

Cheats in online video games: detection, analysis and countermeasures

Samuel Bernard
samuel.bernard@lip6.fr

Maria Gradinariu Potop-Butucaru
maria.gradinariu@lip6.fr

Sébastien Tixeuil
sebastien.tixeuil@lip6.fr

Université Pierre et Marie Curie - Paris 6
LIP6/CNRS UMR 7606
104 avenue du Président Kennedy
75016 Paris, France

Abstract—Video games now represent a large part of the entertainment industry and as with sports, important competitions are held. As games become more frequently played online, cheating against human players turns into a huge problem. Cheats create a serious unbalance in the gameplay among players, upsetting the honest ones and damaging the games fame. Video game developers and then academic researchers have started to design new solutions against cheating, rough at the beginning and then more advanced, to fight the new scourge of the online gaming industry.

I. INTRODUCTION

Video games now represent a large part of the entertainment industry and took in about \$7.4 billions in the US in 2006 (ESA annual report) and some games are played by millions of players. Moreover a new category of sport appeared, Electronic-Sport. World competition are being held, like the *World Cyber Games* [1] or *Electronic Sport World Cup* [2] in which players compete to become world champions of their favourite game. Most of them are professionals paid by their team and sponsors, like in traditional sport world.

Because of the importance of game results, cheating become a huge problem [3]. Like in sport where cheat is represented by doping, people want fair fight. But unlike sport the problem of cheat is not in high level competition but in common online match, bothering most of gamers. This is due to the anonymity given by internet and the easiness of cheating. Online games are quite young, so game developers have not yet created the countermeasures needed. As this turns out to represent a huge market, some company are specialized in developing industrial solutions against cheating [4] and proposed their solutions to developers. Thus this research domain is interesting, useful and most of the work is still undone [5].

It is obvious that because games may be totally different from one another, cheat applied to one game may not be directly applied to another. We use the definition of cheating given by [6]: *Any behaviour that a player may use to get an unfair advantage, or achieve a goal that he is not supposed to, is cheating.*

II. ANTI-CHEAT SOLUTIONS

A. Binaries protection

Initially, security in video games was developed to add copy protection and check for genuine versions. When online cheaters appeared, developers used the same techniques to hide the lack of anti-cheat design and to prevent modifications and analysis of memory and network protocols. However, this is an ineffective solution in the long term. A recent academic paper proposed binaries protection by using dynamic mobile agents [7] instead of static protections (which will be broken eventually). An original agent will be periodically downloaded and executed. The result should be different if the game binaries are not genuine. Because the downloaded code is always new, cheaters do not have the time to break it and return the good result.

The problem of binaries protections and other hiding solutions is that security by obscurity has never been effective: there always has been someone to break it. By chance, a hundred percent protection over the time is not needed. Actually only one paper [7] designs such a technique and its limitations is in the feasibility. Each time it is necessary to create a new and effective binary verification which can not be broken fastly. An example is given but its effectiveness remains to be proved.

B. Protocol-level solutions

The main part of anti-cheat techniques in the academic world deals with protocols. The idea is to modify or create protocols which forbid lag-related cheat. Indeed, players have different and variable latencies and to gain performance protocols that were implemented do not have strict timestamp verification: when players send messages, they put a timestamp that is not verified by others. This default trust allows cheaters to earn time by waiting others messages, decide their action and send it with a smaller timestamp thus acting before non-cheater players.

The first major anti-cheat protocol is the Lockstep Protocol [8] by Baughman *et al.* It works in two phases: first, every player commits his action for the current timeslot (time

is always divided into small timeslots where players can do one action) by sending a cryptographic hash representing it. Second when they all received others hash, they send their action, which can be verified by others by computing the hash and comparing it to the hash previously received.

The main problem of such protocols is the increased delay: every player has to wait for the slowest one. It is not acceptable on an unreliable network like the Internet. To address this issue, Baughman *et al.* proposed a second algorithm, the *Asynchronous Synchronisation Protocol*. The idea is to wait only for players that interact with us. In some games, interactions are limited providing AS protocol correct performances. Improvements of these protocols exist, for instance NEO [9] and SEA [10]. They improve performance by adding a maximum wait time, piggybacking and pipeline.

However, performance is still an issue, as they have to make consensus to choose accepted messages which may be very slow with some message loss, making the game unplayable. Moreover their designs are not totally secure (against a coalition) and not really proved (some assertions made are false: [10] resolved some issues which [9] was supposed to have corrected).

C. Detections

If cheaters cannot be avoided, due to the very high constraints in latency, but can be detected soon enough, then it is possible to repair their damages and to ban them. It is obvious that if a cheater loses all 'his' work if he cheats, it will not be interesting in cheating. One mean is by keeping logs and analyzing [11] it later. A good example is Delap *et al.*'s work [12] who proposes to execute runtime verifications of the rules to detect cheaters and then ban them.

Cheat detections depends, in most cases, on the game being played. And as few works exist about detection, few games (or kind of games) and cheats have a detective approach. [12] does not include specific detection algorithms and is just a generic framework to perform detections.

III. OPEN PROBLEMS

A. Protocols

Some research can still be done regarding protocols. The goal is to modify existing algorithms or to create new ones which will be cheat-free enough while preserving low latency. The good point is one hundred percent security or fault tolerance is not required and an approximation may be much faster. Moreover a large number of protocols exist but proofs are not really present or are incomplete. An interesting work would be to verify and prove existing protocols and then perform experimentations. If no acceptable protocols remains, a new one is to be suggested.

B. Massively Multiplayer Online Game systems

A lot of works may be done in MMOG architecture. All current commercial MMOG are running on top of a bunch of servers and are not really scalable. A good peer-to-peer architecture will provide a scalable and affordable solution

and is a great technical and theoretical challenge. An example of new system is FreeMMG [13] and several theoretical works exist on the topic like [11].

C. Cheat detections and reputation system

Another axis of research is to concentrate on cheat detection by designing detection-tools which can not be easily avoided or countered. Including them in a generic framework [12] would lead to interesting results. As a matter of fact, games may be very different from each other and game-dependant solutions could be best. For instance, fighting "aim bots"[14] is completely different from avoiding "maphack"[15].

Once a cheater is detected and banned, what can be done to avoid his return? Detecting cheat is insufficient, specially if the detection is imperfect. An idea is to create a reputation system where honest players have good marks and cheaters are marked as it. Nothing prevents a cheater to come back with another login but an empty profile is always suspicious. Moreover regular players will quickly have sufficient marks to play between them.

REFERENCES

- [1] WCG, "World Cyber Games," <http://www.worldcybergames.com>.
- [2] ESWC, "Electronic Sports World Cup," <http://www.eswc.com/>.
- [3] M. Pritchard, "How to hurt the hackers: The scoop on internet cheating and how you can combat it." Information Security Bulletin, Feb 2001.
- [4] SecurePlay Team, "Game Security Services and SecurePlay Game Programming Toolkits in Flash Java C++," <http://www.secureplay.com>.
- [5] S. D. Webb and S. Soh, "Cheating in networked computer games: a review," in *DIMEA '07: Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts*. New York, NY, USA: ACM, 2007, pp. 105–112.
- [6] J. Yan and H.-J. Choi, "Security Issues in Online Games," *The Electronic Library*, vol. 20, no. 2, 2002.
- [7] C. Mönch, G. Grimen, and R. Midtstraum, "Protecting online games against cheating," in *NetGames '06: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*. New York, NY, USA: ACM, 2006, p. 20.
- [8] N. E. Baughman, M. Liberatore, and B. N. Levine, "Cheat-proof playout for centralized and peer-to-peer gaming," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 1–13, 2007.
- [9] C. GauthierDickey, D. Zappala, V. Lo, and J. Marr, "Low latency and cheat-proof event ordering for peer-to-peer games," in *NOSSDAV '04: Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video*. ACM, 2004.
- [10] A. Corman, S. Douglas, P. Schachte, and V. Teague, "A secure event agreement (sea) protocol for peer-to-peer games," in *ARES 2006, The First International Conference on Availability, Reliability and Security, 2006.*, 20-22 April 2006.
- [11] P. Kabus, W. W. Terpstra, M. Cilia, and A. P. Buchmann, "Addressing cheating in distributed MMOGs," in *NetGames '05: Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*.
- [12] M. DeLap, B. Knutsson, H. Lu, O. Sokolsky, U. Sannapuri, I. Lee, and C. Tsarouchis, "Is runtime verification applicable to cheat detection?" in *NetGames '04: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*. ACM, 2004, pp. 134–138.
- [13] F. R. Cecin, R. Real, R. de Oliveira Jannone, C. F. R. Geyer, M. G. Martins, and J. L. V. Barbosa, "FreeMMG: A Scalable and Cheat-Resistant Distribution Model for Internet Games," in *DS-RT '04: Proceedings of the Eighth IEEE International Symposium on Distributed Simulation and Real-Time Applications*. IEEE Computer Society, 2004, pp. 83–90.
- [14] T. Schluessler, S. Goglin, and E. Johnson, "Is a bot at the controls?: Detecting input data attacks," in *NetGames '07: Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games*.
- [15] C. Chambers, W. Chang Feng, W. Chi Feng, and D. Saha, "Mitigating information exposure to cheaters in real-time strategy games," in *NOSSDAV '05: Proceedings of the international workshop on Network and operating systems support for digital audio and video*. ACM, 2005.