Journal of Theoretical and Applied Information Technology
31st May 2023. Vol.101. No 10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

POSITIVE INTERVENTION TECHNIQUE FOR A COMPUTER VIRUS

ABDULRAHMAN ALKANDARI[1] , MOHAMMAD ALAHMAD[2] , NAYEF ALAWADHI[3] , ABDULLAH ALSHEHAB[4] , ADEL ALFOUDERY[5]

[1245] Basic Education College (PAAET), Computer department, Kuwait
[3] MIS-Department, Communication & Information Technology Regulatory Authority (CITRA), Kuwait

E-mail: [1] aam.alkandari@paaet.edu.kw, [2] malahmads@yahoo.com, [3] n.alawadhi@citra.gov.kw, [4]aj.alshehab@paaet.edu.kw, [5] k.alfoudery@paaet.edu.kw

ABSTRACT

Negative connotations have generally been associated with computer viruses. In particular, popular belief holds that computer viruses are harmful and many people tend to use them for malicious reasons. However, they are not entirely bad as its known. They are codes that can either be used profitable or harmfully. This bias in existing literature has necessitated this research that aims to evaluate the possibility of using viruses for beneficial purposes while simultaneously improving storage and computing efficiency. First, related work on the subject is reviewed. Second, approaches to accessing virus code for editing purposes are reviewed. Third, beneficial aspects of viruses pioneered by Fred Cohen are reviewed. Fourth, computing and storage efficiency with use of beneficial viruses are reviewed. Lastly, Anti-Virus techniques that have been adopted in dealing with contemporary computer viruses have been reviewed. In this research, we present the possibility of changing computer virus from harmful to useful.

Keywords: *computer virus, beneficial virus, source code, self-replication, metadata.*

1. INTRODUCTION

This A computer virus is a malicious code that duplicates itself and alters other computer programs by injecting its own code in the programs source codes. Once the virus has succeeded in modifying and inserting its code in particular programs, the programs are said to have been 'infected' by a computer virus [1]. Developers of computer viruses normally employ social engineering tactics such as phishing and other deceptive methods in order to exploit any vulnerability that may be present on a host computer [1]. A large number of virus developers target computers that run on the Microsoft Windows Operating System because of two main reasons. First, the Windows OS is the most commonly used OS in the world with at least 88% of all personal computers running on it [1]. Second, the Windows OS has numerous security vulnerabilities such as the Eternal Blue vulnerability that paved way for the WannaCry Ransomware exploit in 2017 [1].

Virus have been widely known to cause massive damage over the world to computer systems, files, and programs, thereby translating into huge costs for affected individuals and corporations. In 2017, for instance, the WannaCry Virus infected at least 200,000 computers globally [1]. Affected individuals and corporations had to pay ransom in order to get back their files and programs. In as much as computer viruses are considered undesirable in the computing world, they could potentially harbor something positive [2]. Considering the many negative aspects that have been associated with viruses, most contemporary researchers have particularly focused on developing mechanisms to combat viruses [3]. Particularly, a large portion of research has focused on detection and quarantining of computer viruses. Detection methods that have taken center stage in state-of-the-art literature include signature-based, heuristics, and behavior-based techniques [4]. Quarantining refers to the act of isolating viruses from other programs and files stored on a computer HDD.

This research paper seeks to establish whether virus code can be modified and cause beneficial

uses in order to save space as well as benefit the Central Processing Unit and finding a new method of getting rid reusing the viruses in beneficial way. After quarantine has been performed, it is necessary to access the contents of the supposed virus program in order to perform the necessary edits that will convert it into a beneficial program. One of the novel ways discussed in this article is the use of a Hexadecimal Editor, which allows alteration of key values in the machine code [6]. A hexadecimal editor displays the components of the particular virus program in machine language format thereby allowing easy editing [6]. However, there are challenges, as will be discussed, associated with using a hexadecimal editor e.g. reverse engineering and virus/malware obfuscation difficulties [6]. The paper focus on a variety of techniques that may be applied in boosting storage space and computation efficiency. The particular techniques that will be discussed include metadata editing, source code editing, maintenance virus, and encryption purposes. Metadata editing or removal typically reduces storage space in cases where the file under consideration could also be used for other beneficial purposes. Source code editing is typically done using the hexadecimal editor as indicated earlier. The maintenance virus concept entails substituting a for some beneficial purpose while encryption entails converting the virus program into an encryption algorithm for protection of sensitive computer files.

2. PROBLEM DEFINITION

In the contemporary computing world, computer viruses have often been associated with negative connotations. From the general definition of a computer virus, viruses cause a lot of harm than good to computers, individual users, and the larger society. Therefore, most contemporary researchers in computer security focus on the detection and removal of computer viruses. Some of the advances made so far include the use of behavior-based, signature-based, and heuristic techniques to detect computer viruses and malware. Researchers have also focused on obfuscation of computer viruses [5]. Precisely, obfuscation makes the detection of these computer viruses quite challenging. While these advances have been made, little research has focused on the re-conversion of computer viruses into useful codes that would be beneficial to the computing environment and individual users. These thoughts are extracted from the idea that sometimes; positive intervention techniques are much more effective for solving common problems within the technology world. Preliminary researches on the possibility of converting computer viruses into 'good codes' were pioneered by researchers such as Cohen. However, advanced studies on this idea have not been conducted so far due to existing negative ethical and legal connotations associated with computer viruses. The explicit focus on detection and removal techniques leaves an unexplored dimension of computer virus research the possibility of changing viruses into useful codes.

3. CONTRIBUTION

This research is a great contribution to the current body of knowledge in matters computer security. Essentially, the research offers new dimensions in handling computer viruses. Instead of the traditional approaches that entailed deleting or quarantining viruses, the current research indicates the possibility of modifying the source code or metadata in order to convert the virus into a useful program. Modification of metadata is easily done in Microsoft Windows by navigating to file properties and following the rubric to edit. Source code editing is done through a hexadecimal editor. Through source code editing, a virus could be modified into a maintenance virus. It could also be used for encryption and decryption purposes, we are looking forward to design a new method that serve these techniques in order to get use of computer viruses in beneficial way.

4. OBJECTIVES OF THE PAPER

The purpose of this study is to change a computer virus into a beneficial code that could be used for various computing purposes. The first specific objective of the study is to find out the most effective approach that can be employed in gaining access to a virus source code. In order to modify a virus, the modifier must gain access to its contents, including the source code. Therefore, alteration of the source code through a variety of cracking mechanisms can be employed. The second objective is to investigate the possibility of altering the source code of a quarantined computer virus to reduce the space and computing resources it consumes while simultaneously using it for beneficial purposes. The paper is organized in five sections. The first section talks about related work of different studies. The second section addresses the first objective by reviewing approaches that could be employed in gaining access to a program code. The third section outlines how a virus code could be modified to reduce storage space and

improve CPU performance. The fourth section provides a discussing about Anti-Virus techniques. The last section provides a succinct conclusion regarding the discussion conducted in the paper.

5. RELATED WORK

This section explores existing literature on studies that have been conducted on computer viruses with a particular focus on negative as well as positive intervention techniques. Negative techniques focus on detecting and removing computer viruses from infected machines. On the contrary, positive techniques, which contains a highly unexplored area in computer security research, entail converting the virus to a useful code. Existing negative intervention techniques can be divided into three main categories; behavioral-based, signature-based, and heuristic techniques [6]. Behavior-based techniques are intelligent programs designed to detect any malicious computer program whose behavioral attributes conform to those defined in a typical malware [7]. Signature-based techniques observe patterns in events caused by a computer program, hence determine whether it is a computer virus or not. For instance, a signature-based detection system can detect the events that take place to other programs when a certain program is run [6]. Therefore, signature-based techniques work in much more similar manner like behavior-based techniques [6]. However, signature-based techniques simply observe the events that take place and the order in which they occur in case of an attack, while behavior-based techniques consider the behavior of a virus such as self-replication, integration, insertion, and deletion [8]. Heuristic techniques involve the discovery of hidden or obfuscated virus by use of artificial intelligence program. Precisely, heuristic techniques focus on detecting malware using data learning techniques. After successful detection, such computer viruses can then be removed safely from the machine.

Preliminary studies about good viruses were pioneered by the studies of Cohen. The main beneficial aspect proposed by Fred Cohen in his original hypothesis was that a virus could be altered to be used for compression of computer files that have been infected by the virus [9]. Precisely, this compression would prevent further spread of the virus to uninfected files. Since each executable in every infected file would attach the compression algorithm to the file, the execution process would cause decompression of the files hence limiting the effect of the computer virus. However, some researches consider computer viruses to be generally destructive agents [9]

Although, acknowledges the fact that computer viruses could be beneficial in some instances, he generally holds the opinion that virus are bad. According to [9], viruses are generally bad because they are difficult to control, wasteful, difficult to detect and remove, and consume a lot of resources. Furthermore, virus alter user data without their consent hence are bad from the ethical perspective [10]. Viruses can be differentiated from other harmful computer software such as malware and Trojans in the sense that viruses simply cause mild disturbance. For instance, a virus might keep displaying different messages on the screen. On the contrary, malware cause significant damage through such practices as stealing the user's sensitive information and encrypting the user's files for a ransom. For that matter, there is little justification to classify viruses as bad.

While the notion of using viruses for beneficial purposes is feasible, the viruses would need to be modified to a great extent. One particular area in which computer viruses can be used for beneficial purposes is advertising [11]. The main feature of viruses that makes them suitable for advertising is the self-replicating feature. Viruses have the capacity to make numerous copies of themselves and spread to other computers within a short time [11]. However, several contemporary marketing methods do not require target user consent either. For instance, messages viewed on billboards, mobile phone applications, and televisions are sent to recipients without their consent. Therefore, advertisement viruses are not unethical as purported by some researchers.

Viruses can also be used to detect and eliminate other viruses from a device [9]. While this idea is effective, it has probably been superseded by existing anti-virus programs which have specialized in detection and removal of computer viruses [9]. These anti-virus programs use sophisticated techniques such as heuristics and artificial intelligence to accomplish the role. Nevertheless, using viruses to detect and remove other malicious codes is relatively cheaper and cost efficient for smaller malicious codes. However, successful implementation of such an ethical virus has not been conducted anywhere in the computing world. Researchers that are against the utilization of computer viruses for beneficial purposes cite the

Journal of Theoretical and Applied Information Technology
31st May 2023. Vol.101. No 10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

uncertainty associated with controlling the viruses. Generally, these researchers indicated that a computer virus that finds positive usage cannot be guaranteed of complete safety. Nevertheless, the virus could be scanned using hex editor tools and edited to ensure that it does not contain any bugs that could cause serious problems on the infected machines.

Another important use of computer viruses highlighted [12] is personal gratification. Viruses can also be beneficial to their developers. Virus developers have been mainly considered to have ill intentions [12]. However, development of computer viruses could be regarded as a hobby for some people thus contributing to personal gratification [12]. Virus creation could also be considered an art as long as the original intention is not to harm users and their devices.

Literature explored by [13] introduces the concept of weaponisation of malware. This concept involves the intentional usage of malware to cause harm to perceived adversaries. Malware could be used as a software weapon that is spread to computing devices owned by adversaries. Using malware for such purposes is not a new concept as it has been used in several cases in the past [13]. In the Gulf War, for instance, the United States infected Iraqi computers with malware in order to disorient the aerial control systems thus preventing aerial strikes by Iraqi soldiers [14]. Even though the malware turned out to be beneficial to the Iraqi soldiers, the case indicates the positive usage of malware for war purposes. Cyber warfare simply describes the use of cyber-crime techniques to cause harm to adversaries. However, such risks are minimal hence weaponization of viruses is a beneficial aspect of computer viruses.

There are also arguments centered on the psychological and social risks of investing in 'good virus'. Essentially, the general public disagrees to computer viruses and would not buy the idea of a good virus [13]. Therefore, engagement in programs designed to develop and distribute good viruses is likely to face public backlash. Furthermore, people might possibly fear using computing resources due to the belief that these resources are designed to perpetrate cyber-crimes. No company would risk losing its reputation by engaging in an activity that the public perceives to be dangerous. Nevertheless, large-scale deployment of good viruses may begin by educating the intended users on the benefits of the virus.

6. HACKING AND ACCESSING VIRUS CODE

The first objective of this study was to evaluate the methods that could be used to gain access to the source code of a virus and edit its contents to convert it into a beneficial program. The static analysis approach is the most common method employed in learning the attributes of malware. Static assessment simply contains inspecting the malware program when it is not running hence it is the safest method. In the dynamic approach, for instance, there is risk of the computer used to edit the code being infected with the virus. First, the malware code is allowed run through a variety of anti-virus programs [6]. This is necessary to determine whether the code is actually malicious in any way. Attributes used to validate the malicious nature of the code include opcode frequency and control flow graph. Behavior of the code is not considered in this case because, the behavior can only be learnt after execution of the malicious code. The malicious code is then opened in a hexadecimal editor (as shown in Figure 1) to allow editing and changing of key values hence making the program useless. A hexa-decimal editor is a special program that is used to alter the basic contents of a computer program or file by changing of binary figures represented in hexa-decimal form like the one presented in Figure 1.

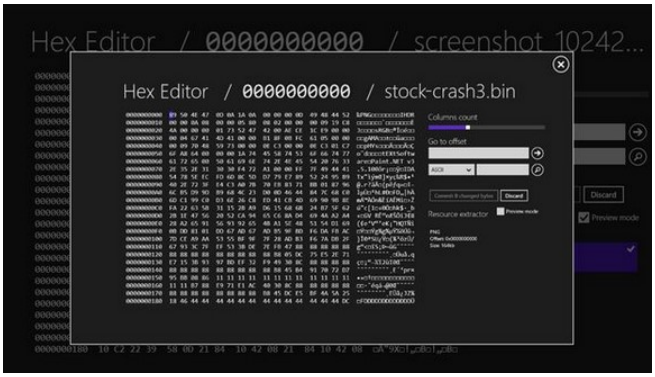

Figure 1: A Typical Hex Editor

If the malware file is using any packer application – e.g. the Ultimate Packer or UPX for executable files – then data compression might be possible [6]. In this way, it is possible to reduce the size of the malware program, hence save on storage space of the quarantined program. However, there are two hidden challenges associated with code editing using this approach. First, using a hexadecimal editor is difficult and requires the reverse engineer to have huge knowledge in assembly language programming. The engineer

must also be extremely careful while editing the code. If care is not taken, the edited version might turn out to be more harmful than the original version, hence a big threat will be created to the computer on which editing is being made. While the issue of hexa-decimal editing is not a significant problem associated with code editing, there is another more serious challenge involving obfuscation. Essentially, the entire process of accessing the malware code requires reverse engineering tactics. However, malware developers have upgraded their tactics as well such that they employ obfuscation in creating malware [6]. Obfuscation techniques are aimed at making the malware code obscure and difficult to decipher. For malware in which a high level of obfuscation has been employed, the process of reverse engineering is often expensive and extremely complicated. Furthermore, binary obfuscation compresses the malware file, hence making further compression even with UPX packer applications impossible. Consequently, obfuscation might cause the reverse engineer not to achieve the goal of file size reduction by compression. However, drawbacks of static reverse decryption are addressed by modern dynamic processes. The virus code is probably a .dll or .exe executable file that is generally non-editable using common text editing tools. Typically, the executable file has its contents followed in binary or machine language notation. Therefore, editing may be achieved in several ways. First, the file can be opened in a hexadecimal editor. Apart from using the hexa-decimal editor, the contents of the executable can be accessed by opening it in a decompiler or resource hacker tool (e.g. Microsoft's Resource Editor). The selection of an appropriate decompiler can only be done when the reverse engineer knows the actual programming language in which the malware was written [15]. However, even if they do not know, there are tools that can be used to decipher the programming language. Once the malware code has been decompiled back into its high-level language equivalent, it becomes easier for the reverse engineers to edit and alter its behavior.

7. VIRUS MODIFICATION AND BENEFICIAL USES

There are some cases in which a virus can be used for beneficial purposes directly without the need for modification. Once a computer virus can be edited, its behavior can also be altered as will, as long as the reverse engineer correctly reads and interprets the code. However, high-level editing can only be done when the malware has been decompiled using a high-level decompiler. There are various ways in which a self-replicating and often destructive computer virus could be edited in order to make it beneficial to the CPU.

Editing to Boost storage space and Computing Efficiency

6.1 Metadata Editing or Removal

If the end-goal is to increase storage space by reducing file size only, then metadata editing, or deletion could be a better approach as compared to code editing because it is much simpler. Metadata refers to a file's hidden information such as authorship, date of creation, and the number of times the file has been executed. Removing metadata results in the reduction of the file size, hence giving out more room for storage. Metadata removal can be achieved by various editing software such as Photo Mechanic and Lightroom [8]. However, metadata removal is only effective in situations where there is no need to alter the actual code and behavior of malware. Precisely, reduction of storage space can be employed if the virus is found to be beneficial. This is because, a virus that is not beneficial should be deleted instead of being edited to boost storage space. Metadata removal and modification can also be employed after code has been successfully edited using a resource or hexadecimal editor. Table 1 below is a summary of how metadata editing is done.

Table 1: Metadata Editing

Understand metadata conventions
Access metadata
Locate metadata tags
Rewrite or remove metadata
Save or apply changes

6.2 Source code editing

Storage space can be boosted through source code editing. The virus source code should be edited to remove the self-replicating feature to increase storage space. However, some researchers argue that self-replication is a principal attribute of a computer virus hence removing this feature will make the virus lose its true meaning [13]. However, the fact remains that it will still have been converted into a beneficial use while simultaneously removing the self-replicating

feature that is sometimes undesirable. In some cases, the self-replicating feature is desired which means that removing the feature will make the virus redundant. For instance, as seen in the studies, a computer virus that is used for advertisement purposes makes use of the self-replicating feature to make several copies of itself and send itself to several contacts in the users address book. In such cases, editing should target the algorithm that defines 'save and send.' Instead of saving and sending, the virus should be modified to only 'send.' In particular, several copies that are saved on the original computers' hard drive typically reduce storage space significantly. Furthermore, a lot of computing resources are required to update the registry as well as perform other automatic file operations. Getting rid of the 'saving' feature should begin by selecting a single virus file and editing the code in such a way that its copies are instantly deleted from the machine's hard drive after being sent to desired destinations. However, a self-replicating virus that constantly sends its copies to external destinations consumes huge amounts of computing resources. In order to counter this disadvantageous trait, the self-replicating virus can be modified to limit the number of replications and encodings made. Limiting the self-replicating behavior optimizes computing power. Furthermore, the virus code can be modified to eliminate random replications. For instance, the code can be modified in such a way that the virus only sends a certain number of copies of itself after every six hours. In this way, the computer's CPU is not overworked through constant replication and encoding of files. The computer can also serve other functions apart from viral advertising. Table 2 below is a summary of how source code editing is done, its benefit, and its applicability.

Table 2: Source code editing

Editing type	Source code modification
How editing is done	Source code accessed via hexadecimal editor
Benefit	Eliminate the self-replicating feature (where necessary) in order to minimize usage of computing resources. Could also be edited in order to perform beneficial encryption/decryption
Applicability	In situations where the virus is beneficial

6.3 The Maintenance Virus

The concept of a maintenance virus has been suggested as a probable approach to the creation of a 'good virus.' A maintenance virus continuously replicates and spreads itself to different computers on a network [9]. Assuming that the main destructive element of the original malware is random deletion of files and programs stored on an infected drive, then the code could easily be edited into a maintenance virus that only temporary files and redundant programs can access. In this way, a computer is continuously kept clean, and CPU performance is maintained at an optimal level. A malware variant that randomly hinders other programs from running effectively can also be converted into a maintenance virus, which actively stops background programs from running and utilizing much of the computer's resources. Too many temporary files consume a significant volume of storage yet are hard to locate and delete. Similarly, a huge number of background programs with minimal benefit to the user only consume a significant portion of the CPU's processing power [9]. Therefore, the conversion of a self-replicating virus into a maintenance virus that deletes temporary files and hinders background programs from running could potentially improve computing power. Table 3 below is a summary of the maintenance virus modification type.

Table 3: Maintenance Virus

Modification Type	Maintenance Virus
How modification is done	Source code accessed via a hexadecimal editor; editing done such that the virus only deletes unwanted files.
Benefit	Deleting unwanted files thereby maintaining the healthy state of the machine. Could also be modified to be constantly closing all background programs that hinder effective performance.
Applicability	In situations where a computer keeps too many log files and background processes

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6.4 Encryption Purposes

A Virus could be edited and converted into a disk encryption and file compressor program. In order to protect computer files from illegal access, computer users employ a number of tactics. Encryption is one of the methods employed. There are computer viruses which encrypt files once they attach themselves on the files. The 2017 WannaCry virus, for instance, was a kind of ransomware that encrypted files and hard drives such that the user could only access them after paying a ransom. Figure 2 indicates the Wannacry ransomware interface. Such a virus could be converted into an encryption program that encrypts hard-drives and files, thereby preventing illegal access to them.

Figure 2: The 2017 WannaCry Virus caption

However, extreme care must be taken during the editing process not to permanently encrypt files and programs. Cohen proposed the use of computer viruses to compress infected files. Compression of infected files serves two main purposes. First, it allows creation of more storage space, particularly when the infected files have been quarantined rather than deleted. Second, compression of infected files can be done in order to separate the infected files from normal files, hence prevent the virus from further replicating. Nevertheless, the conversion of a computer virus into a file compression program is highly probable. However, viruses are difficult to fully tame, are wasteful, and often contain bugs that are probably unknown to the computer user [10]. Therefore, the notion of converting a computer virus into a 'good virus' somehow sounds irrelevant. For instance, [8] states that operating systems have the capability to perform the same file compression roles without appending to the decompression procedure for every infected file. Furthermore, operating systems have the capability to perform perfect encryption better than a modified computer virus [9]. The purported benefits of maintenance viruses could be performed by concurrent processes in the Operating System [9]. However, maintenance viruses consume lesser resources as compared to concurrent processes in memory and operating system [9]. The implementation of maintenance viruses requires minimal human effort and intervention [9]. This forms the basis for the suggestion of converting computer viruses into maintenance viruses. However, a virus could also be modified based on the 'good viruses' model. Precisely, The operating system should be made in such a way that it attracts viruses and suggests the file repositories that the virus should infect or delete [2]. For the beneficial aspects to come out, the operating system must be able to suggest repositories such as temporary files and infected files. However, exchange of digital signatures for trust purposes is necessary. The host needs to validate that the viruses that has been sent is not forged. Table 4 below indicates a summary of all the beneficial modifications that can be made on a computer virus.

Table 4: Summary Table

Maintenance virus	Malware converted into a self-replicating virus that deletes infected files
Encryption type	Malware encrypts files for protection purposes rather than ransom
Compression type	Compresses files to save disk space
Good virus model	Operating system designed to attract virus and suggests files to be deleted or modified

8. ANTI-VIRUS TECHINQUES

Anti-virus Software use different detection techniques to discover malicious programs running in computer devices. Given the rapid changes in technology and the advanced nature that the virus software is taking, more advanced detection methods continue to be developed [16]. Most of the traditional detection systems continue to be phased out due the fact that virus writers now use more sophisticated codes [17]. For example, the signature

scanning method is gradually becoming inapplicable due to difficulties in maintaining signature databases. There are several virus detection methods applied by the anti-virus software. Data mining methods detect viruses through a classification system that categorizes file behaviors as malicious or not [18]. This capability is achieved through the extraction of file features that can be used in the classification system. Another method is the signature technique which operates by storing information about viruses in its database and matching the same with the signatures indicated by detected threats. This technique is the most widely used by anti-virus software [17]. However, there are viruses that do not yet have published signatures, limiting the capabilities of this method. Furthermore, Due to the limitations associated with the signature scanning method, the use of string scanning has been advocated for. In this technique, the signature string undergoes scanning in the new virus using special conditions in byte comparison. This technique is a more advanced version of the signature scanning approach and has been deemed more reliable [17]. Another commonly used detection method is the behavioral based detection. In this detection system, the behavior of the virus is monitored and analyzed to determine any malicious activities. The program or file is then classified as malicious or not depending on the analysis results [17].

Pros and Cons of Virus Quarantine upon detection of a malware, anti-virus programs move the file to the quarantine [19]. One of the advantages of virus quarantine is that there are chances of file recovery if some of the user files or application programs are infected by the malware. Sending the items to quarantine also safeguards the computer by muting all the threats to ensure that they cannot be executed by any command [20]. In this case, the quarantine enables the user to move the malicious program to a location where it cannot be executed, limiting its threat to other files. The quarantine method is also associated with certain limitations. For example, more disk space is occupied by the quarantined files that are not used by the computer.

Uninstalling or deleting an anti-virus program leads to different actions on the quarantined files, depending on the type of program in use. Norton Community replied that uninstalling the anti-virus does can either eliminate the quarantined items or leave them in the quarantine [21], which means the quarantined files will be deleted permanently upon uninstallation. In this case, the files are lost and might not be restored by the user. However, there are anti-virus software that will leave the files quarantined, hence no threats to the user.

Figure 3 is a summary of dealing with an infection. An anti-virus software detected an infection, the admin user shall choice between either delete the file or quarantine, deleting the file will get rid of the infection. While quarantine will suspend the infection from processing. However, if the file has been quarantined, the user shall search for a backup of the file, if backup found, the user shall proceed one of the following three methods, either delete the quarantined file and paste the backup, rename the quarantined file then replace it with the backup or replace the infected file with the backup.

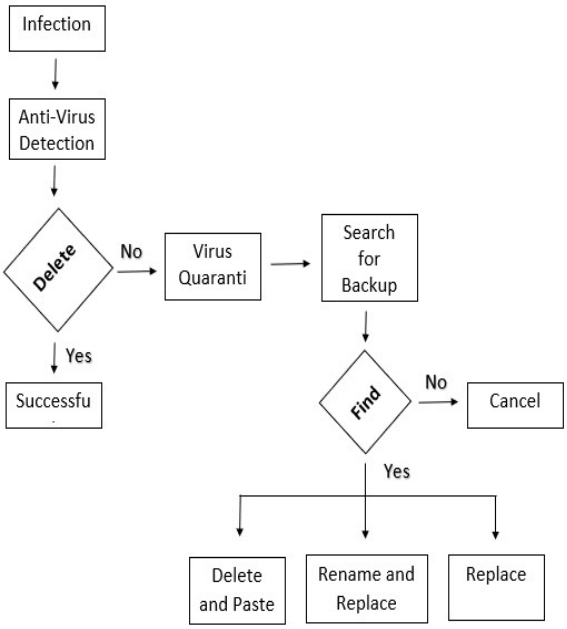

Figure 3: dealing with infection

9. CONCLUSION

The main aim of this paper was to evaluate the good use of computer viruses for beneficial purposes. The modern literature is biased in such a way that it tends to focus more on methods of detecting and removing computer viruses. However, this paper considers the negative connotations associated with computer viruses as a weakness in existing literature. Therefore, the paper seeks to explore the possibility of using viruses for beneficial purposes. The paper begins by exploring related work to the concept of beneficial viruses.

Only a few researchers have explored this area. Ideas suggested including the use of viruses as anti-viruses. Typically, converting a virus into a useful variant must initially involve access to the virus' contents and code. Editing the code could be done in a hexadecimal editor or a high-level code decompiler. However, a high-level decompiler is often preferred over a hexa-decimal editor because; the latter requires deep understanding of machine code hence is tedious and time consuming. Obfuscated and encrypted malware cannot be directly fed into a hex-editor or decompiler. The reverse engineer must first perform decryption to determine the encryption key. Code editing is then performed to convert the virus into a number of useful applications such as maintenance, compression of infected files, and encryption of drives for protection purposes. In some instances, code editing should be done to retain the beneficial aspects of the virus and remove the harmful aspects. For instance, self-replicating features can be modified or removed to improve storage efficiency and computing efficiency. Metadata editing is also effective in improving storage efficiency. While there are arguments that operating systems have capabilities to address most of the purported healthy benefits of viruses, modified viruses have been shown to consume less computer resources, hence much more preferred. In a nutshell, this research indicates that there is a possibility of converting computer viruses into beneficial programs for storage space and CPU optimization.

REFERENCES:

[1] R. James, The Wannacry Virus Analyzed, 2017.

[2] R.T. Guerra, G. Modelo-howard, A. Tongaonkar, L. De Carli, and S. Jha, Symantec Corporation, "Systems and methods for reverse-engineering malware protocols," U.S. Patent Application 15/159,187, 2018.

[3] C. Edge, and D. O'Donnell, "Malware Security: Combating Viruses, Worms, and Root Kits," In Enterprise Mac Security (pp. 221-242) Apress, Berkeley, CA, 2016.

[4] H.M. Deylami, R.C. Muniyandi, I.T. Ardekani, and A. Sarrafzadeh, "Taxonomy of malware detection techniques: A systematic literature review," In 2016 14th Annual Conference on Privacy, Security and Trust (PST) (pp. 629-636) IEEE, December. 2016.

[5] M.R. Asghar, and A. Luxton-Reilly, "Teaching Cyber Security Using Competitive Software Obfuscation and Reverse Engineering Activities," In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (pp. 179-184) ACM, February 2018.

[6] A. Ray, and A. Nath, "Introduction to Malware and Malware Analysis: A brief overview," International Journal, 4(10), 2016.

[7] A. Souri, and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," Human-centric Computing and Information Sciences, 8(1), p.3, 2018.

[8] Y. Ye, T. Li, D. Adjeroh, and S.S. Iyengar, "A survey on malware detection using data mining techniques," ACM Computing Surveys (CSUR), 50(3), p.41, 2017.

[9] M. Ludwig, and D. Noah, "The giant black book of computer viruses," American Eagle Books, 2017.

[10] M. Chowdhury, A. Rahman, and R. Islam, "Protecting data from malware threats using machine learning technique," In 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA) (pp. 1691-1694) IEEE, June 2017.

[11] Y. Yao, D. Lo Re, and Y. Wang, "Folk models of online behavioral advertising," In Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (pp. 1957-1969) ACM, February 2017.

[12] S. Abbasi, A. Naseem, A. Shamim, and M.A. Qureshi, "An empirical Investigation of Motives, Nature and online Sources of Cyberbullying," In 2018 14th International Conference on Emerging Technologies (ICET) (pp. 1-6) IEEE, November 2018.

[13] S. Cobb, and A. Lee, "Malware is called malicious for a reason: The risks of weaponizing code," In 2014 6th International Conference On Cyber Conflict (CyCon 2014) (pp. 71-84), IEEE, June 2014.

[14] D.J. Betz, "Cyberspace and the State: Towards a Strategy for Cyber-power," Routledge, 2011.

[15] J. Arends, and I. Kerstin, "Malware Analysis," 2018.

[16] P. Shahrear, A.K. Chakraborty, M.A. Islam and U. Habiba, "Analysis of Computer Virus Propagation based on Compartmental Model," Applied and Computational Mathematics. 7(1-2): 12-21. Sep. 2017.

[17] S. Amro and A. Alkhalifah, "A Comparative Study of Virus Detection Techniques,"

International Journal of Computer, Electrical, Automation, Control and Information Engineering. 9(6). 2015.

[18] N.A. Nitish, P.D. Shenoy & K.R. Venugopal, "Security in Data Mining- A Comprehensive Survey," Global Journal of Computer Science and Technology: C Software & Data Engineering, 16(5): 52-72. 2016.

[19] Y.W. Muchelule & N.M. Jacob, "Review of Viruses and Antivirus Patterns," Global Journal of Computer Science and Technology: C Software & Data Engineering, 17(3): 1-4. 2017.

[20] S. Sneha, L. Malathi & R. Saranya, "A Survey on Malware Propagation Analysis and Prevention Model," International Journal of Advancements in Technology. 6(2): 1-4. 2015.

[21] Adel Alfoudery, Abdulrahman Alkandari, S. Moein , "Survey on: Nano Technology", Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), Vol 10, No 2, p , 2018.

[22] Abdulrahman Alkandari, Imad Alshaikhli, "Implementation of Dynamic Fuzzy Logic Control of Traffic Light with Accident Detection and Action System using iTraffic Simulation" , Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), Vol 10, No 1, p 100-109 , 2018.