IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 4, Ver. I (Jul - Aug 2018), PP 53-62
www.iosrjournals.org

Malware Analysis and Mitigation in Information Preservation

Aru Okereke Eze and Chiaghana Chukwunonso E.
Department Of Computer Engineering, Michael Okpara University Of Agriculture, Umudike Umuahia, Abia State-Nigeria.
Corresponding Author: Aru Okereke Eze

Abstract: *Malware, also known as malicious software affects the user's computer system or mobile devices by exploiting the system's vulnerabilities. It is the major threat to the security of information in the computer systems. Some of the types of malware that are most commonly used are viruses, worms, Trojans, etc. Nowadays, there is a widespread use of malware which allows malware author to get sensitive information like bank details, contact information, which is a serious threat in the world. Most of the malwares are spread through internet because of its frequent use which can destroy large information in any system. Malwares from their early designs which were just for propagation have now developed into more advanced form, stealing sensitive and private information. Hence, this work focuses on analyzing the malware in a restricted environment and how information can be preserved. So, in other to address the negative effects of malicious software, we discussed some of the malware analysis methods which was used to analyze the software in an effective manner and helped us to control them. Various malware detection coupled with malware propagation techniques were also highlighted. This work was concluded by examining malware mitigation strategies which can help us protect our system's information.*

Keywords: *Malware Analysis, Mitigation, Malware Analysis Methods and Techniques, Malware Softwares, and tools etc.*

Date of Submission: 09--07-2018 Date of acceptance: 23-07-2018

I. Introduction

One of the most dangerous phenomenons we are observing today on the Internet is the unprecedented spreading of malware, a program written with malicious intents. Malware (Andreas, M. et al) is a general term used for programs having malicious code snippet which may cause a major threat to any user. Malware analysis is the study or process of determining the functionality, origin and potential impact of a given malware sample such as a virus, worm, trojan horse, rootkit, or backdoor. Malware or malicious software is any computer software intended to harm the host operating system or to steal sensitive data from users, organizations or companies. Malware may include software that gathers user information without permission. Malware is a malicious code that propagates over the network (Uppal, D. et al, Mehra, V. et al, & Verma, V. et al). It can be considered as the one to which new features can be easily added to enhance its attack. It can also be powerful so as to take full control of infected host and network connection disabling all the firewalls and installed ant viruses. The problem is cumulating with the use of internet as most of the web pages have been infected with various types of malware downloads which are delivered by just opening the web page. According to statistics by Google, 70% of the malware comes from popular sites. According to Osterman research survey, 11 million malware variants were discovered by 2008 and 90% of the malware comes from hidden downloads, pointers in trusted and popular websites. These threats can be delivered in many different variant modes often called blended threats which contain multiple components such as fishing attempts, spams, viruses, worms and Trojan (Honig, A. et al, Michael H.L. et al, Utter, D. et al).

Malware is typically used to steal information that can be readily monetized, such as login credentials, credit card and bank account numbers, and intellectual property such as computer software, financial algorithms, and trade secrets. Although many cybercriminal groups are trafficking in commodities shared by multiple industry sectors, such as credit card numbers, there are some situations wherein a single company is obviously the target of a single adversary, whether it be an organized crime syndicate, nation-state, or a single operative. (Zou C. et al, Towsley D. et al, Gong W. et al).

Everyday critical vulnerabilities are reported on a wide variety of operating systems and applications, and malicious activities perpetrated through Internet are quickly becoming the number one security problem, which ranges between large scale social engineering attacks and exploiting critical vulnerabilities. Recent sophisticated attacks use polymorphism and even metamorphism mixed with cryptographically strong algorithms and self-updating functionality which makes analysis and defense increasingly difficult. Nowadays a fast and reliable mechanism to mitigate, discern and generate vaccines for such attacks is vital for the successful

protection of networks and systems. Also the nature of malicious code shifted recently from trying to disrupt services or cause damage to actively seeking financial gain, as a matter of fact, today malware are designed to steal sensitive information such as credit card numbers, social security numbers, accounts, pin codes, and passwords and send the information to the miscreants for evil purposes including identity theft.

In Figure 1 below are reported malicious code threats detected in 2008 (1,656,227) by Symantec representing over 60 percent of the approximately total (2.6 million) malicious code threats detected in total, over time. The increase of complexity and sophistication of such attacks, the professionalization of attackers and evolution of attack patterns represent the major changes in the current threat landscape, and justify the need for cooperation among academics, security companies and governments to fight these threats. As usual, the first recommendation is always to patch vulnerabilities found in operating systems and applications, install a good anti-virus solution (and keeping it up to date!) and use a fire wall. But, as we know, apply all these solutions is not always enough; malware attacks can be mounted via different vectors or attack methods on a specific weak point, and these methods through which malware can compromise a system are sometimes referred to as threat vectors, and represent the areas that require the most attention when designing an effective solution to help reduce malware risks. Unfortunately, attackers have become skilled in circumventing such as traditional defenses. For example, even encrypted web transactions may not protect sensitive information if the victim's computer has been previously infected. To this end, when developing strategies to help reduce malware effects, it's fundamental to define required operational key points where malware detection and/or prevention can be implemented. Today threats complexity could not be lighted using a single solution or technology as a single line of defense but methods including a layered approach and using proactive, reactive and remediating mechanisms should be preferred (Steven A. et al, Blake H. et al, Matthew R. et al).

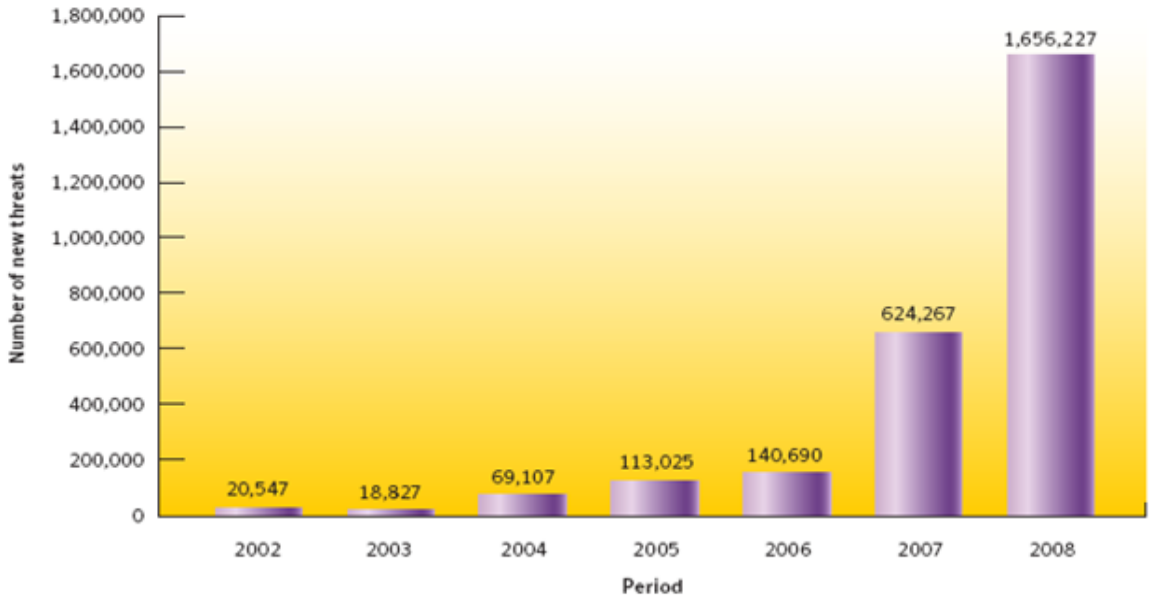

Figure 1-Evolution of new malicious code threats as reported by Symantec, (Mihai C. et al, Somesh J. et al, Douglas M. et al).

Antivirus software plays, still now, a key role in protecting systems from malicious codes; however, it should not be the only instrument used to determine malware attacks, because, as recently seen, it can fail. Finally, if a system has been compromised, there must be a common known way to handle such a situation, being able to minimize the damage and try to remediate it as soon as possible and in the best way. As computer attacks and malware evolve, as quickly as possible new responses and solutions are essential (Miller, T. et al, Cai S. et al).

1.1 Malware Propagation Techniques according to (Bayer, U. et al). The following gives us different ways malware are being propagated in our computer system:'

a. Web browsing - The easiest way of getting infected is through drive-by-download. Malware often spreads through unwanted software downloads, malicious PDF documents, word documents, or fake software. Using this technique, malware authors have no target other than to infect as many computers as possible. Modern browsers like Chromium (the open source project on which Google developed Chrome) include two mechanisms that are designed with security in mind. One component is the browser kernel that interacts with the operating system and the other is the rendering engine that runs inside a sandbox with restricted privileges. This design helps to improve browser security and mitigate attacks from malicious websites (Krugel, C. et al).

b. USB thumb drives - Thumb drives are also used to spread malicious software. This method uses the AutoRun feature to launch malware when the storage device is mounted by the operatingsystem. A common attack scenario is performed by intentionally dropping USB drives in front of targeted organisations (Kirda, E. et al).

c. Email Spear Phishing - Spear phishing is an e-mail spoofing fraud attempt that targets a specific organization, seeking unauthorized access to confidential data. Spear phishing attempts are not initiated by random attackers, but are more likely to be conducted by perpetrators out for financial gain, trade secrets or military information. Similar to e-mail messages used in regular phishing expeditions, spear phishing messages appear to come from a trusted source. Phishing messages usually appear to come from a large and well-known company or Web site with a broad membership base, such as eBay or PayPal. In the case of spear phishing, however, the apparent source of the e-

observe frequently visited websites that they visit and trust, afterwards theyinfect these websites with malware in the hope that a person from the targeted group will get infected (Andreas, M. et al).

II. Review of Related articles on malware analysis

This work reviews some of the related works in malware analysis. There have been many new and powerful malware analysis methods and techniques reported in many literatures. Previously, (Distler, D.) has used static and dynamic analysis for malware analysis. Meanwhile, (Ari, H.N.) also have been doing malware analysis with reverse engineering techniques using biscuit apt1 as a malware sample. Another malware analysis research also done by (Flores, H.) with win32.Kryptic. In the meantime, (Daoud, E. et al) has research regarding technique used by malware to avoid detection from antivirus. Research conducted by (Uppal, D. et al) focus more on technique and tools used in malware analysis.

Most of the literature we came across during our research was either focused on static analysis method or technique used for analysis malware without running the application directly. Whereas our work combines two methods of malware analysis, static and dynamic analysis method to get more detail information for characteristics of malware (Kruegel, C. et al and Tang Yanjun, L. N. et al).

According to [Park, Y.] in malware detection and analysis - The traditional approach for the detection of malicious code is based on signature matching of various complexity. A signature can be a sequence of bytes that identifies pieces of data or code of the malicious program, but even very complex algorithms that test whether a particular program satisfies certain properties. The advantage of using sophisticated detection methods is that signatures become more generic and thus a single signature can be used to detect multiple variants derived from the same family. On the other hand, from the remediation point of view, excessively generic signatures do not allow to distinguish variants. If single variants cannot be told apart, the remediation procedure cannot take variant-specific behaviors into account and cannot perform a complete cleanup (Nazario, J. and Holz, T.).

Purely signature-based approaches have demonstrated their weaknesses when packed, polymorphic and metamorphic malware appeared. The research community started to move toward behavior-based solutions. Behavior-based detection [Lorenzo, M. et al,] and analysis [Andreas, M., Christopher, K. et al] approaches do not focus on the syntactic structure of the analyzed program, but try to consider its semantics. Because these solutions work by observing a concrete execution of the malicious sample, they could provide much more accurate remediation procedures. Recently [Kolbitsch, et al.] proposed and effective and efficient malware detection method that can be used at the end host replacing or complementary to traditional antivirus software. This method is based on fine-grained models obtained by executing the malware program in a controlled environment, monitoring and observing its interactions with the OS resources; detection is done by matching extracted behavior models against the runtime behavior of unknown programs.

III. Types of malware and how they affect the system according to (Nazario, J.).

The best-known types of malware, viruses and worms, are known for the manner in which they spread, rather than any specific types of behavior. A computer virus is software that embeds itself in some other executable software (including the operating system itself) on the target system without the user's

knowledge and consent and when it is run, the virus is spread to other executables. On the other hand, a worm is a stand-alone malware software that actively transmits itself over a network to infect other computers. These definitions lead to the observation that a virus requires the user to run an infected software or operating system for the virus to spread, whereas a worm spreads itself.

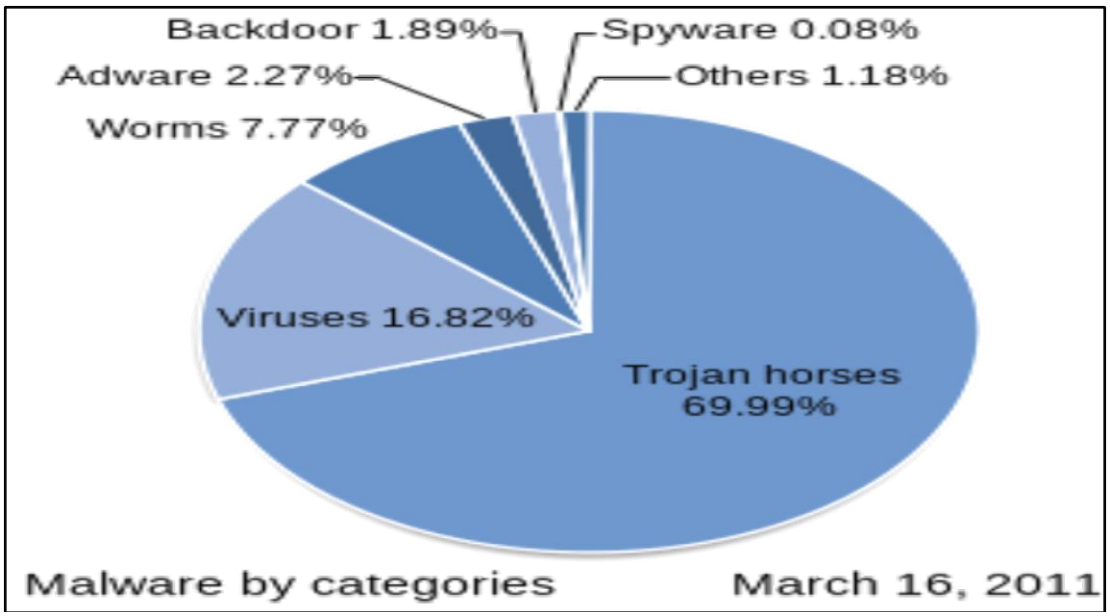

Figure 2 - A Pie Chart Showing in Percentage, the Different Categories of Malware (Guri, M. et al).

- **Viruses** - A computer virus is a software usually hidden within another seemingly innocuous program that can produce copies of itself and insert them into other programs or files, and that usually performs a harmful action (such as destroying data). An example of this is a PE infection, a technique, usually used to spread malware, that inserts extra data or executable code into PE files(Krugel, C. et al).

- **Worms** - Worms are aptly named for their ability to "crawl" through networks. Worms replicate themselves but do not embed themselves in other programs as a virus tends to do. Worms move along a network connection seeking vulnerable machines to infect. For example, in 1988, the "Morris Worm" became so widespread that it managed to slow the entire internet.

- **Trojans** - A Trojan horse is a harmful program that misrepresents itself to masquerade as a regular, benign program or utility in order to persuade a victim to install it. A Trojan horse usually carries a hidden destructive function that is activated when the application is started. The term is derived from the Ancient Greek story of the Trojan horse used to invade the city of Troy by stealth. Trojan horses are generally spread by some form of social engineering, for example, where a user is duped into executing an e-mail attachment disguised to be unsuspicious, (e.g., a routine form to be filled in), or by drive-by download. Although their payload can be anything, many modern forms act as a backdoor, contacting a controller which can then have unauthorized access to the affected computer. While Trojan horses and backdoors are not easily detectable by themselves, computers may appear to run slower due to heavy processor or network usage. Unlike computer viruses and worms, Trojan horses generally do not attempt to inject themselves into other files or otherwise propagate themselves (Krugel, C. et al).

- **Spyware** - Spyware's main function is to monitor what you are doing on your computer, on or off the internet, and send that information to a third party without your knowledge. In some cases, this data harvesting is used solely for marketing purposes. In other cases, the intent is more sinister. A theft might occur when an imposter, posing as a client, directs a CPA to send a payment to an illegitimate recipient.

- **Screen-locking ransom ware** - Lock-screens, or screen lockers is a type of "cyber police" ransom ware that blocks screens on Windows or Android devices with a false accusation in harvesting illegal content, trying to scare the victims into paying up a fee. Jisut and SLocker impact Android devices more than other lock-screens, with Jisut making up nearly 60 percent of all Android ransom ware detections (Kirda, E. et al).

- **Rootkits -** Once malicious software is installed on a system, it is essential that it stays concealed, to avoid detection. Software packages known as rootkits allow this concealment, by modifying the host's operating system so that the malware is hidden from the user. Rootkits can prevent a harmful process from being visible in the system's list of processes, or keep its files from being read.Some types of harmful software contain routines to evade identification and/or removal attempts, not merely to hide themselves. An early example of this behavior is recorded in the Jargon File tale of a pair of programs infesting a Xerox CP-V time sharing system: Each ghost-job would detect the fact that the other had been killed, and would start a new copy of the recently stopped program within a few milliseconds. The only way to kill both ghosts was to kill them simultaneously (very difficult) or to deliberately crash the system (Kirda, E. et al).

- **Backdoors –** A backdoor is a method of bypassing normal authentication procedures, usually over a connection to a network such as the Internet. Once a system has been compromised, one or more backdoors may be installed in order to allow access in the future, invisibly to the user. The idea has often been suggested that computer manufacturers preinstall backdoors on their systems to provide technical support for customers, but this has never been reliably verified. It was reported in 2014 that US government agencies had been diverting computers purchased by those considered "targets" to secret workshops where software or hardware permitting remote access by the agency was installed, considered to be among the most productive operations to obtain access to networks around the world. Backdoors may be installed by Trojan horses, worms, implants, or other methods (Guri, M. et al).

IV. Malware Detection Techniques

There are techniques used in detecting malware activities in the system (Firdausi, I. et al).

a. Static analysis detection technique - It is the procedure of analyzing software without executing it. During static analysis [Bergeron, J. et al] the application is break down by using reverse engineering tools and techniques, so as to re-build the source code and algorithm that the application has created. Static analysis can be done through program analyzer, debugger and disassembler. Various static analysis techniques are as follows:

- **Signature based detection technique** - This technique is also known as pattern matching or string or mask or fingerprinting technique. A signature is a bit of sequence injected in the application program by malware writers, which uniquely identifies a particular malware. To detect a malware in the code, the malware detector search for a previously specified signature in the code (Arun, L. et al).

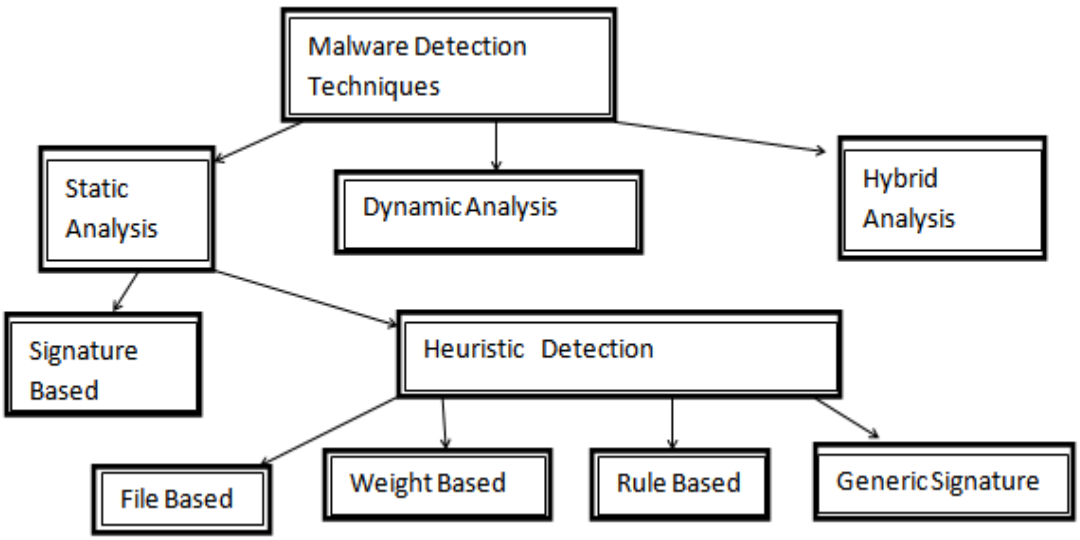

Figure 3 – **Various Malware Detection Techniques** (Savan, G. et al, & Bergeron, J. et al).

- **Heuristic detection technique** - This technique is also known as proactive technique [McGraw, G & Morrisett, G]. This technique is similar to signature based technique, with a difference that instead of searching for a particular signature in the code, the malware detector now searches for the commands or instructions that are not present in the application program. The result is that, here it becomes easy to detect new variants of malware that had not yet been discovered. Different heuristic analysis techniques are:

i. **File based heuristic analysis** - Also known as file analysis. In this technique, the file is analyzed deeply like the contents, purpose, destination, working of file. If the file contains commands to delete or harm other file, than it is considered as malicious. (Muazzam, A. S.).

ii. **Weight based heuristic analysis** - It is the much ancient technique. Each application is weighted according to the danger it may possess. If the weighted value exceeds the predefined threshold value, then the application contains malicious code. (Vasudevan, A. & Yerraballi, R.).

iii. **Rule based heuristic analysis** - The analyzer, here, extracts the rules defining the application. These rules are then matched with the previously defines rules. If the rules are mismatched, then the application contains malware. (Mohammad, N. S.).

iv. **Generic signature analysis** - In this signature, variants of malware are detected. A variant of malware means, the malware are different in behavior but belong to same family like "identical twins". This technique uses previously defined antivirus definition, to discover new variants of malware (Arun, L. et al).

b. Dynamic analysis detection technique - The process of analyzing the behavior or the actions performed by the application while it is executing is called dynamic analysis [Savan, Gadhiya, et al]. Dynamic analysis can be done through monitoring function calls, tracking the information flow, analyzing function parameters and tracing the instructions. Generally a virtual machine or sandbox is used for this analysis; the doubted application is usually run into a virtual environment. If the application behaves unusually it is categorized as malicious. Nowadays, there are behavioral blocking software, which blocks malicious action of the program before their attack.

c. Hybrid analysis detection technique - This technique is the combination of both static analysis and dynamic analysis [Robiah, Y. et al]. The procedure it follows it that it first checks for any malware signature if present in the code under inspection and then it monitors the behavior of the code. Hence this technique combines the advantages of both the above techniques.

4.1 Some static analysis and dynamic analysis tools.

Table 1: Brief overview of dynamic analysis tools of malware.

Dynamic Analysis Tools of Malware (Vasudevan, A., et al)	**Description**
Process Explorer	Monitor currently running process
File Mon	Monitor file operation
Reg Mon	Monitor operation on registry
Reg Shot	Takes snapshot of the registry and associated files
TCP View	Display all TCP and UDP open connections and the process that opened and is using the port
TDI Mon	Logs network connectivity, but does not log packet contents
Ethereal	Packet Scanner that captures packets and support the viewing of contents/payload

Table 2: Brief overview of static analysis tools of malware.

Static Analysis Tools of Malware (Vasudevan, A., et al)	**Description**
Bin Text	Extracts strings from executables, reveal registry keys and IRC , SMTP commands stored in string format
IDA Pro	Disassembles executables into assembly instructions
UPX	Executable packer used by malware writers
Proc Dump	Dumps code from memory
Olly Dbg	Debugger that enables the user to attach to a process and insert breakpoints

Since all the techniques that these tools are using are behavior based analysis, it eliminates Zero-day attacks and also obfuscation, but sometimes malware can destroy the tools by making a DOS attack which will not allow the tool to analyze the malware. Also, these tools should be placed outside the emulated environment in order to avoid DOS attack that can be made by malware author on the tools.

V. Malware Analysis Methods

Malware analysis is a process to perform the analysis of malware and how to study the components and behavior of malware. For analysis malware, there are two main techniques for analysis malware that are the most commonly used methods;Static analysis and dynamic analysis.

1. **Static Analysis:**Static analyzing does not perform execution code instead it checks the control and data flow of the program to determine its characteristics. Here each possibility of the code can be analyzed by using the concept of backtracking and analyzing all the possibilities in which the code can be executed. There are three different static analysis techniques which are widely used, namely:

a. **String Signature:** In this technique, the analyzer looks for specific types of malicious code statements to know whether it is a malware or not (Andreas M. et al).

b. **Control Flow Graph:** In this method, the control flow between the code statements is checked to determine the malicious behavior of the program.

c. **Semantic-Aware Analysis:** Here, semantic of the programs is checked and the analyzer checks the actual meaning of the semantics and sees to it that there is no hidden meaning.

The limitation of this Static Analysis is, it may or may not analyze a self-modifying, obfuscating code which may cause a threat to one's system.

2. **Dynamic Analysis:**Toovercome the limitations of static, dynamic analysis is used. A dynamic analyzer actually executes a code to check its behavior. It uses an emulated environment to run the malicious code so that the actual information in the system is not harmed in the process. Using this method even the self-modifying code can be observed and used to create the signature. The various dynamic strategies to analyze a program are as follows:

a. **Information Flow Trace:** Here, the information flow is logged into a log file to keep a trace on the information and to see who has which information.

b. **Information Flow Tracking:** The data and information collected from one's system is kept track on to see where the information goes so that it does not fall into wrong hands. This technique analyzes whether sensitive information falls into the right hands and is leaked(Bayer, U. et al).

c. **Function Parameter Analysis:** The function calls which are invoked by the program do not give us all the information of the actions performed using that function.

Malware Analysis using the method of static analysis can be divided into two sub stages, namely Basic Static Analysis and Advanced Static Analysis, whereas dynamic malware analysis can also be divided into sub-stages, namely Basic Dynamic Analysis and Advanced Dynamic Analysis respectively. (Sikorski, M.et al, Honig, A. et al, Flores, A. et al).

- **Basic Static Analysis** - The basic method in static analysis, carried out testing against a program which is alleged as malware with doing the scanning using antivirus, moreover also doing hashing, and detection of packed or obfuscated at the program. As well as conducting an analysis of the structure of portable executable which is owned by the program. (Uppal, D. et al).

- **Advanced Static Analysis** - In the advanced method of static analysis, further analysis will be undertaken of the method of static analysis with analysis against the strings, linked libraries and function as well as using IDA disassembler (Zaqaibeh, B. et al).

- **Basic Dynamic Analysis** -The basic method in dynamic analysis, will be build a virtual machine that will be used as a place to do a malware analysis. In addition, malware will be analysis using malware sandbox and monitoring process of malware and analysis packets data made by malware (Zahn, K. J. et al).

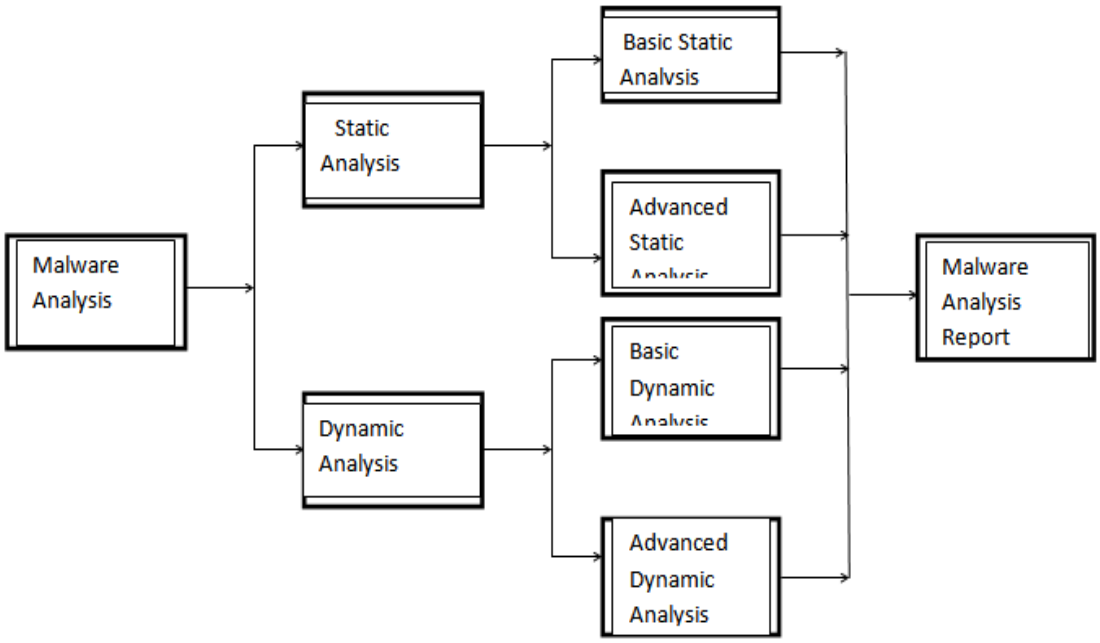

Figure 4- Malware Analysis Method - (Sikorski, M.et al, Honig, A. et al).

- **Advanced Dynamic Analysis** - In the advanced method of dynamic analysis, further analysis will be undertaken of dynamic analysis methods with debugging on malware, analysis the registry and do an analysis on a windows system (Wenhua, L. et al)

- **The Analysis Reports** generated by these tools gives in-depth understanding of the malware behaviour and valuable insight performed by them. The analysis system is required to have an appropriate representation for malwares which are then used for classification either based on similarity measure or feature vectors.

VI. Malware Mitigation Strategies

When developing strategies to help reduce malware, it is important to define necessary operational key points where malware detection and/or prevention can be implemented. When it comes to managing malware risk, a single device or technology should not be solely relied upon as the only line of defense. Preferred methods should include a layered approach using proactive and reactive mechanisms throughout the network. Antivirus software plays a key role in this area; however, it should not be the only instrument used to determine malware attacks.(Daoud, E. A. et al).

The following operational key points are discussed further in detail:

- Assessing malware risks
- Physical security
- Logical security

a. Assessing Malware Risks - When assessing malware risks, midsize businesses need to be mindful of the attack vectors that are most vulnerable to threats. How are they protected and to what extent? The following questions should be considered:

- Does the organization have a firewall installed?
- Does the organization have internal or external vulnerability scan analysis capability? How is the scanned information analyzed?

A tool such as the Microsoft Baseline Security Analyzer (MBSA) is recommended for scanning for misconfigurations or vulnerabilities. It is also possible to outsource the security vulnerability testing process by hiring outside vendors to assess the security environment and provide suggestions for improvement where deemed necessary (Anderson, B. et al).

b. Physical Security - Physical security entails restricting access to equipment for the purposes of preventing tampering, theft, human error, and the subsequent downtime caused by these actions.

Although physical security is more of a general security issue than a specific malware problem, it is impossible to protect against malware without an effective physical defense plan for all client, server, and network devices within an organization's infrastructure. (Nataraj, L. et al)'

c. Logical Security - Software safeguards for information systems in midsize businesses include user ID and password access, authentication, and access rights, all of which are crucial for managing malware risks. These safeguards help ensure that only authorized users are able to perform actions or access information on a particular server or workstation on the network. Administrators should ensure that systems are configured in a way that is consistent with the job function of the computer user.

6.1 Research Methodology – The research method employed is the static analysis method, dynamic analysis method and hybrid analysis method. The static and dynamic analysis method reviews published work on the malware analysis and mitigation in information preservation from the year 2000 to date. Papers were picked from distinguished journals and conferences which vary from the concepts and origin to objective research. The dimension of the researched papers was based on Malware Analysis in details and it's Mitigation Strategies in information Preservation. The research was done extensively. Issues and discoveries related to Malware Analysis and its Mitigation in Information was critically discussed and analysed conceptually. The selected articles were reviewed systematically based on the malware analysis methods and tools discussed in this journal. The analysis identifies and critically evaluates the mentioned methods reflecting what is being practiced and what is documented. This steps and processes conducted differentiate the most frequent analysis methods deemed to be necessary for malware analysis and mitigation in information preservation. A consideration of the whole discoveries (as shown in the matrix) with malware analysis and mitigation strategies developed is then used for final analysis in this study. The considerations or reflection should establish the basis of successful strategies in mitigating malware, which is supported by the analysis methods earlier outlined. Based on the summary on the malware analysis, charts and tables are drawn up to represent the discoveries and findings of the review work.

VII. Discussion and Conclusion

Day to day malware is being spread via network like wildfire. However, preserving information and records in a system involves ensuring they remain accessible, usable and free from malware attacks. Information and records will deteriorate over time, whether they're paper, photographic, digital or audiovisual if they cannot be preserved from possible malware attacks. While the rate of deterioration differs, the lifespan of your information and records will depend on how they are managed and the preservation actions applied throughout their lifecycle.In this work,we had survey a study about various types of malware, malware propagation techniques and categories of malicious software. Although, the rate hazards of malware are increasing at an alarming rate, this paper provides a thorough study of tools for analyzing malware with different techniques. In addition, malware mitigation strategies which forms the major part of this work was discussed in detailed so that information can be properly preserved for a long term. Hence, the need for information preservation in highly vital and in demand.

It is likely that the methods identified in this study would have a significant impact in helping ICT companies and organizations to achieve effective malware mitigation strategies. The study highlighted the steps required for effective and good malware mitigation strategies; there is a need for follow-up research using the tools and different methods to help organization understand what is required to improve the effectiveness of their information preservation policy against malware software.

This study is expected to contribute to both the information communication technology (ICT) and academic world. Practically, the results of this research work will help organizations better implement the use of malware mitigation strategies in information preservation. Furthermore, this study provides recommendations for future research avenues. Firstly, objective study needs to be conducted and the theoretical model's reliability and validity which is to be developed without being tested, needs support too from the studies. Finally, as objective study and research has been done on malware analysis and mitigation in information preservation, the functionality of all methods needs to be developed and further validated.

References

[1]. Ari H, N. et al. (2014). Penerapan Analisa Malware Pada Biscuit apt1 Menggunakan Teknik Reverse Engineering. Journal of KNSI.
[2]. Anderson, B., Quist, D., Neil, J., and Lane, T. (2011). Graph BasedMalware Detection Using Dynamic Analysis. Journal in Computer Virology, 7, 247-258,http://dx.doi.org/10.1007/s11416-011-0152-x.
[3]. Andreas, M. Christopher Kruegel, and Engin Kirda. (2007). Exploring Multiple Execution Paths for
[4]. Malware Analysis. In Proceeding of the IEEESymposium on Security and Privacy, Oakland, California, USA, pages 231.
[5]. Anderson, B., Storlie, C. and Lane, T. (2012). Improving Malware Classification: Bridging the Static/Dynamic Gap.

[6]. Arun, L. et al. (2005). "Are Metamorphic Viruses Really Invincible? Part 2", Virus Bulletin.
[7]. Bergeron, J. et al. (2001). Static Detection of Malicious Code in executable programs. International Journal of Req. Engineering.
[8]. Bayer, U., Moser, A., Kruegel, C. and Kirda, E. (2006) Dynamic Analysis of Malicious Code. Journal in Computer.
[9]. Bayer, U., Comparetti, P.M., Hlauschek, C. and Kruegel, C. (2009). Scalable, Behavior-Based Malware Clustering. Blake, H. et al. (2007). Toward Automated Dynamic Malware Analysis Using Cwsandbox. IEEE.
[10]. Christopher, K. et al. (2013). OPEM: A Static-Dynamic Approach for Machine Proceedings of 5th ACM Workshop on Security and Artificial Intelligence (AISec), 3-14.
[11]. Daoud, E A., and Zaqaibeh, B. et al. (2008). Computer Virus Strategies & Detection Methods. Int. J. Open Problems Compt. Math., Vol. 1, No. 2.
[12]. Flores, A. H. et al. (2012). Malware Reverse Engineering part1 of 2. Static analysis.Technical Report.
[13]. Firdausi, I., Lim, C. and Erwin, A. (2010). Analysis of Machine Learning Techniques Used in Behavior Based Malware Detection.
[14]. Guri, M. et al, Y. (2015). "BitWhisper: Covert Signaling Channel between Air-Gapped Computers Using Thermal
[15]. Manipulations," IEEE 28th Computer Security Foundations Symposium, Verona, pp. 276-289.
[16]. Honig, A. et al. (2012). Practical Malware Analysis. No Starch Press. Retrieved 5 July.
[17]. Holz, T. et al. (2011). Toward automateddynamic malware analysis using CWSandbox. IEEE Security & Privacy.
[18]. Kirda, E. &Kendall, K., (2007). Practical malware analysis. Technical Report. Mandiant, Intelligent Information Security.
[19]. Kolbitsch, et al. (2013). Discriminant Malware Distance Learning on Structural Information for
[20]. Automated Malware Classification. Proceedings of the ACM SIGMETRICS/International Conference on Measurement an Modeling of Computer Systems, 347-348.
[21]. Lorenzo, M. et al. (2007). Practical malware analysis. Technical Report. Mandiant, Intelligent Information Security.
[22]. McGraw, G. & Morrisett, G. (2000). Attacking malicious code: A report to the infosec research council. IEEE Software, 17(5):33–44
[23]. Mohammad N. S, et al. (2009). Malware Analysis Bachelor's Thesis. Helsinki Metropolia University of Applied Sciences. Muazzam A. S. et al. (2008). Data Mining Methods for Malware Detection: University of Central Florida.
[24]. Mihai C and Mattew, W. et al. (2013). "Understanding Threats: Rootkits & Botnets". US-Cert.
[25]. Miller, T. et al. (2008). Unknown Malcode Detection via Text Categorization and the Imbalance Problem.
[26]. Proceedings of the 6th IEEE International Conference on Intelligence and Security.
[27]. Michael, H.L. et al. (2012). Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code.
[28]. Mihai Christodorescu, et al. (2013). Malware threats and mitigation strategies: US-CERT informational whitepaper.
[29]. Nataraj, L., et al. (2011). Malware Images: Visualization and Automatic Classification.
[30]. Nazario, J. Anderson, J. Walsh, R. and Connelly, C. (2001). The future of Internet worms. Technical report, Crimelabs Research.
[31]. Nataraj, L., Yegneswaran, V., Porras, P. and Zhang, J. (2011). A Comparative Assessment of Malware
[32]. Classification Using Binary Texture Analysis and Dynamic Analysis. Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, 21-30.
[33]. Park, Y., et al. (2010). Fast Malware Classification by Automated Behavioral Graph Matching.
[34]. Robiah, Y. & SitiRahayu, S. et al. (2009). A New Generic Taxonomy on Hybrid Malware Technique. International Journal of Computer Science and Information Security, Vol. 5. No. 1.
[35]. Somesh, J. et al. (2011). Automatic Analysis of Malware Behavior Using Machine Learning. Journal of Computer Security, 19, 639-668.
[36]. Sikorski, M., and Honig, A. et al. (2012). Practical Malware Analysis. San Francisco: William Pollock.
[37]. Savan Gadhiya, & Kaushal Bhavshar. (2013). "Techniques for Malware Analysis". International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4.
[38]. Steven, A. et al. (2011). Collective Classification for Unknown Malware Detection. Proceedings of the International Conference on Security and Cryptography, Seville, 18-21, 251-256.
[39]. Towsley, D. (2010). Differentiating Malware from Cleanwares Using Behavioral Analysis. Proceedings of 5th International Conference on Malicious and Unwanted Software (Malware), Nancy, 19-20 23-30.
[40]. Tang Yanjun, L.N. et al. (2001). Data Mining Methods for Detection of NewMalicious Executables. Proceedings of IEEE Symposium on Security and Privacy, Oakland, 14-16, 38-49.
[41]. Uppal, D., and Mehra, V., et al. (2014). Basic survey on Malware Analysis, Tools and Techniques. International Journal on Computational Sciences & Applications (IJCSA) Vol.4, No.1.
[42]. Vasudevan, A., et al. (2006). "SPiKE: Engineering Malware Analysis Tools using Unobtrusive Binary-Instrumentation". Australasian Computer Science Conference. Verma, V. et al. (2014). Antivirus Isn't Dead, It Just Can't Keep Up. Technical Report. Lastline Labs.
[43]. Wenhua, L, Tang Yanjun, L. N. et al. (2012). Reverse Analysis of Malwares: A Case Study on QQ Passwords Collection. Journal of Software, Vol. 7, No. 8.
[44]. Yerraballi, R. et al. (2010). Malware Obfuscation Techniques: A Brief Survey. Proceedings of International conference on Broadband, Wireless Computing, Communication and Applications, Fukuoka, 4-6 November 2010297-300.
[45]. Zahn, K. J. et al. (2013). DC3 Digital Forensic Challenge Basic Malware Analysis Exercise. Journal of SANS Institute.
[46]. Zou, C. Towsley, Gong, W. & Cai.Routing S. (2015). "Worm: A fast, selective attack worm based on IP address information." In Proc.IEEE Work. on Princ. Adv. and Dist. Simul. (PADS), pages 199–206.
[47]. Zaqaibeh, B. et al. (2013). Classification of Malware Based on Integrated Static and DynamicFeatures. Journal of Network and Computer Application, 36, 646-556.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Aru Okereke Eze "Malware Analysis and Mitigation in Information Preservation." IOSR Journal of Computer Engineering (IOSR-JCE) 20.4 (2018): 53-62.