Turkish Journal of Computer and Mathematics Education *Vol.12 No.3(2021), 5776-5787*

Research Article

Macro Based Malware Detection System

Balal Sohail [a], Ma'en Tayseer Ekrayem Alrashd[b], Yaseein Soubhi Hussein[c], Mohammad Tubishat[d], Shounak Ghosh[e], Ahmed Saeed Alabed[f]

[a, b, d, e] Asia Pacific University of Technology & Innovation, Kuala Lumpur, Malaysia
[c, f]Department of Information Systems and Computer Science, Ahmed Bin Mohammed Military College, Qatar
[a] Balalsohail007@gmail.com, [b] dr.maen@apu.edu.my , [c] dr.yaseein@abmmc.edu.qa ,[d] tubishat@staffemail.apu.edu.my
[e]shawn@apu.edu.my, [f] ahmed.alabed@abmmc.edu.qa

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: Macro based Malware has taken a great rise is these recent years, Attackers are now using this malware for hacking purposes. This virus is embedded inside the macro of a word document and can be used to infect the victim's machine. These infected files are usually sent through emails and all antivirus software are unable to detect the virus due to the format of the file. Due to the format being a rich text file and not an executable file, the infected file is able to bypass all security. Hence it is necessary to develop a detection system for such attacks to help reduce the threat.

Technical research is carried out to identify the tools and techniques essential in the completion of this system. Research on methodology is done to finalise which development cycle will be used and how functions will be carried out at each phase of the development cycle. This paper outlines the problems that people face once they are attacked through macro malwares and the way it can be mitigated. Lastly, all information necessary to start the implementation has been gathered and analysed

Keywords: malware detection, Malicious, Macros, Attack, Office.

1. Introduction

As technology and internet gained rise in the last decade, it has offered and delivered several benefits to all kinds of industry or organizations. With the help of technology several processes have been automated in today's time which provides efficiency and enhanced productivity (Cunningham, P. *et al.,* 2018). On the other hand, the usage of internet boosted rapidly to the extent that everyone nowadays uses the internet.

With internet came several benefits such as quick research, cloud computing, Electronic mail and several more but all these benefits gave rise to more online attacks such as social engineering, man-in-the-middle attack, hacking and most importantly, malware attacks.

Malware is a software that is designed by attackers with a malicious intent of stealing private and confidential information, causing harm or damaging the device or any software in the victim's machine. Most of the time a malware is created by a group of hackers who are trying to find a way to make money either by spreading malware themselves and then asking for ransom such as the famous ransomware or by creating a malware and selling it to the highest bidder on the dark web.

Malware has been out there since the late 1980's but most malware programs at that time were simple boot sector and file infectors that were spread via floppy disk. As technology advanced and got standardized, most of the malwares proliferated yet macro malwares gained sudden boost. A macro malware is used to spread virus by taking advantage of the Visual Basic for Applications (VBA) programming in Microsoft Office macros. This virus has been used since the 1990s. Due to the emergence of complex social engineering attacks, macro malware poses a real threat to personal and organization's security.

Research has been done on related works of other researchers to better understand the threat and to get familiar with ideas of how to mitigate this attack. Several research papers have been studied to get an in-depth understanding. Based on the research conducted, the method for how to detect malware in these files was decided. Datasets will be created which would contain data and signatures of several files. Once the signature of the infected file will be extracted, it will be matched to the signatures saved to rate the file as infected or benign.

Recently, macro malware has gained extreme popularity as it is the only way to spread malware while surpassing all antivirus software and firewalls. Attackers are now using emails as the main mean of spreading macro malware. All antivirus software can detect executable files as attachments to emails and block them hence

making the attacker's attempt unsuccessful. With macro malware since it is embedded inside a word document, it is identified as a normal rich text file and that is why it can surpass all security measures.

To mitigate this attack, several systems and search engines have been developed such as VirusTotal to detect such files, yet this attack seems to excel. Hence, a macro malware detection system is to be developed in this project to automatically detect and analyse such macro embedded Microsoft Word files.

The main purpose of this paper is to develop and evaluate a system to detect macro malware which are embedded inside Microsoft Word document macros and can go undetected due to the nature of the file being non-executable. The main contributions of this paper are:

- To investigate and analyse the existing anti-virus software for macro malware detection.
- To design a framework to detect macro based malware.
- To develop a detection system based on the proposed framework for Microsoft Word documents.
- To increase the level of security and protection against macro attacks.
- To evaluate and test developed system to ensure its accuracy.

2. Literature Review

In this section, we will discuss the related works of different ways of spreading the malware, such as Macro Malware, Spam Email and Phishing Email.

Macro Malware:

Cyber-attacks aimed at organizations have increased drastically whereas in 2013, 91% of the organizations were hit by cyber-attacks. According to (Cohen *et al.*, 2016) and (Ucci, Aniello and Baldoni, 2019), email has become the main platform to execute such attacks as majority of the organizations rely on email to be the main means of internal and external communication. Non-executable files such as Microsoft Office documents attached to emails have been the basis of several recent cyber-attacks. Attackers can embed a malicious code inside these documents. The author further points out that there is several anti-virus software that can scan and filter any sort of executable files attached to emails, yet they fail here as to the non-executable nature of the documents. Hence, this study presents a novel structural feature extraction methodology (Cohen *et al.*, 2016) for Microsoft Word based documents. This methodology makes use of the hierarchical nature of office documents to convert them into unique paths as each path represents the file's properties and actions. Figure 1 below shows how these paths look like:

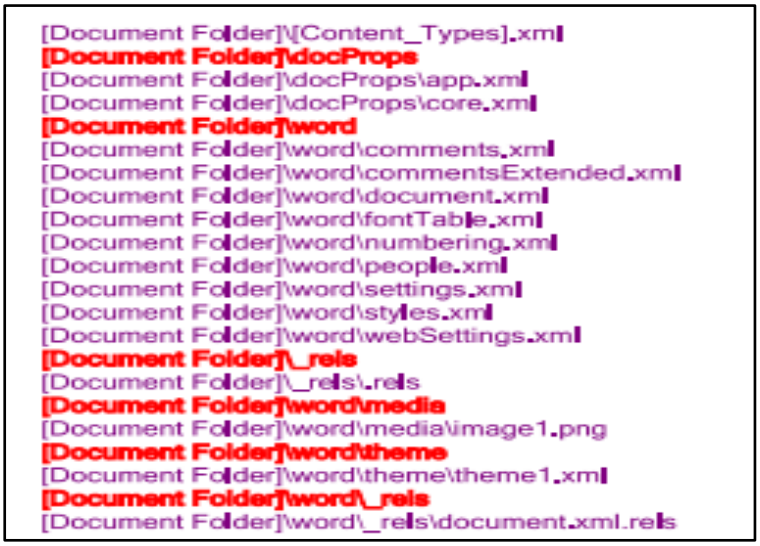

Figure 1. Demonstration of file paths

Moving on a java library has been used to scan all the available paths. During scanning, this methodology yields a set of unique features that have been extracted along with their occurrences in a specific document. Then the most unique feature will be selected, and a training set is built which is then used in training the machine learning classifiers. Different datasets are created, and the results are evaluated accordingly.

Attackers these days have changed their attack vector from operating system level to application level and that is why most attackers now are concentrating on finding vulnerabilities in common office applications like Microsoft Word. (Schreck, Berger and Göbel, 2013) from Siemens in 2013 presented a new approach called

Binary Instrumentation System for Secure Analysis of Malicious Documents (Schreck, Berger and Göbel, 2013). This approach consists of three purposes including distinguishing malicious documents from benign documents, extracting malicious embedded shellcode in the malicious documents and detecting and identifying the vulnerability exploited by the malicious document. This approach automatically determines the vulnerability that is targeted by the code. The vulnerability can either be for which a patch already exists as well as a new security flaw that has recently been identified. Hence, this approach helps to remedy the vulnerability by installing the correct patch if it exists. According to (Daryabar, F. et al., 2011) embedded malware is one of the effective ways to bypass an anti-virus gateway. Once the malware is added into documents or other files, the chances for it to be discovered decrease radically. Microsoft Word documents allow several executable files to be attached as objects and that is how they reach the victims system. As soon as the victim opens the file or clicks on this object, the malware is executed. Figure 2 shows the percentage of malwares that bypassed an anti-virus via Microsoft Word documents.

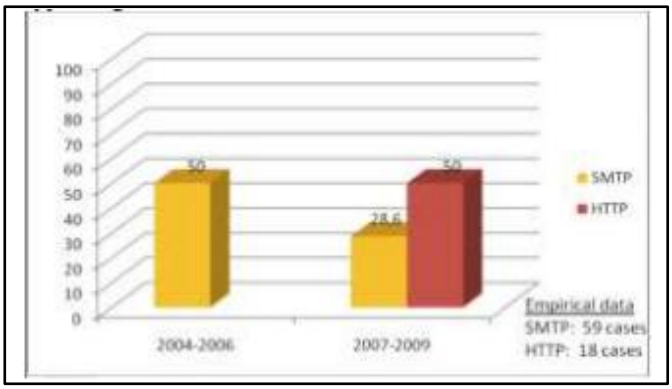

Figure 2. Percentage of Malware that bypassed an anti-virus via MS Word documents (Ding, L., et al., 2014)

The proposed solution to such attacks is to implement a honeypot system which requires three different levels of honeypot with each honeypot having its own role and responsibilities. The first layer is to be considered as an Intrusion Detection system (IDS) which will check the data received, whether executable or not and files received from external parties. This layer will run the received files to capture the activities and behaviour for further use. The second layer checks the data and the running processes using an anti-virus. If a known virus is detected at this layer, it will be stopped from entering the network and if no virus is detected, the data will be categorized as a new malware. The final layer will then delete the malicious code making the system or network secure. Figure 3 below shows a proposed design on this system.

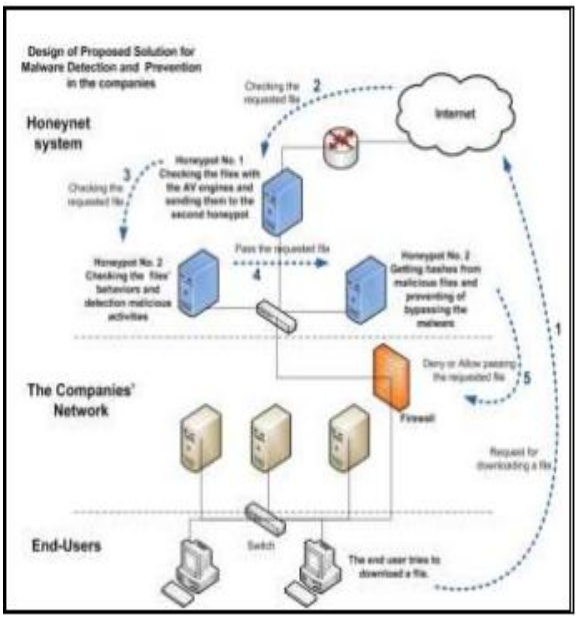

Figure 3 Proposed Design

Bradley (2018) also believes that macro-based threats are mostly spread via email campaigns where the malware is included as an attachment to the email. Attackers choose to use this method because the malware can be hidden behind layers of data making it difficult for anti-virus engines to detect it. One of the famous macro-based malwares is Adnel that downloads the runs files on the computer as soon as the infected file is open. According to Bradley (2018) the first step to prevent such attacks is to design an effective email security policy that would include training the employees so that they would not open or download malicious attachments that come from unknown sources. It is extremely important to spread awareness of some common cyber-attacks so that the employees are not manipulated easily. Furthermore, any word document by default has macro disabled yet there are malwares that can make the user do a different task that would automatically enable macro in that document hence, training would be a good way to prevent such careless mistakes. Besides training, sanitizing files through file type conversion or any other method is a good way to strip out any potentially dangerous macro while leaving behind the benign content on the file. This technique will help block known and unknown macros from entering.

Spam Email:

Spam is data intended to be sent to a huge group of individuals without them wanting to, whereby a spam channel is an automated instrument to recognize spam to guarantee greater profitability of the client (Cormack, 2008). As indicated by Jorgensen, Zhou and Inge (2008), assaults on spam channels have become a significantly greater test for the counter spam network. In light of their examination a 'good word assault' is the most successive spamming system utilized by spammers and the most ideal approach to counter it is through a guard procedure utilizing various case learning that has indicated extraordinary precision based on all the tests led by them. A decent word assault is the utilization of words, that are a piece of authentic messages (Jorgensen, Zhou and Inge, 2007).

As per Basavaraju and Prabhakar (2010), there is another method of utilizing content bunching dependent on vector space model which separates the spam/non spam messages and recognize spam messages all the more productively by examining designs. This procedure works by isolating the information into bunches with comparative examples. As indicated by (Cunningham et al., 2004), a case-based way to deal with spam sifting beats Naïve Bayes as it binds together substance based and cooperative methodologies. In their exploration synergistic methodology alludes to all the clients teaming up for example, in the event that one of them gets a spam, he will share the hash mark of that email with the rest which will make mindfulness for that email. In the light of the examination done, a case-based methodology appears the most ideal approach to channel spam messages in such a case that it has been identified by a client and made open to public.

No spam channel is sheltered, as Lowd and Meek (2014) and Robert and Kołcz (2020)accept that any great word assault can bring about half of the right now blocked spam to go past through any channel as no suspicious words are utilized to be recognized by the channel and subsequently the main cure is to constrain the harm they can cause. This method is powerful, however, it requires steady updates to coordinate with all the new words being utilized by the aggressors which is tedious and requires persistent exertion. Email spam is probably the most concerning issue nowadays causing enormous monetary loses to associations and irritating the people and they over flood the email accounts that outcomes in significant messages being ignored.

There are several studies and researches available that mark macro malware as one of the most stealth ways to deliver a malware to the victim. Most anti-virus software scan executable files to look for viruses whereas since macro malware is embedded inside Office Documents hence it can go undetected. Due to its nature it can also go around Gmail security making it extremely easy for attackers to send macro infected files to their victims through emails. According to a study, Microsoft macros is still the top vector for malware delivery. In the third quarter of 2018, 45% of the malwares were delivered using Microsoft macros (Seals, 2018). These macro scripts can carry any kind of malware ranging from a simple worm or a Trojan horse up to malwares like ransomware that caused so much of damage to several major companies in 2019. Although macro in any Office Documents are disabled by default but there are macro scripts written which can automatically enable the macro on the victim’s machine as soon as the file is opened. These malicious scripts embedded inside Office Documents can be executed either when the file is opened or if any hyperlink in form of a text or picture is clicked inside the document (Brook, 2018). Malware does not spread unless the file is opened hence simply downloading it on to your machine is not harmful at all.

Most of the people easily fall prey to such attacks as it is very hard to identify the attack. Macro infected files are mostly sent through phishing and spam email making them look like a legitimate email provoking the victim to download and open the file. Most of the people out there are not aware of macro malwares which is why no precaution is taken by them while downloading and opening these files from unknown sources. Most people think

that having an anti-virus is enough to prevent against hacking, but it is not. Attackers evolve as technology advances and they come up with new ways to hack, ways that were unknown to the world.

The Macro Based Malware Detection System is a desktop application that has been developed for detecting macro scripts in infected Office Documents. The main aim of the project was to only focus on Microsoft Word Documents, but later enhancements were made to the system and more formats were added such as Microsoft Excel and Microsoft PowerPoint files.

The developed system provides countless benefits to the users. Firstly, unlike other anti-virus that are unable to detect macro malware, the developed system has unparalleled accuracy to detect macro infected files. This would help the user to effectively detect malicious files to avoid falling prey to such attacks. Efficient detection of malicious files would ensure the safety of user data and hardware as well as data privacy and confidentiality. User's will not have to face the agony of their files being deleted, or the system getting super slow, or their files being encrypted along with a ransom demand etc.

Secondly, the developed system not only detects macro scripts in files but produces a detailed result regarding as to why a file has been marked as malicious. This will benefit security experts, lecturers and students of the related field to analyse the results to better understand the nature of these malwares and how they work.

Thirdly, the system has been developed to cater to all kinds of users specially novice users which is why the interface of the system was designed to be simple and user-friendly. This system will also help generate and create awareness amongst people regarding macro malwares which might result in people taking this threat seriously and making it hard for the attackers to hack using this method.

Phishing Email:

Phishing assaults involve authentic looking messages from genuine associations that threaten the client to enter in their private data. When the client enters this data, it is legitimately sent to the assailants who have started this assault. Fette et al., (2006) presented pilfer which is an artificial intelligence (AI) based methodology for grouping messages dependent on whether they are authentic messages or non-genuine messages. Recently, scientists had the option to get high exactness with just ten highlights incorporated into the framework (Wombat Security, a., 2018) . In addition, since assailants can change strategy, this channel can without much of a stretch adjust to the new highlights added to its classifier. What's more, phishing assaults undermine client trust in the legitimacy of messages and dependent on an exploration directed in April 2004, a colossal populace in America succumbed to these phishing assaults which cost the banks and organizations about 1.2 billion dollars in misfortune (Litan, 2004).

Chang et al. (2008) took a couple of highlights which to them were of most extreme significance when distinguishing phishing messages. Thorough testing was done on all highlights and progressions were made whereby two new highlights were presented DMZ (Dynamic Markov Chains) which decreases the memory necessities by two third and CLTOM (Class-Topic Models) include that joins class-explicit data into the point model. Classifiers with these highlights beat past methodologies. As per (San Martino and Perramon, 2011), a multifaceted confirmation framework will help against phishing assaults as it will check the computerized marks of the two gatherings followed by an asymmetric encryption applied to all data being shared between two substances which will guarantee that the data is protected and must be decoded utilizing a common key. This proposed framework will likewise give discovery against malwares and infections on the customer side to thoroughly make sure about the client's gadget as an assailant can utilize the client's gadget to hack into the association system or server. These highlights make this examination incredibly effective as having an awry encryption upgrades the security by an indent followed by an in-fabricated enemy of infection.

Rule-based phishing assault strategy involves fifteen guidelines which were created after an exceptionally broad research on website pages, perceptions led and dependent on AI highlights proposed in different looks into. These guidelines are then utilized in Decision Tree alongside Logistic Regression learning calculations to recognize whether a page is real or a phishing trick subsequently utilizing guess procedure to arrive at a resolution (Basnet et al., 2012). It tends to be gotten from this examination paper that a multifaceted verification is most reasonable strategy to counter a phishing assault, as referenced by San Martino and Perramon (2011). Lopsided encryption utilizes an open key and a private key to unscramble so regardless of whether data is lost or taken. It cannot be decoded without a private key which makes it of no worth yet making sure about the safety of the private key is another crucial component.

3. Materials and Methods

In this section, we will present a justification on the most suitable language for developing this system.

3.1 Programming Language Chosen

Python was created by Guido van Rossum in 1991. Python is an interpreted, general purpose, high level language and its design emphasizes code readability as it uses significant whitespace. Its object-oriented approach helps the programmers to write codes that are logical and clear regardless of the size of the project. Python supports multiple programming paradigms such as functional programming, object-oriented and functional programming. It is dynamically typed and has an extremely comprehensive library (Rongala, 2015).

Python interpreters are available for several operating systems. The greatest strength of python stems from the large library that provides countless tools suitable for several tasks. It includes areas like web services tools, string operations, internet protocols and operating system interfaces. Several programming tasks have been scripted inside this library hence there is no need to write the whole code as these scripts can simply be called as functions or imported as libraries. The official repository for third-party software contains more than 200,000 packages such as graphical user interfaces, web frameworks, multimedia, databases etc. Python is easy to learn and has vast available resources to provide support. It is extremely user-friendly. Lastly, python has several libraries that can be used to analyse data and for DE obfuscating macros (Chou, 2019).

Table 3.1.1 Comparison between JavaScript and Python

	JavaScript	**Python**
Type	Object oriented	Object oriented high-level programming language
Static Typing	Dynamic	Dynamic
Platform	Cross-platform compatible	Cross-platform compatible
Community Support	Easy to find tutorials	Has large support
Development Speed	Moderate	Fast
Capabilities in System Development	For stand-alone systems	For stand-alone systems
Learning Curve	More complex and hardcore skills needed.	No hardcore skills are needed

When comparing with other languages python has a better and a faster execution speed compared to Java or R which would improve the overall productivity of the system. Python and Java both are used for building stand-alone systems, yet python does not require hardcore coding skills making it a preferred choice when compared to other languages.

Therefore, based on all the reasons mentioned above, python has been chosen as the most suitable language for developing this system.

3.2 IDE (Interactive Development Environment)

PyCharm is a framework that has been designed by programmers to allow other programmers with all the tools necessary for a productive Python development. PyCharm allows programmers to write neat codes as it helps control the quality with PEP8 checks, smart refactoring and testing assistance (Aditya Sharma, 2019). PEP8 checks highlight any coding violations and give suggestions instantly as you type. It provides intelligent python assistance with code inspections, error highlighting and quick fixes to make development easier. Furthermore, it

offers great framework support for modern web development. Apart from integrated with scientific tools, PyCharm also supports other programming languages such as JavaScript, Cython, SQL etc. to allow cross-technology development. Lastly, PyCharm provides numerous features built-in developer tools such as an integrated debugger, test runner, python profiler, built-in terminal (Sharma,A., 2019) .

Compared to other IDE's PyCharm is the most suitable as it compromises of several functions such as run configuration, remote debugging etc, which cannot be found in other frameworks. PyCharm supports the greatest number of frameworks when compared to Visual Studio and Eclipse. Therefore, PyCharm has been chosen as the most suitable IDE to develop this system.

3.3 Libraries/Tools chosen

Olevba:

Olevba is a script to analyse OLE and XML files such as Microsoft Office documents to detect VBA Macros, extract them, detect patterns such as executable files or keywords and potential IOCs. Moreover, it detects and decodes obfuscation methods as well including Base64, Dridex, Hex encoding etc. It is a part of python oletools package.

It supports several formats such as Word 97-2003, Word 2007+ (.docm), Excel 97-2003 (.xls), Excel 2007+ (.xlsm), PowerPoint 97-2003 (.ppt) and PowerPoint 2007+ (.pptm). Different versions of each program cater to different file extensions. It main features include detecting VBA Macros, extracting macros, detect auto=executable macros, detect suspicious keywords, extract IOCs and scan all of them to check whether they are malicious or not.

Olevba works by checking the file type to ensure that the file type is supported by this package. Next, it identifies all VBA projects and each project is analysed to find its corresponding OLE stream which contains the macro code. The code is then extracted and compressed and the it checked for specific strings, algorithms, executable files, specific keywords etc.(Pippi, G., 2020).

Class:

- **VBA_parser:** This class is used to parse the files once the name of the file is provided to open as a parameter.
- **VBA_scanner:** This class is used to scan the source code to find strings, keywords etc. The scan results consist of a list of tuples containing type, keywords, descriptions etc.

Methods:

- **Detect_vba_macros**: This is a method that returns true if macros have been found in the chosen file.
- **Extract_macros:** This is a method that extracts the source code for each macro found in the file. It functions as a generator yielding a tuple for each macro found.
- **Analyze_macros:** This is a method to scan the source code to find obfuscated strings, suspicious keywords, IOCs, auto-executable files etc.

Functions:

- **Detect_autoexec:** This function is used to check whether the macro code contains such code that can be triggered if the document is open or if something is clicked inside the document. It returns a list containing the detected keyword and the description of the triggered identified.
- **Detect_suspicious:** This function is to check whether the macro code contains specific keywords that are often used by attackers to design malwares or viruses. It returns the detected keyword and its description.
- **Detect_patterns:** This function checks the macro code for containing specific patterns that maybe useful when analysing the malware and its detection. It returns the pattern type and the extracted value.

3.4 Operating System Chosen

An Operating System provides the users with a user interface to use the hardware and software. In this paper Windows 10 by Microsoft is selected to be the chosen operating system.

Windows follow a directory structure to store files. Windows has very less restrictions and allows third party applications to be installed as well compared to Mac operating system (macOS). Windows can support a vast number of tools in the operating system(Grabham, 2019). Moreover, Windows is the most used operating system as all offices and organizations mostly use Windows as it has an easy user interface to interact with. Any devices can be easily connected to Windows as compared to macOS that requires Apple products to run. On the other

hand, Linux has very less features and it is complicated to use. Lastly, NetBeans is fully supported by Windows Operating System.

4. Results and Discussion

The macro malware detection system will allow users to stay protected against such malware attacks. It will detect files with macro malware to prevent the malware from being deployed successfully. These malicious files will be marked as malicious only after being compared with a dataset. Signatures of several files and several macro scripts will be used to create these datasets and then characteristics of the test file will be compared with the signatures inside the datasets. Once detected the system will inform the users about the presence of a malware which would help users and their files to be safe and not be infected by this malware.

4.1 User Acceptance Testing Result Analysis

Interface Design

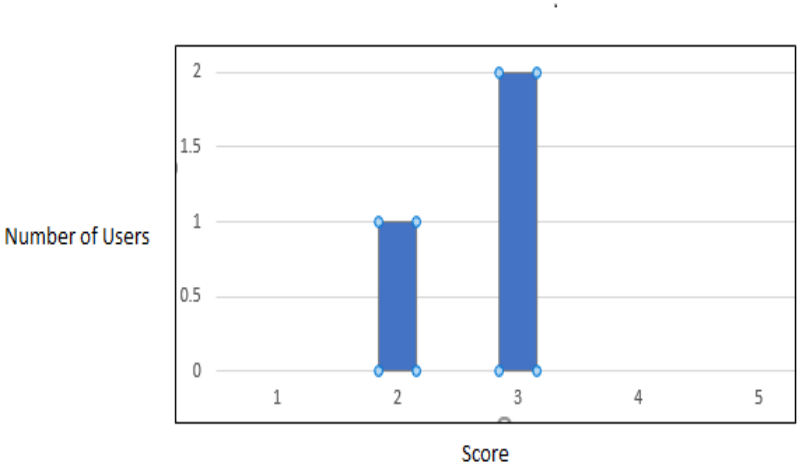

Figure 4 Interface Design UAT results

Figure 4 illustrates the results of the UAT regarding the interface design of the developed system. There are two users that gave a score of 3 which stands for average whereby 1 user gave a score of 2. These results show that the interface can be improved and should be improved to make it more attractive.

Figure 5 illustrates the results of UAT regarding objectives being met by developing the project. 2 users gave a score of 5 that means that they are extremely happy with the functionality of the system whereas 1 user gave a score of 4. Thus, the users are extremely satisfied with the functionality of the system as it meets the project objectives.

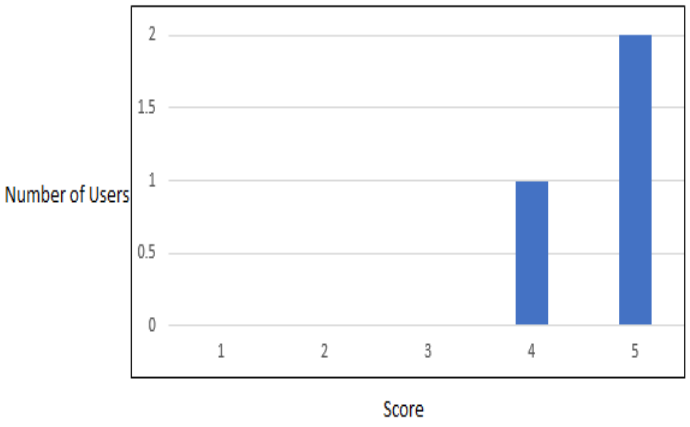

Figure 5 Meeting Objective UAT results

Figure 6 illustrates the UAT testing results for system validation. We found that there are two users gave a score of 5 and 1 user gave a score of 4. Thus, users are fully satisfied with the system as it exactly functions how it is supposed to and as expected by the users. It is extremely accurate and highly precise when detecting macro scripts which is a huge positive point for the developer and the system.

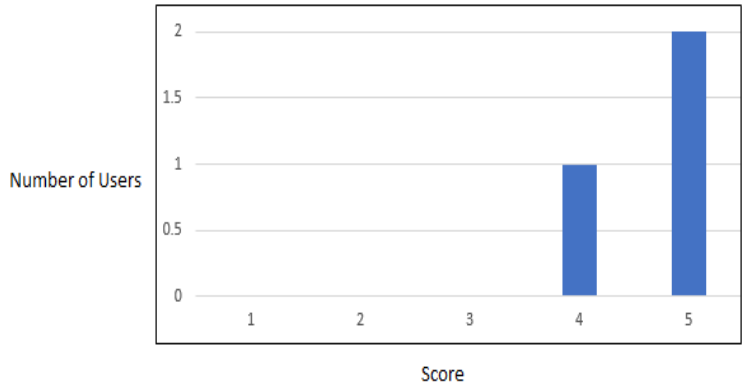

Figure 6 System Validation UAT results

Figure 7 illustrates the results of UAT maintainability testing. All 3 users gave a score of 4, hence the users are quite satisfied with the maintainability of the system. As a system equipped with a simple GUI, consequentlyto maintain it and overcoming failures will relatively be easier.

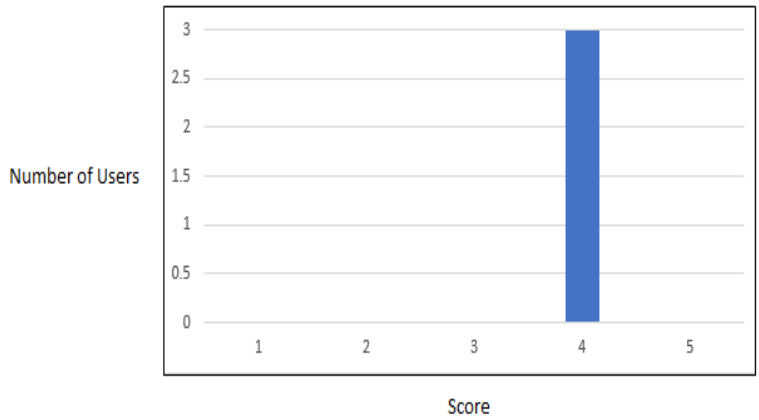

Figure 7 Maintainability UAT results

Figure 8 illustrates the results of UAT Free from bugs test. There are two users gave a score of 5 and 1 user gave a score of 5. These results are considered extraordinary meaning that the users are satisfied with the system as there are so visible or existing bugs in the system and it is functioning perfectly according to the proposed method.

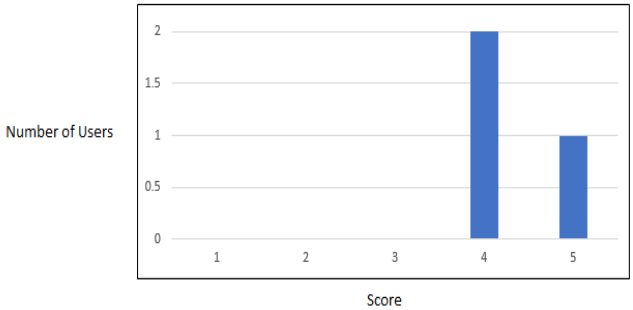

Figure 8 Free from Bugs UAT results

The feedback received from the users was extremely satisfactory. There were several recommendations given by the users such as:

- The result should also show the file path and file type of each tested file to avoid confusions.
- The interface should inform the user regarding which file is being tested if the user has uploaded multiple files to be checked at the same time.
- When multiple files are being tested at the same time, results of all files are saved in one file too, but the result of each file should be properly segregated to avoid confusion as to which result is for which file.

All recommendations were taken into consideration and changes were made to the system to further make the system user-friendly.

5. Conclusion

In conclusion, several types of tests were conducted on the developed system to validate the system and to locate errors or bugs in the system. Testing was done for all functions of the system to ensure that everything is functioning properly. Secondly, accuracy testing was conducted to test the authenticity of the system for which a file pool of 50 files per format was created and tested with VirusTotal and the developed system. Lastly, user acceptance testing was done in which 3 users were given the system to use and their feedback was recorded. Based on the analysis of the feedback, all 3 users overall were satisfied with the system in terms of interface, meeting the objectives, maintainability, validation and no bugs. The users also suggested a few recommendations that can be applied to the system which would further improve it.

Moreover, macro malware is a major issue of this era where technology is growing rapidly yet the developed system will hopefully contribute a lot in staying secure from such possible attacks. The developed system is extremely accurate in detecting macro for a few file formats making it hard for the attackers to execute these macro malware scripts. Hence, the developed system is successful in solving the problem of malwares going undetected by being embedded inside Office Documents.

Whereas, the developed system is very accurate in detecting macros in files, but like any other system, it has certain limitations. The developed system can only detect macro in three file formats including Word, Excel and PowerPoint files. This makes the functional of the system limited as macro malwares can also be delivered using several other file formats such as pdf or jpeg files.

The most important future enhancement is to modify the system so that it can check and detect macros in several formats especially the ones that have not been added yet. In future research directions, AI can be used to make a customer inference engine that will help detect macro scripts based on keywords, shell commands etc. The current system is using a pre-developed library, but using AI, a more extensive system can be developed. While, real-time scanning can be added to the system like every other anti-virus software to scan a few files automatically being saved on the system so that the user does not have to upload the check the file manually.

References

Aditya Sharma (2019) Top Python IDEs for 2019 (article) - DataCamp. Available at: https://www.datacamp.com/community/tutorials/top-python-ides-for-2019 (Accessed: 13 February 2020).

Baker, S. E. and Edwards, R. (no date) How many qualitative interviews is enough? Expert voices and early career reflections on sampling and cases in qualitative research. Available at: http://eprints.ncrm.ac.uk/2273/4/how_many_interviews.pdf (Accessed: 8 January 2019).

Basavaraju, M. and Prabhakar, R. (2010) 'A Novel Method of Spam Mail Detection using Text Based Clustering Approach', International Journal, 5(4), pp. 15–25. doi: 10.5120/906-1283.

Basnet, R. et al. (2012) Rule-Based Phishing Attack Detection Kattis View project Phishing View project Rule-Based Phishing Attack Detection. Available at: https://www.researchgate.net/publication/265919217 (Accessed: 15 December 2018).

Bradley, T. (2018) Introduction to Intrusion Detection Systems (IDS). Available at: https://www.lifewire.com/introduction-to-intrusion-detection-systems-ids-2486799 (Accessed: 23 January 2019).

Branson, T. (2019) 8 Major Advantages of Using MySQL. Available at: https://www.datamation.com/storage/8-major-advantages-of-using-mysql.html (Accessed: 13 October 2019).

Chang, J. H. et al. (2008) Improved Phishing Detection using Model-Based Features. Improved Phishing Detection using Model-Based Features. Available at: http://www.webcallerid.net (Accessed: 15 December 2018).

Cohen, A. et al. (2016) 'SFEM: Structural feature extraction methodology for the detection of malicious office documents using office learning methods', Expert Systems with Applications. Elsevier Ltd, 63, pp. 324–343. doi: 10.1016/j.eswa.2016.07.010.

Cormack, G. V. (2008) Email Spam Filtering: A Systematic Review, Foundations and Trends® in Information Retrieval. doi: 10.1561/1500000006.

Cunningham, P. et al. (2018) A Case-Based Approach to Spam Filtering that Can Track Concept Drift. Available at: www.detritus.org/spam/skit.html (Accessed: 15 December 2018).

Ding, L., Wei, R. and Che, H., 2014. Development of a BIM-based Automated Construction System. Procedia Engineering, 85, pp.123-131.

Daryabar, F., Dehghantanha, A. and Ghasem Broujerdi, H. (2011) Investigation of Malware Defence and Detection Techniques. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.967.6837&rep=rep1&type=pdf (Accessed: 6 August 2019).

Drumond, C. (2019) Scrum - what it is, how it works, and why it's awesome. Available at: https://www.atlassian.com/agile/scrum (Accessed: 13 October 2019).

Fette Norman Sadeh Anthony Tomasic, I. et al. (2006) Learning to Detect Phishing Emails. Available at: https://apps.dtic.mil/dtic/tr/fulltext/u2/a456046.pdf (Accessed: 11 December 2018).

Global spam volume as percentage of total e-mail traffic from January 2014 to March 2018, m. (2018). Spam statistics: spam e-mail traffic share 2018 | Statista. [online] Statista. Available at: https://www.statista.com/statistics/420391/spam-email-traffic-share/ [Accessed 3 Dec. 2018].

Grabham, D. (2019) Windows 10: the best Windows OS review | TechRadar. Available at: https://www.techradar.com/reviews/pc-mac/software/operating-systems/windows-10-1267364/review (Accessed: 13 October 2019).

Hong, J. and Hong, J. (2012) 'The Current State of Phishing Attacks The Current State of Phishing Attacks', Communications of the ACM, 55(1), pp. 74–81. doi: 10.1145/2063176.2063197.

Jorgensen, Z., Zhou, Y. and Inge, M. (2008) A Multiple Instance Learning Strategy for Combating Good Word Attacks on Spam Filters, Journal of Machine Learning Research. Available at: http://www.jmlr.org/papers/volume9/jorgensen08a/jorgensen08a.pdf (Accessed: 11 December 2018).

Litan, A. (2004) Phishing Attack Victims Likely Targets for Identity Theft. Available at: https://www.social-engineer.org/wiki/archives/IdTheif/IdTheif-phishing_attack.pdf (Accessed: 11 December 2018).

Lowd, D. and Meek, C. (no date) Good Word Attacks on Statistical Spam Filters. Available at: http://www.utdallas.edu/~muratk/courses/dmsec_files/125.pdf (Accessed: 27 November 2018).

Medium. (2018). Classify emails into ham and spam using Naive Bayes Classifier. [online] Available at: https://medium.com/swlh/classify-emails-into-ham-and-spam-using-naive-bayes-classifier.

Pippi, G., 2020. Advanced VBA Macros: Bypassing Olevba Static Analyses With 0 Hits. [online] Certego. Available at: <https://www.certego.net/en/news/advanced-vba-macros/> [Accessed 14 September 2020].

Robert, R. and Kołcz, A., 2020. Improving Spam Filtering By Detecting Gray Mail. [online] Scottyih.org. Available at: <http://scottyih.org/files/Improve_Spam_Filtering_by_Detecting_Gray_Mail.pdf> [Accessed 14 September 2020].

Ramzan, Z. (2010) 'Phishing Attacks and Countermeasures', in Handbook of Information and Communication Security. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 433–448. doi: 10.1007/978-3-642-04117-4_23.

Rongala, A. (2015) Benefits of Java over Other Programming Languages - Invensis Technologies. Available at: https://www.invensis.net/blog/it/benefits-of-java-over-other-programming-languages/ (Accessed: 13 October 2019).

Sahami, M. et al. (no date) A Bayesian Approach to Filtering Junk E-Mail. Available at: http://robotics.stanford.edu/users/sahami/papers-dir/spam.pdf (Accessed: 11 December 2018).

San Martino, A. and Perramon, X. (2011) Phishing Secrets: History, Effects, and Countermeasures, International Journal of Network Security. Available at: https://pdfs.semanticscholar.org/ce2c/79720dc4eaca6d1263323b04d09262fe392c.pdf (Accessed)

Schreck, T., Berger, S. and Göbel, J. (2013) 'BISSAM: Automatic Vulnerability Identification of Office Documents', in. Springer, Berlin, Heidelberg, pp. 204–213. doi: 10.1007/978-3-642-37300-8_12.

Spirin, N. and Han, J. (2012) 'Survey on web spam detection', ACM SIGKDD Explorations Newsletter. ACM, 13(2), p. 50. doi: 10.1145/2207243.2207252.

Susan Rose, N. S. and A. I. (2015) An introduction to using Microsoft Excel for quantitative data analysis. Available at: http://office.microsoft.com/en-001/support/?CTT=97 (Accessed: 8 January 2019).

Ucci, D., Aniello, L. and Baldoni, R., 2019. Survey of machine learning techniques for malware analysis. Computers & Security, 81, pp.123-147.

Wielenga, G. (2014) Top 5 Benefits of 'NetBeans Platform for Beginners' | Oracle Geertjan's Blog. Available at: https://blogs.oracle.com/geertjan/top-5-benefits-of-netbeans-platform-for-beginners (Accessed: 13 October 2019).

Wombat Security, a. (2018). 2018 State of the Phish. [online] Wombatsecurity.com. Available at: https://www.wombatsecurity.com/state-of-the-phish [Accessed 3 Dec. 2018].

Youn, S. and McLeod, D. (2007) 'A Comparative Study for Email Classification', in Advances and Innovations in Systems, Computing Sciences and Software Engineering. Dordrecht: Springer Netherlands, pp. 387–391. doi: 10.1007/978-1-4020-6264-3_67

Zareapoor, M. and K. R, S., 2015. Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection. International Journal of Information Engineering and Electronic Business, 7(2), pp.60-65