

**Title :** Smart Security Software

**Project Summary:** This project is designed to meet the security need in the online gaming industry with the highest efficiency. In the project, “FPS” was chosen as the target online video game genre. First-Person Shooter (FPS) is a sub-genre of shooter video games centered on weapons and other weapon-based combat and it is played through the player character's own eyes (Wikipedia, 2021).

The library of this project is equipped to cover all attack methods against online games, from the oldest to the newest, as a result of the necessary literature searches.

Project Content:

1. Intelligent Security Expert
2. Smart Security Software

Functions of the intelligent security expert:

1. It can detect vulnerabilities that are likely to be found in game security systems.
2. It can successfully detect DLL files which are injected into the memory area by the person(s) who attack the security system of the games and memory injection techniques by using advanced signature-based intrusion detection systems and memory scanning methods.
3. It can create techniques to prevent newly developed attack methods that are not in its library with the help of artificial intelligence and it can add these attack methods to its library together with the current blocking techniques that it has successfully created.
4. It can constantly update and improve itself.
5. If it detects a vulnerability in the relevant game security system, it can make in-depth analyzes on how to integrate the defenses against all old and new generation attacks into the code lines, and it can simplify and report these analyzes and present them to the relevant game security developers.

Functions of the smart security software:

1. It was developed by using intelligent security expert and it can simultaneously block attacks on game security systems.
2. It can learn and it can be updated continuously.
3. It can be integrated into all online games of the FPS genre.
4. It protects FPS games from in-game cheats and it also protects them from possible game piracy (crack) by making the necessary encryptions.

**Keywords:** FPS, Game Security Systems, Artificial Intelligence, In-Game Cheats, Crack

## 1. BACKGROUND RESEARCH & ACADEMIC SURVEY

The scope, limits and importance of the subject addressed in the project proposal are explained with qualitative or quantitative data as well as a critical evaluation of the literature.

### 1. XignCode3

Xigncode is a kernel-mode anti-cheat that protects the games from injecting, debugging and other hacking/cheating relating activities.

XignCode works based on server -> client, client -> server. It sends some security packets from client to server. These security packets are received by the server and sent back to the client. This interaction is called "heartbeat" and if XignCode is disabled, after a while the client will be disconnected from the server.

Unfortunately, I must inform you that this interaction is actually one of the main weaknesses of XignCode. The main weakness here is that XignCode can be easily disabled by a simple manipulation of its initialization code (asm: RETN). Even if you will be disconnected from the server since XignCode does not send the required security packages to the server when it was disabled, your intervention in XignCode to be finalized by the server will take about a few minutes and during this time you will have full access to the target game you want to attack. With the few minutes you spend in the game without getting caught by the heartbeat, in order to interfere with the game system; you can dump the necessary modules from memory, reverse engineer them or just find pointers. Later, you can code a hack with this informations.

There's more! XignCode is not ready to run as soon as the computer starts like some game security softwares. It is initialized during the execution of the target game you want to attack. This means that the client of the target game is vulnerable during the installation of XignCode and is open to remote intervention. In other words, you can successfully inject the hack you have written to manipulate the game while the installation of XignCode is not exactly completed.

As you can see, you can disable XignCode with a very simple intervention, collect the necessary information for the hack you will code without getting caught by the heartbeat and then you can simply integrate this hack into the game by using a simple vulnerability of XignCode, without even the needing for any intervention.

Sometimes, some bypass methods may not work for every game which is under the protection of the any security software. However, these methods I mentioned above will work in any game that uses XignCode, since they take advantage of XignCode's general vulnerabilities.

Finally, let's talk about whether we can disable XignCode fully or not.

1. We can disable XignCode to run in all games, albeit temporarily.
2. We can successfully inject code while the installation of XignCode is not exactly completed, even without the needing for bypass.

These are the general weaknesses of XignCode that we mentioned above. So, can XignCode be completely disabled for an unlimited time? In other words, can any bypass be coded that works in every game which is under XignCode protection and allows us to access the clients of these games illegally "for an unlimited time" (Can heartbeat be bypassed)? Yes heartbeat can be bypassed.

As you know, if XignCode is disabled, this can only be detected by effective communication between client and server. Then we need to intervene in this communication. If we can detect and imitate the packets that XignCode sends and receives to the server over the game module, we can exchange client-server security packages even though we disable XignCode. So, we actually design a heartbeat emulator. In this case, the server will suppose that XignCode is running and will keep us on the server.

As a result, XignCode can be fully bypassed and each coded successfully bypass can perfect work in any game which is under XignCode protection.

In my project, there will be a direct link between my game security software and the related game client. If my game security software is tried to be disabled in any way, the related game will be closed directly. There will be no loss of time. To solve this relation, of course, you can try to get into kernel mode. However, if you try to interfere with this interaction, the system will turn itself off again. In addition, it will be close to impossible to fully unravel such relations. This is because such interactions that directly affect game security will be carefully encrypted and hidden very well.

### 2. Easy™ Anti-Cheat

Easy™ Anti-Cheat is the kernel-level anti-cheat service, countering cheating and hacking in multiplayer PC games. It is one of the most successful software available in the market in terms of preventing cheats. It would be really tiring to fully bypass this software. It contains many control mechanisms and these mechanisms are controlled very successfully at the kernel-level.

The most important weakness of this software is that they have not been able to successfully hide the interactions at the kernel-level. This means that we can easily enter the kernel and access the process of the relative game. If our aim is to screw around comfortably in any game client which is protected by EAC, unfortunately, this is very easy task to achieve. In other words, once you get into the kernel, unfortunately, it's not hard to dump the modules and start reversing.

Besides, I see a lot of user complaints about EAC. Even though users do not use any third party software, they are banned by EAC out of the blue. This is a very serious problem that EAC needs to fix.

My project can very successfully encrypt and effectively hide kernel-level game security operations compared to EAC. Therefore, it is almost impossible to fully solve these interactions by using reverse engineering. This means that, compared to EAC, it is almost

impossible to just screw around the client of any game which is protected by my software, even without changing anything. Besides, Smart Security Software (SSS) never bans any of its users for no reason. The reason for this is that it can sharply distinguish between applications that try to affect the game illegally and applications that will not have a direct effect on the game by using its enhanced artificial intelligence.

### 3. BattlEye

BattlEye is one of the most popular game security software. It is a kernel-level anti-cheat. BattlEye has most of the controls that EAC has, but it is less popular and easier to bypass compared to EAC.

When you are in the kernel, you can manipulate BattlEye pretty easily. You can use a virtual machine (VM) or hypervisor to dump the BattlEye module and reverse engineer it. Although there are some emulation detectors in BattlEye, bypassing this system is not too hard.

As a result, it will not be a very difficult task for you to fool the detecting system emulators after analyzing and emulation the game security packages and to manipulate the BattlEye at kernel-level.

As can be seen from the above, BattlEye can easily be disabled (bypassed) completely. The main reason for this is that BattlEye could not hide kernel-level processes well enough, just like EAC, and the BattlEye's heartbeat system, which can be easily intervened remotely, has an easily decryptable encryption structure.

Smart Security Software (SSS) performs kernel-level operations that are almost impossible to monitor and interfere with (Although it is very unlikely, in case of any interference with the kernel-level operations which is performed by SSS, this is detected very successfully and the relative client is shut down. Thus, the security of the target game is ensured.). Besides, SSS encrypts the communication between client and server using very advanced protocols and makes remote access to this communication nearly impossible.

### 4. PunkBuster

PunkBuster is a game security software that is designed to detect software used for cheating in online games. It does scanning the memory contents of the local machine to detect the suspicious activities. PunkBuster, a kernel-level anti-cheat software, has a heartbeat system. In addition, PunkBuster takes screenshots of the game at completely random time intervals in order to detect in-game cheats.

Unfortunately I must inform you that if you have enough coding knowledge, you can completely bypass PunkBuster. You can develop a code that will prevent PunkBuster from taking screenshots, easily make reversing PunkBuster by using IDA, manipulate PunkBuster's use of kernel mode driver and completely bypass its heartbeat system.

Smart Security Software (SSS) has protected itself comprehensively and effectively against all these manipulations.

### 5. nProtect GameGuard

nProtect GameGuard is an anti-cheating "rootkit". It is widely installed in many online games to block possibly malicious applications, protect client and prevent common methods of cheating.

nProtect GameGuard (GG), a kernel-level game security software, uses rootkits to prevent cheating software from running. This anti-cheat program monitors the entire memory range, hides the game application process, blocks certain calls to Direct X functions/Windows APIs and terminates applications identified as cheating by the game vendor/INCA Internet.

Since this anti-cheat program works like a rootkit, players experience potentially unwanted side effects.

nProtect GameGuard (GG) may block any installation or activation of hardware and peripherals (e.g., a mouse) while it is running.

Since this anti-cheat program monitors any changes in the computer's memory, it causes performance issues when the protected game loads multiple or large resources all at once.

When installing nProtect GameGuard some antivirus programs will alert the user that it is a rootkit and block the installation. They recommend uninstalling or disabling both antivirus and firewall programs while running it. Unfortunately, this shows that nProtect GameGuard has failed to be a user-friendly game security software.

nProtect GameGuard, failing to be a user-friendly game security software, also falter in game security. When you do a little research on the internet, you can see that most of the bypass codes you can find belong to nProtect GameGuard. Some of these bypass codes continue to work in certain games and unfortunately, nProtect GameGuard seems to have stopped developing itself in order to take precautions against such manipulations.

My project absolutely includes an user-friendly game security software. Smart Security Software (SSS) minimizes performance degradation, does not compromise your computer security, and does not work like a rootkit. In addition, it can be easily integrated into games and constantly improves itself compared to nProtect GameGuard against new generation attacks.

To sum up, it can be clearly seen that Smart Security Software (SSS) is much more successful than nProtect GameGuard in terms of both user-friendliness and game security.

---

Popular game security softwares have to do kernel-level (ring 0 level) operations to prevent cheats. This is because any ring 1-2-3 level anti-cheat software cannot block a ring 0 level threat (all professionally designed cheats are written at ring 0 level). Therefore, non-kernel-level game security softwares can be bypassed very easily.

Smart Security Software (SSS) is a kernel-level game security software. We only use your data to ensure the security of the target games successfully and we never store it. With Smart Security Software (SSS), both related games and user data will be completely safe. Some non-kernel-level game security software are listed below (2 pieces). All of them can be easily bypassed. For this reason,

Smart Security Software (SSS) will provide much more successful protection than each of the game security software listed below.

6. Valve Anti-Cheat (VAC)

Valve Anti-Cheat (VAC) is software running on the client and server that attempts to detect cheaters. It is made by Valve and has been around since the early days of Counter Strike, most known for its usage in CSGO. Valve Anti-Cheat (VAC) is a user-mode anti-cheat (ring 3 level) and it does not have a kernel-mode driver.

7. Warden

Warden (also known as Warden Client) is a user-mode (ring 3 level) anti-cheating tool integrated into most Blizzard Entertainment games.

Some actions are taken to ensure game security in online games. These actions are similar in each game security software, but since some game security software takes these actions in a more professional way, it becomes difficult to bypass them. There has been a cold war between game security teams and cheat makers for years. Even if game security softwares are improved with various algorithms, cheaters always manage to find new ways to disable these softwares. In the face of this situation, Smart Security Software (SSS) continues to improve the algorithms it uses every time.

8. Here is the summary list of actions taken by Smart Security Software (SSS) to ensure game security in online games:

1. Block all interaction with game process.
2. Block creation of process handles.
3. Scan for known suspicious drivers / DLL modules.
4. Scan for disks & devices.
5. Detect debuggers.
6. Check for kernel patches.
7. Detect hooks.
8. Checks all services.
9. Scan all threads & system threads.
10. Hypervisor & VM detection.
11. Instrumentation callbacks.
12. Controlling data exchange between client and server (Heartbeat).

### Aims and goals

The purpose and objectives of the project proposal are written in a way that is clear, measurable, realistic and achievable throughout the project.

**This project is designed to meet the security need in the online FPS gaming industry with the highest efficiency.**



abbreviations is important in the process of understanding and analyzing the function of the flowchart.

Special abbreviations and their expansions which are in the flowchart:






**GAP**= Game Access Permission

**SSS.sys**= Smart Security Software.sys

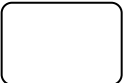
**AC-1**= Anti-Cheat-1


**AC-2**= Anti-Cheat-2

In the flowchart, modules (symbols) which have some special meanings are used. These modules are universally valid and their meanings are the same everywhere.

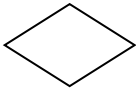
Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

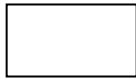
In the flowchart, some colorings are used. The purpose of this colorings is giving a beautiful appearance to the flowchart. Each color has a different purpose of usage.

 → It is represented by **blue color**.

 → It is represented by **yellow color**.



 → It is represented by **purple color**.

 → It is represented by **green color**.

**Red color** is used to highlight the necessary places.

In my flowchart, attention was paid to simplicity. Very complex details about the project were not mentioned. The whole process from start to end was gathered under general topic titles and presented in summary form.

Smart Security Software has a driver named "SSS.sys" which is loaded when the computer starts. SSS.sys is a very comprehensive driver consisting of 2 separate anti-cheat systems (AC-1 and AC-2), 1 Heartbeat system and many control mechanisms. This driver comprehensively monitors the processes which take place on computer, reports it and transmits it to the game server. It has kernel-level (ring 0) authority on the target computer (fully authorized). So, Smart Security Software is a kernel-level anti-cheat software. It is unfortunately necessary for Smart Security Software to be fully authorized on a computer in order to "full" protect the respective games. Because non-kernel-level security software can be bypassed easily because it cannot completely detect the activities which take place on the kernel.

The confidentiality of user data is of the utmost importance. The sole purpose of Smart Security Software's having full authority on the computer is to ensure the security of the respective games. It is guaranteed through contracts that the data received from users will be processed by the SSS only for the purpose of ensuring game security and then completely deleted. Also, SSS.sys only works while the corresponding game is running. It never processes data after the game is closed. Flowchart symbolizes the process from start to end of a game which is protected by Smart Security Software. When a game start request is received, a boot process is initiated. In this process, the game files are checked, if there is any missing/damage in the files, they are corrected, if the files are required to be updated, the update process is performed. In addition, during this process, all possible interventions which is to manipulate the game system are prevented. Thus, the game can be started with up-to-date and flawless files. A procedure called "initial value assignment" is applied during the boot process. This procedure sets the Game Access Permission (GAP) value to 2.

GAP can have 3 different values. These values are 0, 1 and 2. Smart Security Software starts or ends the game based on the values of the GAP. If the value of the GAP is equal to 2, the game start process will be paused for 1 second periods until the value of the GAP equals 0 or 1. AC-1, which is part of SSS.sys, has the authority to directly change the value of the GAP. Therefore, the game will not be started until AC-1 is started and functioning. After the "initial value assignment procedure" is applied, the boot phase is completed and SSS.sys is started. SSS.sys starts AC-1 and is ready to take the relevant actions by checking the GAP value. AC-1 checks the computer before and during the game start and if it detects a game violation, it sets the GAP value to 0. If everything is as it should be, it will set the GAP value to 1.

SSS.sys has a system that controls the game access permission of the user. The GAP value is controlled directly at the start of the game, and after the game is started, it is checked periodically by the AC-2 and Heartbeat system. If the GAP value is 0 before the game starts, the game is definitely not started and the user is banned since AC-1 has detected a violation. If the GAP value is 2 before the game starts, it means that the first scan of AC-1 is not fully completed and the game is put on hold ("wait 1 second") until the first scan of AC-1 is completed. If the GAP value is 1

before the game started, it means that AC-1 did its first scan and did not find game violations. So the game has become startable. AC-1 continues to scan in a loop (repeatedly) until the game is closed. If it detects a violation after the game starts, it will set the GAP value to 0 and this change in the GAP value will be detected by the AC-2 and the Heartbeat system and the ban procedure will be applied to the user. In summary, as long as the GAP value remains 1, AC-1 has not detected a game violation.

AC-2 and the Heartbeat system are also part of SSS.sys, just like AC-1, and AC-2 periodically (repeatedly) checks for interactions with the game during the game. The Heartbeat system, periodically (repeatedly) checks the data traffic between the client and server and SSS.sys and server. If everything is as it should be according to AC-1 (GAP value), AC-2 and Heartbeat system, the game stays open until the user requests to close it. However, if any violation is detected by SSS.sys before the game (AC-1) or during the game (AC-1, AC-2 and Heartbeat system), then the game client is closed and a ban procedure is applied to the user.

When the game is closed, SSS.sys is stopped. So AC-1, AC-2, Heartbeat system and game security controllers are all turned off. Thus, Smart Security Software never processes data while the game is closed.

Note: If Smart Security Software detects a game violation, it implements a ban procedure to the user. This procedure takes place as follows:

- 1) "Violation detected" message is sent to the user.
- 2) The user's data is sent to the server (HWID, IP, Account ID).
- 3) With the account that was found to have violated the game security, all accounts with the same IP or HWID as it are banned permanently.



### 3. PROJECT MANAGEMENT

#### 3.1. Work Packages (WP), Task Distribution and Duration

##### 3.1.1. Timeline

The duration of the main work packages to be included in the project is given by filling out the "Work-Time Schedule".

**WORK-TIME SCHEDULE (\*)**

WP id*	WP Name*	The Importance of the success of the Project(%)**	MONTHS																							
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	To search the literature and collect data.	20%	X	X	X	X	X	X																		
2	To come to the forefront by eliminating the projects which are developed by possible rival institutions/organizations.	20%							X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	Finding sponsorships to invest in the project.	20%							X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4	To manage risk and income.	20%							X	X	X	X	X	X							X	X	X	X	X	X
5	To become a fully corporate company by agreeing with the relevant companies (to bring this project to life).	20%																			X	X	X	X	X	X

(\*) The rows in the chart can be expanded and multiplied as needed.

(\*\*) The column total must be 100.

### 3.2. Risk Management

The risks that may adversely affect the success of the project and the measures to be taken to ensure the successful execution of the project when these risks are encountered (Plan B) are outlined in the Risk Management Table below by specifying the relevant work packages. Possible risks related to the research question and/or hypothesis of the project are taken into consideration. The implementation of plan B should not deviate from the core objectives and original value of the project. If there is a method change in case of switching to plan B, this situation should be detailed. Work packages for which no risk is foreseen are not included in this section.

WP ID*	Definition of Risk(s)	Action(s) to be Taken (Plan B)
1	Game server crash	If the game does not have a strong game server, the game may crash no matter how strong the game security is. Therefore, the servers of the games which have weak servers should be strengthened.
2	Security vulnerability of game sites	The most basic platform that introduces a game to users is game site. Therefore, a gaming site actually represents the quality of that game. Nobody wants to play a game which has a site full of security vulnerabilities. Therefore, the security vulnerabilities of the game sites are as important as the security vulnerabilities in the games, and all kinds of security gaps of the game sites should be closed in order to prevent this problem.

(\*) Rows in the table can be expanded and multiplied as needed.