International Journal of Advanced Engineering and Management Research

Vol. 9, No. 05; 2024

ISSN: 2456-3676

Unveiling the Veil: hacking using debugger GNU

Dr Chetanpal Singh[1]
Ass Professor Rahul Thakkar[2], Jatinder Warraich[1] and Numan Ahmed[1]

[1]Future Technologies Industry Cluster College of Vocational Education RMIT University, Melbourne
[2]Computer and engineering science school, Victorian Institute of Technology, Melbourne

doi.org/10.51505/ijaemr.2024.9510 URL: http://dx.doi.org/10.51505/ijaemr.2024.9510

Received: Oct 08, 2024 Accepted: Oct 15, 2024 Online Published: Oct 23, 2024

Abstract
In this present research, the tools such as debuggers and their use for ethical debugging and unethical, malicious use are analyzed. Debugger tool, in general, is used to analyze a program and make code debugging to detect vulnerabilities. But in reality, malicious people like cyber attackers are using such tools to misuse the system, find weak areas, and infect the system for malicious interests. Application-based interferences, containing based transformation of variables, malicious code penetration, security mechanism bypass, and the exploitation of weakness are the main functions of misusing a debugger tool. GNU or GDB debugger is used as a demo in this report to show how it is used. The paper also finds out ways to give security to the application by detecting the problem, avoiding any unauthorized intrusion, making preventive measures, and using roust coding policies. Developer awareness must also be enhanced to detect and predict such misuse of their program and take security measures to lower threats.

Index Terms—debugger tools, ethical debugging, malicious use, cybersecurity, vulnerability detection, reverse engineering, code integrity, security mechanisms, application security, developer awareness, GNU Debugger (GDB), runtime verification, secure coding practices, access control, software development, threat prevention.

Introduction

In the field of software development often developers make use of tools called debuggers which aids in resolution of the challenges in the program and code. Such tools are used in different application functionalities for analysis, understanding and making changes in the program for its execution. However, debuggers can be maliciously used in reverse engineering activities to analyse how an application behaves and steal sensitive information about algorithms made by a propriety enterprise and make program duplications. These malicious individuals use debugger tool for locating any weakness or vulnerability if it exists in the software application. These are then used to insert infected codes, intrude the system, enhance the privilege weakness and breach the security and integrity of the program. So detection of such tools is important for detecting the

Vol. 9, No. 05; 2024

ISSN: 2456-3676

weak points and develop robust defence mechanisms for security and integrity of the applications.

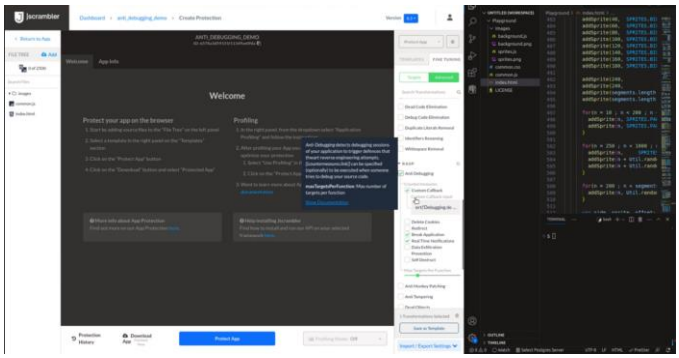


Figure 1: Anti-debugging [2]

In this research the hacking method which is conducted using debugger is simulated. Such methods can be categorised in the transformation of logic of program, malicious code addition, penetrating security measures and others. As a result of these acts the application slows down or fails and the security is lost [1]. Hence, developers must use awareness and implement preventive protocols such as verification of the integrity at runtime, safe practices, and access controls.

There are three main role players in this process and they are the code engineers, the security stakeholder and the end user who are at risk in debugger tool uses in reverse engineering actions undertaken for malicious intent. Developers must be ahead of the cyber attackers and hence widen their knowledge and use of tools and prevention techniques.

Research Objective

This research identifies the need for a cheat engine and a debugger to change game values and variables. This paper shall explore Cheat Engine's functionalities, what it can and cannot do, and how it can be used appropriately for single-player offline games. Also, it will touch on the immoral issues and probably misuse of the stated tools by wrongdoers for the wrong reasons.

Research Motivation

Software programs, such as Cheat Engine, are commonly used in software development and gaming communities as debugging mechanisms for various purposes. Specifically, in gaming, the engine allows players to alter game values, such as the number of bullets, item health, and quantity. Since this research focuses on the functioning of the Cheat Engine and its use in improving or modifying the gaming experience, it is deemed relevant for practical studies.

Furthermore, the study acknowledges the risks of hackers and other cybercriminals abusing such tools to recode, reverse engineer, discover, and usurp software programs' weaknesses [1]. Through an analysis of Cheat Engine, this research will examine the methods involved and explain why it is necessary to consider the ethical use of debuggers when building up security measures to counter these tools.

Vol. 9, No. 05; 2024

ISSN: 2456-3676

From this research, readers will understand the tactics used by Cheat Engine as well as the difficulties and constraints involved in its use. This research will help expand the body of knowledge regarding debugging tools and their effects on games and software, thus raising awareness of the implications of their usage and encouraging a more responsible approach.

Literature Review

Reviews on Debugger Tools and Their Ethical and Unethical Uses

Debugging tools are essential in software development as these are the techniques that offer developers techniques in analyzing, debugging, and even improving the software. However, these tools can also be abused by cyber attackers, making the tools a double-sided weapon, which is double-edged. This section parallels the positive and negative aspects of the various debugger tools highlighted and a call for increased security and awareness among application developers.

Moving forward, [1] explores the concept of 'reverse engineering,' which involves the use of 'debugger tools' to understand the workings of 'malicious code.' This paper describes the workings of these tools in the hands of cyber attackers to reveal the software systems' flaws to enable them to overhaul the security systems. Analyzing these tools is necessary to improve the overall Information Security and foster the creation of proper countermeasures against their misuse.

[2] extensively explains the existence of anti-debugging technologies and techniques aimed at averting unauthorized disruption of applications. Anti-debugging principles include methods that facilitate the detection of a debugger and ways in which the software's operation can be changed so that it cannot be debugged. These techniques help protect applications from being reverse-engineered or subjected to other forms of evil.

Debuggers have been crucial to SD software practices since the middle of the 20th era. According to [3], the progression of debugging tools is also analyzed in the framework of the issue with the resolution of software complications. Initially, these tools helped improve code quality to such an extent that with their help, developers could inspect their code and treat it to eliminate defects and shortcomings. As software programs emerged more elaborate, the debugging tools also evolved to offer intricate characteristics for deep scrutiny and deciphering.

Reverse engineering is another area that benefits from debugger tools. To sum up, debugger tools are widely used by professionals in various areas related to software development. [4] point out two parameters of debugging behavior in programming. Some programmers employ them for a noble cause, such as enhancing the quality of their software. In contrast, others use debugging tools for the wrong reasons, like gaining access to proprietary software codes. This may result in the replicating of programs and embezzlement of ideas, which is why there is always a need to employ high-level security systems to anticipate such risks.

Vol. 9, No. 05; 2024

ISSN: 2456-3676

B. Misuse of Debugger Tools

Common misuses of debugger tools include altering program logic, inserting additional code, and bypassing security. These activities can cause considerable intrusions into application operations and potential breaches and threats to its security. Thus, cyber attackers employ these tools to independently discover flaws in the software that may enable the injection of contaminated code or circumvent security features.

Another example worth mentioning is Cheat Engine, used for ethical software testing; however, it is often employed for unethical purposes, such as cheating in games. The tool is designed to facilitate control over data in the memory of running processes, which can be utilized to modify a particular game or any other application. It is also interesting to note that Cheat Engine has numerous legitimate uses, including for testing and debugging, despite being primarily a game utility.

[6] have evaluated most tools, such as Cheat Engine, with implications for game design. They point out some ethical concerns that developers should observe, especially given that these tools can potentially be used for cheating or other wrongdoing. They expounded that game developers should make head-and-tails efforts to develop anti-cheating measures that would combat the issue.

C. Security Measures and Developer Awareness

Due to these problems, developers should develop measures to increase their awareness of the misuse of debugger tools and take preventive measures. This involves runtime integrity checks that ensure it is free from vulnerabilities such as buffer overflows, suitable access control mechanisms, and adherence to code security standards of coding. Developers should also be required to have a clue on signs of insecure activities and how they could protect their applications from such insecure threats.

[7] has pointed out identity components in games, that is, the crucial question of identification to enhance protection from unauthorized entry. The paper pointed out various measures for preventing any user, other than the authorized ones, from accessing or modifying any game data to maintain the game's sanctity and avoid cheating.

The research study [8] focuses on the exploitative processes in monetizing video games and their possible prevention using consumer protection mechanisms. They claim that due to social engineering practices, developers should know how the games they developed can be misused and find ways of preventing users from becoming victims. This includes applying proper security measures in coding and, at the same time, informing the users of the dangers involved in using counterfeit tools.

[9] Recently, a study on cheat engines was conducted, explaining the menace and the need to develop measures to counter their use. Based on the study's findings, recommendations are

Vol. 9, No. 05; 2024

ISSN: 2456-3676

provided on how and when cheat engine risks can be identified and prevented, with the chief recommendation being to increase the developer's awareness of such threats and security measures that can be undertaken.

Resilient self-debugging software protection methods described by [10] and [11] can help protect applications against malicious debugging. These techniques include integrating security features at the code level and designing the software to make it hard for an attacker to alter the code or circumvent existing security features. They are equally crucial in protecting high-value applications from advanced attacks.

D. Advancements in Debugger Tools

Special tools like debuggers have also evolved over the years. Unlike in the past, when they provided simple functionalities to ethical hackers to debug their code, advanced debuggers helped carry out other complexities, such as hacking activities. While these tools become more enhanced, there's an even greater need for security and developer sensibility regarding the issues.

[12] Introduced the tool to study software debugging processes, focusing on the constructive approach to improve and not mistake them. As you would infer from these papers, the work done in the area and the debugger tool is still progressive and still in a stage where attempts to maintain ethical use while considering the security aspect of it. One of the dangerous aspects of these powerful tools is that they can be more hazardous in the wrong hands; the industry, on its part, can assist by offering better debugging tools to the developers and including security features.

Methods and Materials

Cheat Engine:

Cheat Engine installation is done in the following manner:

- Cheat Engine link is given here which can be used to be downloaded for operating systems Window or Mac. URL- https://cheatengine.org/downloads.php.
- Cheat Engine is utilised to change product quantity in an inventory by making aspect numbering changes.
- Some game might be working less efficiently using Cheat Engine. However in single player games this is used in an optimised manner [5].

Vol. 9, No. 05; 2024

ISSN: 2456-3676

Figure 2: Cheat Engine [5]

- "Download cheat engine" is tapped which is situated in mid portion of page.
- This has cheat engine latest version which is 7.5.
- "Download cheat engine 7.4 for OS Mac" is tapped if for Mac it is being used.

Cheat Engine Installation:

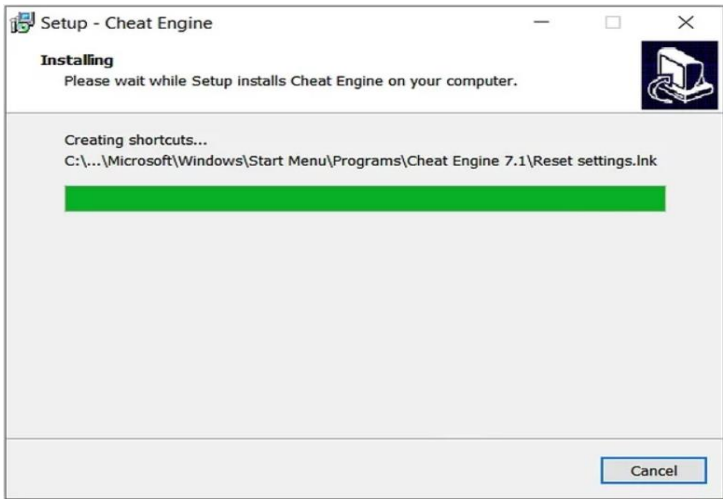


Figure 3: Install Cheat Engine [5]

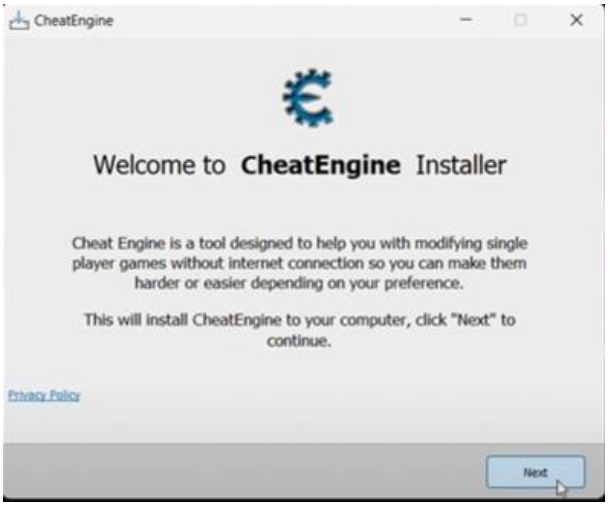
On the basis of the operating system one is using the cheat engine installation process can be different.

Windows OS: Cheat Engine setting file is clicked. As a prompt comes click "Yes". Then tap on "Next". A box comes and clicks "I agree" here. More three time tap "Next". The box called "I agree to install McAfee Web Advisor" must be uncheck and then tap "Next". Then click "Install" As configuration is finished tap "Next" and then "Finish".

Vol. 9, No. 05; 2024

ISSN: 2456-3676

Mac OS: Cheat Engine based DMG file is there and double click on it. The installation must be confirmed here. Then drag logo of Cheat Engine and paste in "Applications" folder, The directions given next must be followed.



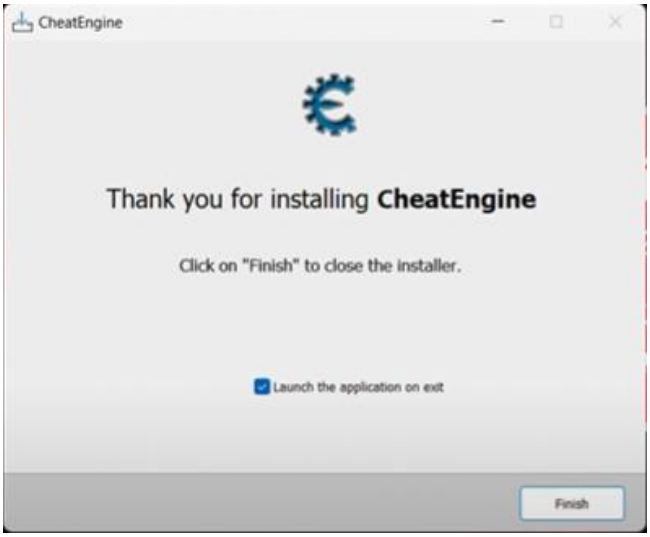


Figure 4: Mac [4]

Cheat Engine installation as done you can open the program. Start the program.

Vol. 9, No. 05; 2024

ISSN: 2456-3676

- "Cheat Engine" button is then clocked after one is selecting the operating system- Mac or Windows.
- As Cheat Engine program is opened tap wither "Open" or "Yes".

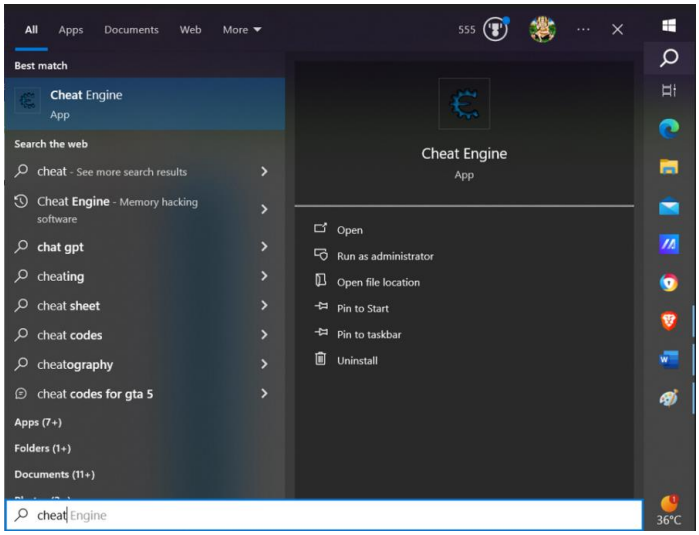
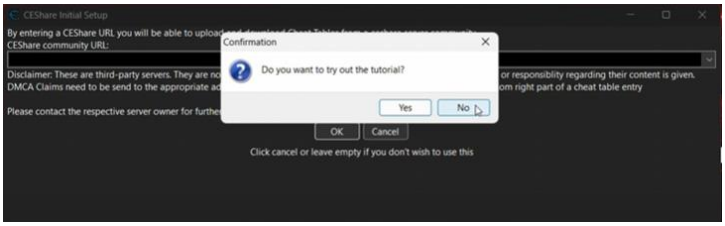


Figure 5: Cheat Engine [8]

Cheat Engine Basic Setup:



Vol. 9, No. 05; 2024

ISSN: 2456-3676

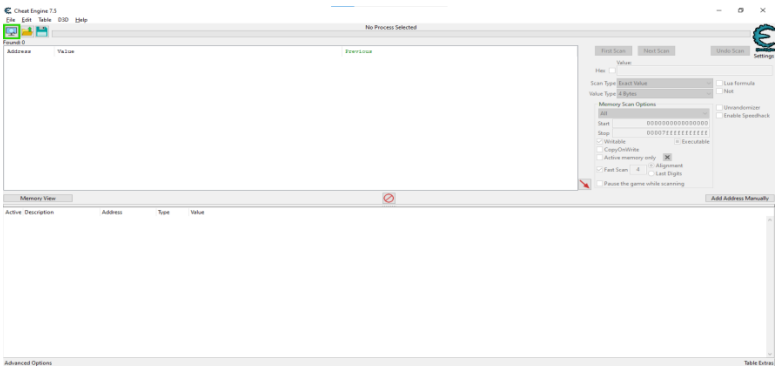


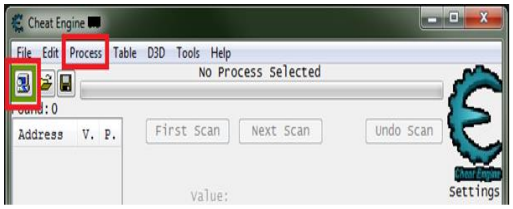
Figure 6:• Cheat Engine Initial Setup [3]

Offline Game Chosen (Plant VS Zombies):

Figure 7: Offline Game [11]

Game Starting: As set up is finished the game starts as per the preferences given and this is done by using Cheat Engine. Do not select server or multiplayer based game here.
Cheat Engine access process is summarised below:

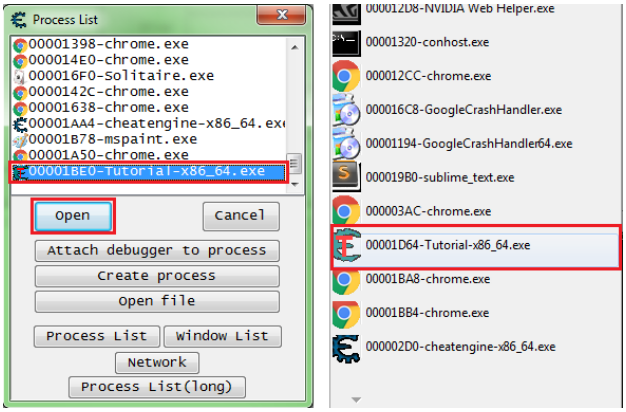
- If this is not opened yet run it as per preferences.
- Cheat Engine is then opened if not opened before.
- Cheat Engine if not opened in a recent time will be opened using icon saved on Desktop.



Vol. 9, No. 05; 2024

ISSN: 2456-3676

- Use list and select the respective process.



- "Open" is clicked once or two times. This is conducted in the format beginning phase.

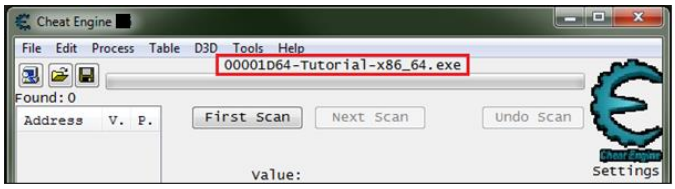


Figure 8: Cheat Engine [1]

- Access for reading or writing game process is now there.

Cheat Engine (x64) Tutorial

Cheat Engine (x64) tutorial is denoted here:

- Cheat Engine is opened and here on top region click on "help". Select "Cheat Engine Tutorial" from the primary menu.

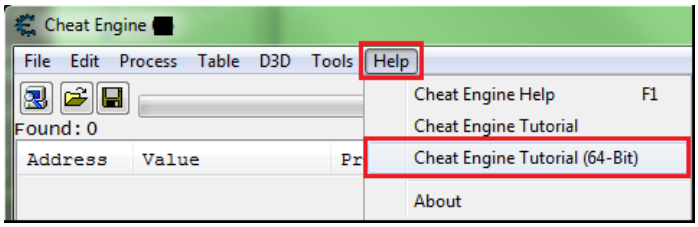


Figure 9: Cheat Engine (x64) Tutorial [7]

- "Tutorial-x86_64.exe" is then attached to initiate the process of tutorial.

  Step 1: Welcome to tutorial,
- Dialogue is opened on launching of the tutorial.
- The help text is opened. On reading a button comes up called "next". Tap on it.
- Password is saved here. The program is restarted.

Vol. 9, No. 05; 2024

ISSN: 2456-3676

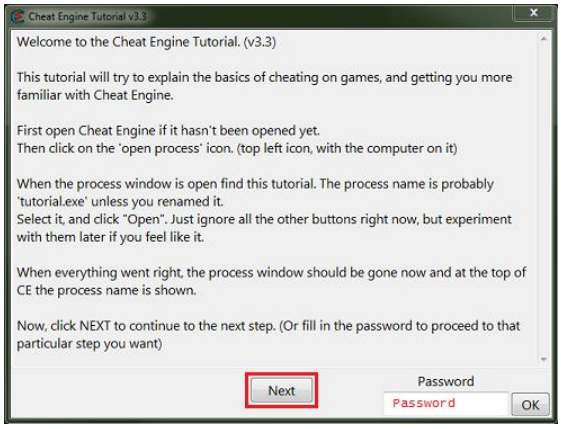


Figure 10: Tutorial of Cheat Engine [9]

Step 2: Scanning

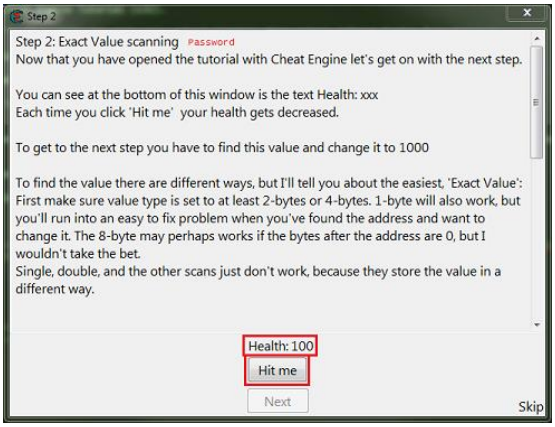


Figure 11: Exact Value Scanning [12]

- The memory scanner is being used for discovering the integer 4-byte. The present health is used for finding value of health value. This is stored in the form of byte, int16, int32, or int64.

Methodology used:

Opening Application Cheat Engine:

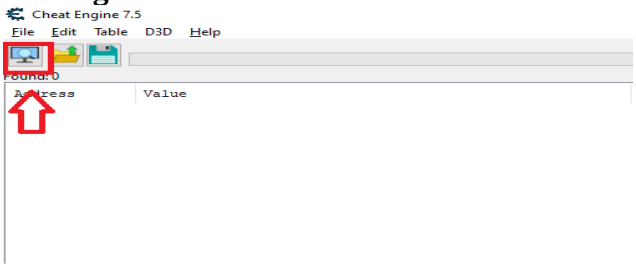


Figure 12: Opening Application Cheat Engine [2]

Vol. 9, No. 05; 2024

ISSN: 2456-3676

- The icon “Processes” is trapped inside Cheat Engine application window.
- From upper left hand side corner, icon looking similar to a computer is found. The window pops up which shows the running application list in this machine.

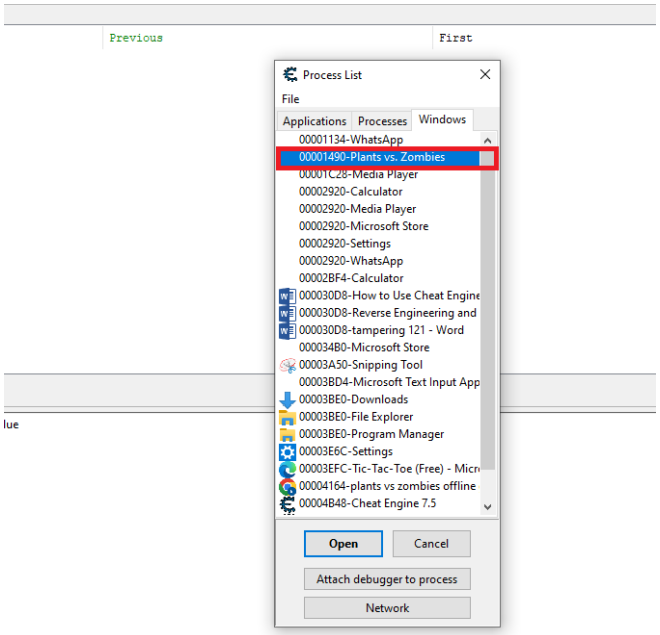


Figure 13: Opening Application Cheat Engine [5]

Game process is selected:

- The process list is scrolled till the game preferred is selected and it is tapped.
- Cheat Engine if it is being used in the browser in game will require you to click over browser name.
- The “Processes” list if it do not contain name of game preferred then the Cheat Engine cannot be edited.

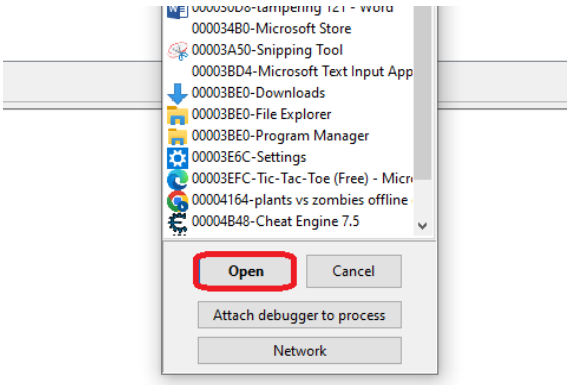


Figure 14: Opening Cheat Engine Application [10]

Vol. 9, No. 05; 2024

ISSN: 2456-3676

- "Processes" is the tab needed to be opened which is in top portion of window.
- For Cheat Engine window launch "ok" must be clicked in window bottom segment in main page.

In-Game Value Detection and Change:

Figure 15: In-Game Value Detection and Change [4]

- The game element is chosen where a number based representation is done using item list or bullet list.
- Ensure that the number shown in screen has no other associated activities such as the menu item or the inventory page.

Cheat Engine transitioning for game window minimization:

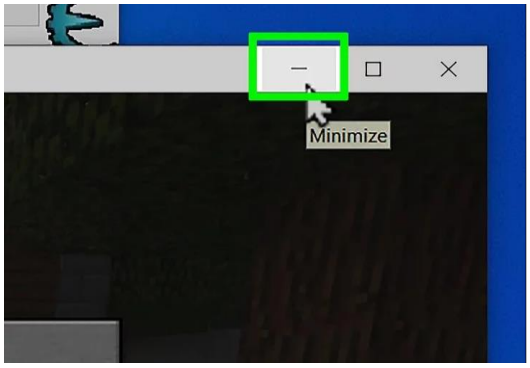


Figure 16: Cheat Engine transitioning for game window minimization [7]

- Game window is lowered and navigation is made in the Cheat Engine and here game is not paused.
- As game remains minimised the overall cheat engine is kept in control.
- Inventory valuation is made in Cheat Engine.

Vol. 9, No. 05; 2024

ISSN: 2456-3676

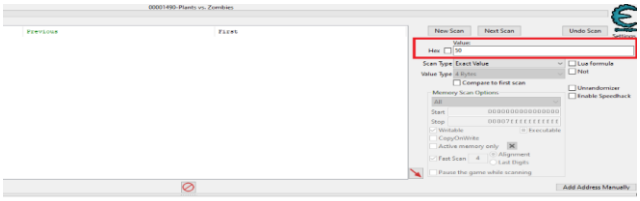


Figure 17: Cheat Engine transitioning [8]

- New value is being searched. “New Scan” is chosen which is at top position on Cheat Engine screen. On left side panel this is located.

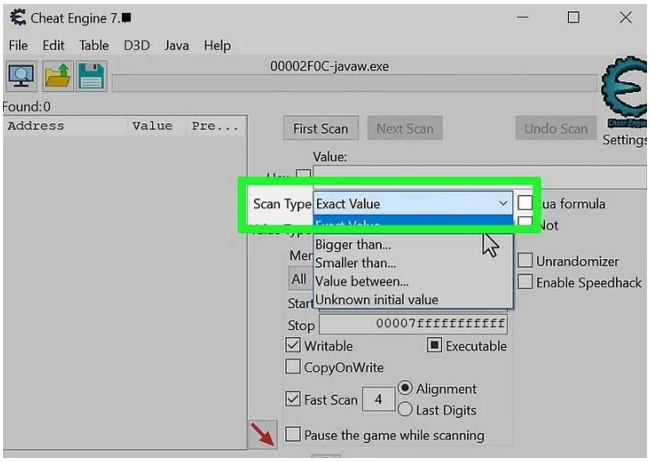


Figure 18: Finding and Changing In-Game Values [6]

Scan Type Selection:
As the new scan is being initiated, one can select from the five scan types. This is mainly utilising the “Exact Value” and the "Unknown Initial Value" for two of these. Selection of the scan is needed to be made in a precise manner. For this menu which is drop down is used and this is situated by scan type. Scans are explained in-depth below:

Exact valuation: This type of scan is used for finding the quantity in precise manner which can be transformed. Example is making teleparameterly based control in life number or ammunition quantity this is being used precisely.

Unknown initial valuation: When playing the game there can be some situations where one does not have a number to make value representation. Health based meter for example cannot always show user health but use information in figure metrics. There is no definite metrics to make measurement of user health when playing the game. Use of artificial intelligence or AI however this information can be found. The screen is not displaying the exact health of the user

Vol. 9, No. 05; 2024

ISSN: 2456-3676

in an accurate manner. The number denotes the user health here [6]. So a starting value can hence be set as a starting point and based on this gauging of health changes can be possible.

Starting health - Bigger than...:' Here you are having a choice to set lower most figure. This is the lowest one can receive and beyond it one cannot have another.
Starting health - Smaller than...: Option is now blank with regards to maximum health denoting amount. Hence borrowing health can be made here even if a certain figure is not achieved.
Starting health—Value between...: The user can choose a value range to search for a price[7].

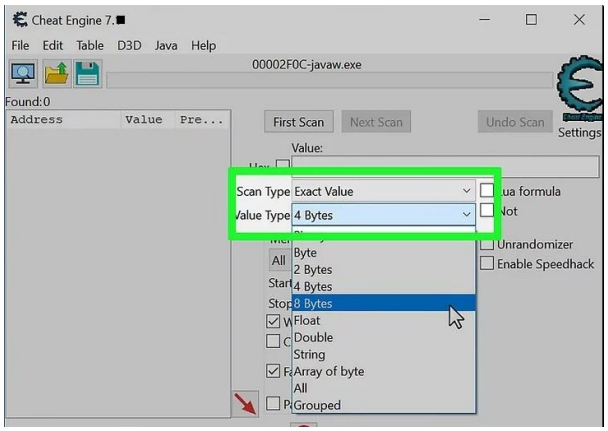


Figure 19: Choosing a Scan Type [4]

Data type selection:
"Data type" here compries of a value which is stored inside memory. This is not an easy or straight process and there are challenges about selecting this not being veyr sure about type of the data. The data found amongst "bool", "byte", "string" or "a number" is used as well as "Arrow". This can be used for finding the value of the scan.

Typical types of data with which one can work are being summarised here as follows:
Bytes number: The use of two types of data is made here in a frequent manner and these are 2 bytes and 4 bytes. Begin this with the 4 bytes as most applications in the Windows are using this as a default metrics. 2bytes here is also used for detecting values and 1 as well as 8 bytes is used as a function here.

Float data: Data type float is where the values make use of decimal points inside the item. The decimal point albeit is not very visible on screen. For the prevention of the scans made for routine memory are saving the values for data floating points. One makes effort in the floating point if the right values are not being found by making byte examinations.
Double data: The doubles are having two times more digits in the float points. These are in relation to floating data types where value is found in double form. This is used if one is unable to find float or byte.

Vol. 9, No. 05; 2024

ISSN: 2456-3676

All data: The alternative here is searching using values and these are linked with different kinds of data. If one is not sure about what they are searching then the best way is production of the search outcomes which must be also filtered [8].

Value is entered:

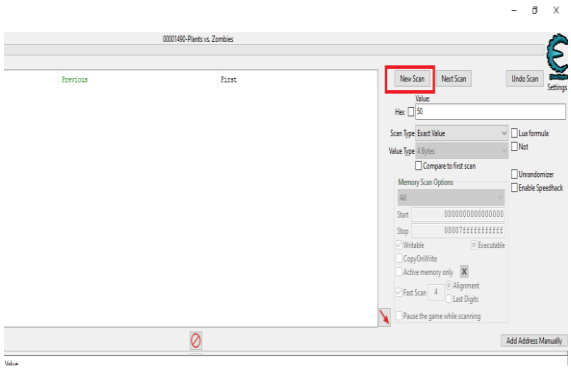


Figure 20: Value Entering [1]

When the desired value is being selected it is required to be changed. Select “First Scan”. An example is made here. One must enter here a value 50 for text box named “Value”. This means there are 50 rounds available in ammunition segment. The search is made using “50” digit. One nudges this somewhat to find out result for entry list which are lengthy in list of address.

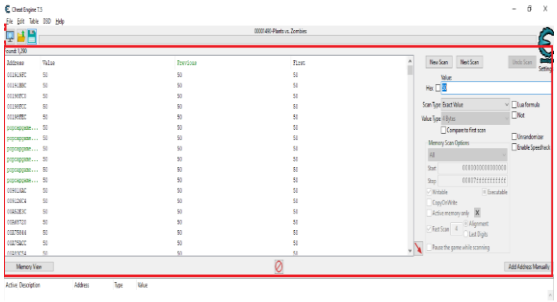


Figure 21: Values Entering [3]

Value transformed and returning in game:

Gameplay is then resumed. The value change is made and this is different as per game variances. To enhance the bullets for an example can be used to increase count of bullet or for additional ammo or fire shot tool. This improves ammo total count as a result.

Vol. 9, No. 05; 2024

ISSN: 2456-3676

Figure 22: Value transformed and returning in game [6]

Cheat Engine opened and scanned using current value:

- Game size is lowered and one can now return to window Cheat Engine.
- New number is added here as "Value" in field. Here "Next Scan" is used for reviewing value address.
- If one does not know the initial value then select "Increased value" or select "Decreased value". Use "Scan type" for making value increase or decrease. All changes made in correspondence is searched which helps in making substation of the number chosen precisely.

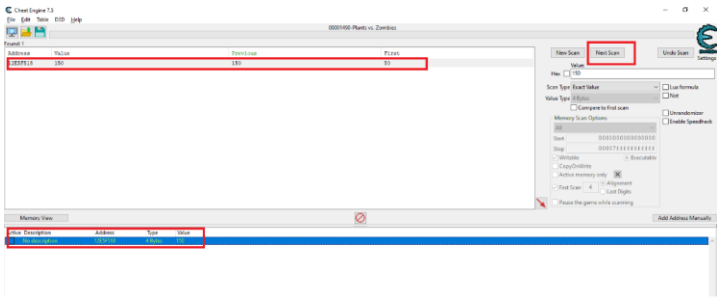


Figure 23: Cheat Engine opened [4]

Vol. 9, No. 05; 2024

ISSN: 2456-3676

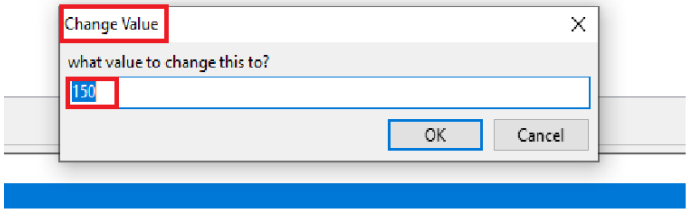


Figure 24: Return to the Cheat Engine

Change the Value:

- The number is double clicked using option list. Cheat Engine window at the bottom is found in "Value".
- New figure is typed in field "Value".
- Select the button "ok".

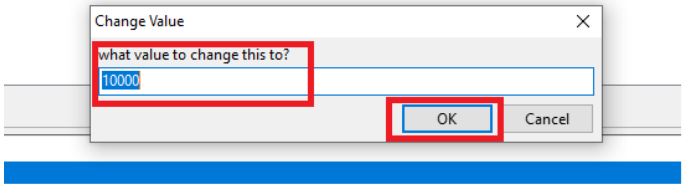


Figure 25: Return to the Cheat Engine [8]

- Cheat Engine is opened. The value preferred is then discovered in this list.
- The number is then double tapped from the "Value and this is changed.
- In "Value" field the number required is entered.
- All transformations is approved and "Ok" is tapped.
- In the game test is made to find that updating of value is made in a successful manner.

Figure 26: Change Value [11]

Vol. 9, No. 05; 2024

ISSN: 2456-3676

- Give confirmation in edited value as it is updated in the current game.
- The number which is entered is then shown in this game.
- It is to be noted that some value adjustments are to be made to make updation and then the value is shown in screen.

Results and Discussion

Results:

Cheat Engine is used and the value increased in bullet number is found which increases from value 50 to value 10000.

Figure 27: Result [1]

"Active" menu button is clicked here beside the target or goal address. This is added for making pointer activation. It is found that underneath "Active" inside address list located at bottom region in display. Point is created easily using this.

Vol. 9, No. 05; 2024

ISSN: 2456-3676

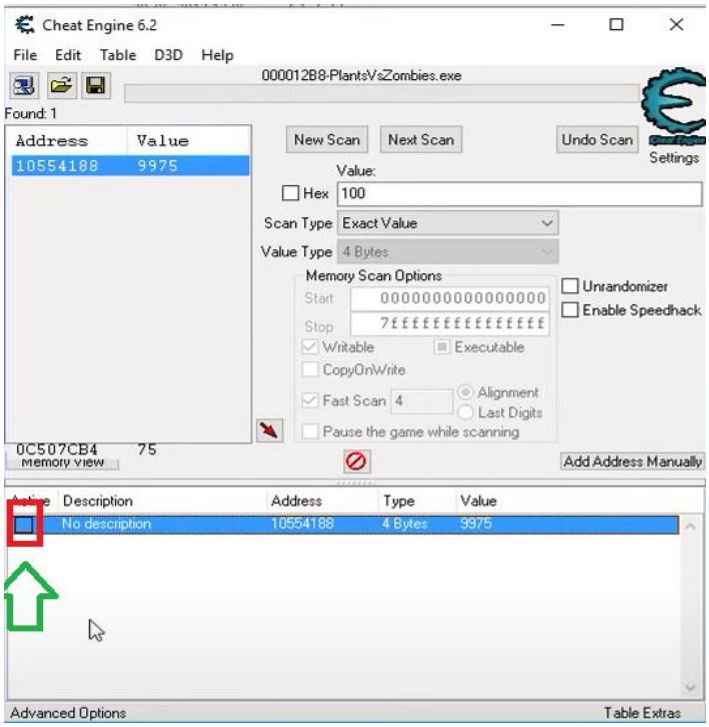


Figure 28: Outcome [9]

The active button is tapped. If not change is made in value count this is done. These points are hence not permanent in nature.

Figure 29: Result [12]

Vol. 9, No. 05; 2024

ISSN: 2456-3676

Cyber attackers and hackers use this type of activities for executing attack type. They make hacks to such games, enhances point and the life of user. For this they use the tool of Cheat Engine.

Cheat Engine Understanding

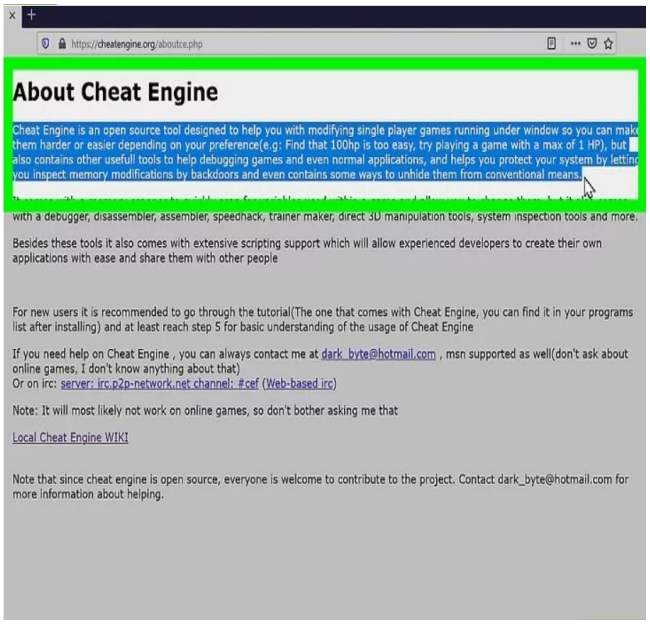


Figure 30: CHEAT ENGINE UNDERSTANDING [10]

Cheat Engine Functionality:

Functions of Cheat Engine are found in observing the stored data in RAM or Random Access Memory segment of computer system. The game is nothing but software. When it is running data is then moving from its static storage segment such as the computer Hard Drive to Ram segment where it can be written or read. Here Ram based data stored is scanned by Cheat Engine window and this data can be changed as well. As connection is made using Ram all users get the skill sets to make changes in the variable of game. This is related to health of player or count of ammunition, make changes in experience of game play in an effective manner and in a practical manner. Such a methodology helps user have better control on the function and behaviour of application which is run over the system [9].

Vol. 9, No. 05; 2024

ISSN: 2456-3676

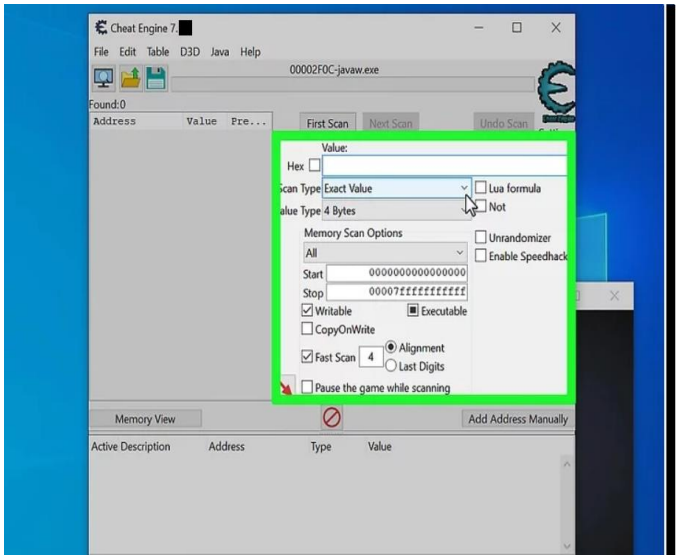


Figure 31: Understanding the Cheat Engine [11]

Relevant Terms and Concepts Understanding

Cheat Engine window is where computer programs are run which are of top status. To ensure that everything and every concept and term is understood this segment is created.

Value: For Cheat Engine there is a value which can be relate to just any numerical information in the computer system. This covers the traits and attributes related to the count of ammunition, percentage of user health or the quantity of stock. The user can use this tool for scanning variables and changing them when needed.

Address: Here the location related to the Ram is signified with regards to the computer system. Here the data is being stored. Location can be vried and hence as per time the address can be changed.

Data Type: Type of data is about methodology utilised for storing value inside memory. This covers the double level precision or format of floating point and covers 2, 4 or 8 bytes.

Pointer: The pointer is address where value gets in written format using another level of address. Such address is dynamic in nature and changes as load changes on game or when playing is made. Understanding must be made what words are used for understanding concepts of Cheat Engine.

Vol. 9, No. 05; 2024

ISSN: 2456-3676

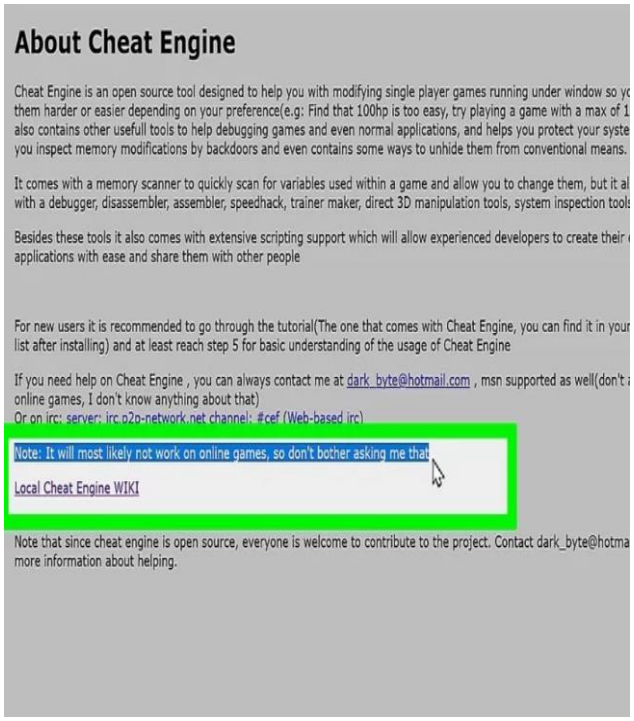


Figure 32: Relevant Terms and Concepts Understanding [5]

Limitations of Compatibility:

The Cheat Engine is noted to be not having high compatibility with all type of games. If these are with the games where protection of cheating is given or if there is online components or multiplayer game type this is not at all compatible. The use and application of Cheat Engine is possible for game where there can be a ban on the account or if online players are not give the access and are stopped from playing. Cheat Engine additionally can be used for collection of resources since some games engage real payment made by users and hence if such user is stopped then there can be legal obligations. The game in general uses robust security mechanisms to give high security for all user profiles. This showcases that every game has some rules integrated without which there can be major legal outcomes.

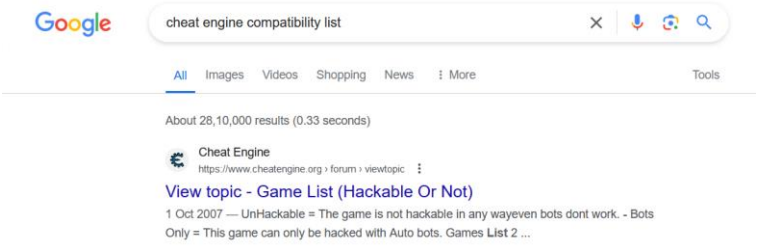


Figure 33: Cheat Engine Compatibility [2]

Vol. 9, No. 05; 2024

ISSN: 2456-3676

Cheat Engine Compatible Games:
The games where there are single player component are compatible with Cheat Engine. They comprise of such titles where single players play the game and they feature values on screen which users can view and change. Moreover, these are the digital flash games which do not have multiple players or have features of making very high scores.

Discussions:
In the above an example is found where applications are explored using debuggers. Hence for cyber protection of the varied software and applications it is important to find out and these re summarised as below:

Debuggers Dual Nature: The debuggers are major tools with which developers can bring improvement in the quality of their programs and applications. But the can be exploited for wrong reasons. Debuggers are detected hence to find how vector attacks are made and ho developers must use powerful tools to detect and prevent such incidents.

Techniques and Methods used: Techniques and strategies can be used to prevent debugger negative uses. The cyber attackers can use different processes such as using manipulation of variable or code infection injection or reverse engineering. It is important to understand these to create robust defence systems.

Vulnerabilities and Security Challenges: Attacks made using debugger tools can compromise the security, integrity and privacy of the programs developed. Loopholes are made in the logics or the flaws and weakness in management of memory are used by attackers. Some program might not use robust security mechanisms. Since systems fall prey to the cyber attacker malicious actions.

Preventive mechanisms need: Debugger related exploitation and threats are increasing exponentially. Such risks landscapes must be investigated and severity must be lowered. For this it is recommended that audits and inspections are made regularly in the developer circle, run the integrity at run time to validate the appliances and also take some steps to detect and prevent any tampering related efforts made in an illegitimate manner.
Develop awareness and collective efforts: The enterprises engaged in development activities must develop a high level of awareness about such risk and threats. This will require the team member support as well. All awareness, knowledge must be shred so all can use the safety mechanisms.

Conclusion
The above report analysis is made to find how applications and programs are tampered using debugger tools. The research has helped in developing understanding and insights about the interactions which take place in the programs being developed and security issues which are highly complicated. The debugger comprises a nature that portrays duality and is used in a

Vol. 9, No. 05; 2024

ISSN: 2456-3676

positive manner by software developers. But for the malicious cyber attackers these are also beneficial tools as it helps them uncover program vulnerabilities. The techniques which are being used by such malicious individuals for making unethical changes in the application are analysed here. With this research approach better measures can be created and implemented to stop such exploitation incidents [10].

Attacks and cyber hack incidents which take place are real life examples about the need of having robust security and safety audits and mechanisms to make the software system strong and protected. There are securities protocols which are genuine and strong but are sometime neglected. When such approaches are used in program and application development it can be a reason of penetration made by malicious hackers using debugger tools. This will compromise the privacy, confidentiality and integrity of the system. Enterprises who are engaged in application development hence must develop security first as their new organisational culture.

Enterprises hence must use better access control mechanisms, regular auditing of the systems and verification measures to maintain runtime security and integrity. All these strategies are used by organisation so they can detect and give a preventive approach to stop such tampering incidents. The integrity of the entire digital environmental system is maintained in this manner.

The work of protecting the company and its programs and software need a collaborative approach [11]. The team must share the knowledge and information they are gathering. The team members must create a defence system and for this their understanding must be shared. The developers and project managers must also analyse risks regularly so any new risk can be eliminate as it crops up. Applications can be tampered using the debugger tool in different levels and degrees. The culture of using protective measures must be proactive and not responsive. Hence before any malicious attacks take place protection must be added to the applications being developed and for this enterprises will use efforts from the team members. The enterprise culture must be that of access control, privilege and authorisation and this will make the software development a resilient process.

Future Works:

Here are some opportunities which can be leveraged to strengthen the application and software development and avoid debugger based intrusions.

Use mechanisms for anamoly detection: The team is recommended that they can use sophisticated tools and techniques so they can detect as well as avoid debugger based exploitation further. More research must be made so integrity of run tine is enhanced, algorithm anomaly is uncovered fast and any malicious behaviour must be detected. **Analyse vulnerability in an autonomous manner:** The future work must revolve around finding more tools and technologies to audit and identify the weakness in the applications being developed. There are advanced technologies such as AI and ML which can detect debugger related activities and malicious requests autonomously.

Vol. 9, No. 05; 2024

ISSN: 2456-3676

Security supervision must be dynamic: The controls applied to give the applications more secured there must be dynamism since the threats are also fast changing its formats. Research must be made to find out how better access related control can be implemented along with robust security policies to ensure that tampering efforts are prevented [12].

Profile the behaviour and detect the anomalies: The firm and the team must develop better skill sets to detect the debugger exploitations in an early stage by detecting profile of behaviour. Algorithms in machine learning, use of statistical analytical method and behaviour based profiling are used further for these purposes.

Higher awareness and knowledge creation: With a higher awareness the team must develop more knowledge about such threats and develop know how. This education is used to protect the applications from any such risks and threats and give the developers higher empowerment. The developers, managers and researchers must hence use the processes suggested above to create higher resilience.

Vi. References

V. Bhardwaj, V. Kukreja, C. Sharma, I. Kansal, and R. Popali, "Reverse Engineering-A Method for Analyzing Malicious Code Behavior," 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3), Dec. 2021, doi: https://doi.org/10.1109/icac353642.2021.9697150.

B. Zhang, "Research Summary of Anti-debugging Technology," Journal of Physics: Conference Series, vol. 1744, no. 4, p. 042186, Feb. 2021, doi: https://doi.org/10.1088/1742-6596/1744/4/042186.

D. Abramson and R. Sosic, "A debugging tool for software evolution," Nov. 2002, doi: https://doi.org/10.1109/wcre.1995.514716.

M. Beller, N. Spruit, D. Spinellis, and A. Zaidman, "On the dichotomy of debugging behavior among programmers," Proceedings of the 40th International Conference on Software Engineering,

"Cheat Engine," www.cheatengine.org. https://www.cheatengine.org/

A. Kultima and A. Sandovar, "Game design values," Proceedings of the 20th International Academic Mindtrek Conference on - AcademicMindtrek '16, 2016, doi: https://doi.org/10.1145/2994310.2994362.

D. P. Rowbottom, "Identification in Games: Changing Places," Erkenntnis, vol. 77, no. 2, pp. 197–206, Apr. 2012, doi: https://doi.org/10.1007/s10670-011-9357-0.

D. L. King, P. H. Delfabbro, S. M. Gainsbury, M. Dreier, N. Greer, and J. Billieux, "Unfair play? Video games as exploitative monetized services: An examination of game patents from a consumer protection perspective," Computers in Human Behavior, vol. 101, pp. 131–143, Dec. 2019, doi: https://doi.org/10.1016/j.chb.2019.07.017.

Vol. 9, No. 05; 2024

ISSN: 2456-3676

Totally_Not_A_Haxxer, “Reverse Engineering: Introduction to cheat engine,” Medium, Feb. 08, 2023. https://read.martiandefense.llc/introduction-to-cheat-engine-e49ee19544e (accessed May 23, 2024).

B. Abrath, “Resilient Self-Debugging Software Protection,” Journal, vol. 1, no. 1, pp. 1–2, 2020.

B. Abrath, B. Coppens, Stijn Volckaert, J. Wijnant, and Bjorn De Sutter, “Tightly-coupled self-debugging software protection,” Dec. 2016, doi: https://doi.org/10.1145/3015135.3015142.

Viktor Shynkarenko and Oleksandr Zhevaho, “Development of a toolkit for analyzing software debugging processes using the constructive approach,” Eastern-European Journal of Enterprise Technologies, vol. 5, no. 2 (107), pp. 29–38, Oct. 2020, doi: https://doi.org/10.15587/1729-4061.2020.215090.