doi/10.1108/02640470410520122 /full/html

[5] Beykent University, Department of Computer Engineering, Ayazaga, 34396, Istanbul, Turkey. (2020) Evaluating a Player's Network Class in a Multiplayer Game with Fuzzy Logic. DOI: 10.17714/gumusfenbil.518689