Національний університет «Полтавська політехніка імені Юрія Кондратюка»
(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій та робототехніки
(повна назва інституту)

Кафедра комп'ютерних та інформаційних технологій і систем
(повна назва кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістра
(рівень вищої освіти)

на тему

Програмний модуль захисту комп'ютерних ігор від вбудованого коду

Виконав: студент 2 курсу, групи 601–дТН
спеціальності
122 Комп'ютерні науки
(шифр і назва спеціальності)
Бєлан Ф.О.
(прізвище та ініціали)
Керівник Зінченко А.О.
(прізвище та ініціали)
Рецензент Микусь С. О
(прізвище та ініціали)

Полтава – 2021 року

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ « ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ

КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

спеціальність 122 «Комп'ютерні науки»

на тему

«Програмний модуль захисту комп'ютерних ігор від вбудованого коду»

Студента групи 601-дТН Бєлана Федора Олеговича

Керівник роботи
доктор технічних наук,
професор Зінченко А.О.

Завідувач кафедри
кандидат технічних наук,
доцент Головко Г.В.

Полтава – 2021

РЕФЕРАТ

Дипломна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, додатків, загальним обсягом робота складає 7272 сторінки, має 31 рисунків, 4 таблиці, 2 сторінок додатків. Список використаних джерел містить 57 найменувань і займає 6 сторінок.

Метою дипломної роботи є розробка програмного продукту для виявлення вбудованого коду в комп'ютерних іграх.

В дипломній роботі я проаналізував доступні модулі захисту інформації. Розглянув їх надійність, плюси та мінуси.

Розроблений програмний продукт підвищує захист комп'ютерних ігор від вбудованого коду що можливо спостерігати зупинками в іграх, а це унеможливлює виграш.

Ключові слова: пірати, таблетка, зловмисник, злом, трейнер, шахрай, програмний модуль, ігри.

ABSTRACT

Thesis consists of an introduction, three chapters, general conclusions, a list of sources, appendices, the total volume of the work is 72 pages, has 31 figures, 4 tables, 2 pages of appendices. The list of used sources contains 57 names and occupies 6 pages.

The aim of the thesis is to develop a software product for detecting embedded code in computer games.

In my dissertation I analyzed the available information protection modules. Considered their reliability, pros and cons.

The developed software product increases the protection of computer games from the embedded code that can be observed by stops in games, and this makes it impossible to win.

Key words: pirates, tablet, attacker, breaking, trainer, scammer, software module, games.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІВ – інтелектуальна власність.

ТБ – технічна база.

КІ – комп’ютерні ігри.

АП – авторські права.

ВІ – внутрішньоігрові.

АЧ – античит.

ВСТУП

Актуальність. На сьогоднішній день ігрова індустрія перейшла на новий рівень. Якщо раніше ігри були потрібні лише для розваги, то вже зараз їх прирівнюють до спорту та влаштовують багатомільйонні турніри. На жаль, будь-який розвиток несе за собою нові способи обману та шахрайства, кіберспорт не став винятком. Так, з абсолютною впевненістю можна сказати, що саме так звані «кіберзлочини» є однією з найважливіших проблем сучасності, адже з кожним роком «кіберзлочини» стають дедалі масовими й небезпечними.

Метою дипломної роботи є розробка програмного продукту для виявлення вбудованого коду в комп'ютерних іграх.

Для досягнення мети, потрібно розв'язати такі **завдання**:

- аналіз нормативно-правової бази України з захисту інтелектуальної власності;
- аналіз програмних продуктів для зламу програмного модулю комп'ютерних ігор;
- розробка програмного продукту для виявлення вбудованого коду.

Об'єкт дослідження: процес захисту комп'ютерних ігор від вбудованого коду.

Предмет дослідження: методи та засоби захищеності ігор від вбудованого коду.

Галузь застосування. Розроблений програмний модуль захисту може активно себе проявити в захисті комп'ютерних ігор та в інших комп'ютерних програмах, які потребують захисту.

Практична цінність полягає в отриманні результатів, в ході дослідження та розробки коду програмного модуля для захисту комп'ютерних ігор.

РОЗДІЛ 1. ЗАХИСТ КОМП'ЮТЕРНИХ ІГОР, ЗА ЧИННИМ ЗАКОНОДАВСТВОМ УКРАЇНИ

1.1 Регулюючі закони України

Будь яка інтелектуальна праця охороняється законом. Законодавство про захист авторських прав базується на основі Конституції. Завдяки законам можна хоч якось захищати свою працю, але на жаль не завжди. Йдеться про наповнення різних сайтів, авторське програмне забезпечення, книгах, кіно або наукових роботах. Саме захист авторського права дозволяє забезпечити недоторканність власності з боку третіх осіб.

На сьогоднішній день законодавство дозволяє максимально ефективно боротися з людьми, які користуються чужою інтелектуальною власністю. Захист авторських прав дає можливість отримати солідну компенсацію в разі використання захищених продуктів, якщо вина зловмисника буде доведена. Серед винятків інтелектуальної власності варто відзначити:

- ідеї імена персонажів або назва глав книг;
- чисті факти;
- будь-яку інформацію, яка не була зафіксована на папері або інших носіях інформації.

Для захисту роботи за допомогою авторського права немає необхідності в додаткових і тривалих приготуваннях, як це буває у випадку з патентами. Необхідно просто залишити свою роботу на носії інформації і скористатися підписом «Copyright», ім'я, дата. В даному випадку захист авторських прав буде полягати в наступних процедурах:

- потрібно взяти плід інтелектуальної праці, вказати на носії всі атрибути, описані вище;
- носій повинен бути максимально довговічним.

Що тягне за собою порушення закону за ступенем важкості:

Кримінальна (див. Рисунок 1.1.)

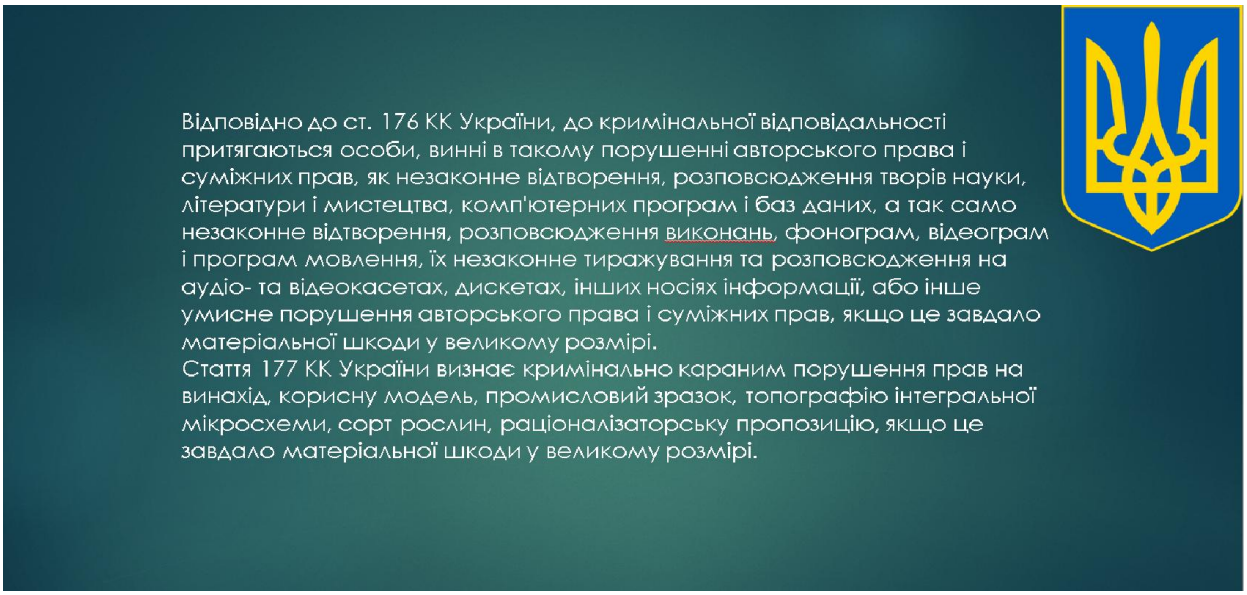

Рисунок 1.1 – Кримінальна відповідальність.

Адміністративна (див. Рисунок 1.2.)

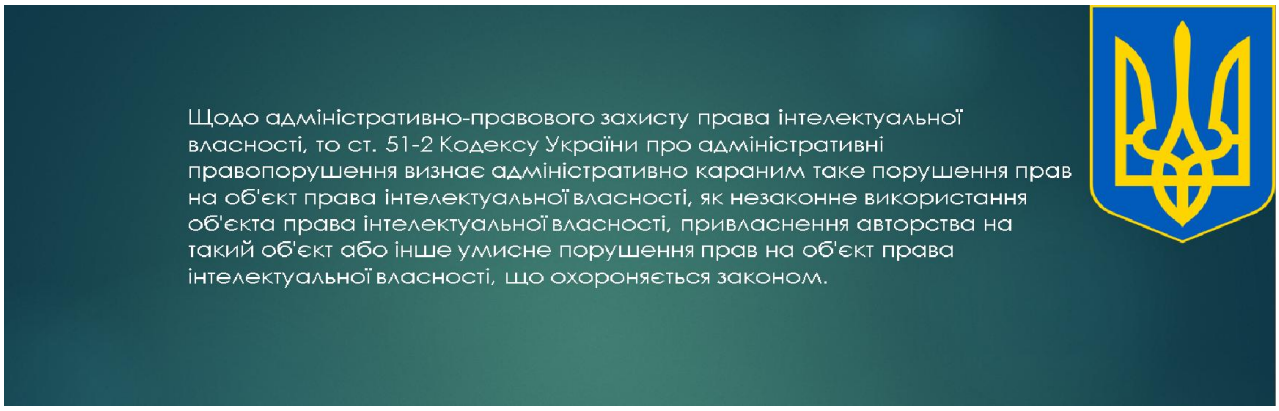

Рисунок 1.2 – Адміністративна відповідальність.

Цивільно-правова (див. Рисунок 1.3.)

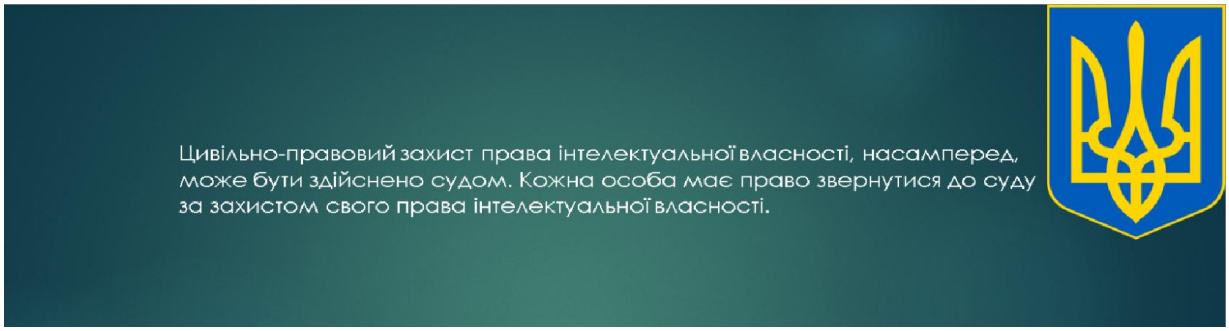

Рисунок 1.3 – Цивільно-правова відповідальність.

1.2. Загальні стандарти забезпеки інформації в світі

Згідно зі своїми основними завданнями Всесвітня організація: забезпечує адміністративне співробітництво союзів, створених у межах конвенцій, а також інших договорів, адміністративні функції стосовно яких виконує Всесвітня організація інтелектуальної власності, сприяє охороні інтелектуальної власності у всьому світі завдяки безпосередньому співробітництву з іншими організаціями з різних держав.

Збереження в цілісності ІВ в світі дана організація забезпечує багатьма способами, а саме:

- допомога країнам в розвитку ТБ;
- залучення нових міжнародних інвесторів, для розширення меж ІВ;
- розповсюдження нових даних;
- збір та контроль служби моніторингу, їх робота спрямована на охорону ІВ;
- розвиток усіх видів адміністративного співробітництва між країнами.

Всесвітня організація ІВ реалізує захист в 23 міжнародних організаціях, в різних галузях захисту прав ІВ. Ці договори об’єднують у три групи: договори, що забезпечують правову охорону різних об'єктів інтелектуальної власності в

різних державах; договори, що встановлюють міжнародні класифікації для різних об'єктів інтелектуальної власності; договори з охорони інтелектуальної власності, що визначають міжнародні стандарти з цього питання.

1.3. Жанри комп’ютерних ігор

Жанри КІ поділяються за рахунок характерних рис, властивій тій або іншій КІ. Цей підхід не є загальноприйнятим, але, цим підходом користуються як і творці КІ так і звичайні користувачі (див. Рисунок 1.4.) [1].

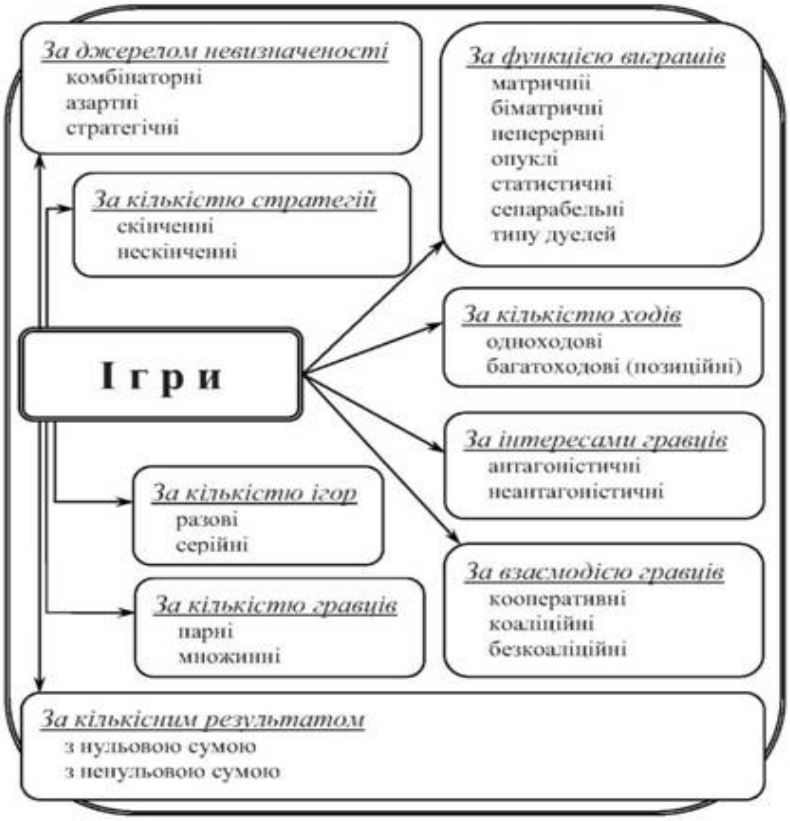

Рисунок 1.4 – Загальна класифікація комп’ютерних ігор.

Існує ряд основних жанрів КІ, а саме:

- Аркади (див. Рисунок 1.5.);

Рисунок 1.5 – Опис жанру «Аркади».

- Пригоди (див. Рисунок 1.6.);

Рисунок 1.6 – Опис жанру «Пригоди».

- Рольові ігри (див. Рисунок 1.7.);

Рисунок 1.7 – Опис жанру «Рольові ігри».

Розвиток широкосмугового Інтернету призвів до появи різновиду РПГ – масових багатокористувацьких рольових онлайн-ігор (ММОРПГ). У цьому виді ігрових програм віртуальні персонажі, керовані реальними гравцями, здатні взаємодіяти один з одним [2].

- Симулятор (див. Рисунок 1.8.);

Рисунок 1.8 – Опис жанру «Симулятор».

- Стратегічні ігри (див. Рисунок 1.9.);

Рисунок 1.9 – Опис жанру «Стратегічні ігри».

- Бойовики (див. Рисунок 1.10.);

Рисунок 1.10 – Опис жанру «Бойовики».

1.4. Проблеми читерства в кіберспорті

Кіберспорт є одним з найбільш популярних та масових видів спорту. Крім цього, кіберспорт – це бізнес, пов'язаний з отриманням прибутку від процесів пов'язаних з ігровою діяльністю, в тому числі незаконними шляхами, до яких можна віднести створення та розповсюдження чит-програм.

Читерство (англ. cheat – шахрайство,обманювати) – практика отримання нечесної переваги в багатокористувацьких комп'ютерних іграх зовнішніми програмами та нестандартне апаратне забезпечення. Це сприяє широкому поширення чит програм і як наслідок, неспортивної поведінки на змаганнях.

Гіпотеза дослідження: передбачалося вивчення питань знання проблеми читерства серед кіберспортсменів дозволить визначити ставлення кіберспортивної спільноти до досягнення ігрової переваги з допомогою чит-програм [3].

Мета дослідження: вивчити питання ставлення кіберспортсменів до читерства на етапі розвитку кіберспорту.

Завдання дослідження: вивчити громадську думку з питань: чи буде подальше вдосконалення ігрових технологій сприятиме розвитку читерства. Визначити найефективніші способи боротьби з читерством.

Об'єкт дослідження: суспільні відносини, що виникають при поширення та використання чит-програм.

Предмет дослідження: протидія розповсюдженню чит програм у кіберспортивному співтоваристві.

Методи дослідження:

- аналіз спеціалізованих інформаційних джерел;
- включаючи мережу Інтернет;
- метод анкетування.

Результати дослідження

На останньому турнірі з дисципліни, «counter-strike global offensive», організаторами було проведено опитування серед гравців та глядачів з наступних питань. У опитуванні прийняли участь близько 1000 чоловік:

1. чи буде подальше вдосконалення ігрових технологій?
2. сприяти розвитку читерства?
3. визначте ступінь покарання за читерство у сфері кіберспорту?
4. як ви вважаєте, який метод боротьби з читерами є найефективнішим?
5. проранжуйте античит-програми за рівнем ефективності.

Були отримані такі відповіді:

- на 1 питання 76% опитаних відповіли “так”, 20% “ні” та 4% "важко відповісти" ;
- на 2 питання 36% відповіли "довічна дискваліфікація", 28% "дискваліфікація на 1 рік", 20% "дискваліфікація на 2 роки", 12% "дискваліфікація на 3 роки" та 4% "усунення від гри на турнірі";
- на 3 питання 48% відповіли “розробка більш досконалих античит програм”, 40% “кримінальне покарання за розробку та розповсюдження чит програм”, 8% “просвітницька діяльність” та 4% “штраф від 1 000 000 карбованців”;
- на 4 питання: проранжуйте античит-програми за ступенем ефективності, були отримані такі результати:

1. Esl Wire;
2. Mail.ru Anti-cheat;
3. ESEA Anti-cheat;
4. Punk Buster;
5. Easy Anti-cheat;
6. Faceit Anti-cheat;
7. Battle Eye;

8. Valve Anti-cheat (VAC).

1.5. Висновок до першого розділу

На сьогоднішній день законодавство дозволяє максимально ефективно боротися з людьми, які користуються чужою інтелектуальною власністю. Захист авторських прав дає можливість отримати солідну компенсацію в разі використання захищених продуктів, якщо вина зловмисника буде доведена. Серед винятків інтелектуальної власності варто відзначити:

- ідеї імена персонажів або назва глав книг;
- чисті факти;
- будь-яку інформацію, яка не була зафіксована на папері або інших носіях інформації.

Для захисту роботи за допомогою авторського права немає необхідності в додаткових і тривалих приготуваннях, як це буває у випадку з патентами.

Необхідно просто залишити свою роботу на носії інформації і скористатися підписом Copyright, ім'я, дата. В даному випадку захист авторських прав буде полягати в наступних процедурах. Потрібно взяти плід інтелектуальної праці, вказати на носії всі атрибути, описані вище. Носій повинен бути максимально довговічним.

РОЗДІЛ 2. ВИДИ ПРОГРАМ ДЛЯ ПРОТИДІЇ ВБУДОВАНОМУ КОДУ ТА ВИДИ ВБУДОВАНОГО КОДУ

2.1. Види античитів

Можна, звичайно, поставити систему на кшталт «Denuvo» або аналогічної [4]. Або ж можна додати сильну прив'язку до серверів – тобто перевіряти активацію не тільки при запуску, але і після кожної дії. Але, як би там не було, рано чи пізно і це зламають. На мій погляд, кращий спосіб захистити гру від злому – надавати додаткові сервіси, робота яких можлива тільки при покупці гри (див. Таблицю 1.1)

Таблиця 1.1– Варіанти захисту ігор.

Система рейтингів і досягнень	Для гравців які люблять досягнення, це буде вагомим приводом купити оригінальну гру
Записувати збереження в хмарі	такий сервіс надають багато ігрових онлайн магазинів
Постійно робити тематичне оновлення	Наприклад під тематику Нового року, восьмого березня.

Проте кожного разу любителі безкоштовних ігор тільки посміювалися над ними. Зараз розробники схоже нарешті прозріли і почали все частіше відправляти своєрідні послання тим, хто відмовиться купувати ліцензію, а замість цього встановить піратську копію [5]. Поясню на прикладі однієї гри під назвою «Alan Wake», Замість того, щоб закодувати гру і просто закрити можливість запуску, боси проекту наказали вставити піратську пов'язку на око головного героя, якщо гравець запускає піратську версію. Оскільки сюжет переповнений драматизму і гострими моментами, ролики з пов'язкою на оці

виглядають просто безглуздо [6]. Але Алан Уейк не єдиний, хто сміється з «піратів» в буквальному сенсі. Як з'ясувалося, багато компаній хочуть підвищити популярність своєї гри, але не бажають втрачати потенційну клієнтську базу. Тому в подібних випадках, грати, можна, але дискомфорт все частіше змушує вас замислитися про придбання ліцензійної версії [7].

Надійним способом захисту інформації є інтернет магазин «Steam». (див. Рисунок 2.1.)

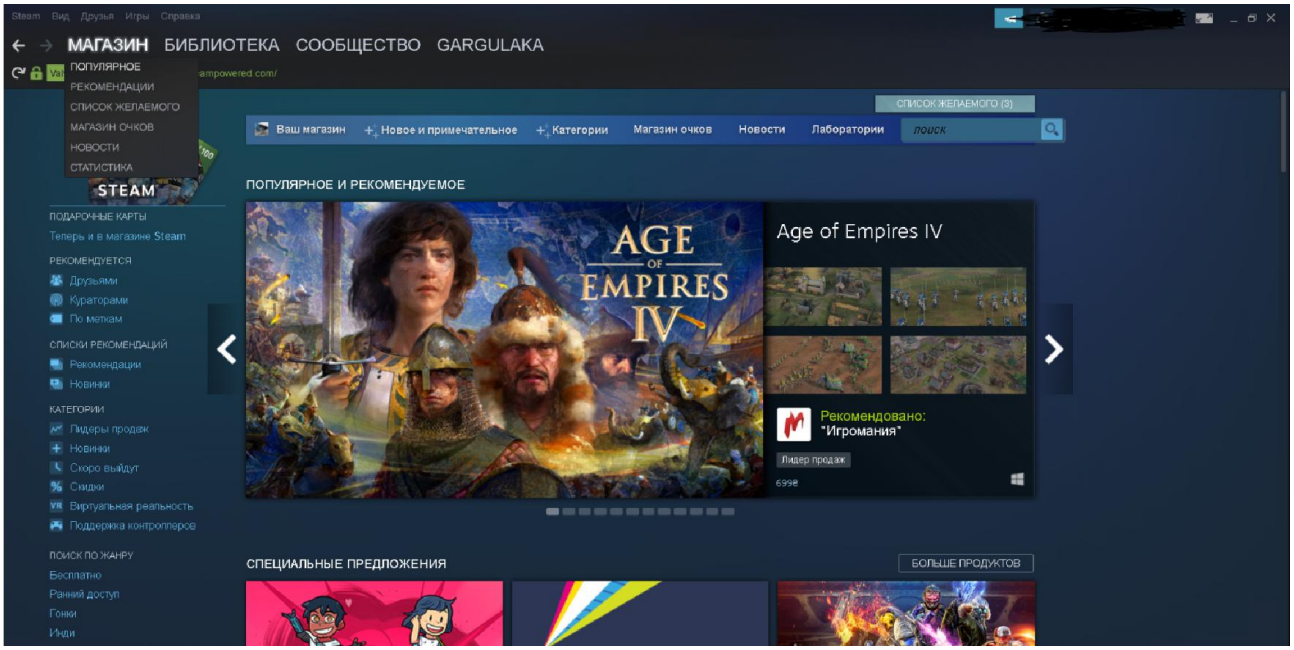

Рисунок 2.1 – Магазин «Steam».

Він виступає в ролі програмного засобу захисту АП (DRM). Надання регіональних цін – в кожному регіоні ціна відрізняється на одну й ту саму гру. Тому КІ, куплені в Україні (на фізичному носії), не запустяться в Європі. Такий хід, дає змогу видавцям унеможливити перепродаж своєї ІВ.

Активація ключів ігор. CD-key – це свого роду набір з 13 символів для активації гри в магазині «Steam» [8]. Як правило «CD-key» знаходиться в коробці з фізичним носієм, на зворотному боці номенклатури. Також в деяких випадках дані ключі можна використовувати і на сайтах розробників ігор [9]. Наприклад, при покупці в Steam гри, або будь-якого доповнення до неї

видається ключ (окремий для самої гри і окремий для кожного доповнення), який можна активувати на офіційному сайті гри для отримання SimPoints – валюти електронного магазину, в якому продаються додаткові предмети для даної гри.

Повернення можливе тільки протягом 14 днів і тільки якщо користувач провів у грі не більше двох годин.

Потокове завантаження – Steam підтримує потокову завантаження контенту. Це дозволяє розподілити пріоритети завантаження вмісту. Таким чином, спочатку завантажується частина гри, необхідна для запуску. Інші файли (в правильному порядку) – у фоновому режимі [10]. Завантаження рівня гри припиняється, якщо ще не завантажені необхідні файли. Потокове зміст вимагає додаткових зусиль з боку розробника, тому її активно використовують не так багато ігор.

Режим Big Picture – 10 вересень 2012 вступив в бета-тест, а 3 грудня вийшов офіційно. Big Picture – це режим роботи Steam, оптимізований для великих екранів телевізорів і управління геймпадом [11]. Натисканням однієї кнопки Steam буде перемикатися в повноекранний режим, оптимізований для зручності читання і використання на телевізорі, без клавіатури і мишки, хоча вони теж підтримуються [12]. Спеціально розроблені версії браузера і Товариства дозволять спілкуватися з друзями та подорожувати по інтернету без клавіатури, не відчуваючи ніяких проблем.

Магазин «Steam» має право на блокування аккаунтів користувачів, на свій розсуд, за:

- шахрайство з банківськими картками. Будь-які сумнівні дії з банківськими картками при купівлі в цьому магазині, включаючи використання кредитних карток інших людей, дані яких були викрадені;
- внесення на рахунок подарунків, придбаних карткою;
- піратство та шахрайство (хакерство). Сюди входить використання викраденого аккаунту онлайн магазину;

- крадіжка, обмін або торгівля рахунками інших людей;
- сумнівні дії з іноземними рахунками без відому власників;
- крадіжка та заміна пароля, використання облікового запису для входу в систему, будь-які операції з обліковим записом, розкриття інформації про акаунт тощо. Але обмін можливий по черзі;
- хакерські атаки та навмисні шахрайства користувачів. Це стосується шахраїв, які видають себе за іншу людину, та випитують інформацію, нібито працівниками магазину або компанії, щоб ввести пароль та іншу особисту інформацію облікового запису. Через це був введений фільтр псевдонімів, який не дозволяє створити прізвисько адміністрації магазину, підтримка, тощо [13];
- купівля та продаж викрадених рахунків. Відповідальність за використання та безпеку облікового запису покладається насамперед на оригінального користувача – усі облікові записи, які були придбані чи продані, компанією можуть заблокувати після з'ясування продажу;
- будь-яке інше порушення «Угоди про підписку з магазином» або «Кодексу поведінки Steam Online».

Крім того, іноді обліковий запис може бути тимчасово заблокований, якщо, наприклад, обліковий запис було вкрадено, а компанія заблокує його, поки не буде встановлений поточний власник [14]. Після блокування облікового запису користувач позбавляється можливості завантажувати та грати у всі раніше придбані ігри, пов’язані зі Магазином, включаючи фізичні версії. Гроші не будуть повернені, якщо рахунок заблокований.

Величезна кількість нечесних гравців з нечесними методами гри розгулює по сей день на просторах онлайн ігор. Щоденним приборканням їх популяції займаються анти-чити [15].

Найчастіше методи виявлення читів куди складніше і хитріше, ніж сам алгоритм читов, однак і ті, і ті, вічно удосконалюються і іноді без людського фактора не обійтися.

В одній з минулих статей я описав алгоритми роботи декількох популярних читів, зараз же опишу які бувають античити, як вони працюють і причому тут злом одиночних ігор. Античити умовно можна розділити на 3 типи:

- серверний;
- клієнтський;
- гібридний.

2.1.1. Серверний античит. Серверний античит працює як фільтр і створити його під силу звичайному розробнику, бо це лише набір перевірених умов, часто багатоетапний, щоб навантажувати планомірно і конкретно по відношенню до підозрюваного.

Беремо проміжок часу і знаходимо різницю в векторах позиції, якщо він вище будь-яких норм – бан [16]. Можна безпосередньо запитати у клієнта, яке значення змінної, що відповідає за швидкість і видати бан. Але і чити не дрімають – давно навчилися викликати штучні лаги, які незримо для читера і природні для сервера, в яких він не бачить нічого дивного [17].

Aimbot, наприклад, визначити теж досить просто:

Напрямок погляду це теж вектор і різка його зміна в бік голови ворога абсолютно очевидно викличе підозри навіть у бездушної машини, яка виконує математичні обчислення [18].

2.1.2. Клієнтський античит. Запускається разом з грою і працює на тлі, забороняючи взаємодію на процес з грою. Для більш повного розуміння давайте згадаємо як можна було отримати перевагу в одиночній грі за допомогою ArtMoney [19]. Гра містить у собі купу динамічно змінюваних значень, таких як наприклад здоров'я, мана, швидкість, гроші, вообщем залежить від гри. Такі значення зберігаються в оперативній пам'яті, доступ до якої ArtMoney і отримує (див. Рисунок 2.2).

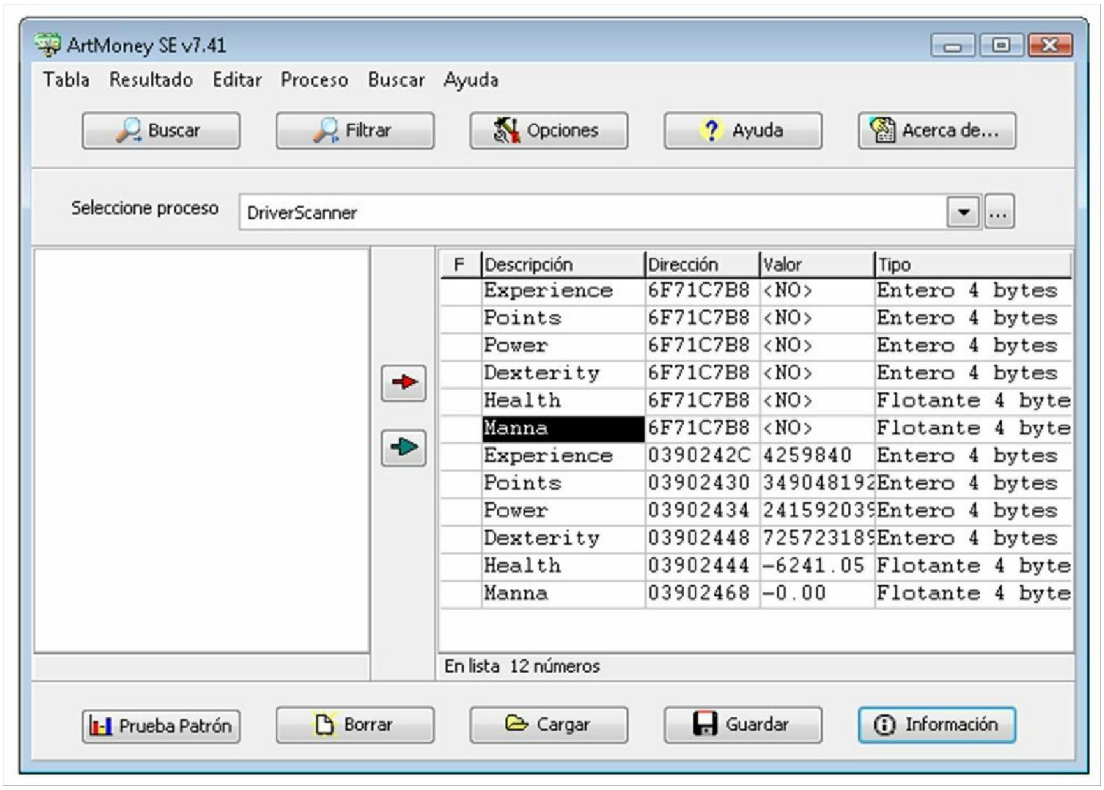

Рисунок 2.2 – Інтерфейс програми «ArtMoney»

Найпростіше редагування цих значень незахищеного процесу гри, як раз і дозволяє отримати перевагу. Як ви вже здогадалися, вектор напрямку погляду, швидкість та інші приблуди гри зберігаються саме там і чит їх змінює.

Захист процесів існує і на рівні ОС, проте цей захист не ідеальний: античити її доповнюють, забороняють доступ, приховують процес від вторгнень або просто посилають сервера сигнал, що мовляв тут один чоловічок намагається порушити природний хід буття, а значить – бан.

Значить це – ідеальний захист?

Приходить на пам'ять випадок, коли чит маскувався під популярний Easy Anti-Cheat просто змінивши назву, працювало все це досить довго і зараз можна зуміти і обдурити клієнт гри, давши йому необхідний мінімум даних, що дає зазвичай античит.

2.1.3. Гібридний античит. Гарним прикладом гібридного анти-чита буде розробка Valve – VAC (Valve Anti-Cheat). Це гарний приклад не як анти-чит, а як гібридний анти-чит (див. Рисунок 2.3.)

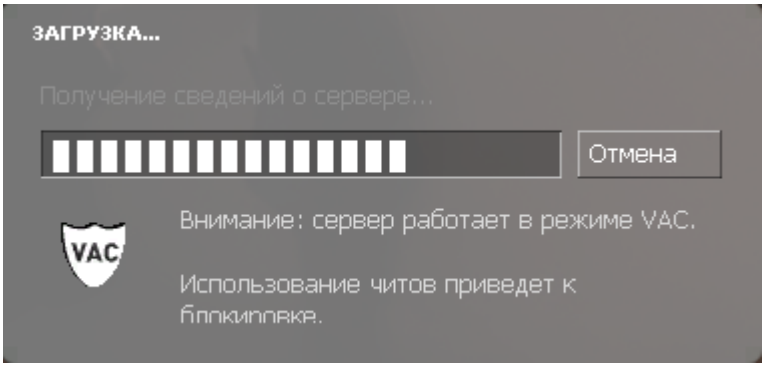

Рисунок 2.3 – Valve – VAC (Valve Anti-Cheat)

Багатоетапний метод обробки даних, який пролазить все далі, аж до перегляду DNS-адрес підозрілого користувача. Це означає, що компанія теоретично може дізнатися адреси сайтів, куди ви заходили, проте думаю їм це не потрібно. На місці машини, побачивши слово cheats серед купи мотлоху, в сукупності з купою попередніх підозр, я б дав бан [20].

До того ж зараз VAC вдає із себе самонавчальну машину по винищенню читера і працює на мій погляд цілком стерпно. Допомагають навчатися їй звичайні гравці, вдивляються в тисячі реплєєв підозрілих осіб щодня.

Шифрувати мережеві пакети, відстежувати всі непередбачені дії з боку гравців, банити гравців, у яких підозрілі процеси які лізуть в ваш процес [21].

Нормальні анітічити працюють на рівні драйверів, такі як GameGuard, зокрема для того що б можна було виявити і припинити втручання в процес, через те що ті, хто пишуть чити теж мають певні серйозні навички [22].

Загальний принцип написали, шифровка трафіку, захист пам'яті процесу – і це вимагає глибоких, часто не документованих знань [23].

2.2. Порівняння анти-читів

Зазвичай програмне забезпечення для читів вимагає активності, коли гравець хоче грати в Інтернеті з іншими гравцями [24]. Анти-чит може бути наданий власником гри, тому він може бути вбудований у гру, або його можна буде встановити як спеціалізоване програмне забезпечення, перш ніж гравець зможе приєднатися до певного ігрового сервера або взяти участь у турнірі. Різні програмні засоби проти обману можуть відрізнятися декількома способами, тому існують різні категорії служб проти обману (див. Таблицю 2.2.)

Таблиця 2.2 – Існуючі компанії та їх продукт.

Назва анти-читу	Компанія
«Valve Anti-cheat»	«Valve Corporation»
«Punkbuster»	«Even Balance»
«Warden»	«Blizzard Entertainment»
«GameGuard»	«Inca Internet»
«HackShield»	«AhnLab»
«ESL Wire»	«Turtle Entertainment»
«ESEA Client»	«E-Sports Entertainment»
«EasyAntiCheat»	«EasyAntiCheat»
«BattleEye»	«BattlEye Innovations»

Інтеграція на стороні клієнта – Залежно від того, наскільки інтегрована клієнтська частина анти-читів на гравця, технічно ми можемо виділити три категорії:

- нульова інтеграція клієнтів. Анти-чит без будь-якого коду, що працює з боку клієнта. Зазвичай на основі аналізу поведінки клієнта шляхом перевірки мережевих даних. У такому випадку клієнт не має доступу до захисної програми, кодів і тому, чіт не можуть бути виявлені;
- інтеграція клієнта в режимі користувача. Анти-чит, реалізований як запущений процес або системна служба. Має обмежений доступ до пам'яті комп'ютера;
- інтеграція клієнта в режим ядра. Анти-чит, реалізований як драйвер, працює в адресному просторі ядра, а отже, має доступ до всієї пам'яті комп'ютера. Найбільш ефективне рішення, але можливо джерело нестабільності системи.

Anti-cheat також можна розділити на кілька модулів, при цьому кожен модуль має різний рівень інтеграції. Наприклад, гра може бути захищена послугою режиму користувача, з додатковим рівнем режиму ядра та аналіз на стороні сервера вхідних даних клієнтської мережі [25].

Інтеграція ігрових серверів

Щоб бути ефективним, анти-чит повинен бути інтегрований в ігрові сервери де б він забороняв доступ до гри, гравцям, яким заборонено грати в Інтернеті. Він також використовується для підтвердження того, що частина клієнта пройшла ігрову активацію та є функціональною під час підключення до захищеного ігрового сервера. Цю мінімальну функціональність можна також розширити функціями, які дають адміністраторам ігрових серверів можливість виконувати додаткові анти-чит перевірки конкретного гравця. Наприклад, просити захоплення гравців ігровий екран [26].

Система блокування користувачів

Як тільки анти-чити виявляють чит, дія може бути негайно виконана, або, скоріше, може бути виконана в майбутньому. Можуть бути різні причини, чому затягують дії проти гравця, який був виявлений, щоб все обдурити. Це може бути або технічною причиною, щоб створити простір для ручного аналізу виявлення, щоб переконатися, що виявлення було правильним, або це може бути партнером довгострокової стратегії протидії обману. Зазвичай заборони постійні або тривають останній рік [27].

Ліцензія та ціни

Зазвичай розробник ігор покладається на службу сторонньої протидії обману розвитку власного продукту. Існують різні постачальники проти обману, які мають різні ліцензії та ціни. Для них типово надавати свої рішення проти читів. В одній грі зазвичай є один анти-чит, але залежно від вимог власник ігрового сервера може використовувати анти-чити, надані розробником гри, та додатково придбати ліцензію на інший анти-чит в якому більш високий рівень інтеграції на стороні клієнта, щоб, наприклад, забезпечити кращу безпеку на турнірі. Якщо буде виявлено іншу службу проти обману [28].

Активні в комп'ютерних анти-читах гравця, як правило, коригують свою поведінку, щоб запобігти помилковим виявленням.

Користуючись сторонніми послугами проти обману, розробник гри, як правило не має доступу до жодної адміністративної частини анти-читів. Якщо він хоче, щоб анти-чит виявив конкретного шахрая, він повинен надіслати її компанії хто надає анти-чит – сервіс для аналізу [29].

2.3. Види читів

Для тих, хто не хоче чесно грати в ігри, створено чимало програм, які можуть зламати будь-яку гру і змінити в ній що завгодно, таких людей називають шахраями (Читерами). Читери – «це люди, які нечесним шляхом

отримують перевагу в багатокористувацьких та одиночних іграх за допомогою зовнішніх програм і нестандартного ПЗ». З різних причин. Заради грошей – зламані ігри продають дешевше, ніж у авторів, плюс дохід від реклами. З задоволення – багатьом хакерам подобається сам процес. Для тренування – треба ж на чомусь практикуватися, тим більше захист постійно вдосконалюють. Ще бувають замовлення конкурентів або ж самих авторів для перевірки вразливост [30].

Навіщо вони це роблять? Все заради перемоги. Найпоширеніша причина використовування програм для несанкціонованого доступу до ігор – небажання програвати. Для цих гравців сенс гри в перемозі, і не важливо, що досягнута вона без особливих зусиль і розвитку. На відміну від чесних гравців, які отримують задоволення від самої гри і прикладають зусилля для досягнення рівнів / рангів / титулів, обманщики не можуть прийняти поразку і тішать своє самолюбство нечесними перемогами.

Як вони це роблять? Онлайн-ігри, на відміну від одиночних, складаються з клієнтської частини, що працює на комп'ютері користувача, і сервера. Змінити сервер не можна, тому читери модифікують сам клієнт гри або втручаються в обмін даними між клієнтом і сервером. Зробити чит для онлайн-гри складніше, ніж для одиночної. Сервер перевіряє багато дій гравця, тому доводиться шукати слабкі місця в цих перевірках.

2.3.1. Огляд програм для несанкціонованого доступу. Існують різні види стороннього коду. Можна розділити їх на кілька груп.

- **External** – зовнішній сторонній код, який працює в окремому процесі. Якщо ж ми приховуємо наш external–код, завантаживши його в пам'ять іншого процесу, він перетвориться в hidden external;
- **Internal** – внутрішні коди, які вбудовуються в процес самої гри за допомогою інжектора. Після завантаження в пам'ять гри в окремому потоці викликається точка входу програми;

- **Pixelscan** – вид програмного коду, який використовує картинку з екрану і патерн розташування пікселів, щоб отримати необхідну інформацію від гри;
- **Network proxy** – програмний код, якій використовують мережеві протоколи, ті, в свою чергу, перехоплюють трафік клієнта і сервера, отримуючи або змінюючи необхідну інформацію.

Шахрайство одиночних іграх – це нешкідливі шахраї, так як нікому вони шкоду заподіяти не можуть. Так що ж ними рухає?

- По-перше, налагоджувальними кодами користуються самі розробники для тесту локацій, рівнів і зброї, як приклад;
- По-друге, шахраї, в одиночних іграх, ризикують тільки одним – в багатьох іграх, які поширюються через інтернет магазини, вони позбавляться досягнень, хоча використання трейнерів в одиночній грі не було зафіксовано.

Як правило, використовуються такі програми, як:

2.3.2. Cheat Engine. «Cheat Engine» – призначена переважно для взлому КІ, розроблена для операційної системи Windows. Розповсюджується безкоштовно у вигляді «Open Source» проекту. Програма сканує оперативну пам'ять КІ з широким спектром опцій для знаходження в пам'яті ігрових значень (здоров'я, гроші, золото та ін.) І можливості їх зміни. Cheat Engine включає в себе сканер пам'яті, засіб для перегляду пам'яті, відладчик, дизассемблер, асемблер, можливість управління Direct3D, інструменти для контролю системи та інші (див. Рисунок 2.4.). Cheat Engine за допомогою відладчика, дизассемблера і інших інструментів дозволяє вносити зміни в програмний код, що може дати такі переваги в іграх як нескінченне здоров'я, час і боєприпаси. Також програма має Direct3D засобами, які дозволяють бачити в грі крізь стіни, збільшувати / зменшувати зображення і з певною розширеної конфігурацією.

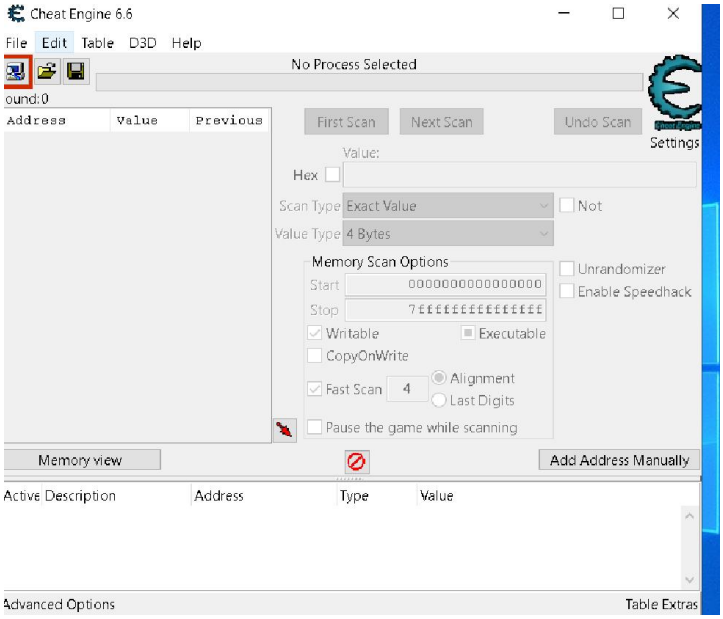

Рисунок 2.4 – Інтерфейс програми «Cheat Engine»

Cheat Engine впроваджує свій програмний код в ігри на комп'ютері, що може привести до конфлікту в системі, і антивірусне програмне забезпечення може порахувати програму шкідливою. Програма Cheat Engine може самостійно на основі таблиць сконструювати свій трейнер, який буде працювати автономно від самої програми і ця функція постійно розвивається.

Однак, трейнери, створені за допомогою програми Cheat Engine, дуже великі в розмірі і повільно виконуються, тому в основному використовуються для тестових цілей, так як повної гарантії роботи дати не можуть, у зв'язку зі своєю недопрацьованою програмною частиною.

2.3.3. ArtMoney. «ArtMoney» – комп'ютерна програма з закритим вихідним кодом, призначена для модифікації параметрів комп'ютерних ігор, для отримання нескінченних віртуальних грошей, життів, патронів і тому подібних ресурсів (див. Рисунок 2.5). ArtMoney не може редагувати параметри в мережевих або онлайн-іграх, бо в цьому випадку дані зберігаються на сервері, а програма може змінювати дані, що зберігаються тільки на локальному комп'ютері користувача [31].

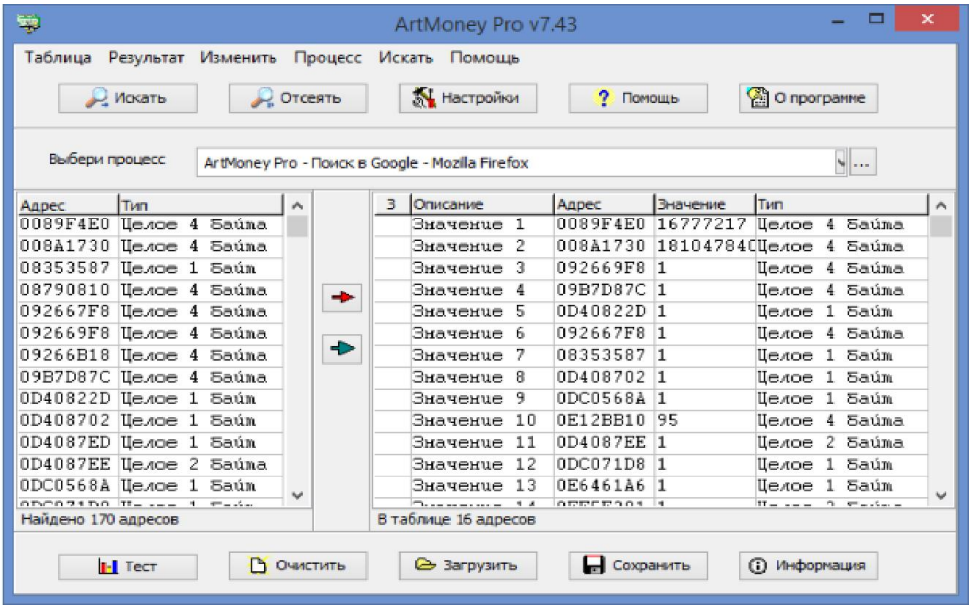

Рисунок 2.5 – Інтерфейс програми «ArtMoney»

Програма ArtMoney не порушує законодавства, вона не призначена для злому паролів, серійних номерів, захисту від копіювання. Робота програми здійснюється двома основними способами:

- **пошук в пам'яті процесу гри**. Користувач повинен вибрати в спеціальному вікні (аналогічному вікна «Диспетчера завдань» Windows)

запущений процес гри;

- **пошук в файлі**. Гра може зберігати призначені для користувача дані у файлі конфігурації. Користувачеві потрібно виявити такий файл, і проводити операції вже з ним.

Для використання програми не потрібно ніяких спеціальних знань в галузі інформатики, програмування або архітектури адресації пам'яті. Інтерфейс запропонує користувачеві покрокові дії і допоможе досягти позитивного результату за кілька хвилин.

У загальному сенсі, принцип роботи програми побудований на методі «грубої сили» (Брутфорс) – користувач вказує програмі необхідне значення (або діапазон значень), а програма перебором шукає осередки пам'яті, що містять вказані значення, відсіваючи непотрібні.

ArtMoney і гра повинні бути одночасно запущені. Шляхом впровадження в процес або в файл гри, ArtMoney виявляє осередки адрес пам'яті, які відповідають за зберігання призначених для користувача налаштувань гри: час (таймер), кількість грошей, сили, життів, патронів і т.д [32]. Припустимо, користувачеві потрібно змінити в грі кількість патронів , що дорівнює 100. Спочатку потрібно провести пошук осередків пам'яті, які містять значення «100», потім в грі змінити кількість патронів, припустимо до 80. Після цього зробити відсів тих осередків, значення яких змінилося з 100 до 80. Якщо остаточних осередків буде занадто багато, процедуру потрібно повторити спочатку. Коли потрібну адресу виявлено і зафіксовано, користувач може змінити його в інтерфейсі.

Трейнери. Модифікація ігрових даних – Переваги можна легко домогтися шляхом зміни даних гри. Подібні методики часто менш надійні, ніж використання чит-кодів, включених в гру її творцями. Це пов'язано з тим, що деякі стилі програмування або химерна відео-ігрова логіка, різні версії гри або навіть запуск однієї і тієї ж гри в різний час або на різних апаратних конфігураціях, можуть привести до конфліктів і помилок. Отже, програма

трейнера може не набути ніякого ефекту або взагалі не дозволить гравцю запустити гру [33]. Також слід пам'ятати, що зміна ігрових даних зазвичай є порушенням ліцензійної угоди, в якому часто прописують заборона на модифікацію файлів. Найпростіший спосіб вклинитися в ігрові дані – заручитися допомогою програмного забезпечення для редагування пам'яті, яке дозволяє гравцеві безпосередньо редагувати числові значення за певною коміркою пам’яті [34]. Цей вид програмного забезпечення зазвичай включає функцію, яка дозволяє виконувати пошук в пам'яті, щоб допомогти користувачеві знайти області, в яких розташовані відомі значення (такі як кількість життів, кількість патронів, шкоди від зброї в чисельному еквіваленті). Якщо ресурси гри не зашифровані, можна виконати модифікацію і вручну, відкривши потрібний файл за допомогою текстового редактора. (див. Рисунок 2.6.)

Рисунок 2.6 – Інтерфейс трейнера.

До модифікації ігрових даних також відноситься підробка мережевого трафіку, яка реалізується набагато складніше гри з програмним модулем і кодами. Подібний метод обману в онлайн-іграх включає в себе редагування, відправляються пакети для зміни вихідного мережевого трафіку, що робить прямий вплив на гру. У минулому шахраї активно використовували цю лазівку,

але в наш час ігри розробляються з захистом проти мережевих і пакетних модифікацій.

Незвичайні ефекти – програмні коди іноді можуть викликати незвичайні або цікаві ефекти, які не обов'язково роблять гру більш легкою. Наприклад, один код міняє зображення. Інший показовий приклад – Dungeon Siege, де активування чита для розширення діапазону лука також дозволяє ворогам стріляти на такій же відстані, тим самим усуваючи перевага, яке дав чит. Введення чита може навіть ускладнити гру: дати ворогові особливі здібності, збільшити загальну складність, змусити нейтральних спостерігачів атакувати гравця або зробити персонажа вразливим, занизивши його здоров'я до мінімально допустимого рівня. У цю ж категорію можна віднести приховані послання залишені розробниками гри. Хоча такий прихований контент не впливає на гру, він може натякати на майбутні ігри в серії або давати більше різноманітної інформації. Деякі приховані предмети можуть бути знайдені тільки за допомогою обманних заходів, таких як режим noclip (проходження крізь стіни) або розтин ігрових ресурсів стороннім софтом. (див.Рисунок 2.7.)

Рисунок 2.7 – Чит «Noclip».

Шахрайство в мережевих іграх – це онлайн-гра, здебільшого з іншими реальними гравцями, маючи на увазі деякі соціальні аспекти обману в онлайн-іграх. Було встановлено, що мотивація для шахраїв індивідуальна всі шахраї, однак мотивації можна розмістити в трьох різних великих групах. Перша мотивація – суто грошова.

Призовий фонд в онлайн-іграх збільшився протягом останніх років, а також точності віртуальних товарів, які можна придбати в більшості онлайн-ігор. Віртуальні товари – це, наприклад, скіни на зброю в CS: GO які можуть коштувати дуже дорого коштувати. (див. Рисунок 2.8.)

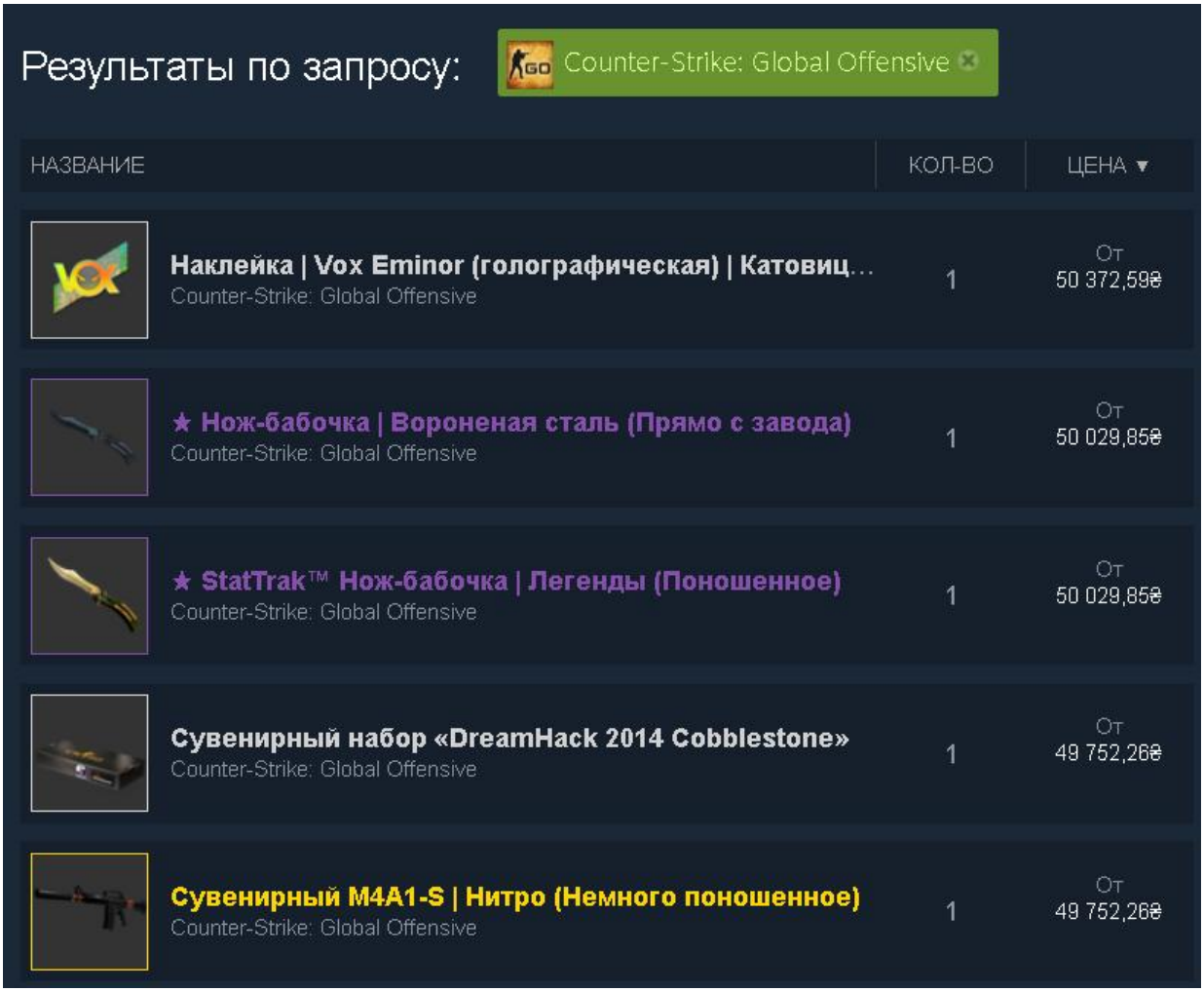

Рисунок 2.8 – Офіційний магазин скінів в «STEAM».

Майже неможливо знайти успішну онлайн-гру, в якій немає шахраїв. Іншими словами, якщо у вашій публічної грі немає шахраїв, вона або недостатньо популярна, або розпізнавання шахраїв працює не дуже добре. У всіх інших випадках вам доведеться мати справу з «Читерством».

Друга мотивація – отримати перевагу, змагаючись у потягу до перемоги, це часто дуже конкурентоспроможні люди. Нарешті, третя мотивація для

шахраїв – просто вміти розважатися та ставати кращими у грі, не вкладаючи часу та сил на гру. Компанія-розробник чіт-коду складає близько 50 000 доларів щомісяця [36].

Щось можна побачити, це те, що у шахраїв більше друзів, які обманюють, ніж у звичайних геймерів. Якщо шахрай потрапляє в будь-яке програмне забезпечення, яке захищає від шахрайства, вони більш обмежують налаштування конфіденційності, оскільки не хочуть показувати це ігровому співтовариству. Це свідчить про те, що шахрай знає, що обман не приймається ігровою спільнотою. Мережеві ігри, на відміну від одиночних, складаються з клієнтської частини, що працює на комп'ютері користувача, і сервера.

Змінити сервер не можна, тому шахраї модифікують сам клієнт гри або втручаються в обмін даними між клієнтом і сервером. Зробити програму для онлайн-гри складніше, ніж для одиночної. Сервер перевіряє багато дій гравця (все не дозволяє поточний рівень інтернету) – доводиться шукати слабкі місця в цих перевірках.

Серед найпопулярніших видів програм або стороннього коду, є:

- Wallhack – завдяки йому чітери наперед знають, які позиції потрібно перевірити і як уникнути небезпеки, що знаходиться попереду. Як би ви не намагалися, за рахунок інформації про ваше місцезнаходження нечесний гравець із ймовірністю в 99% здобуде над вами перемогу у перестрілці. Комунікація та передача інформації – важлива складова геймплея Counter-Strike. Без цього вам важко досягти перемоги. Проте гравці з Wallhack обходять це стороною: вони й самі можуть обіграти всю ворожу команду. Проти них марно щось вигадувати та намагатися їх здивувати. Вони завжди на 10 кроків попереду;
- Aimbot – за прикладом Aim Assist у мобільних та консольних шутерах, спрощує прицілювання. З ним приціл відразу наводитись на модель суперника, і так чітери можуть відразу почати стрілянину, не боячись промахнутися хоча б навіть на міліметр від своєї мети. Найгірше наступне:

Aimbot можна налаштувати так, що ви навіть не відразу зрозумієте присутність читера. Не всі вони ставлять прицілювання на голову; не кожен закріплює приціл на ворогові, щоб отримати ідеальний постріл;

- Spinbot – найдурніший чит із усіх існуючих. При активації Spinbot гравець починає безперервно крутитись на 360 градусів і після кожного пострілу ставить хедшот. Коли побачиш таке, чинити опір марно. Якщо у випадку з Wallhack і, в деяких випадках, Aimbot ви ще можете зреагувати швидше за читера, то тут результат матчу вирішено відразу;
- Speed Hack – дає змогу читеру пересуватися та виконувати дії набагато швидше ніж інші чесні гравці;
- Чит 3D-BOX – за допомогою цього софту гравці наперед бачать своїх ворогів, адже навколо них малюється спеціальна «коробка». Завдяки їй чітери можуть помітити опонентів, що стоять за довколишніми стінами та кутами. Примітно, що схожий на цей чит функцію можна налаштувати завдяки консольним командам;
- GOD MOD (Noclip) – цей чит значно полегшує ігровий процес для нечесних гравців. З софтом God Mode вашого ворога неможливо засліпити, і він завжди бачитиме крізь димові гранати. Також програма прибирає йому спрей. Всі кулі летять у напрямку прицілу, і це робить гру неймовірно легкою навіть для новачків. Однак у цьому випадку чітери не вбивають вас за 1 секунду, як за Aimbot або Spinbot, і вони не знають про ваше місцезнаходження. Їх можна обіграти;
- Спливаючі оповіщення – у грі існують сповіщення, про початок та кінець раунду, про встановлення бомби, тощо… В згаданій функції, у вас на екрані періодично спливають різні оповіщення. Тільки ось з читом вони говорять вам не про те, що ви забули бомбу, а про ворога, що знаходиться за стіною. Уявімо, що ви виходите з ями на Mirage. І якщо за рогом стоїть гравець CT, чит вас про це попередить. І тоді вже нескладно вийти і зробити смертельний постріл по опоненту;

- Чит ESP – можна вважати це Wallhack версії 2.0. Крім розташування ворога, чит розкриває масу іншої корисної інформації. Софт показує, скільки у гравця залишилося очок здоров'я та патронів, яка зброя в нього в руках і навіть які гранати він купив перед раундом. Навіть звичайний гравець може отримати з цього користь. Він побачить пролом у захисті суперника: опорника з низьким запасом HP або дешевою зброєю. Через таких пробитися набагато простіше, ніж цілком здорових та озброєних захисників з іншими гвинтівками [37] (див. Рисунок 2.9.).

Рисунок 2.9 – Інтерфейс включення читів.

Шахраї модифікують або машинний код гри, або скрипти, або дані. Зміни вносять або в файли на диску, або в пам'ять запущеної гри. Для автоматизації внесення змін пишуть спеціальну програму, яку і називають «читом». Конкретні місця для внесення змін знаходять шляхом реверс-інжинірингу(дослідження деякого готового пристрою або програми), або шляхом пошуку закономірностей в даних гри та в пам'яті. Як правило, гра заснована на одному з відомих програмних забезпечень, що спрощує дослідження файлів [38].

Від шахрая немає якогось універсального методу боротьби, який раз і назавжди змусить його грати чесно. Але виробники ігор в силах контролювати ситуацію. Вони захищають свої сервера і вихідні коди, розробляють програмні модулі захисту і встановлюють «пастки» для нечесних гравців. Розробники створюють цілі відділи по боротьбі з шахраями, і в ньому не тільки розробники. Покладатися виключно на програмні модулі захисту, марно. Розробники стежать за відомими майданчиками по поширенню шкідливих програм(Читів), виявляють ошуканців за скаргами гравців, аналізують нові програми і програмне забезпечення, відразу вносячи зміни в програмний модуль захисту. Щоб максимально швидко отримати доступ до «закритих» програм, також розробники впроваджуються в закриті групи і купують ці самі програми, щоб розібратися з її кодом і знайти спосіб захистити гру [39].

Є три основні тактики модифікації поведінки гри:

- зміна пам'яті гри. API операційної системи використовується для пошуку і зміни ділянок пам'яті, що містять потрібну нам інформацію (наприклад, життя, патрони);
- симуляція дій гравця: додаток повторює дії гравця, натискаючи мишкою в заздалегідь зазначених місцях;
- перехоплення трафіку гри. Між грою і сервером встає чит. Він перехоплює дані, збираючи або змінюючи інформацію, щоб обдурити клієнт або сервер.

Покупець відправляє гроші на електронний гаманець продавця, де звичайно ж не обов'язково проходити верифікацію, і чекає, коли продавець відправить предмет – що насправді виливається в відмова та блокування всіх контактів. У разі відмови покупця оплачувати товар, шахраї можуть заблокувати користування пристроєм і почати вимагати кошти, оскільки користувач не виявив пильності і скачав разом із зображенням шкідливе ПЗ. Згідно з даними дослідження, команда відділу захисту «MY.GAMES» [40] виділила 4 типи потенційно заборонених програм за складністю механізму їх

роботи і зробила вибірку для подальшого дослідження. Фахівці порівняли із річною статистикою блокувань у грі. І ось результати: Найбільш прості чити – 13,21% від загальної вибірки. Вони вбудовуються в процеси клієнта гри після впровадження (інжекту) [41].

Чити середньої складності – 16,98% від загальної вибірки. Вбудовуються в інший процес, а доступ до клієнта гри отримують, використовуючи проломи в античиті, або за допомогою драйверів. Чити високої складності – 3,77% від загальної вибірки. Ця категорія вбудовується в інший процес, а доступ до клієнта гри отримує за допомогою самостійно створених і встановлених програмних засобів в ядрі ОС. "Фейкі" – 66,04% від загальної вибірки склали файли, які позиціонувалися як чити, однак самих читів у собі не містили. Більшість використовуваних читів їх модифікацій, додаткового контенту не є офіційним доповненням розробників, а значить може піддати загрозі особисті дані, що містяться в інвентарі профілю або файлів на персональному пристрої. Розробники даних читів не дають гарантій щодо безпеки їх використання, можливості не виявлення систем безпеки. Оскільки опис і реклама всього на все – фікція, для продажу та поширення свого забороненого ПЗ [42]. Будь-який чит у існуючий час виявляється, «патрулем» чи античитом, рано чи пізно – питання часу. Для боротьби і читерами ігрові компанії створили античит – програмне забезпечення, що контролює ігровий процес користувача (геймера) припиняє сторонній вплив на гру. Антич працює з процесами гри, звіряючи їх зі своєю базою даних для виявляючи стороннього ПЗ, які вбудовуються в процес гри або має зовнішню програму для обробки даних, яку Досить важко виявити.

Процес виявлення читів залежить від виду античита, розповімо про найпоширеніші. Перший це клієнтський, скачується і працює на сам комп'ютер, пристрій перевіряє систему на наявність сторонніх програм. Такий античит сприяє додатковому навантаженню на потужності. комп'ютера. Другий вид популярного античита – серверний, містить у собі сукупність інструментів обмежень та звіряє потік даних від користувача. Так само є нейронна мережа, часті античита, що розробляються для замовлення або продажу ігровим.

компаніям, організаторам турніру. Даний вид читів рідко використовується, але більш ефективний метод боротьби. Великі компанії уникають цього, тому що їм важлива кількість онлайн гравців у мережі. Не всі компанії вживають достатніх заходів, щодо стримування чітерів, покращення ігрового сервісу, так як це позначиться на онлайн-гравців, серверах гри. Тобто популярність та додаток на прибуток дуже важливі для компанії. Найкращий варіант – це просто додавання нових внутрішньогрових предметів для продажу та обміну між гравцями, покращення візуальних локацій та ефектів для насиченої деталізації та атмосфери. Ще однією проблемою є низька «кіберспортивна свідомість» геймерів, відсутність засобів поширення та навчання користувачів основ безпеки та правил поведінки (етики) в кіберспортивному просторі та протидії різним віртуальним загрозам [43].

Оскільки код налагодження не є безпечним. Більшість доступних читів, що скачуються і купуються в інтернет-просторі, містять у собі:

- рекламне ПЗ;
- приховані трояни для відстеження дії користувача та віддаленого керування комп'ютером;
- крадіжки облікових записів та платіжних даних;
- проникнення у файли програм та ігор для віддаленого доступу;
- встановлення та скачування інших шкідливих програм, що уповільнюють швидкодію комп'ютера;
- використання вашого пристрою для майнингу.

Найгірша в світі на даний момент ігрова захищеність від чітерів так і ігрового сервісу – є шутер CS:GO (Counter Strike Global Offensive), компанія Valve володіє нею, досить мало приділяє час безпеки античита і його поліпшення, так як куди вигідніше зробити гру доступною для всіх – безкоштовною, додавати нові всередині ігрові предмети та контент для різноманітної візуалізації профілів гравців, виправлення багів локацій.

Тому навіть античит гри VAS виявить налагоджувальний код (чит), і заблокує акаунт. Ні особливих складнощів для його купівлі нового на сторонніх сайтах, що заборонено угоду між Valve та користувачем. Тому продаж прайм акаунтів (довірених профілів), не є проблемою щоб зайти до гравців, які пройшли перевірку та використовувати проти них чити. Адже функції читів досить різноманітні, і рекомендацій щодо їх використання теж – знімаються блоги та є групи та форуми у соціальних мережах та сайтах з уникнення виявлення блокування внутрішньогрових механізмів захисту чи недоліки самої гри. Але рано чи пізно згідно з великими кількостями скарг або знаходження процесу впровадженим у код гри, античит виявляє даний процес звіряє його зі своєю базою даних та видає блокування користувача гри [44].

Покарання на даний момент залежить від локальних правил компанії або гри. наприклад, у Steam, крім блокування та заборони грати на захищених серверах користувач акаунту не зможе обмінюватися та купувати всередині ігрові предмети, тобто вони також блокуються назавжди. Бездіяльність даної компанії щодо кібербезпеки гравців, змушує ігрові компанії та організаторів турнірів, наймати та використовувати сторонні системи безпеки, які ефективно показують себе інших платформах. Наприклад наприкінці 2019 року, розробники гри «WARFACE» та "ЛАБОРАТОРІЯ КАСПЕРСЬКОГО" об'єднали зусилля зі створення свого спільного античита, під назвою – Kaspersky Anti-Cheat, який здатний виявити 100% заборонених програм. Що набагато ефективніше, на відміну від VAC компанії Valve. Продукт «Лабораторія Касперського» почнуть використовувати на турнірах від StarLadder, включаючи PUBG Europe League, що говорить про серйозної репутації продукту [45].

2.4. Майбутнє античитів

Valve використовує глибоке навчання, щоб ловити чітера. За словами Джона Макдональда з Valve. Поточна система машинного навчання Valve

VACnet виявляє шахраїв і потім відправляє випадки для системи Overwatch для гравців, щоб розглянути і судити [46].

Макдональд заявив що реалізація VACnet підвищила рівень переконаності шахраїв в Система спостереження від 15% – 30% до 80% – 95% [47].

Це приголомшливе збільшення, яке явно показує, що VACnet виявляє людей, які явно обманюють. Проте, Макдональд також зазначив, що поточна реалізація «VACnet» може зловити тільки гравців, які явно обманює зі значним відривом. Наприклад, якщо шахрай встановлює зброю, щоб вдарити після кожного пострілу система машинного навчання виявить це. Якщо шахрай робить зброя вражає тільки 60% часу, тоді система може не зловити його, так як воно не дуже очевидно (див. Рисунок 2.10.) [29].

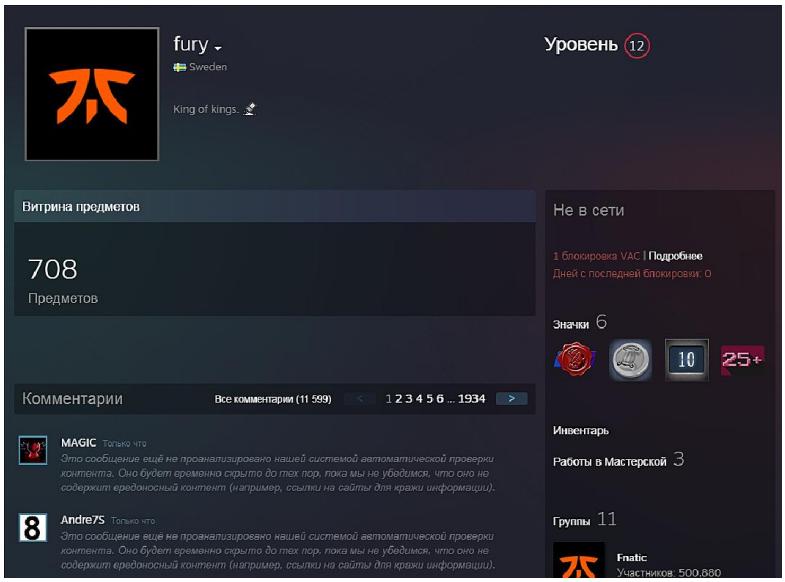

Рисунок 2.10 – Заблокований гравець.

На цьому наголошує слабкість поточної реалізації. З великою кількістю часу і більше даних, точність системи повинна покращитися. Easy Anti-Cheat – хороший приклад античит-системи SaaS (Програмне забезпечення як послуга) частково працює в хмарі і частково на клієнтському комп'ютері. Це

використовується в багатьох популярні ігри, такі як Fortnite і чимось схожі на старі програми, такі як:

- «nProtect»;
- «GameGuard»;
- «Hack Shield».

Багато старі системи, такі як nProtect GameGuard, діяв як руткіти, коли вони були встановлені на клієнтському комп'ютері, що сьогодні сильно піддався критиці з-за дуже шкідливою інвазивної функціональності. У розділі 5, розділ 4, ми Коротко розглянув Fairfight, Антич-систему, яка працює виключно на стороні сервера. Він також використовується в багатьох популярних іграх, таких як Battlefield V. Easy Anti-Cheat і Fairfight [48] представляють собою наступне покоління антічита, які хоча б частково працюють в хмара і які використовують великі обсяги даних для виявлення шахраїв. В епоху, коли конфіденційність стає все більш важливою, машинне навчання Підхід набагато краще системи, яка постійно сканує клієнтський комп'ютер. Якщо система на основі машинного навчання експлуатується в хмарі сторонньою компанією, це також хороша угода з точки зору вартості розробки і масштабованості. Ігрова компанія сама по собі не потребує спеціалізованих людей, які зосереджені виключно на боротьбі з шахрайством, оскільки це має був виведений в іншу компанію. Зовнішня компанія буде повністю спеціалізованої в розробці анти читов і мають глибокий досвід у цій галузі в порівнянні з більшістю ігор компанії, в яких мало хто розвиває свої анти-чіти. Основна слабкість підхід машинного навчання полягає в тому, що може виникнути ситуація, коли система вчиться не так, і це може бути важко помітити потім. Існує також можливість прийняття хибних рішень на основі статистики.

Немає сумнівів, що ці типи повністю Антич-системи на стороні сервера, що використовують науку про дані, будуть ставати все більш поширеними Чити на стороні клієнта стають все більш витонченими. Проте, тому що Антич,

заснований на машинному навчанні, повністю на стороні сервера, йому дуже важко виявити чити на стороні клієнта, такі як «Wallhacks», якщо читер використовує їх таємно. Використовуючи «Wallhacks», все залежить від гравця, та його вміння не видавати себе. Умілого читера, важко роздивитися навіть гарному гравцю [49]. (див. Рисунок 2.11.)

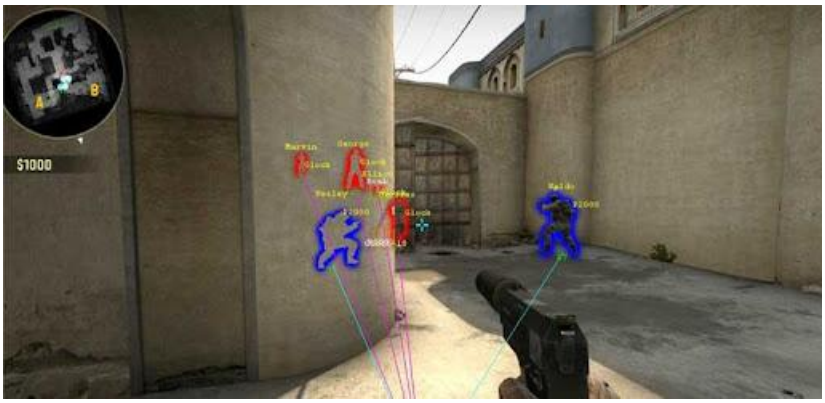

Рисунок 2.11 – «Wallhacks».

2.5. Система збору даних

Архітектура розроблена в рамках проекту системи збору мультимодальних. (див.Рисунок 2.12.)

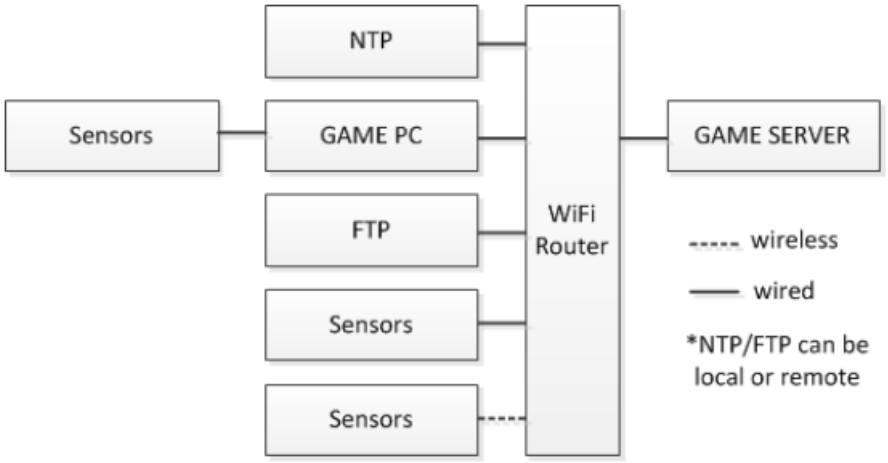

Рисунок 2.12 – Схема збору даних.

Система включає кілька компонентів: виділений сервер зберігання даних FTP, комп'ютер користувача, набір різнорідних датчиків (заснованих на одноплатних комп'ютерах Raspberry PI). Виділений сервер точного часу NTP також є частиною системи. Сервер заснований на одноплатному комп'ютері Raspberry PI та отримує сигнал точного часу від підключеного до нього приймача GPS/PPS сигналу [50]. Високопродуктивний роутер забезпечує з'єднання всіх пристроїв у єдину локальну мережу та вихід до мережі Інтернет. Ряд датчиків має пряме провідне підключення до комп'ютера користувача (миша, клавіатура, мікрофон, вебкамера, ЕЕГ та окулограф). Зовнішні датчики розділені на аналогову групу (пульс, IMU, КГР, ЕКГ) та цифрову групу (температура, вологість, освітленість, рівень вмісту CO2 у повітрі). Розроблене програмне забезпечення для стенду дозволяє здійснювати локальний або віддалений запуск збору даних, стежити за ходом експерименту та забезпечує зберігання зібраних даних як локально, так і із застосуванням хмарних технологій зберігання даних. Аналогів створеної системи збору даних (за кількістю та різнорідності використовуваних датчиків) у відкритих літературних джерелах виявлено не було [51].

2.5.1.Схема синхронізації даних. Важливим завданням, яке було успішно вирішено у рамках виконання першого етапу проекту, було забезпечення синхронності всіх даних, що збираються. Синхронізація в нашій системі побудована на протоколі NTP із загальним локальним сервером часу. Як сервер був обраний одноплатний комп'ютер Raspberry PI 3B, а як джерело точного часу сигнал GPS. Наявність виділеного сигналу PPS, заведеного на окрему GPIO Raspberry PI [52], дозволило забезпечити точність часу в межах 10 – 5 – 10 – 6 с (точність часу 1 – 10 мкс).

Версія 1607 [5]. У 53 випадків запропонованих налаштувань через деякий час швидкість відходу часу в бік компенсується внутрішніми алгоритмами Windows і годинник стає синхронним з сервером точного часу в межах 2 – 3 мс.

2.5.2.Приклад використання. Досягнута точність синхронізації дозволяє знайти кореляцію між мультимодальними даними. В одному із тестових експериментів проводився одночасний запис координат курсора миші та запис показань датчика руху IMU, закріпленого на правій руці користувача (див. Рисунок 2.13.).

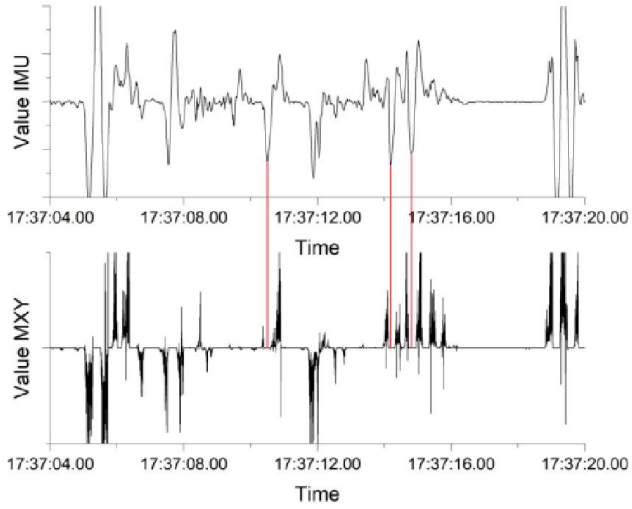

Рисунок 2.13 – Схема збору даних.

По осі Y відкладено значення датчика, по осі X – поточний час. Верхній графік відповідає даним з IMU, нижній координаті миші. Графік наведено для відрізка часу завдовжки 15 секунд. Дані датчиків синхронні з точністю 10 мс. Червоними лініями відзначені ситуації пропуску даних запису координат миші. Видно, що аналіз двох синхронних графіків є інформативнішим, дає менше втрат даних і дозволяє виявити додаткові особливості поведінки.

У ході роботи над першим етапом проекту було створено синхронну систему збору мультимодальних даних. Використовувалися різні класи датчиків, що реєструють наступну інформацію: дані про навколишнє

середовище (температура, вологість, освітленість, вміст CO2 у повітрі), дані комп'ютерної телеметрії (руху та натискання клавіш миші та 54 клавіатури, запис голосу та відео), фізіологічні параметри (КГР, ЕЕГ, ЕКГ, окулографія, IMU) [53]. Усі зібрані дані отримані синхронними з точністю 10 мс. Досягнутої точності вистачає для комплексного аналізу контекстної інформації, телеметрії та фізіологічних показників гравця. Варто підкреслити, що запропонована та реалізована схема синхронізації не вимагає наявності специфічного апаратного забезпечення та може бути застосована до звичайного комп'ютера користувача. Розроблена система збору та схема синхронізації даних можуть бути застосовані в різних галузях психології та фізіології для збору даних та аналізу поведінки кіберспортсменів.

2.6. Висновок до другого розділу

У цьому розділі були описані найпоширеніші програмні прийоми обману. Описано кілька методів введення коду та підключення функції, використовується для розробки простого програмного забезпечення, для читання Source Engine. Особлива увага приділялась поточному програмному забезпеченню анти-читів. Чіт і анти-чит розробники обидва починають використовувати переваги реалізації їх програмного забезпечення для режиму ядра. Режим Kernel пропонує доступ до всієї пам’яті комп'ютера і дає величезну перевагу анти-чит над найбільш поширеними читами.

Заходи проти обману не повинні залежати лише від цього у доступі до режиму ядра, але слід включити стратегію проти обману цикл розвитку гри. Розробники ігор повинні знати про можливості клієнтських модифікацій, які дозволяють гравцям обманювати. Баланс слід знайти між клієнтською та серверною обробкою гри середовище, щоб мінімізувати можливості обману, але не різко зростає вартість запуску ігрового сервера. За цією роботою може слідувати опис читів режиму ядра та методи їх виявлення.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ЗАХИСТУ КОМП'ЮТЕРНИХ ІГОР

3.1. Захист дескриптора драйвером

Деякі античити використовують власний драйвер. Він дозволяє задіяти більш широкий спектр можливостей для захисту програми. Часи хуков SSDT пройшли через високу ймовірність конфлікту з іншим програмним забезпеченням. Прикладом таких драйверів є – «Faceit AC». (див. Рисунок 3.1.)

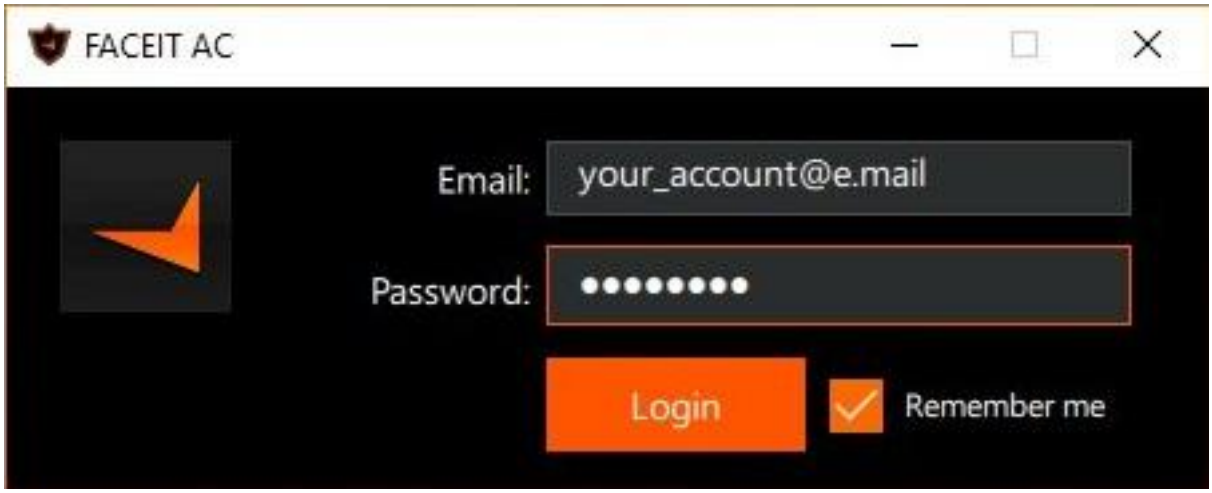

Рисунок 3.1 – Інтерфейс «Faceit AC».

У Windows з'явилася спеціальна функція для перехоплення деяких подій системи – «ObRegisterCallbacks». Драйвер античита урізує права дескриптора процесу, встановлюючи callback на його отримання. При спробі запросити повний доступ до захищеної грі додаток третього кільця отримає лише доступ до загальної інформації про процес [54].

Існують і BI перевірки: гра сама може перевіряти чи модифіковані окремі змінні або код в цілому. Простий приклад: якщо патронів в обоймі буде більше, ніж максимальна кількість патронів в обоймі, значить, щось тут не так.

3.2. Види захисту від внутрішніх читів

Для обходу внутрішнього захисту доведеться реверсити код.

Захист: хук функції «Load Library».

Обхід: «Manual Mapping».

«Manual Mapping» – це ручне завантаження бібліотеки в адресний простір процесу. Вона включає в себе:

- парсинг заголовків;
- аллокації пам'яті;
- запис;
- ручний імпорт бібліотек;
- виклик точки входу бібліотеки.

Виконуючи «Manual Mapping», ми повністю імітуємо функцію «Load Library», але не залишаємо інформації про завантажену бібліотеку.

Захист: моніторинг активних потоків і трейсинг адреси бібліотек.

Знаходячи потік, який не відноситься до процесу гри, античит намагається перевірити цифровий підпис бібліотеки, код якої виконує цей потік. Якщо це не вдається зробити, користувач позначається як читер.

Обхід: хукі і code caving.

Перехоплення викликів функцій дозволяє вбудувати наш код в існуючі функції гри. Нам не потрібно мати власний потік для виклику коду чита. Рано чи пізно гра сама виконає чужорідний код, і чит зробить свою справу.

Code cave – ділянку нулів в пам'яті програми, який ніколи не використовується ним під час виконання. В цю ділянку можна вбудувати код чита. Виконавши перевірку, чи стосується код до адресного діапазону гри, античит пропустить його [55].

3.3. Види захисту від зовнішніх читів

Для захисту від зовнішніх читів використовується драйвер.

Захист: моніторинг відомих процесів або моніторинг всіх процесів і пошук читерскіх програм по їх сигнатурам.

Обхід: обфускація.

Обфускація змінює, заплутує, виртуалізує код, змінює сигнатури. Античит шукає тільки відомі йому сигнатури, і версії коду, що були піддані обфускації будуть проігноровані [56].

3.4.Античит від зовнішніх читів

Хакери розробляють чити, геймери їх купують, компанії наймають інженерів, щоб розробляти нові способи захисту. Хакери знову знаходять лазівку, і коло замикається. Подивимося, як працюють (і чи працюють!?) Різні оборонні методи, і спробуємо створити свою систему захисту від читерства [57].

Напишемо античит. В реальності античити – це комплексні програми, які стежать за багатьма аспектами системи.

Ми будемо шукати непідписані процеси в системі – тому що чити рідко підписують, – отримувати їх хеш і порівнювати з хешами відомих читів. Для пошуку процесів скористаємося Process, а для валідації підготуємо wrapper для функції WinVerifyTrust з wintrust.dll.

Список відомих нам читів:

```
private static readonly string[] CheatHashes =
{
"30BD612FF7FF2D809255364F04B6A9361061BA4E3AA46CD99FDF
1FEF0DA04CC0"};
```

Напишемо просту функцію вибору всіх непідписаних процесів з системи, до яких у нас є доступ.

```
Private static IEnumerable <string>
FindNotSignedProcesses()
{ return Process.GetProcesses()
.Where(prc =>
{
try
{
return
!AuthenticodeTools.IsTrusted(prc.MainModule.FileName);
}
catch
{
return false;
}
})
.Select(x => x.MainModule.FileName)
.Distinct();}
```

Функція отримання хешу SHA-256 файлу за його шляху:

```
public static string GetChecksumBuffered(string path)
{
 var stream = File.OpenRead(path);
  using (var bufferedStream = new
BufferedStream(stream, 1024 * 32))
 {
var sha = new SHA256Managed();
var checksum = sha.ComputeHash(bufferedStream);
```

```
stream.Close();
return
BitConverter.ToString(checksum).Replace("-",
string.Empty);
}
}
```

Створюємо функцію і шукаємо всі процеси:

```
public static void DoWork()
{
Console.WriteLine("Searching for not signed processes..\n");
var prcs = FindNotSignedProcesses();
```

Перебираємо всі процеси, отримуємо їх хеш, порівнюємо зі списком відомих читів:

```
foreach (var process in prcs)
{
Console.WriteLine($"CHECKING:
{Path.GetFileName(process)}");

 var hash = GetChecksumBuffered(process);

 if (CheatHashes.Contains(hash))
{
Console.WriteLine("\nCHEAT DETECTED!");
    }
}
}
```

Якщо чит знайдений, виводимо повідомлення на екран.

Для тестування я створив пустий додаток і вніс його хеш в список. Перевіряємо роботу античита (див. Рисунок 3.2.).

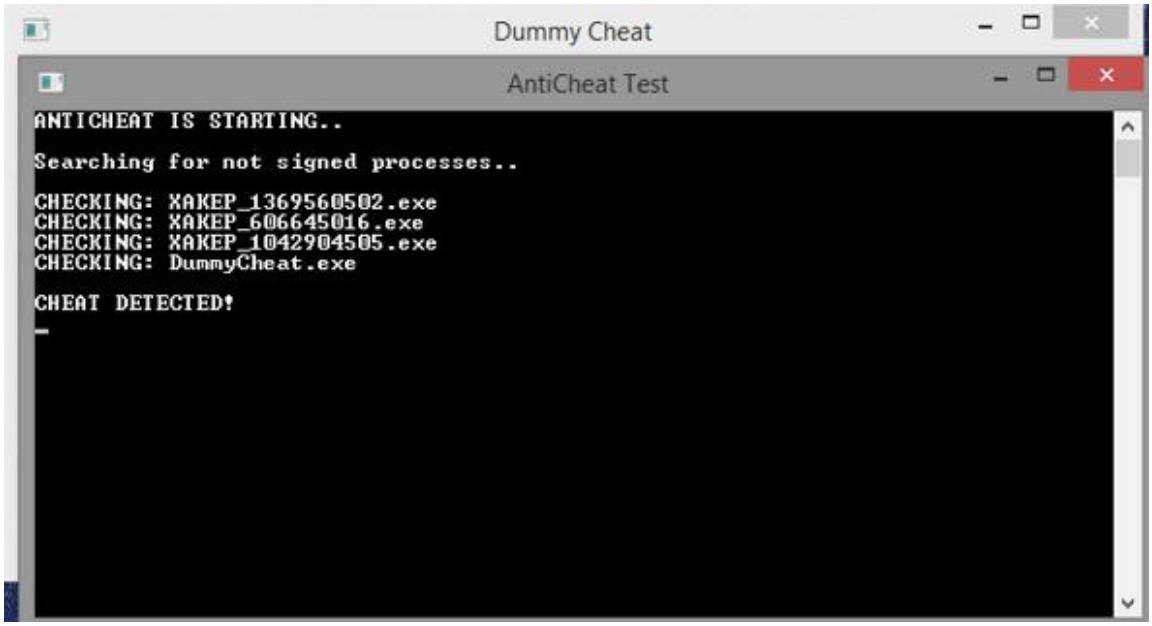

Рисунок 3.2 – Тест античиту.

Успіх – процес чита знайдено.

3.4.1.Доповнюємо античит захистом від внутрішніх читів. Для початку доповнимо нашу функцію отримання шляхів процесів кодом, який буде відправляти на перевірку ще й список непідписаних модулів нашого процесу.

```
private static IEnumerable<string> FindNotSignedProcessesAndModules()
{
var modules = Process.GetCurrentProcess().Modules;
foreach (ProcessModule module in modules)
{
var fn = module.FileName;

if (AuthenticodeTools.IsTrusted(fn))
{
```

```
continue;
}
prcsAndModules.Add(fn);
}
return prcsAndModules;
}
```

Тепер ми записуємо повертається раніше набір IEnumerable <string> в змінну prcsAndModules, додаючи в неї все непідписані модулі нашого процесу. Потім компілюємо порожню бібліотеку, яка виводить повідомлення про свою завантаженні, і вносимо її хеш в список відомих читів.

Вона завантажується в точці входу античита за допомогою LoadLibrary з kernel32.dll.

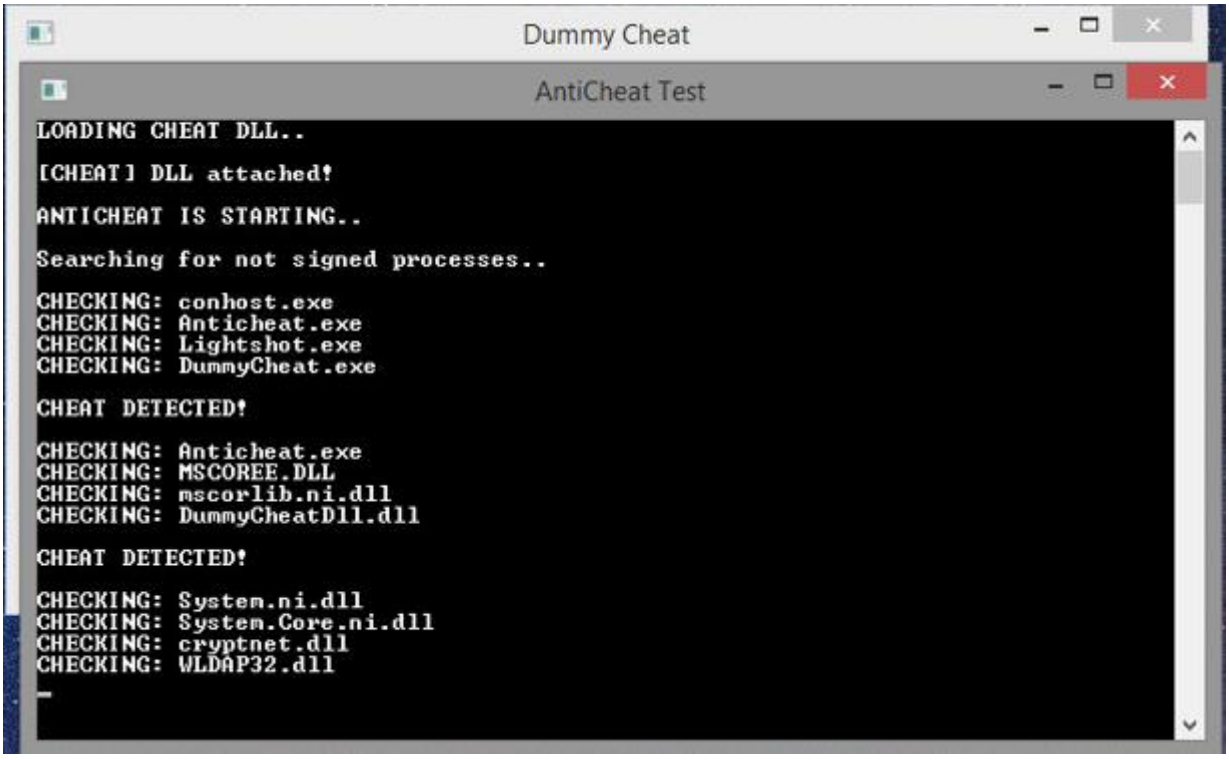

Рисунок 3.3 – Тест античиту після доповненого захисту.

Успіх, адже процеси знайдено.

Одержаний античит зможе знайти відомі копії публічних читів.

3.4.2.Працюємо з кільцями захисту. Привілеї коду всередині Windows контролюються системою UAC. Код розділяється на кільця захисту [58].

Ring 0 (kernel mode) – режим супервізора, або режим з максимальним доступом до всього і вся аж до фізичної пам'яті. Домігшись можливості виконувати свій код в ring 0, читер може отримати доступ до пам'яті гри без обмежень.

Ring 3 (user mode) – кільце, в якому запускаються програми. У них мінімальний набір прав.

У Windows тільки драйвери і ядро системи виконуються в ring 0, а значить, нам потрібно завантажити свій драйвер.

Починаючи з Windows 7 в Microsoft ввели перевірку підписів драйверів. Хочеш свій код в ring 0 – плати за підпис. Це захищає античити, але тільки частково.

3.4.3. Пробиваємо вікно в kernel mode. Деякі Читери помітили, що навіть драйвери з підписом уразливі. Іноді читерам вдавалося отримати доступ до фізичної пам'яті і виконання коду в kernel mode з легітимним драйвером.

Після цього почалася ера кастомних драйверів і автоматичних ManualMapper для них. Деякі умільці робили handle spoofer, який крав дескриптор з повним доступом у легітимного системного процесу.

Так можна провернути виконання будь--якої функції kernel mode прямо з user mode. Послідовність дій проста.

1. Завантажується уразливий драйвер;
2. Знаходиться адресу дуже рідко використовуваної функції, доступної з user mode, але викликає функцію kernelmode;
3. Код марної функції ядра зберігається і замінюється кодом, який перенаправляє нас на потрібну нам функцію;
4. Викликається функція usermode, перенаправляється на kernel mode;

5. Через трампліну виконання перенаправляється на потрібну нам функцію, вона отримує всі аргументи;

6. Пам'ять функції kernelmode відновлюється, трамплін видаляється.

Трампліном називається код операції асемблера, який перенаправляє виконання.

Таким методом можна отримувати доступ до віртуальної пам'яті процесу, не маючи відкритого дескриптора для нього, використовуючи для читання і запису функцію MmCopyVirtualMemory.

Архітектура як Античит

Будь-код, який виконується на клієнті, можна модифікувати. Будь-який код, який можна перенести на сервер, краще перенести на сервер.

Уявімо мультикористувацьку гру в хрестики-нулики. Гра не встановлює порядок того, хто і як ходить. Читер може підмінити дані в пакеті, сказавши серверу, що він сходив хрестиком і виграв, хоча всю гру ходив нуликами. Сервер повірить клієнту, і перемога дістанеться читерам.

Щоб уникнути цього, сервер повинен визначати порядок ходів, призначати, який клієнт ходить і чим, і самостійно вирішувати, який клієнт переміг.

Реальні приклади

Існує така гра – Rust. На сервер відправляється швидкість пересування і позиція гравця. Чит дозволяє телепортуватися або пересуватися дуже швидко.

У грі CS: GO архітектура продумана краще. На сервер відправляються тільки натиснуті кнопки, що відповідають за пересування.

Рух вперед в присяді буде виглядати так:

```
cmd->buttons = IN_FORWARD | IN_DUCK;
```

Гравітація і швидкість пересування прораховуються на сервері, щоб виключити можливість телепортації або зміни швидкості пересування.

Тактики читерства для ігор з уразливою архітектурою

Розглянемо їх на прикладі Source Engine.

Атака № 1: packet spam, або speedhacking. Ця тактика передбачає відправку великої кількості пакетів пересування. Так був реалізований speedhack для Counter Strike 1.6 / Counter Strike: Source.

Захист: підрахунок пакетів, відправлених клієнтом, і стеження за інтервалом відправки. Виправлення додано в нових версіях Source Engine.

Атака № 2: packet invalidation – тактика зміни параметрів пакета так, щоб сервер відкидав пакет і не обробляв тик для цього клієнта.

У читах для SE використовується параметр tick_count. Його значення встановлюють на INT_MAX, змушуючи сервер ігнорувати пакет, пропускати прораховування гравітації, наприклад залишаючи гравця висіти в повітрі.

Захист: симуляція гравітації, життів і інших параметрів гравця окремо від отримання пакетів.

Атака № 3: packet choke, або lag switch. Вона затримує пакети і одночасно відправляє їх через деякий проміжок. Викликає смикання рух всередині гри або в деяких випадках навіть телепортацію через всю карту.

Захист: ввести систему репортів і записи ігор.

Знайти вразливість складно, а ось виправити – іноді навіть занадто легко. Єдина проблема – лінь розробників.

3.5.Висновок до третього розділу

Обман в онлайн-іграх сьогодні є проблемою як для окремих гравців, так і для громади. Оскільки призові фонди та онлайн-турніри починають збільшуватися, більше уваги потрібно спрямовувати на розробку програмного модуля проти читів, щоб змусити користувачів змагатися на одних і тих же умовах.

РОЗДІЛ 4.ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ

4.1.Експлуатація

Для експлуатації та перевірки працездатності розробленого АЧ, було створено свій власний багатокористувацький сервер в грі «Counter-strike global offensive», під назвою,-«HACKERSPACE.SITE – AIM and AWP», при використанні свого власного АЧ. На сервері встановлене обмеження в 16 гравців, тобто по 8 гравців на команду. (див. Рисунок 4.1.).

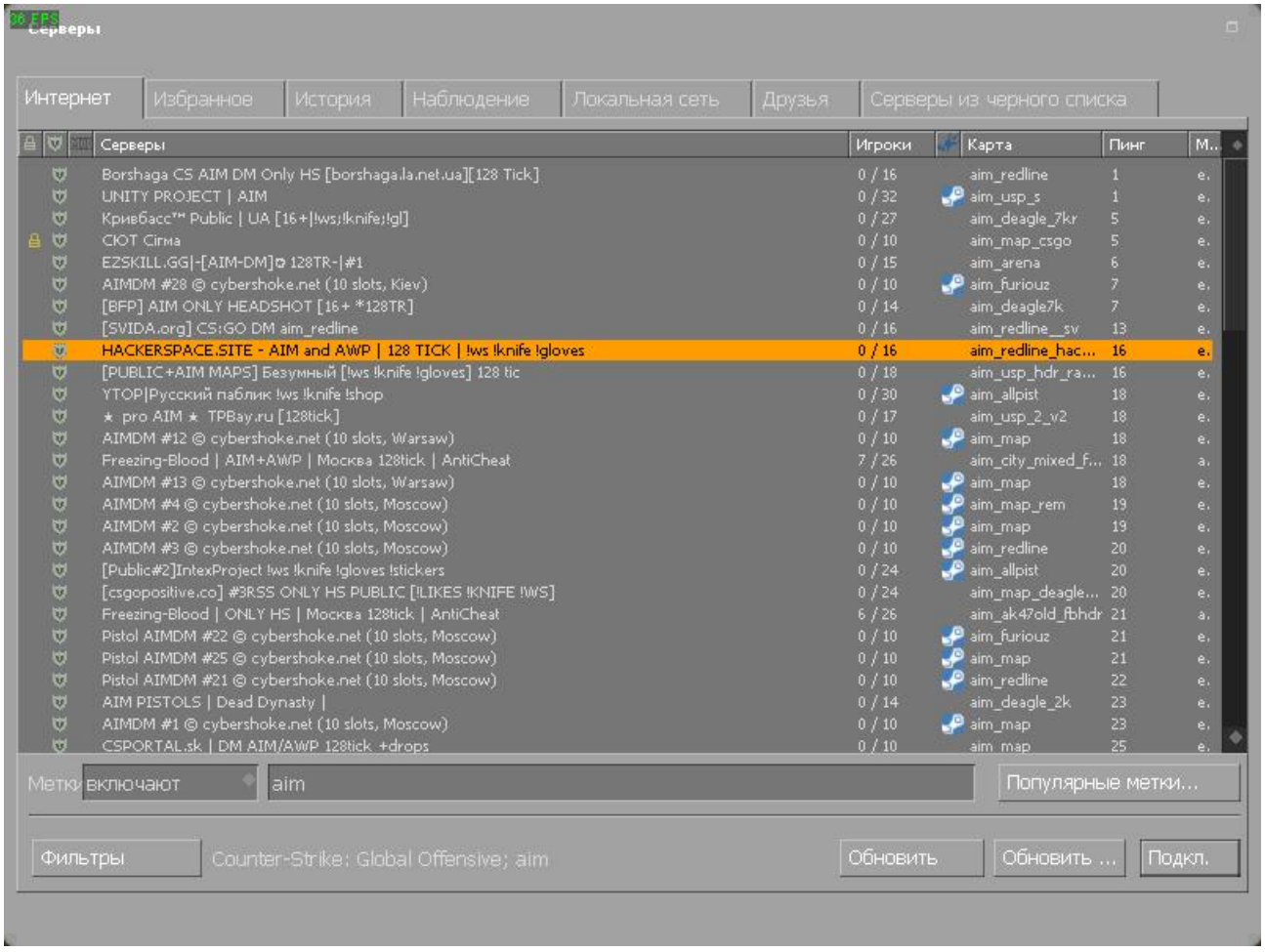

Рисунок 4.1 – Створений сервер.

На сервері був реалізований запис ігрових сесій, для детальної перевірки адміністратором (мною). Тобто, сервер налаштований таким чином, якщо

гравці поскаржилися на підозрюваного, то адміністратор отримує відеозапис ігрової сесії з підозрілим моментом, для перегляду та винесення вердикту (див Рисунок 4.2.).

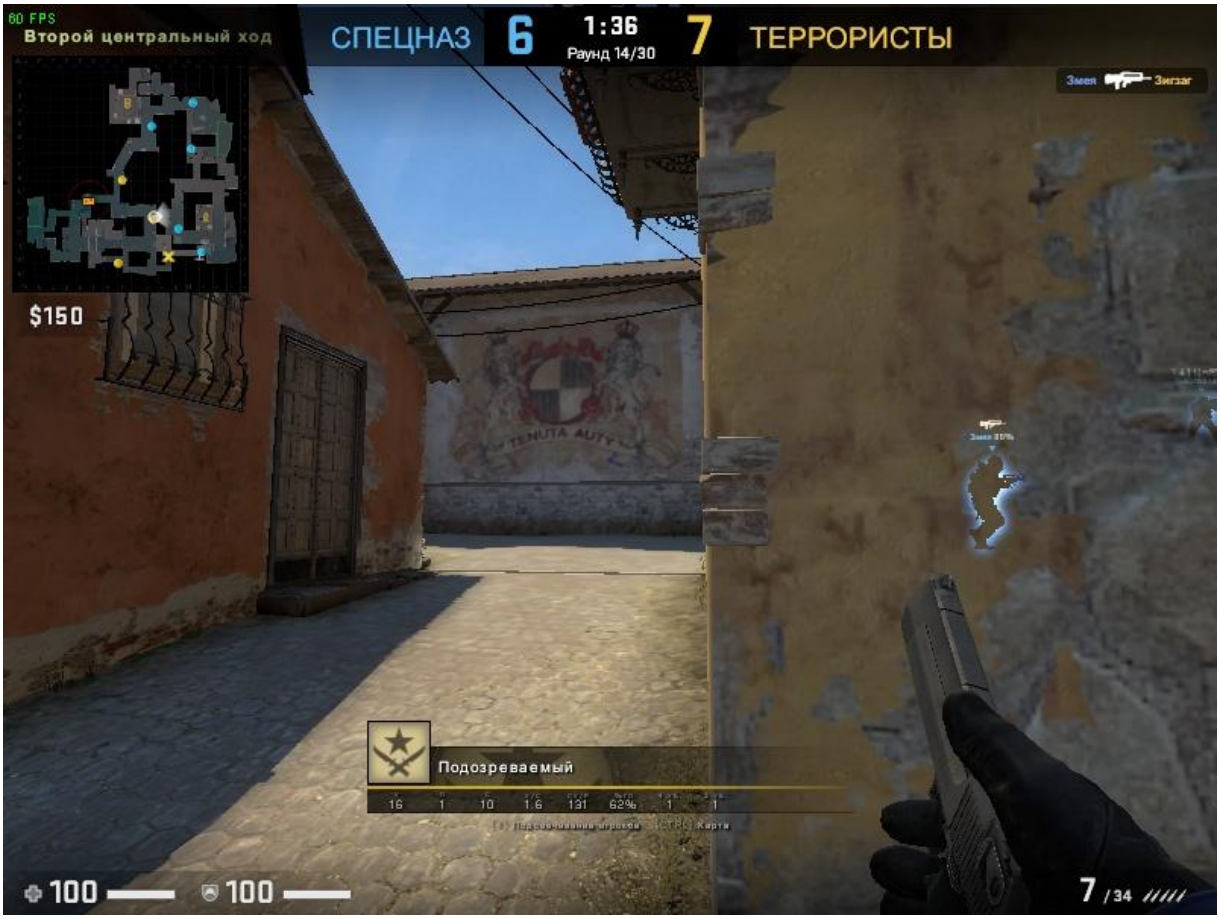

Рисунок 4.2 – Підозрюваний гравець.

На цього гравця поскаржилися супротивники, на їх думку він використовував чит. При перегляді ігрової сесії було помічено використання читів, а саме,-«Wallhack» [58], за що він був заблокований особисто мною, без права оскарження.

Якщо АЧ помітила підозрілі файли, або дії гравця мають характерні ознаки застосування читів, АЧ самостійно виключає гравця з ігрової сесії та блокує, назавжди забороняючи йому підключатися та грати на сервері. В разі якщо, заблокований гравець не визнає своєї провини, він має право оскаржити

вердикт протягом 7 днів з моменту блокування, рівно стільки часу зберігалися записи ігрових сесій.

Сервер було відкрито 8 жовтня 2021 року в 18:54, завершив свою роботу 1 грудня 2021 року в 00:00 за київським часом. За весь час існування сервера на ньому змагались 250 гравців, серед них було виявлено 41 читера, це 16,4% гравців (див. Рисунок 4.3).

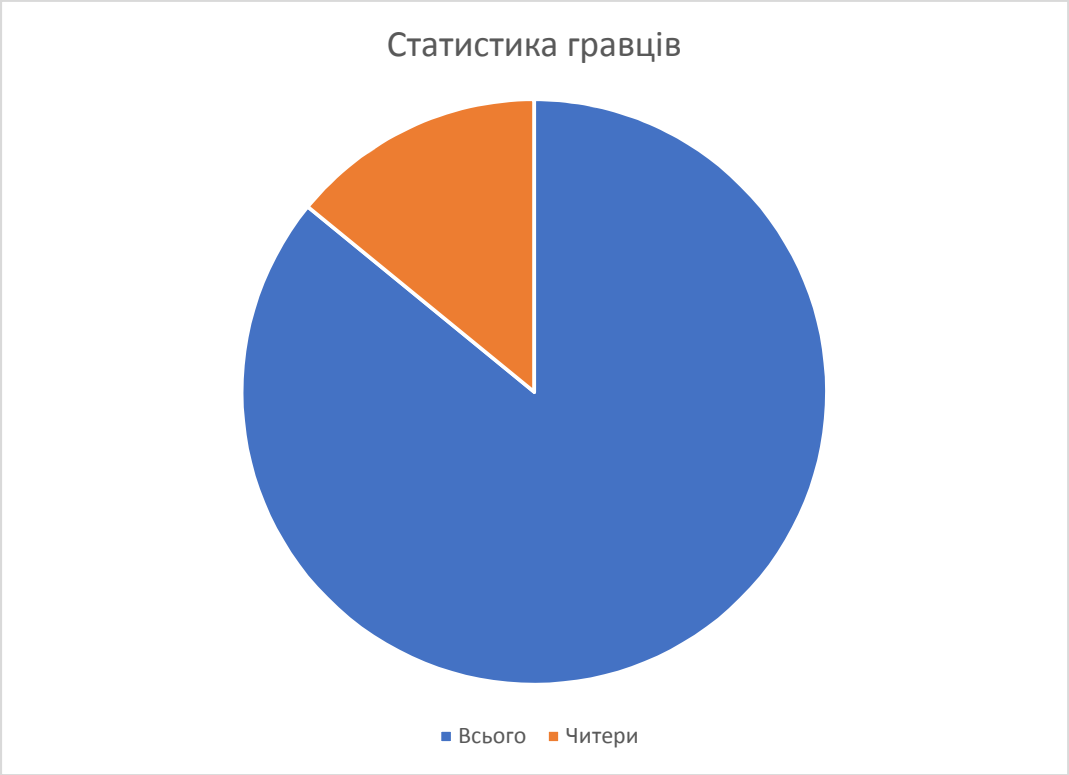

Рисунок 4.3 – Діаграма гравців.

Чити які використовувалися на сервері (див. Таблицю 4.1), (див. Рисунок 4.4).

Таблиця 4.1 – Статистика використаних читів.

Назва читу	**Кількість гравців**
Wallhack	33
Aimbot	6
Spinbot	2

Рисунок 4.4 – Діаграма читів.

Найпоширенішим з читів є,«Wallhack», пояснити це дуже легко, цей чит при вмілом використанні дуже важко помітити звичайному гравцю. Цим читери і користуються, система АЧ в офіційній грі працює лише при умові, що читер себе сам видасть безкоштовним читом, код якого внесений в чорний реестр офіційного АЧ, або велика кількість гравців поскаржаться на підозрюваного, але шанс того, що його заблокують назавжди, мізерний.

Під час експлуатації мого АЧ на сервері, АЧ самостійно виявив та заблокував 41 читера, в коді яких знайшов чит. При кожному запуску чита, АЧ реагував майже миттєво, та блокував читерів протягом 5 хв. Та все ж один читер зміг ненадовго обдурити АЧ, та протримався 15 хвилин. Це був справді вмілий читер, який дуже вміло маскував використання чита, але дуже багато гравців на нього скаржились. Переконатись в тому що він справді читер, мені допомогли записи ігрових сесій та особиста гра проти підозрюваного. В кожній грі він займав перше місце серед гравців за кількістю балів (див. Рисунок 4.5).

Рисунок 4.5 – Статистика матчу проти читера.

На кожного читера був відправлений репорт розробникам гри, проте на момент 10 грудня 2021 року тільки один з них був заблокований офіційним АЧ. А таких як він, тисячі, тому це дійсно нагальна проблема (див. Рисунок 4.6).

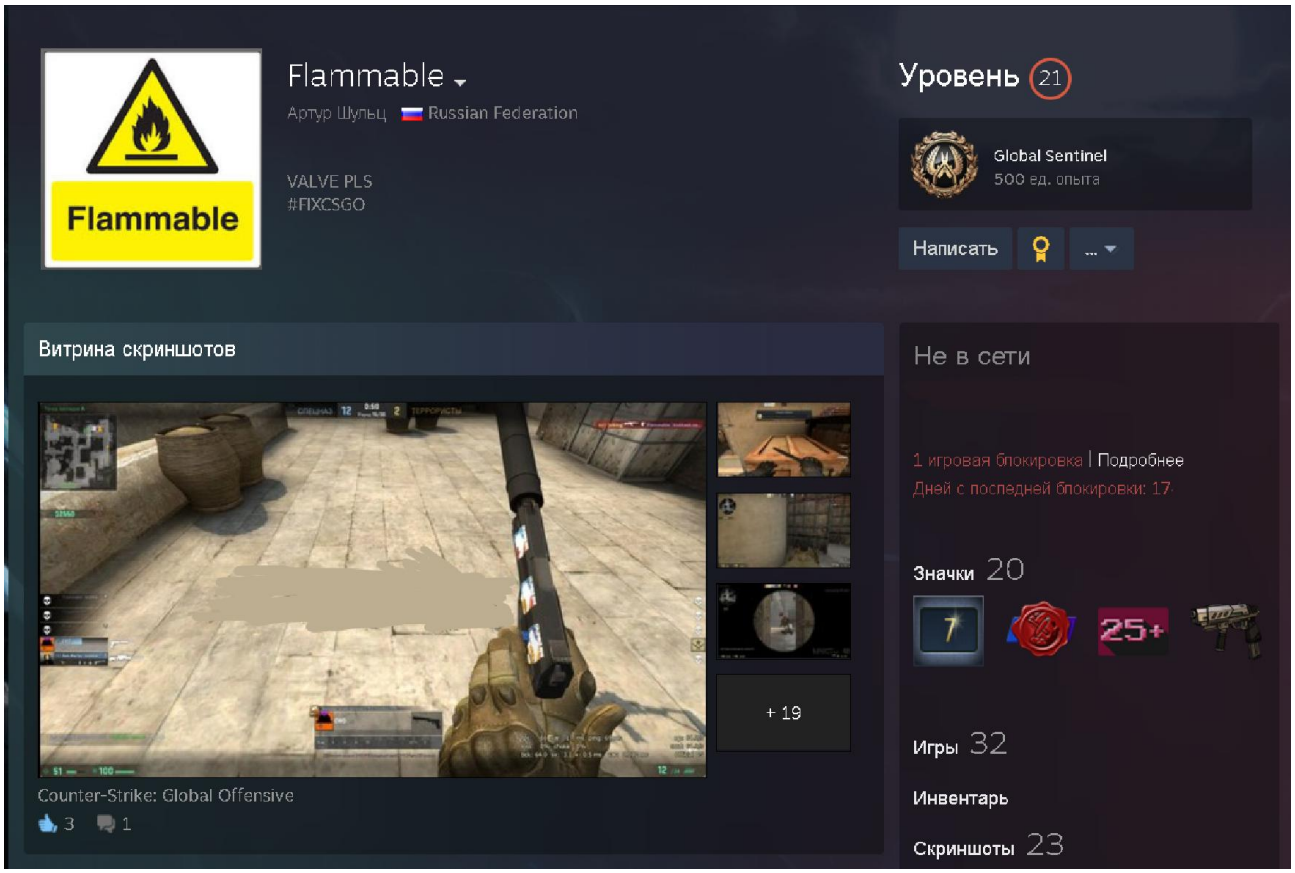

Рисунок 4.6 – Один заблокований офіційним АЧ.

4.2.Висновок до четвертого розділу

Читерство в онлайн-іграх сьогодні є проблемою як для окремих гравців, так і для громади. Оскільки призові фонди та онлайн-турніри починають збільшуватися, більше уваги потрібно спрямовувати на розробку програмного модуля проти читів, щоб змусити користувачів змагатися на одних і тих же умовах.

ВИСНОВКИ

Метою дипломної роботи є розробка програмного продукту для виявлення вбудованого коду в комп’ютерних іграх.

Для досягнення мети, потрібно було розв’язати такі **завдання**:

- проаналізувати нормативно-правову базу України з захисту інтелектуальної власності. З якої слідує, що інтелектуальна власність регулюється чинним законодавством України;
- Проаналізувати програмні продукти для зламу програмного модулю комп’ютерних ігор, що дозволяє обрати оптимальний метод для авторського алгоритму захисту ігор;
- розробка програмного продукту для виявлення вбудованого коду. На основі хєш-кодування;
- експлуатація та порівняння з офіційним АЧ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Makuch, Eddie. US government recognizes League of Legends players as pro athletes. 2013–7–12. [Електронний ресурс]. – Режим доступу: https://www.gamespot.com/articles/us–government–recognizes–league–of–legends–players–as–pro–athletes/1100–6411377/.

2. Grayson, Nathan. Top Counter–Strike Players Caught In Big Cheating Scandal. 2020–11–24. [Електронний ресурс]. – Режим доступу: https://kotaku.com/top–counter–strike–players–caught–in–big–cheating–scand–1662810816.

3. Microsoft. Virtual Address Space [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/ru–ru/windows/win32/memory/virtual–address–space?redirectedfrom=MSDN.

4. Microsoft. User mode and kernel mode [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/ru–ru/windows/win32/memory/virtual–address–space?redirectedfrom=MSDN.

5. Selna, James. Blizzard Entertainment Inc v. Ceiling Fan Software LLC et al [Електронний ресурс]. – Режим доступу: https://www.ceilingfansoftware.com.

6. Campbell, David. MDY Industries, LLC v. Blizzard Entertainment, Inc. et al [Електронний ресурс]. – Режим доступу: https://docs.justia.com/cases/federal/districtcourts/arizona/azdce/2:2006cv02555/322017/82/.

7. Microsoft. Data Execution Prevention [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/ru–ru/windows/win32/memory/data–execution–prevention?redirectedfrom=MSDN.

8. Microsoft. Driver Signing Requirements for Windows [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/ru–ru/windows/win32/memory/data–execution–prevention?redirectedfrom=MSDN.

9. Howard, Michael . Address Space Layout Randomization in Windows Vista [Електронний ресурс]. – Режим доступу: https://hackmag.com/uncategorized/deceivingblizzard–warden/.

10. Valve. SDK Docs [Електронний ресурс]. – Режим доступу: https://hackmag.com/uncategorized/deceivingblizzard–warden/.

11. Valve. Valve Anti–Cheat System (VAC) [Електронний ресурс]. – Режим доступу: https://www.easy.ac/en–us/.

12. Valve. Steam Family Sharing [Електронний ресурс]. – Режим доступу: https://www.easy.ac/en–us/.

13. Meer, Alec. Valve offers free game after 12,000 false Steam bans [Електронний ресурс]. – Режим доступу: https://support.steampowered.com/kb_article.php?ref=2117–ILZV–2837.

14. Valve. An issue with your computer is blocking the VAC system. You cannot play on secure servers [Електронний ресурс]. – Режим доступу: https://steamcommunity.com/app/730/discussions/0/624076027434619597/.

15. Newell, Gabe. Valve, VAC, and trust [Електронний ресурс]. – Режим доступу: https://steamunpowered.eu/gabe-newell-valve-vac-and-trust/.

16. Battle.net [Електронний ресурс]. – Режим доступу: https://www.blizzard.com/en–gb/legal/fba4d00f–c7e4–4883–b8b91b4500a402ea/blizzard–end–user–license–agreement.

17. A Letter from the Anti–Cheat Team. Feb. 2019 [Електронний ресурс]. – Режим доступу: https://www.pubg.com/2019/02/26/a–letter–from–the–anti–cheat–team/.

18. Luigi Auriemma and Donato Ferrante. Multiplayer Online Games Insecurity (Never Feel Safe While Playing Online). [Електронний ресурс]. – Режим доступу: https://media.blackhat.com/eu–13/briefings/Ferrante/bh–eu–13–multiplayer–online–games–ferrante–wp.pdf.

19. Jan Cappaert et al. “Towards Tamper Resistant Code Encryption: Practice and Experience”. In: Information Security Practice and Experience. Ed. by

Liqun Chen, Yi Mu, and Willy Susilo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 86– 100. isbn: 978–3–540–79104–1.

20. Cheat Engine. [Електронний ресурс]. – Режим доступу: https://www.cheatengine.org/.

21. Client Server Model. [Електронний ресурс]. – Режим доступу: https://docs.unrealengine.com/udk/Three/ClientServerModel.html.

22. Bootie Cosgrove–Mather. Cheats Could Ruin Online Gaming. Dec. 2002. [Електронний ресурс]. – Режим доступу: https://www.cbsnews.com/news/cheats–could–ruin–online–gaming/.

23. Deceiving Blizzard Warden. [Електронний ресурс]. – Режим доступу: https://hackmag.com/uncategorized/deceivingblizzard–warden/.

24. Diablo. [Електронний ресурс]. – Режим доступу: https://diablo.elis.ugent.be/.

25. Julian Dibbell. The Life of the Chinese Gold Farmer. June 2007. [Електронний ресурс]. – Режим доступу: https://www.nytimes.com/2007/06/17/magazine/17lootfarmers–t.html.

26. Wenliang Du. Computer Security: a hands–on approach. CreateSpace, 2017 [Електронний ресурс]. – Режим доступу: https://www.handsonsecurity.net.

27. Easy Anti–Cheat. [Електронний ресурс]. – Режим доступу: https://www.easy.ac/en–us/.

28. EnumProcesses function (psapi.h). [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/en–us/windows/win32/api/psapi/nf–psapi–enumprocesses.

29. Ettercap. [Електронний ресурс]. – Режим доступу: https://www.ettercap–project.org/.

30. Jon Fingas. Bungie pulls popular gun from ’Destiny 2’ after discovering exploit. Oct. 2019. [Електронний ресурс]. – Режим доступу: https://www.engadget.com/2019/10/20/bungie–pulls–telestofrom–destiny–2–after–exploit/.

31. Lorenzo Franceschi–Bicchierai. For 20 Years, This Man Has Survived Entirely by Hacking Online Games. July 2017. [Електронний ресурс]. – Режим доступу: https://motherboard.vice.com/en_us/article/59p7qd/this–man–has–survived–by–hacking–mmo–online–games.

32. GameBlocks – Server Side Anti–Cheat. [Електронний ресурс]. – Режим доступу: https://gameblocks.com/.

33. GDPR Key Changes. [Електронний ресурс]. – Режим доступу: https://eugdpr.org/the– regulation/.

34. Ghidra. [Електронний ресурс]. – Режим доступу: https://www.nsa.gov/resources/everyone/ghidra/.

35. Jill Grodt. Epic Acquires Easy Anti–Cheat Company For Fortnite. Oct. 2018. [Електронний ресурс]. – Режим доступу: https://www.gameinformer.com/2018/10/09/epic–acquires–easy–anti–cheat–company–for–fortnite.

36. Guest Diary (Etay Nir) Kernel Hooking Basics. [Електронний ресурс]. – Режим доступу: https://isc.sans.edu/forums/diary/Guest%20Diary%20Etay%20Nir%20Kernel%20Hooking%20Basics/23155/.

37. Список найпоширеніших програм-читів [Електронний ресурс]. – Режим доступу: https://zikurat.media/kakie-byvayut-chity-v-csgo-samye-rasprostranennye-programmy/.

38. IT технології [Електронний ресурс]. – Режим доступу: https://dlod.ru/osobennosti–antichit–sistemy–mrac–kak–rabotaet–antichit–vac–dlya–ks–pozna–m/.

39. Види античитів [Електронний ресурс]. – Режим доступу: https://mkr–novo2.ru/at/kak–rabotayut–vse–antichit–programmy–osobennosti–antichit.html.

40. Система VAC [Електронний ресурс]. – Режим доступу: https://counterstrike.fandom.com/ru/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_VAC.

41. Новосьолов М.А., Скаржинська О.М. Актуалізація наукового супроводу комп'ютерного спорту / Фізична культура: виховання, освіта, тренування. 2017. № 2. С. 39–40.

42. Новосьолов М.А. Спеціалізація теорія та методика комп'ютерного спорту (кіберспорту) / У збірнику: Національні програми формування здорового способу життя. Міжнародний науково–практичний конгрес. Неверкович С.Д., Годіна Є.З., Бєліченко О.І., Смоленський А.В., БуликінаЛ.В., Рубцова Н.О., Мельникова Н.Ю., Биховська І.М., Козирєва О.В., СоповВ.Ф., Гоніянц С.А.. 2014. С. 630–632.

43. Новосьолов М.А., Олекмінська П.М. Сучасні проблеми вітчизняного комп'ютерного спорту / У збірнику: Матеріали Всеросійської науково–практичної конференції з питань спортивної науки у дитячо–юнацькому спорті та спорті вищих досягнень Збірник матеріалів конференції 2016. С. 329–332.

44. Чому чити для ігор стають все більше схожими на шкідливі програми. [Електронний ресурс]. – Режим доступу: https://www.kaspersky.ru/blog/malware–like–cheats/24073/.

45. MY.GAMES – співдружність із десятків талановитих студій–партнерів, які розробляють ігри найрізноманітніших жанрів. https://my.games

46. РИЗИКИ ВИКОРИСТАННЯ ЧИТІВ 27.11.2019 р. [Електронний ресурс]. – Режим доступу: https://ua.warface.com/news/1007608.html.

47. «Лабораторія Касперського» запустила античит для турнірів з CS:GO, Dota 2 та PUBG» [Електронний ресурс]. – Режим доступу: https://www.championat.com/cybersport/news-3843781-kaspersky-chity--eto-virusy-a-s-nimi-my-boremsja-luchshe-vseh-v-mire.html.

48. Новини кіберспорту [Електронний ресурс]. – Режим доступу: https://www.cybersport.ru/games/news/laboratoriya–kasperskogo–zapustila–antichit–dlya–turnirov–po–cs–godota–2–i–pubg.

49. Методичні рекомендації щодо здійснення прокурорського нагляду за виконанням законів при розслідуванні злочинів у сфері комп'ютерної інформації. 30 травня 2014.

50. Valve Biofeedback Ambinder [Електронний ресурс]. – Режим доступу: https://steamcdna.akamaihd.net/apps/valve/2011/ValveBiofeedbackAmbinder.pdf.

51. Krzysztof Kutt, Wojciech Binek Towards the Development of Sensor Platform for Processing Physiological Data from Wearable Sensors. // ICAISC, 2018, P. 168–178.

52. Simone Tognetti, Maurizio Garbarino Modeling enjoyment preference from physiological responses in a car racing game. // CIG, 2010, P. 321–328.

53. Georgios N. Yannakakis, Héctor Perez Martínez Psychophysiology in Games // Emotion in Games, 2016, P. 119–137 5. [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/en–us/windows–server/networking/windows–time–service/configuring–systems–for–high–accuracy.

54. Aisenk G. Stryktyra lichnosti. Spb.: Uventa, 1999. 464 s.

55. Brain SPECT Imaging in Complex Psychiatric Cases: An EvidenceBased, Underutilized Tool, Amen DG, Hanks C, Prunella J. Predicting positive and negative treatment responses to stimulants with brain SPECT imaging. J Psychoactive Drugs. 2008;40:131–8.

56. Bush G., Luu P., Posner M. I. Cognitive and emotional influences in anterior cingulate cortex (англ.) // Trends in Cognitive Sciences. — 2000. — Vol. 4, no. 6. — P. 215—222. — ISSN 1879–307X. — PMID 10827444.

57. Miller EK, Freedman DJ, Wallis JD. The prefrontal cortex: categories, concepts and cognition // Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 357 (1424). — 2002. — С. 1123—1136.

58. «Ob ytvergdenii programmi razvitija vida sporta «computernii sport» v Rossiiskoi Federatzii» Prikaz Minsporta ot 21.05.2018 № 468.