Swinburne University of Technology

Faculty of Science, Engineering and Technology

ASSIGNMENT AND PROJECT COVER SHEET

Unit Code : COS30015   Unit Title : IT Security

Assignment number and title : Research Assignment Due date : **11 Sep 2021 23:59**

Lab group : Class 9   Tutor : Rory Coulter   Lecturer : (Sky) Yuantian Miao

Family name : Hudson   Identity no : 102533320

Other names : Cobie

To be completed if this is an INDIVIDUAL ASSIGNMENT
I declare that this assignment is my individual work. I have not worked collaboratively, nor have I copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for me by another person.

Signature : C.Hudson

To be completed if this is a GROUP ASSIGNMENT
We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

ID Number	Name	Signature

Marker's comments:

Total Mark:_________________

Extension certification:

This assignment has been given an extension and is now due on ________________________

Signature of Convener:___________________________________ Date: __________/ 2021

Abstract

Solely designed to carry out malevolent actions, malware will continue to become a major threat to users of the internet. As the internet has become more prevalent in our everyday lives, this invasive and exploitive software has derived to a strikingly intricate problem for researchers within the industry, encompassing viruses, trojans, spyware, ransomware, and other invasive code. Capable of stealing sensitive information or adhering performance, most malware programs are substantial and complex in scope, making it hard for any individual to fully grasp the impacts. However, through the examination of available literatures on malware analysis, we can ultimately educate internet users of the critical steps to overcome the many challenges that malware presents and understand malware attacks to a much greater extent. This is an essential concept of this study as we compare and analyse each malware preventions technique.

Keywords : Malware, Internet, Viruses, Trojans, Spyware, Ransomware, Attacks, Analysis

Introduction

Formed from primitive measures, malware has evolved into a far more malignant code with the purpose of intruding and harming a computer or network. Initially spread via floppy disks in 1971, malwares presence has been consistent throughout the rise of technology within our society (Rankin, 2018). Specifically in 1988 when the Morris Worm undergone a process that would infect over 60,000 computers and malfunctioning the majority, ultimately leading to the first ever convicted malware author Robert Morris and provoking an uprising in malware attacks (Vaughan-Nicholas, 2018). Since this repercussion, there has been an obligation for researchers to altercate malware authors and persistently devise new methods of defending against attacks (Danchev, 2006). Surfing the internet absent of any form of firewall or antivirus protection for a single day is ample evidence to extend any individuals understanding of the risks involved with malware attacks and its consequences (Gutierrez, 2020). Consequently, San Diego Supercomputer Centre (SDC) conducted an experiment to evaluate the contingency of malware attacks. Initially having no breaches and security setups, the Linux based computer was connected to the internet. In a span of 8 hours the computer had been hacked and ultimately regarded as “compromised” after a 40-day period of undergoing countless of attacks (Idika, & Mathur, 2007). Malware is an extremely malevolent software capable of hindering network and computer speeds, stealing sensitive information, operation disruptions and dire altercations involving the loss of human life. Reported by the Spanish newspaper by El Pais, they surmise that a Spanair central computer was trojan-infected causing 154 people to tragically die due to an alarm failing to trigger (Storm, 2010). Thus, with ample evidence about the versatile nature of malware, the disclosure of malware detection is an essential matter of concern to both researchers and the general populace. This literature review covers this crucial subject, aiming to educate individuals of the various malware detection techniques and disclose distinct comparisons among them. Hence, the next section will interpret the content of the literature review, after which the succeeding segment will comprise of the various malware detection techniques along with their respective comparisons.

Overview / Background

Malware Analysis Techniques

Malware analysis involves the interpretation of a suspicious file or URL, this ensures an understanding of the behaviour or purpose is determined. The output of this analysis is a crucial step to discern the urgency of the attack and damage it may provoke (Baker, 2020). This indicates a definitive approach of removal, ensuring appropriate defensive approaches are employed to protect vulnerable systems (N-able, 2019). There are three fundamental approaches to malware analysis composed of distinct skills and time required, each capable of disclosing the malware involved and their impacts on the system.

Static Analysis

Commonly known as code analysis, static analysis involves an examination of source code without executing the application (Richardson, 2021). This technique confirms whether file is malicious, while detailing its functionality and other important information allowing network signatures to be produced. Useful for understanding the structure of malware within an executable, static information composed of the header data is examined, while further assessing the extent of maliciousness through a sequence of bytes. Consequently, the operation code is retrieved as a feature to evaluate the relevant behaviour to disclose the malware.

Dynamic Analysis

Also known as behavioural analysis, this involves the examination of a suspicious file while actively executing (Medeiros, 2016). This analysis exists in an isolated environment such as a virtual machine, emulator, simulator, or sandbox. This allows experts to monitor any active vulnerabilities and behavioural patterns without running the risk of exposing their system or network to infection. However, this technique can be time-consuming due to the development efforts of the isolated environment. Although this is a stagnant process, dynamic analysis provides experts with a deeper knowledge of the respective file, ensuring the true purpose of the threat is documented.

Hybrid Analysis

Superior in all aspect, hybrid analysis involves combining both techniques discussed above. Through combining both static and dynamic approaches, experts can devise better and more sophisticated means of detection. This allows experts to surmount the challenge that static and dynamic analysis presents, ensuring no setbacks or limitations are associated with the process (Roundy, & Miller, 2010).

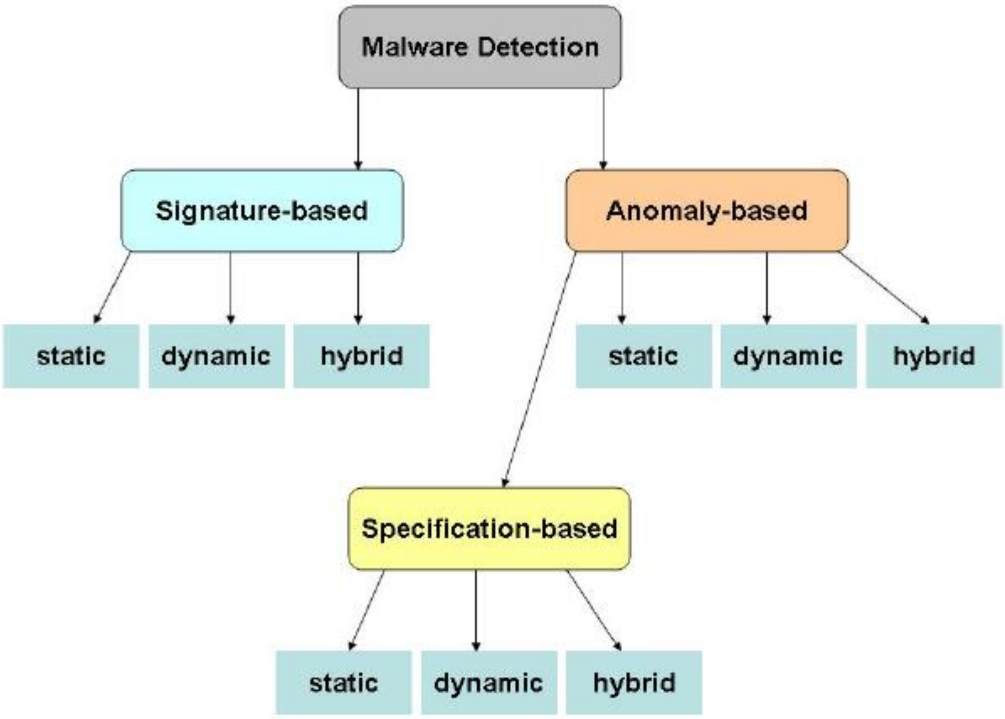
Data-mining Analysis

Considered just as important as other analysis techniques. Data mining is a crucial technique utilised with other analysis types and detection methods, allowing new and innovative approaches to malware detection to be possible. This is achieved by extracting and uncovering patterns contained within a suspicious executable.

Malware Detection

The existing amount and diversity of malware is unmatched when compared to malware detection. According to McGraw and Morrisett (2000), this unfortunate reality is due to the ceaseless combination of distinct types of malware characteristics, conceiving more advanced and competent varieties. Consequently, this has provoked researchers to devise new and innovative approaches as malware detection is progressively compromised every day. These techniques can be categorised into two expansive sections including signature-based and behaviour-based (anomaly-based), both capable of static, dynamic and hybrid approaches. Although this technique will not be covered, specification-based detection consists of similar aspects involved in behaviour-based detection. (*Refer to Figure 1*). Additionally, tools such as malware detectors ensures the detection process occurs, relying heavily on the type of malware detection method being used.

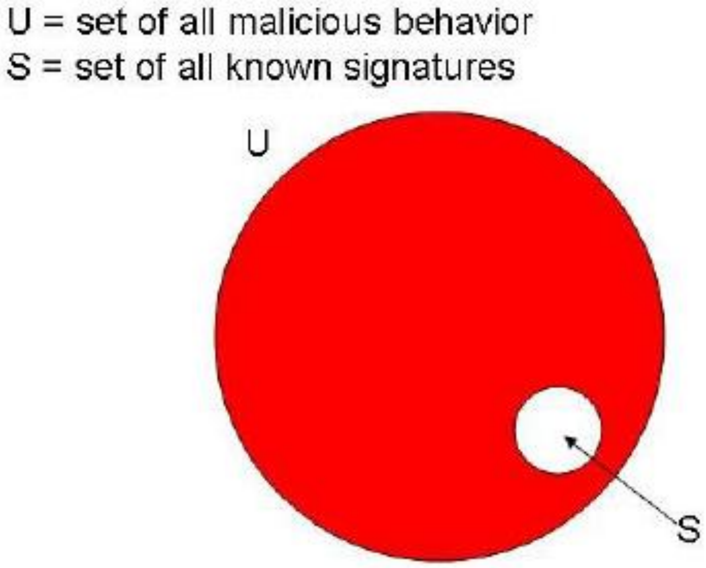
Figure 1 : Depicts a visualisation of the concept discussed above



Signature-based Technique

Signature-based detection involves the establishment of distinct characteristics about the maliciousness of malware. Composed through an examination of the dismantled code of malware binary, a signature is defined by malicious behaviour, allowing experts to detect malware through a comparison analysis against the corresponding database (Uddin, & Rahman, 2010). This ensures suspicious or abnormal files are uncovered of any form of malware as outlined by the signature and its defined attributes. This signature may consist of a common set of malicious code or unique identifiers. As a result, if that distinct pattern or signature is disclosed, the suspected file will be flagged as infected (Bricata, 2021). Similar to other substantial quantities of data, signatures also require a means of storage, known as a repository. Representing every aspect of the signature-based technique, a repository is crucial for comparative purposes during malware detection. This is achieved once an expert conceives and adds the necessary signatures to the corresponding repository. Consequently, when a suspicious file is being assessed of existent malware, the repository retrieves and interprets corresponding signatures that may represent it. Although immensely useful, signature-based techniques heavily depend on human expertise and their creation of signatures that represent malicious behaviour. This also introduces other inevitable drawbacks such as human error and a lack of signature automation. Considering the versatility of malware, the potential to promptly develop an authentic signature becomes paramount. Another major drawback of signature-based detection refers to the inability to detect zero-day attacks, an attack that is absent of any corresponding signatures in the repository (*Refer to Figure 2*). Although having extensive drawbacks, based off patterns in its library, signature-based detection can expose malware more accurately compared to other detection methods whilst having much lower resource usage.

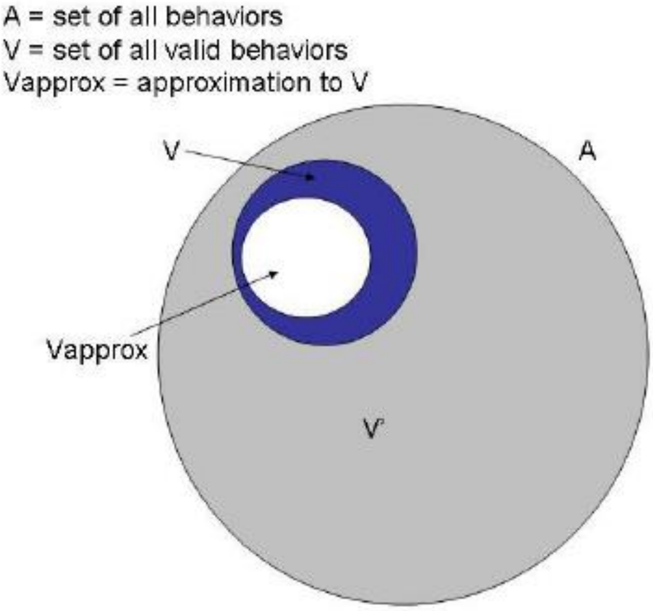
Figure 2 : Depicts a visualisation of the limitations involved with signature-based detection


U = set of all malicious behavior
S = set of all known signatures
U
S

Behaviour-based Technique

Also known as anomaly-based, behaviour-based detection is derived from its fundamental process of observation (Mujumdar, Masiwal, & Meshram, 2013). This technique ensures suspected files are actively examined, alerting users of any unknown or suspicious activity, ultimately determining the extent of maliciousness. Unlike signature-based procedures, behaviour-based detection employs machine learning to form a normalized baseline, useful for improving the detection system. Through comparative analysis of the subject, this baseline is an effective tool to identify any unusual behaviour that may occur. These behaviour parameters comprise of a multitude of components ranging from the source or destination of malware, attachment types and among other distinct measurements (Jerlin, & Jayakumar, 2015). Consequently, this technique functions through two crucial phases – a training (learning) phase and a detection (monitoring) phase. Attempting to monitor the behaviour of the system, the training phase ensures a profile of normal behavioural activity is constructed. This profile is then obtained and compared against the current behaviour of the subject. Known as the detection phase, this procedure documents any shift in variation, flagging them as possible attacks. This detection approach is highly flexible, allowing experts to apply a single behaviour signature and ultimately identify a range of the malware alterations. Unlike signature-based detection, behaviour-based has the ability to detect zero-day attacks. Although retaining some advantage over other techniques, this detection method also comes with some limitation including higher resource throughput and high false alarm rate (*Refer to Figure 3*)

Figure 3 : Depicts a visualisation of the limitations involved with behaviour-based detection



Literature Review – Signature-based Technique

A study by Eu et al. (2014), determined to solve limitations of signature-based detection, the study proposed artificial immune based smartphone malware detection models (SP-MDM) could be applied to confront the techniques inability to detect day zero attacks. Through combining static and dynamic analysis in conjunction with SP-MDM, this could allow the generation of antigens. Defined by the encoding extracted characteristics from malware, mature detectors could be formed. Additionally, by integrating clonal selection algorithms, experts could introduce superior offspring through the selection of detectors with higher affinity. This selection of detectors would then undergo a proliferation and somatic hyper-mutation process. Consequently, the researchers uncovered that the method had enhanced detection rates for foreign smartphone malware, concluding that an increase in detection performance could be possible through more iterations of clone generation.

A study by Bat-Erdene et al. (2017), devised an innovative approach to overcome the rapid growth of packed malware. Through the classification of packing algorithms, this method would ensure an increased probability of detecting malware within unknown executables. Known as symbolic aggregate approximation (SAX), this involves scaling entropy values of an executable and converting a distinct region of its memory into symbolic depictions. Consequently, these symbols would then be distributed through learning classification methods supervised by both the Naïve Bayes algorithm and support vector machines. As a result, the experiment consisting of 324 packed benign programs and 326 packed malware programs and 19 packing algorithms, the method was able to pinpoint the packing algorithms with an accuracy of 95.35%, recall of 95.83% and precision of 94.12%. Additionally, through the use of SAX similarity measurements for detection, the fidelity similarity measurement able to best other techniques with a rate of accuracy of 95% - 99.9%, a value 2 to 13 higher in comparison. Identified through entropy analysis, this study confirmed that the detection of packing algorithms is viable even without prior knowledge the executable.

A study by Cui et al. (2015), formulated a novel detection scheme based on cloud computing and packet analysis. Through data mining packets within mobile malwares, the system ensures the detection of malicious behaviour. Additionally, by implementing a new clustering strategy called contraction and multimodule detection schemes, this ensured an overall improvement in system performance, accuracy, and overall reduction of data set sizes. Effectively avoiding all limitations of traditional approaches, this innovative method allows service providers to send alerts to users who have malwares on their device.

A study by Fan et al. (2016), intending to surmount the common limitations of signature-based detection and its lack of detecting day zero-attacks. Through the introduction of automatic malware detection, this study clarifies the importance of sequence mining. This form of data mining allows frameworks to detect malicious behaviour based off extracted sequential patterns. Additionally an All-Nearest-Neighbour (ANN) classifier is constructed based off these discovered pattens. Through the integration of these techniques, experts can effectively expose foreign malware existing within the corresponding executable. As a result, it was revealed that the framework of this proposal exceeded other alternative data mining-based detections and their ability to disclose day zero-attacks.

A study by Santos et al. (2013), advocated a new data mining approach to detect unknown malware attacks, also known as “zero-day attacks”. Through extracting characteristics existing in opcode sequences, experts could assess their recurrence within each grouping. Consequently, this approach proved to be an appropriate and effective tool for recognising obscure malware.

Evidently, the contents in the review above outline the central limitation of signature-based detection and its inability to detect “zero-day attacks”. Although a perpetual problem, as demonstrated by the countless of conducted studies above, researchers have devised a myriad of methods and techniques to overcome this. Through data mining, combining of detection schemes and algorithms, the limitations of signature-based detection has become minimal. Additionally, disadvantages such as the reliance of human expertise and the clarification of a signature has also been resolved by the efforts of researchers. Specifically the study by Fan et al. (2016) and Eu et al. (2014), by introducing automatic malware detection and higher affinity detectors, this has ensured the absence of any setbacks. Although having many limitations, signature-based detection has many advantages over other alternative detection methods such as reduced overhead resources, faster execution times and more accurate malware detection. Overall, signature-based detection is a desirable and effective technique that can be used when uncovering malware within a foreign executable due to the dedication and efforts of the literature above.

Literature Review – Behaviour-based Technique

A study by Mohaisen et al. (2015), devised a behaviour-based malware analysis and labelling system. Also referred as AMAL, this high-fidelity technique addresses the many shortcomings of existing systems. Consequently, this system is comprised of two sub systems, AutoMal and MaLabel. Through the collection of low granularity behavioural artifacts within virtual environments, AutoMal provides the tools to characterise malware usage of the executable such as its memory, network, and registry. Thus, MaLabel then exploits these artifacts to generate classifiers that group malware samples into families with comparable behaviours. Through the detection of malware at high precision and accuracy, AMAL proved to be a suitable and effective malware detection tool.

A study by Yuan et al. (2015), expressed the importance of integrating deep learning techniques with malware detection. An experiment that analysed thousands of apps within the Android store, the researchers sought to demonstrate the impact that deep learning can have on behaviour-based detection. Hence, by replacing traditional methods of machine learning with an online deep-learning-based Android malware detection “DroidDetector”, this ensured the result of 96.76% detection accuracy, outperforming alternatives.

A study by Ding et al. (2013), formulated an innovative data mining approach involving API call sequences in order to characterise malware. Unlike previous approaches that focus on detection rather than explanation, this method ensures malware is detected but also construed. Through fusing both dynamic and static API sequences into a hybrid sequence based on semantics mapping, ensures the construction of a hybrid feature vector space. Consequently, by carrying out this process, an explainable malware detection framework is established, known as a MalDAE. As a results, the detection and classification accuracy of MalIDE achieved up to 97.89% and 94.39% respectively, successfully outperforming other comparable studies.

A study by Eskandari et al. (2013), advocated the introduction of a novel hybrid approach known as a “HDM-Analyzer”. Through associating the advantages of both static and dynamic analysis, a majority of decision-making points were predicted by the HDM-Analyzer. Therefore, by utilizing the statistical information, which is collected by dynamic analysis, this ensured no overhead execution occurred. Thus, the results demonstrated that HDM-Analyzer attains superior accuracy and time complexity compared to the isolated methods of static and dynamic analysis methods.

A study by Ming et al. (2017), outlined an approach that utilises comparison mechanisms to detect behavioural-based features. Through substitution attacks that involve providing the system with call dependence graphs, allowing the identification of malware variants based on the features of malicious groups. Overall, the study examines a range of infiltrating approaches through data mining over 5200 malware behavioural specification tests, while establishing a compiler-level model to prompt the replacement attacks.

Evidently, the contents in the review above highlight the viability of behaviour-based detection and its ability to detect “zero-day attacks”. Exceeding other detection techniques, this method of malware disclosure been further enhanced due to the efforts of the studies above. Through data mining, deep-learning, revised systems and among other approaches, the countless advantages of behaviour-based detection has exceeded its capabilities of malware exposure. Although this method involves disadvantages such as higher resource usage and high false alarm rate. However, due to the discussed literature above, specifically the study by Eskandari et al. (2013), ensures that there is no overhead execution during the procedure, lowering the number of resources when using behaviour-based detection. Additionally, the study by Yuan et al. (2015), further improved resource efficiency through the implementation of deep-learning techniques. Overall, behaviour-based detection is an exceptional detection technique with very few drawbacks, a malware disclosing approach improved even further due to the studies discussed above.

Conclusion

In conclusion, this paper has effectively demonstrated a literature review about the various methods of malware detection, exhibiting their distinct aspects, advantages / disadvantages and potential to uncover malicious code. Through an in-depth discussion of available literatures, background knowledge and illustrations, this paper has successfully outlined the various findings that experts have established. Overall, this study has shown the efforts that researchers have conducted to ultimately overcome the rapid growth of malware. Consequently proving the importance of malware detection and the hardships that malware authors subject upon researchers.

References

Rankin, B. (2018). *A Brief History of Malware—Its Evolution and Impact*. [online].
Available at: https://bit.ly/2Xa4lry.

Vaughan-Nichols, S.J. (2019). *The day computer security turned real: The Morris Worm turns 30*. [online].
Available at: https://zd.net/2VAQ0nO.

Danchev, D. (n.d.). *Malware -future trends*. [online].
Available at: https://bit.ly/3E43Hgx.

Gutierrez, R. (n.d.). *3 Top Risks of Not Having a Firewall*. [online].
Available at: https://bit.ly/3Ea7ecY.

ResearchGate. (n.d.). *(PDF) A survey of malware detection techniques*. [online].
Available at: https://bit.ly/3nk7AYF.

Storm, D. (2010). *Murder by malware: Can computer viruses kill?* [online].
Available at: https://bit.ly/2YNTXqv.

N-able. (2019). *Malware Analysis Steps and Techniques*. [online].
Available at: https://bit.ly/3C0IbqR.

Richardson, A. (2021). *What is static analysis?* [online].
Available at: https://bit.ly/3tBuv2S.

Medeiros, N. (2021). *What is Dynamic Analysis?* [online].
Available at: https://bit.ly/2X3xW65.

Roundy, K. and Miller, B. (2010). *Hybrid Analysis and Control of Malware*. [online].
Available at: https://bit.ly/2X7kRIX.

Baker, K. (2020). *Malware Analysis Explained | Steps & Examples | CrowdStrike*. [online].
Available at: https://bit.ly/3CgYkZJ.

McGraw, G. and Morrisett, G. (2000). *Attacking Malicious Code: A Report to the Infosec Research Council*. [online].
Available at: https://bit.ly/3jZIw7e.

Uddin, M. and Rahman, A. (2010). Dynamic Multi Layer Signature based Intrusion Detection system Using Mobile Agents. *arXiv:1010.5036 [cs]*. [online].
Available at: https://bit.ly/3l8dWHG.

Bricata. (2018). *Signature Detection vs. Network Behavior | Bricata*. [online].
Available at: https://bit.ly/3A4s1wg.

N-able. (2021). *Intrusion Detection System (IDS): Signature vs. Anomaly-Based*. [online].
Available at: https://bit.ly/3tznvDF.

Mujumdar, A., Masiwal, G. and Meshram, D.B.B. (2013). *Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches*. [online].
Available at: https://bit.ly/3k4HQ0t.

Owino, Z. (n.d.). A Literature Study on Malware Detection Techniques. *www.academia.edu*. [online].
Available at: https://bit.ly/3noGfEA.

Fan, Y., Ye, Y. and Chen, L. (2016). *Malicious sequential pattern mining for automatic malware detection*. [online].
Available at: https://bit.ly/3llhZ3Q.

Cui, B., Jin, H., Carullo, G. and Lui, Z. (2015). *Service-oriented mobile malware detection system based on mining strategies.* [online].
Available at: https://bit.ly/3A6xdQ9.

Bat-Erdene, M., Park, H., Li, H., Lee, H. and Choi, M. (2017). *Entropy analysis to classify unknown packing algorithms for malware detection.* [online].
Available at: https://bit.ly/3nn7gbz.

Wu, B., Lu, T., Zheng, K., Zheng, D. and Lin, X. (2014). *Smartphone malware detection model based on artificial immune system.* [online].
Available at: https://bit.ly/3hqtq91.

Santos, I., Brezo, F., Ugarte-Pedrero, X. and Bringas, P. (2011). *Opcode sequences as representation of executables for data-mining-based unknown malware detection.* [online].
Available at: https://bit.ly/3k1S1mp.

Mohaisen, A., Alrawi, O. and Mohaisen, M. (2015). *AMAL: High-fidelity, behavior-based automated malware analysis and classification.* [online].
Available at: https://bit.ly/3tvOWy5.

Yuan, Z., Lu, Y. and Xue, Y. (2016). *Droiddetector: android malware characterization and detection using deep learning.* [online].
Available at: https://bit.ly/3A3eK7f.

Weijie, H., Jingfeng, X., Yong, W., Lu, H., Zixiao, K. and Limin, M. (2017). *MalDAE: Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics.* [online].
Available at: https://bit.ly/38VgjIq.

Eskandari, M., Khorshidpour, Z. and Hashemi, S. (2013). *HDM-Analyser: a hybrid analysis approach based on data mining techniques for malware detection.* [online].
Available at: https://bit.ly/3lfsaGX.

Ming, J., Xin, Z., Lan, P., Wu, D., Lui, P. and Mao, B. (2017). *Impeding behavior-based malware analysis via replacement attacks to malware specifications.* [online].
Available at: https://bit.ly/3A7kSeK.

Bhojani, N. (2014). *Malware Analysis*. [online]
Available at: https://bit.ly/3A4ZYge.