The Malware Detection Challenge of Accuracy

[1]Mohammad Akour
Yarmouk University, Jordan
mohammed.akour@yu.edu.jo

[2]Izzat Alsmadi
University of New Haven, USA
ialsmadi@newhaven.edu

[3]Mamoun Alazab
Macquarie University, Australia
mamoun.alazab@mq.edu.au

[4]Mohammad Z. Al-Saad
JUST University, Jordan
Mzalsaad15@cit.just.edu.jo

Abstract—**Real time Malware detection is still a big challenge; although considerable research showed advances of design and build systems that can automatically predicate the maliciousness of specific file, program, or website, Malware is continuously growing in terms of numbers and maliciousness. Web-based Malware detection is also growing with the expansion of the Internet and the availability of higher speeds and bandwidths. In this paper, we design, develop and evaluate an application that able to determine whether targeted website is malicious or not by utilizing available detection APIs. These APIs are able to communicate with several public scanners and Malware repositories. While the availability of many public scanners can help utilize those public services, however due to the fact that in most cases, they produce conflicting decisions, the process to make a final detection inference is not a trivial task. We conducted experiments to evaluate the different decision outcomes that come from the different scanners that utilized machine learning, data mining and other techniques. We also evaluated the issue of "unrated" decision based on the different Malware scanners.**

Keywords— Malware analysis; Malware detection, signature base; Machine Learning

I. Introduction

Malware threats continue to expand vertically (i.e. In numbers and volumes) and horizontally (i.e. in types and natures). The internet, social networks, smart phones, and e-commerce websites are all providing enormous opportunities for the creation of smart and sophisticated Malware.

Malware is the most generic term that is used to refer to all types of software that include a harmful or human-undesired goal. Malware is masked behind several forms such as Viruses, Worms, Trojan Horses, Backdoors, Spyware, Rootkits, Botnets, Keyloggers, etc.[1]. In dealing with Malware, there are several challenges related to detection, prevention and eradication [2].

The process of Malware detection can be accomplished based on different approaches including:

1. Dictionary based-detection: This is the first approach that is used by Malware detection systems. Previously known Malware can be identified based on certain attributes. If the subject or the software under investigation or analysis matches or includes those same attributes, it is positively identified as a Malware.

2. Role-based or signature-based detection methods. Signature or dictionary based detection is effective for known Malware. However, if a Malware is new, it will be unlikely detected based on the previous method unless if it's a copy-cat from a known virus. Some smart and sophisticated Malware try to follow a different or unique approach to make it hard to detect them.

Both categories can be further divided into static or dynamic analysis methods. Unlike dynamic methods, in static methods, the tested software is not executed or reverse engineered.

In both, cases false positive and negative detection outcomes maybe produced. In the Malware detection process one of the four following outcomes will be produced:

- True Positive (TP): This is the case where the detection method identifies the subject software as Malware and it is in fact a Malware.
- True Negative (TN): This is the case where the detection method identifies the subject software, correctly, as normal software (i.e. not a Malware).

The previous two outcomes are called true, because the detection outcome was correct and matches reality.

- False Positive (FP): The detection method identifies the subject software as a Malware, while it is normal software.
- False Negative (FN): The detection method identifies the subject software incorrectly as normal software (i.e. while it's a Malware).

A "perfect" detection method is the one that produces the following outcomes: TP: 100%, TN: 100%, FP:0% and FN 0%. Notice also that TP and FP complement each other and also TN and FN complement each other.

978-1-5090-4580-8/16/$31.00 ©2016 IEEE

In reality, such 100 % accuracy is impossible to achieve. This is particularly true as there is an important compromise between detection accuracy and speed of detection or performance. With a frequently large number of software applications or files to test, it is important to achieve the detection process quickly. This will not allow, for example, for rigorous detection methods that may take more time (while improve detection accuracy).

Malware detection processes are continuously evolving. Many existing websites allow users to check files or web links against predefined or new Malware. Those websites themselves make requests to popular Malware detection engines such as: AutoShun, PhishLabs, Kaspersky, StopBadware, Sophos, Netcraft, etc.

In this paper, we built an application to connect to different freely available web-based Malware detection systems. Those systems offer APIs to be used to interact with them and call their services. We noticed that detection results vary specially based on the time giving to wait for or process the detection or analysis process.

We used two different datasets. The first dataset includes popular links that are known to be hosts for malware or otherwise malicious. The second dataset includes government and educational websites from the country of Jordan.

II. RELATED WORKS

In this section, we will describe a selected list of closely related research papers to the subject of this paper.

Ye et al. [9] employed Objective-Oriented Association (OOA) mining based classification to build their signature based detection technique. Their model is improved by utilizing associative classification method based on the analysis of application programming interface (API) execution calls [10].Zolkipli and Jantan [11] built their model by combining signature-based technique and genetic algorithm technique, they addressed three types of Malware which are; Viruses, Worms and Trojan Horses.

Signature based detection is also applied at run time (i.e. dynamic analysis) where the researchers traced API calls and then built the suspected file signature [12].

Guo et al. [13]tried to build hybrid detection technique by integrating static and dynamic analysis to go behind the drawbacks of each technique. Their model is able to detectMalware and avoid its execution by combining static and dynamic binary translation features.

Others ideas were applied to solve the drawbacks of the signature-based detection such as utilizing graphs (Control Flow Graph) Bonfante et al. [14], Call graph Lee et al. [15], machine learning techniques Rieck et al. [16] and data mining techniques [17,18], Objective-Oriented Association (OOA) Yanfanget al. [19]. Others researchers apply techniques like finite automaton, HMM, data mining [20; 21], (Naïve Bayes, Support Vector Machine (SVM) and Decision tree) and neural network [22] to improve behavior-based detection.

Akour and Alsmadi [23] performed a vulnerability assessment on 20 Websites of universities in Jordan. They conducted passive penetration testing methods for confidential reasons. Their results showed that a significant number of those evaluated universities have critical or sever level vulnerabilities. Such vulnerabilities can be relatively easily be exploited by security attacks or attackers.

Tran et al. [24] tried to predict spam Emails containing Malicious Attachments and URLs. Their prediction model is based on descriptive sets of text features that allow identifying emails with malicious attachments and URLs. To evaluate their prediction model two real-world data sets of spam emails sourced from a manually labelled corpus (Habul) and automated collection from spamtraps (Botnet). The result reveal that emails with malicious attachments can be reliably predicted using text features extracted only from emails, without requiring external resources. However, this is not the case with emails with malicious URLs as their text features do not differ much from emails with non-malicious URLs.

Soskaand Christin [25] utilize several data mining techniques to build new classification system. The proposed system was able to predict whether a given, not yet compromised website will become malicious in the futureot not. They train their system on 444,519 archives sites containing a total of 4,916,203 webpages. The proposed prediction system accomplished positive rate of 66% and a false positive rate of 17%. The performance result was promising.

Kapravelos et al. [26] feveloped a tool called Hulk. The aim of this tool is to detect the malicious behavior in Google Chrome extensions. The development of Hulk is mainly based on dynamic execution of extensions and the usage of several techniques to trigger malicious functionality during execution. HoneyPages is developed to elicit malicious behavior, Furthermore; a fuzzer is built to drive the execution of event handlers registered by extensions

III. MALWARE DETECTION

In this section, we will expand the earlier discussion about Malware detection methods.

A. Static and Dynamic Analysis Techniques

Malware Detection methods use two main inputs to the detection process: 1) Training dataset: The signature or the behavior of common and well known Malware, 2) The suspicious or subject software: The software under study.

In the static analysis, the infected suspicious software files are investigated without executing them. Although static analysis may fail in analyzing unknown Malware that use code obfuscation techniques [6], still it could reveal more safety and rapid process in addressing the suspicious software and files. In terms of performance or speed, static detection methods are faster to accomplish than dynamic methods.

If the suspicious software/file is to be analyzed at run time then dynamic analysis is applied. Here the software under study is executed within isolated/fabricated environment to address its malicious behaviors.

If the suspicious software is able to change its behavior by using trigger conditions, dynamic analysis may also fail in detecting the malicious activities. This is considered as one of the main challenges of the dynamic analysis methods [6].

B. *Signature and Behavior Based Technique*

A signature is a sequence of bytes at specific locations within the executable, a regular expression, a hash value of binary data, or any other format created by Malware analyst which can accurately identify Malware instances [7].

Signature based techniques are widely used to complement earlier described dictionary-based techniques. For unknown Malware, or Malware that frequently change their behaviors, dictionary-based techniques will fail or create a lot of false positive and negative detection outcomes.

This process can complement dictionary based-methods as signatures are then extracted frequently and been added to the main dictionary repository. These repositories are checked by anti-Malware tools to procure the recent signatures for latest Malware instances.

Malware detection based on behavior techniques works on addressing the behavior of malicious code. These behaviors could be the source and destination addresses of the Malware, the attachment types in which they are embedded, the port in which they access the system or statistical anomalies in Malware infected systems [2]. In terms of speed of detection, those detection methods may require more time in comparison with either static or dictionary-based methods.2

IV. RESEARCH METHODOLOGY

Malware detection processes evolve from several different perspectives. Several websites now offer APIs through which users can create their own interface or application to call an available detection service. Those websites can have their own repositories or can just call public Malware repositories such as AutoShun, PhishLabs, Kaspersky, StopBadware, Sophos, or Netcraft. We namely identified the following problems in those automatic Malware detection systems:

1: **Multi-faceted detection decision:**

As most of those web-based automatic detection systems connect to several Malware repositories, varied decisions could be collected from the different repositories (i.e. whether the suspect file/program is Malicious or not). Ultimately, there should be another engine that makes the final decision (e.g. based on the aggregated votes from the different repositories). Which repository can be more accurate?! Which repository can produce more cases of FP or FN?! What is the most appropriate collective decision to make specially as in most cases we will not have a full agreement from all repositories (whether the suspect file/program is malicious or not). As all or most cases fall within this inclusive decision, studied should be made to evaluate different algorithms based on collective analysis.

2: **Removal decisions:**

The detection process is usually a pre-process for Malware removal or eradication. However, due to problems with FP and FN, such actions can cause significant problems especially if those decisions are to be made automatically without user-intervention. They may cause damages for important system, database, or user files. Malware action component should then have also different scenarios and should evaluate decisions and their impacts. For example, if a decision to stop or remove a file is made, this decision should be temporary (i.e. a backup of that file should be created). This makes it easy to undo or roll-back such actions if problems occurred afterward.

V. RESULTS AND DISCUSSION

This paper reports results for an ongoing research to address the three previously mentioned goals (i.e., Described in the research methodology section).

In the first experiment, we developed an application to use an API from (www.virustotal.com). The application can then use a dataset of files or links to be passed to the website and retrieve detection outcome based on a long list of Malware repositories or scanners. We evaluated a simple detection decision algorithm based on scanners' vote. The infection percentage is calculated as follows.

Infection Percentage = No_Infect ÷ Total scanners

Where No_Infect indicates the number of scanners that infer that suspect link is Malicious.

In the first link datasets, we used a publicly known malicious links dataset [27]. We observed the following based on running several cycles of Malware detection using listed scanners in Table 1:

- The experiment retrieved a list for each malicious link. This list consists of the scanner who voted as clean or as malicious and the target link. The result may vary (usually through increasing the number of infected decisions) when more time is given to the detection engine or algorithm.
- The list of (infected/clean) are varied, however if a scanner decides a link is infected, the decision is stable and repeated. This indicates that such decisions are made based on (solid) information. For example, those links or their malicious behaviors are previously known to those scanners.

Table 1 shows a sample of those links and their infection percentage. The Table shows that while those are known to be infected pages (i.e. all scanners should be able to detect that), all infection percentages are less than 50 %. Usually if a link is listed as malicious in only one Malware scanner, it is considered malicious although all other scanners may list the same link as clean or unrated.

TABLE1: A SAMPLE OF INFECTION PERCENTAGE FOR [27]

Link	Infection Percentage	link	Infection Percentage
1	0.134	11	0.030
2	0.119	12	0.044
3	0.029	13	0.045
4	0.014	14	0.030
5	0.014	15	0.044
6	0.074	16	0.074
7	0.102	17	0.030
8	0.147	18	0.045
9	0.015	19	0.014
10	0.121	20	0.047

In our second experiment, we decided to select a list of "normal" websites or links to evaluate infection percentage formula. We selected for the study more than 30 public websites from Jordan. Those websites are related to two main categories: Government websites and websites for public and private Universities.

Malware scanners vary in their decisions for the same evaluated files or links. In many cases, certain links can be reported as "clean" or "malicious" by some scanners while "unrated" by others. In the next experiment, we evaluated the number/percentage of unrated links in the evaluated websites' dataset.

Table 2 shows those Jordanian websites and summary of scanning results. We scanned a selected list of links in each website. Following are two constraints related to the number of selected links in each evaluated website:

1. Selected links within the website should be within the same domain name. We focused in this study to evaluate only internal links. However, we are aware that based on many references that most malicious links come from external or foreign links.
2. For large websites, we limited the crawling process to stop after 10 minutes. Evaluated links in large websites depend on how many links can be collected within this (10 minutes period). Unrated percentage is calculated based on the number of unrated links by the total number of crawled or evaluated links. The number of scanners that report a link as unrated can vary from one link to another. Nonetheless, we noticed that some particular scanners continuously report most links as unrated (see Table 3).

TABLE 2: RESULTS OF SCANNING JORDANIAN WEBSITES

NO of links	Unrated %	NO of links	Unrated %
5624	2	260	2
4989	20	6285	6.3
4510	2	2245	31
385	7.5	616	6.5
44	2	1116	3.5
475	0.2	885	2
6871	0.7	217	11.5
147	2	572	0.1
155	2	1310	0.3
538	0.2	1253	4
192	0.5	1138	0.1

Many of the evaluated websites report zero links as either malicious or unrated. In other words, all links in those websites are reported as "clean" in all scanners.

We also noticed that there is a pattern of scanners that report infected links. Those scanners are: AutoShun, Kaspersky, PhishLabs, StopBadware, Sophos, Netcraft, URLQuery. This may indicate that those scanners in particular have larger Malware repositories which make them detect those links as infected more than other scanners. In the future, we are planning to build a large dataset of known infected links and files. We will use this to build an accuracy model for each public scanner. This can be eventually used to vary the weight of each scanner in the final Malware detection decision.

4.1. Unrated Decisions

We noticed that many Malware detection systems continuously report detection decisions as unrated. Unrated decisions indicate lack of enough information in the particular detection system to judge whether a link or a file clean or not. This may also indicate a limitation in the existing dataset that can help in dictionary or signature based detections. Table below shows the top rated Malware systems with respect to the number or percentage of reported "unrated" decisions. Those numbers are extracted based on the popular malicious links that we used. However, we noticed also the same patterns in all other evaluated datasets.

TABLE 3: TOP SYSTEMS WITH "UNRATED" DECISIONS

Mal-system	Unrated	percent
StopBadware	230	0.987124
Netcraft	227	0.974249
AutoShun	205	0.879828
URLQuery	189	0.811159
Sophos	143	0.613734
PhishLabs	123	0.527897

Kaspersky	91	0.390558
Wepawet	86	0.369099
Websense ThreatSeeker	76	0.32618
Fortinet	66	0.283262
Trustwave	46	0.197425
Trustwave	46	0.197425
SecureBrain	2	0.008584

4.2. Malicious links

In the evaluated dataset of public websites in Jordan, we observed few cases of malicious links. We expected this number to be small due to the following two reasons:

- Links collected from each website are not comprehensive. They represent a sample of the total links. The size of this sample varies based on two factors. The first one is related to the website size and the number of links in that website. The second factor is related to our crawling process and how much links can be crawled within the fixed allocated time (i.e. 10 minutes).
- We decided to eliminate, in the crawling process, any external link within the evaluated websites. External links to any website are those links that are not part of the name of the main website. We acknowledge that this constraint eliminates a significant number of website related links (that are not necessary part of its main domain name). Additionally, and based on several references, most malicious links come from external ones. Table 4 showed a sample of those malicious links, their websites and Malware scanners that decided that such links are malicious. 8

TABLE4: A SAMPLE OF MALICIOUS LINKS FROM JORDANIAN WEBSITES

Web-site	link	Scanners
JU	http://goo.gl/gBAX1q	CLEAN MX, Dr.Web
JU	http://goo.gl/S936Mq	CLEAN MX, SCUMWARE.org, Sophos
JU	http://goo.gl/ICkvsy	CLEAN MX, Antiy-AVL, SCUMWARE.org, CRDF, Fortinet
Jrc-jordan	http://goo.gl/0XA7NB (Many links in the same website with the same Scanners)	Sucuri, SiteCheck
HU	http://goo.gl/gGdk5G (Many links in the same website with the same Scanners)	ParetoLogic
JUST	http://goo.gl/GXRgIC (Many links in the same website with the same Scanners)	Yandex, Safebrowsing

VI. CONCLUSION

The process of Malware detection continues to be very important to insure that information systems can be used only by intended users to perform desired tasks. Nonetheless, challenges to conduct this process successful continue to grow as well. The evolution of Malware detection is moving toward online and real time. Different scanners or detection repositories can be accessed via public APIs.

This paper presents an ongoing study to evaluate the future of Malware detection and reaction algorithms. We showed in this study that many public websites that provide important services can have Malware.

REFERENCES

[1] Christodorescu, M., Jha, S., Maughan, D., Song, D., Wang, C.: Malware Detection. In: Advances in Information Security. Verlag New York, Inc. Secaucus, NJ, USA: Springer (2007).

[2] Goertzel, Karen Mercedes. "Introduction to software security." Build Security In (2009).

[3] Goertzel, Karen M., Theodore Winograd, Holly L. McKinley, Lyndon J. Oh, Michael Colon, Thomas McGibbon, Elaine Fedchak, and Robert Vienneau. Software security assurance: a State-of-Art Report (SAR). information assurance technology analysis center (iatac) herndonva, 2007.

[4] Mell, P., Kent, K., &Nusbaum, J. (2005). Guide to Malware incident prevention and handling (pp. 800-83). US Department of Commerce, Technology Administration, National Institute of Standards and Technology

[5] M. Eskandari, Z. Khorshidpour, and S. Hashemi, “Hdm-analyser: a hybrid analysis approach based on data mining techniques for Malware detection,” Journal of Com-puter Virology and Hacking Techniques, vol. 9, no. 2, pp. 77–93, 2013.

[6] Egele, M., Scholte, T., Kirda, E., and Kruegel, C. (2012). A survey on automated dynamic Malware-analysis techniques and tools. ACM Computing Surveys (CSUR), 44(2), Article 6, 1-42.

[7] Bidgoli, Hossein. "Handbook of Information Security, Threats, Vulnerabilities, Prevention, Detection, and Management. vol. 3." (2006): 290.

[8] Aycock, John. Computer viruses and Malware. Vol. 22. Springer Science & Business Media, 2006.

[9] Ye, Y., Wang, D., Li, T., Ye, D., and Jiang, Q. (2008). An intelligent PE-Malware detection system based on association mining. Journal in Computer Virology, 4(4), 323-334.

[10] Ye, Y., Li, T., Jiang, Q., and Wang, Y. (2010). CIMDS: Adapting postprocessing techniques of associative classification for Malware detection. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 40(3), 298-307.

[11] Zolkipli, Mohamad Fadli, and AmanJantan. "A framework for Malware detection using combination technique and signature generation." InComputer Research and Development, 2010 Second International Conference on, pp. 196-199. IEEE, 2010.

[12] Vinod P., H. J., Yashwant K. Golecha, Manoj Singh Gaur, Vijay Laxmi. (2010). MEDUSA: Metamorphic Malware Dynamic analysis Using Signature from API. Proceedings of the 3rd international conference on Security of information and networks, ACM, 263-269.

[13] Guo, H., J. Pang, Y. Zhang, F. Yue and R. Zhao, 2010. HERO: A novel Malware detection framework based on binary translation. Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems, Oct. 29-31, IEEE Xplore Press, Xiamen, pp: 411-415. DOI: 10.1109/ICICISYS.2010.5658586

[14] Bonfante, G., Kaczmarek, M., and Marion, J.-Y. (2007). Control flow graphs as Malware signatures. International Workshop on the Theory of Computer Viruses, 1-6.

[15] Lee, Jusuk, KyoochangJeong, and Heejo Lee. "Detecting metamorphic Malware using code graphs." In Proceedings of the 2010 ACM symposium on applied computing, pp. 1970-1977. ACM, 2010.

[16] Rieck, K., Trinius, P., Willems, C., and Holz, T. (2011). Automatic analysis of Malware behavior using machine learning. Journal of Computer Security, 19(4), 639-668.

[17] Kephart and toilet JO Arnold, "Automatic extraction of computer virus signatures," in Proceedings of the 4th International Conference Bulletin, Abingdon Virus, UK, 1994. JO Kephart and services Arnold, 1994.

[18] Schultz, M. G., Eskin, E., Zadok, E., and Stolfo, S. J. (2001). Data mining methods for detection of new malicious executables. Symposium on Security and Privacy, 2001. S&P 2001. Proceedings. IEEE, 38-49

[19] Ye, Yanfang, Dingding Wang, Tao Li, Dongyi Ye, and Qingshan Jiang. "An intelligent PE-Malware detection system based on association mining."Journal in computer virology 4, no. 4 (2008): 323-334.

[20] Schultz, M. G., Eskin, E., Zadok, E., and Stolfo, S. J. (2001). Data mining methods for detection of new malicious executables. Symposium on Security and Privacy, 2001. S&P 2001. Proceedings. IEEE, 38-49.

[21] Siddiqui, M. A. (2008). Data mining methods for Malware detection: ProQuest. Sipser, M. (2006). Introduction to the Theory of Computation: Thomson Course Technology Boston, MA

[22] Tesauro, G. J., Kephart, J. O., and Sorkin, G. B. (1996). Neural networks for computer virus recognition. IEEE expert, 11(4), 5-6.

[23] Akour, Mohammed, and Izzat Alsmadi. "Vulnerability assessments: a case study of Jordanian universities." Open Source Software Computing (OSSCOM), 2015 International Conference on. IEEE, 2015)

[24] Tran, Khoi-Nguyen, Mamoun Alazab, and Roderic Broadhurst. "Towards a Feature Rich Model for Predicting Spam Emails Containing Malicious Attachments and URLs." 11th Australasian Data Mining Conference, Canberra. 2013.)

[25] Soska, K. and Christin, N., 2014. Automatically detecting vulnerable websites before they turn malicious. In 23rd USENIX Security Symposium (USENIX Security 14) (pp. 625-640).

[26] Kapravelos, A., Grier, C., Chachra, N., Kruegel, C., Vigna, G. and Paxson, V., 2014. Hulk: Eliciting malicious behavior in browser extensions. In 23rd USENIX Security Symposium (USENIX Security 14) (pp. 641-654).

[27] Malwared 2013-2015 c&c Full List, https://malwared.malwaremustdie.org/db/fulllist.php, retrieved, June 1st 2016.