УДК 004.056.5:004.421:004.3:681.5
DOI https://doi.org/10.32782/tnv-tech.2024.6.5

ВРАЗЛИВОСТІ ІГРОВИХ АНТИЧИТІВ: АНАЛІЗ БЕЗКОНТАКТНОЇ АВТОМАТИЗАЦІЇ ІГРОВИХ ДІЙ ІЗ ВИКОРИСТАННЯМ ЗОВНІШНІХ ПРИСТРОЇВ

Клєпцов А. А. *– магістр інженерії програмного забезпечення, головний інженер програміст, 28Software*
ORCID ID: 0009-0008-3147-0618

Гусєва-Божаткіна В. А. *– старший викладач кафедри програмного забезпечення автоматизованих систем Національного університету кораблебудування імені адмірала Макарова*
ORCID ID: 0000-0002-1117-3391

В 21 сторіччі ігри стали невід'ємною частиною нашого суспільства[1]. Якщо раніше вони сприймались як щось екзотичне, то зараз важко знайти людину, яка б жодного разу не грала у комп'ютерні або мобільні ігри. Більше того, у деяких людей комп'ютерні ігри стають альтернативним способом дозвілля, або навіть хобі. Зараз ігри сприймаються так само нормально, як фільми, серіали або книги. Тобто це вже стало буденністю. Поріг входження став настільки маленький у деяких продуктів, що навіть старше покоління все частіше розглядає ігри як щось нормальне. До прикладу цільова аудиторія Candy Crush Saga[2] – це жінки 35+. Часто можна побачити якусь бабусю у метро чи в автобусі, яка грає в match 3, word games або щось подібне.

Проблема використання програмного забезпечення, яке дозволяє отримати перевагу у грі (в майбутньому чити), є дуже актуальною. Великі компанії можуть випускати гарні ігри, мати високий рейтинг, але стрімко втрачати його, користувачів та прибутки, через збільшення кількості нечесних гравців. При тому, що розробники намагаються виправляти вразливості та покращувати захист, чити також не стоять на місці і проблема за часту не вирішується.

У даній роботі буде розглянута можливість автоматизації ігрових процесів, за допомогою зовнішніх простоїв, які зазвичай ігноруються програмами античитами, та як можна запобігти цьому з перспективи розробників. В статті будуть описані самі популярні рішення захисту ігор від стороннього втручання, їх переваги та недоліки. Окрім цього, дослідимо вже існуючі рішення, які є загальнодоступними, та ставлення розробників ігор до таких девайсів.

Під час дослідження та розробки, будуть використані; мова програмування C#, платформа Arduino Pro Micro, зовнішня HDMI capture card, розгалужувач HDMI сигналу та платформа Raspberry Pi 5.

Ключові слова: *античит, автоматизація, ігри, C#, HDMI, Arduino, Raspberry.*

Klieptsov A. A., Guseva-Bozhatkina V. A. Weaknesses of gaming anti-cheat systems: an analysis of contactless automation of gameplay using external devices

In the 21st century, games have become an integral part of our society [1]. While they were once perceived as something exotic, it is now difficult to find someone who has never played computer or mobile games. Moreover, for some people, video games have become an alternative form of leisure or even a hobby. Today, games are viewed as just as ordinary as movies, TV series, or books, making them part of everyday life. The barrier to entry for some products has become so low that even older generations are increasingly considering games as a normal activity. For example, the target audience of Candy Crush Saga [2] consists of women aged 35 and older. It's not uncommon to see an elderly woman on the subway or bus playing match-3 games, word games, or something similar.

The issue of using software that provides an unfair advantage in games (referred to as cheats hereafter) is highly relevant. Large companies can release great games with high ratings but quickly lose their reputation, users, and revenue due to an increase in dishonest players. Even though developers strive to fix vulnerabilities and improve protection, cheats continue to evolve, leaving the problem unresolved in most cases.

This paper explores the possibility of automating gameplay using external idle systems, which are often overlooked by anti-cheat programs, and discusses how developers can address this issue. The study will outline the most popular solutions for protecting games from third-party interference, along with their advantages and disadvantages. Additionally, we will examine existing solutions that are publicly available and analyze developers' attitudes toward such devices.

During the research and development process, the following tools will be used: the C# programming language, Arduino Pro Micro platform, external HDMI capture card, HDMI splitter, and Raspberry Pi 5 platform.

Key words: *anti-cheat, automation, games, C#, HDMI, Arduino, Raspberry.*

Постановка проблеми. Що таке античит? Це автоматична система, що створена для виявлення читів, встановлених на користувацьких комп'ютерах [3]. Простими словами, це вбудована, або окремо встановлена програма, яка запускається разом з грою, та націлена на те, щоб визначати нечесних гравців, які використовують інші сторонні програми для отримання переваг у грі.

Як часто гравці намагаються оманювати ігри? У сфері ігор 57% опитаних зізналися, що використовували чити в однокористувацьких або багатокористувацьких іграх. Більша частка припадає на чити для однокористувацьких ігор, які становлять 37% від загальної кількості [4]. Статистика вже вражаюча. Якщо розмовляти про конкретні античити, то одним з самих популярних є Easy Anti-Cheat [5]. Ця античит-система, розроблена для захисту багатокористувацьких ігор від читерів. Вона була створена компанією **Kamu** (Фінляндія), а у 2018 році її придбала **Epic Games**. Цю систему використовують такі відомі тайтли як Fortnite, Apex Legends, Rust і багато інших. Наприклад у *Apex Legends* було заблоковано понад **1 мільйон акаунтів** за читерство. Що є величезною цифрою. Більше того, якщо дослідити публікації від гравців в інтернеті, то можна прослідкувати тенденцію зростання кількість скарг на нечесну гру. З цього можна зробити висновок, що розробники читів вдосконалюють свої продукти. В свою чергу, розробники античитів не встигають вживати контр заходів. Який наслідок такої тенденції? На прикладі того самого Easy Anti-Cheat, та Apex Legends. Проблема нечесної гри там стала настільки гострою, що рейтинг гри за останні роки стрімко знизився [6]. В той же самий час, кількість гравців також стрімко падає [7].

Аналіз досліджень і публікацій. Існує дуже багато науковців та дослідників, які цікавляться проблемою читів та античитів. Як приклад, у статті «Cheating in Online Games: A Social Network Perspective» [8] авторів Джеремі Блекберн (Jeremy Blackburn), Ніколас Кортеліс (Nicolas Kourtellis), Джон Скворетц (John Skvoretz), Матей Ріпеану (Matei Ripeanu), Адріана Ямнітчі (Adriana Iamnitchi), досліджується проблема читерства в онлайн-іграх з точки зору соціальних мереж. Автори намагаються зрозуміти, як читерам вдається інтегруватися в ігрові спільноти. Основна увага приділяється взаємодіям між гравцями, а також способам, якими читери намагаються уникати виявлення.

Також у статті «NGUARD: A Game Bot Detection Framework for NetEase MMORPGs» [9] авторів Лінся Гун (Linxia Gong), Чжунвень Цзя (Zhongwen Jia), Сяо Лі (Xiao Li), Чжао Лі (Zhao Li), Хунань Фан (Hunan Fang), Лей Чень (Lei Chen) описується система **NGUARD**, яка є фреймворком для виявлення ботів у багатокористувацьких онлайн-іграх. NGUARD використовує методи машинного навчання для аналізу ігрової поведінки та виявлення ботів, забезпечуючи захист від автоматизованих дій, що шкодять ігровому досвіду.

Мета статті. Завданням роботи – є спроба створення апаратно програмного забезпечення, яке дозволить автоматизувати примітивні дії у багатокористувацькій

грі з використанням вразливості античит системи, а також опису методів, як можна виявляти гравців, які користуються такими вразливостями.

Виклад основного матеріалу. Розглянемо самий банальний метод автоматизації процесу. Без прив'язки до конкретної багатокористувацької гри, або античита. Уявімо, що нам треба забирати якусь нагороду, кожні 2 хвилини. Нагорода забирається шляхом наведення курсора на кнопку, натисканням лівої кнопки миші, та скажімо, клавіши Enter для підтвердження вибору.

Для такої банальної задачі можна використати програму AutoIt [10]. З її допомогою можна емулювати рухи миші, натискання кнопок і так далі. Проблема в тому, що більшість античитів з легкістю визначать, що у вас запущена дана програма. Як результат, античит, або вимкне вам гру, або взагалі заблокує вам акаунт.

Якщо створити свою програму, наприклад на мові C#, шанси не бути поміченим стають більшими. Ось тут я надав банальний приклад, як можна автоматизувати цей процес на мові C# (код не повний і тільки для ознайомлення) [11].

А що, якщо античит може визначити і таку поведінку? Звичайно, коли курсор рухається дуже швидко, то це може виглядати підозріло. Можна зробити поведінку більш натуральною, за допомогою рандомних рухів, не прямих переміщень, а також затримок. Але можна піти далі.

Наступна ітерація – використання Arduino Pro Micro.

Рис. 1. плата Arduino Pro Micro

Цей мікроконтрлер може поводити себе як пристрій HID (Human Interface Device), наприклад, клавіатура або миш. Відповідно, за допомогою тієї самої програми на C#, ми можемо передавати інформацію по Serial порту до Arduino, а Arduino Pro Micro буде виконувати натискання або переміщення. З таким підходом, більшість античитів вже не будуть вважати цю поведінку підозрілою, бо з їх перспективи, користувацький ввід робиться людиною.

Щоб ускладнити задачу, і додати умову зчитування інформації з екрану, нам може знадобитись GetPixel [12]. Вона дозволяє зчитати колір пікселя в конкретних координатах з об'єкта Bitmap (можна зробити скріншот екрану). Повертаючись до першочергової задачі, нам треба забирати нагороду кожні 2 хвилини. Але якщо таймер не фіксований? В такому випадку нам треба опиратись на якийсь візуал. Можна зчитувати колір пікселя на кнопці і тільки тоді виконувати дію. Тобто припустимо, що кнопка не активна і має сірий колір, а коли зелена, то вона активна і можна забрати нагороду. Метод дійсно працюючий, але робити скріншоти кожну 1 секунду також підозріло для програм античитів. Тому наступною ітерацією буде використання Raspberry Pi 5. За допомогою цього одноплатного комп'ютера можна реалізовувати більш складні задачі.

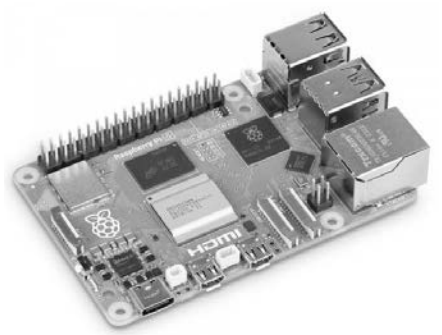

Рис. 2. плата Raspberry Pi 5

Якщо підключити монітор гри через HDMI розгалужувач, інший кінець кабелю через HDMI capture card до Raspberry, ми цілком безпечно можемо аналізувати зображення у ізольованому середовищі. Враховуючи те, що Raspberry також може виступати в ролі HID, нам навіть не потрібна прокладка на ПК в ролі C# програми. Бо всі дії будуть відбуватись всередині Raspberry, абсолютно незалежно, базуючись на зображенні. Програма на C# (або іншій мові) все одно треба, але вона буде запущена саме всередині Raspberry.

Як боротися з подібними девайсами? Не так давно компанія Activision-Blizard забанила дуже багато геймерів у грі Call of Duty: Warzone, за використання девайсу Collective Minds Cronus Zen [13; 14].

Рис. 3. девайс Collective Minds Cronus Zen підключений до Xbox

Цей девайс дозволяв знизити або взагалі прибрати віддачу при стильбі і грі. Зазвичай у іграх є вертикальна і горизонтальна віддача, яку гравцю треба контролювати (рухати мишку або стік в протилежну сторону, щоб попадати в ціль), цей девайс робив це автоматично.

Враховуючи вищесказане, деякі античити, все ж можуть виявляти стороні девайси. Поки що, вони ніяк не реагують на Arduino та Raspberry. І скоріш за все, користувачі, в яких вони підключені до пк, будуть і надалі в безпеці. Бо по суті, це просто плати для розробки, вони не обов'язково можуть робити щось погане. Тому розробникам античитів слід подумати над більш розвинутими механізмами захисту. Хорошим варіантом буде просто перевіряти підключені девайси. Наприклад перевірка USB Vendor ID (VID) та Product ID (PID). Тобто порівняння ідентифікаторів VID та PID пристроїв. Arduino або Raspberry Pi, налаштовані як HID-пристрій, часто використовують нестандартні або відомі ідентифікатори.

Можна створити список дозволених VID/PID і відхиляйте пристрої, які не відповідають цьому списку. Або виявлення нестандартних або відсутніх драйверів. Arduino або Raspberry Pi можуть використовувати загальні драйвери USB HID, що відрізняються від драйверів для сертифікованих пристроїв.

Висновки. У цьому досліджені, була проведена робота з емуляції дій користувача за допомогою програми на C#, платформ Arduino та Raspberry. Наведено пояснення, чому ці методи можуть безпечно працювати наразі і які дії розробники ігор та античитів можуть прийняти, щоб убезпечити свої ігри від шахрайства.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Горбань, О. В., Мартич, Р. В., Малецька, М. О. Феномен відеоігрової культури в сучасному суспільстві. *Studia Warmińskie*, 2019, № 56, с. 123–135. URL: https://elibrary.kubg.edu.ua/id/eprint/31142/.

2. Candy Crush Usage and Statistics. *Helplama*. URL: https://helplama.com/candy-crush-usage-and-statistics/.

3. Anti-cheat software. *Wikipedia*. URL: https://en.wikipedia.org/wiki/Category:Anti-cheat_software.

4. YouGov and PLITCH Study Shows that Over Half of Americans Use Cheats While Gaming. *UberStrategist*. URL: https://uberstrategist.com/press-release/yougov-and-plitch-study/.

5. Easy Anti-Cheat. *Easy Anti-Cheat*. URL: https://www.easy.ac/en-US/.

6. Apex Legends Receives Mixed Reviews on Steam. *Game Rant*. URL: https://gamerant.com/apex-legends-steam-reviews-mixed/.

7. Apex Legends Hits Lowest Player Count in Years as Season 22 Disappoints. *Dexerto*. URL: https://www.dexerto.com/apex-legends/apex-legends-hits-lowest-player-count-in-years-as-season-22-disappoints-2954901/.

8. Blackburn, J., Kourtellis, N., Skvoretz, J., Ripeanu, M., Iamnitchi, A. Cheating in Online Games: A Social Network Perspective. *University of British Columbia*. URL: https://people.ece.ubc.ca/matei/papers/toit-final.pdf.

9. Tao, J., Xu, J., Gong, L., Li, Y., Fan, C., Zhao, Z. NGUARD: A Game Bot Detection Framework for NetEase MMORPGs. *NetEase Fuxi AI Lab, Zhejiang University*. URL: https://linxiagong.github.io/misc/myPapers/KDD2018_NGUARD.pdf.

10. AutoIt Script. *AutoIt*. URL: https://www.autoitscript.com/site/.

11. AutoClicker: Perform Action Every Two Minutes. *GitHub*. URL: https://github.com/MrlCrosl/AutoClicker/blob/main/PerformActionEveryTwoMinutes.

12. Метод Bitmap.GetPixel. *Microsoft Learn*. URL: https://learn.microsoft.com/ru-ru/dotnet/api/system.drawing.bitmap.getpixel?view=net-8.0.

13. Cronus Zen. *Cronus Shop*. URL: https://cronus.shop/collections/cronus-zen.

14. Players complain for being banned for no reason, I call it BS they are using the Cronus Zen Device. *Steam Community*. URL: https://steamcommunity.com/app/1938090/discussions/0/5230393378279357245/?l=ukrainian.

REFERENCES:

1. Horban, O. V., Martych, R. V., Maletska, M. O. (2019) Phenomenon of Videogame Culture in Modern Society. *Studia Warmińskie*, No. 56, pp. 123–135. URL: https://elibrary.kubg.edu.ua/id/eprint/31142/.

2. Candy Crush Usage and Statistics. *Helplama*. URL: https://helplama.com/candy-crush-usage-and-statistics/.

3. Anti-cheat software. *Wikipedia*. URL: https://en.wikipedia.org/wiki/Category:Anti-cheat_software.

4. YouGov and PLITCH Study Shows that Over Half of Americans Use Cheats While Gaming. *UberStrategist*. URL: https://uberstrategist.com/press-release/yougov-and-plitch-study/.

5. Easy Anti-Cheat. *Easy Anti-Cheat*. URL: https://www.easy.ac/en-US/.
6. Apex Legends Receives Mixed Reviews on Steam. *Game Rant*. URL: https://gamerant.com/apex-legends-steam-reviews-mixed/.
7. Apex Legends Hits Lowest Player Count in Years as Season 22 Disappoints. *Dexerto*. URL: https://www.dexerto.com/apex-legends/apex-legends-hits-lowest-player-count-in-years-as-season-22-disappoints-2954901/.
8. Blackburn, J., Kourtellis, N., Skvoretz, J., Ripeanu, M., Iamnitchi, A. Cheating in Online Games: A Social Network Perspective. *University of British Columbia*. URL: https://people.ece.ubc.ca/matei/papers/toit-final.pdf.
9. Tao, J., Xu, J., Gong, L., Li, Y., Fan, C., Zhao, Z. NGUARD: A Game Bot Detection Framework for NetEase MMORPGs. *NetEase Fuxi AI Lab, Zhejiang University*. URL: https://linxiagong.github.io/misc/myPapers/KDD2018_NGUARD.pdf.
10. AutoIt Script. *AutoIt*. URL: https://www.autoitscript.com/site/.
11. AutoClicker: Perform Action Every Two Minutes. *GitHub*. URL: https://github.com/MrlCrosl/AutoClicker/blob/main/PerformActionEveryTwoMinutes.
12. Bitmap.GetPixel Method. *Microsoft Learn*. URL: https://learn.microsoft.com/ru-ru/dotnet/api/system.drawing.bitmap.getpixel?view=net-8.0.
13. Cronus Zen. *Cronus Shop*. URL: https://cronus.shop/collections/cronus-zen.
14. Players complain for being banned for no reason, I call it BS they are using the Cronus Zen Device. *Steam Community*. URL: https://steamcommunity.com/app/1938090/discussions/0/5230393378279357245/?l=ukrainian.