REA605 ASSIGNMENT 6

Literature Review Draft 3

Gaston Carvallo - 048530133, Loyd Rafols - 022827158

REA605: Research Methodologies

February 22, 2021

Contents

Abstract

This paper is written to establish a current understanding of the state of the art in detecting malware using machine learning algorithms, particularly using features from network-based traffic and actions done on a computer host. The state of the art in the reviewed literature suggests that network-based detection methods can be effectively augmented using machine learning algorithms, in particular by detecting DNS subdomains and determining malicious payloads in encrypted traffic, but does not consider host-based features that would help in detecting malware. The state of the art with host-based detection revolves around the concept of detecting malware through features found within file information such as file signatures or entropy. The knowledge base is well-established but lacking in features that would indicate a compromised host on a system-level. In host-level detection, detection through features other than file signatures lack novel research and its feasibility is unknown. We identified a gap in regards to hybridization: detection of network-level and host-level features are separated in the reviewed literature and thus does not effectively provide a comprehensive detection scheme for malware that uses the network, the host, or both as its means of compromise.

Introduction

The present state of detecting malware through network-based and host-based methods through machine learning is well-developed, but is lacking in hybridization. With a majority of malware payloads taking place through the network and through the compromised host, it is important to consider both types of levels when implementing a malware detection scheme. This is important, because a lot of information is missed when only considering one level in developing a solution, which may be crucial in identifying that a malware attack is underway or

has occurred, as well as estimating the damage done by the attack. Additionally, the number of recent disruptive ransomware attacks is growing significantly (Cook, 2021). A recent example of one attack is Polish game developer CD Projekt Red, where an attacker stole and sold their proprietary information, while the ransomware disrupted their regular business operations (Abrams, 2021). The state of the art suggests that network-level detection is effectively augmented using machine learning, and provides additional help in differentiating malware from benign applications in encrypted traffic. Host-level detection is also augmented using machine learning, primarily through features presented in files, such as file signatures. An opportunity presents itself in creating a hybrid malware detection scheme, by utilizing features presented by a malware's network traffic and the changes it makes to a host, whether by file interactions or some other method. The purpose of this paper is to analyze current literature to establish a state of the art in utilizing machine learning to detect malware, as well the state of the art of methods used to detect malware in a host-based and network-based environment using machine learning.

Literature Review

We will first look at some of the general challenges presented by the use of machine learning in detecting malware, then we will look at some literature around the network-based and host-based methods to detecting malware with machine learning algorithms.

Machine Learning

Machine Learning (ML) has been used in many malware detection methods. Das and Morris (2017) emphasize the importance of extracting the features from the raw data. This is a two-step process, first identifying the features and then processing the raw data, for example within a packet capture. The effectiveness of the ML methods can then be analyzed using the

receiver operating characteristics curve, by plotting the true positive rate vs the false positive rate (Das and Morris, 2017, 4). Babaagba and Adesanya conducted a study on the effectiveness of selecting features and not selecting features for a machine learning algorithms in a supervised and unsupervised environment. They found that feature selection improved the true positive rate in both supervised and unsupervised machine learning algorithms at the cost of some training time, with the EM algorithm (the unsupervised machine learning algorithm) improving its accuracy from approximately 54% to 75% (Babaagba and Adesanya, 2019, 4).

Supervised Machine Learning methods require that the training data is labeled as features, which can be difficult to obtain. Noorbehbahani and Saberi (2020) explored the use of semi-supervised methods for detecting ransomware, i.e. methods where only part of the training data is labeled. While the semi-supervised methods were effective, they had to use supervised feature selection methods to create the models, and the use of simplified silhouette filter (an unsupervised feature selection method) to select the features did not provide acceptable results (Noorbehbahani and Saberi, 2020, 5).

While machine learning methods have been prove to be effective in malware detection, they are vulnerable to adversarial machine learning. Generative adversarial networks are an example of adversarial machine learning, which seeks to create samples designed to confuse the machine learning classifiers and lead them to misclassification (Martins et al, 2020). Both Suciu, Coull and Johns, and Liu and others and created adversarial examples designed to fool malware visualization detectors. (Suciu, Coull and Johns, 2019, and Liu et al, 2019)

Malware Detection

Malware detection has traditionally been classified in static and dynamic analysis. Static analysis looks at the source code of the malware in isolation. Signature based detection is one of

the main approaches to detect malware in this manner, while it can be fast and efficient for known malware it is ineffective for novel attacks and is susceptible to obfuscation attempts, like making changes to the source code or encrypting the file (Aslan and Samet, 2020, 6253). Dynamic analysis, on the other hand looks how the malware behaves, e.g., what system call it makes or how it changes the filesystem. While network analysis can be considered a subset of dynamic analysis, Manzano, Meneses and Leger (2020, 1) instead propose to classify detection methods as host-based and network-based contexts.

Network-based Detection

Some malware families require a connection to a command and control server in order to grab data needed for delivering its payload.[1] After the victim is infected, it establishes a connection to a server under the attacker's control. Through this connection, the attacker can issue direct commands to the malware and extract data. Researchers have proposed different methods that seek to determine the presence of a malware by trying to detect and classify these connections.

Modern malware tends to use DGAs[2] to establish a channel to its C2 server instead of hard coded IPs to prevent defenders from blocking the specific IP or domain used by a family of malware. Zhang (2020) proposed a Deep Learning method to detect DGAs. This method is based on the assumption that domain names generated by DGAs are by their nature more random that benign domains and therefore should have a character distribution different enough that it can be detected (Zhang, 2020, 464). The deep learning models achieved over 0.95 precision in the binary categorization, that is, whether the domain was from a DGA or not. Salehi and others

[1] Also known as C2 or C&C - a method of controlling multiple infected hosts through a centralized server.
[2] Domain Generation Algorithm - instead of using a static IP to create a C2 channel, pseudorandom generated subdomain names are used.

(2018) studied and showed success in detecting ransomwares based on their used of DGAs. They identified 3 classes of features: gibberish domains, the frequency of requests to different domains and re-generation of domains by the algorithm. Their detection engine is supplemented by a black/white list module to reduce false positives. However, the methods proposed cannot detect encrypted traffic as they need to be able to read the traffic's payload to make their classification.

Most of the research to detecting DGAs is under the assumption that the traffic is in plain text (Patsakis, Casino and Katos, 2019) however there are several protocols being evaluated to offer encrypted DNS services. Patsakis, Casino and Katos (2019) developed indicators of compromise that could distinguish legitimate DNS from those generated by a malware DGA. They only looked at DNS over HTTP and TLS, on the assumption that other encrypted DNS protocols are exotic enough that their presence alone can be detected and blocked by organizations firewalls or IDS (Patsakis, Casino and Katos, 2019, 4). Besides encrypting traffic, adversaries are starting to use word lists in their algorithms and make them harder to detect through the presence of entropy in the domains. Ren and others (2020) proposed a deep learning method to specifically detect these types of algorithms.

While the presence of a DGA might reveal the presence of a malware, the network traffic to the C2 server may also help an incident report detect the presence of the malware. Alhawi, Baldwin and Dehghantanha (2018) proposed a model to detect ransomware on Windows machines called NetConverse. They extracted their features using tshark (a network protocol analyzer) from the network conversations. their experiments had 97.1% detection rate accuracy using a Decision Tree algorithm (J48) but their model cannot detect ransomware using real-time data. In contrast Almashhadani and others (2019) created a prototype with two network detectors, one at packet level and the other at flow level for the Locky family of ransomware.

They looked at 3 aspects of its network traffic: through its distinguishable use of RST, ACK-flagged packets to terminate connections, its use of POST[3] requests and DGA-generated subdomains.

Instead of looking at the payload to obtain the features, Zhu and others (2018) proposed a model to detect Remote Access Trojans[4] that looks at the TCP[5] headers, this makes the model more efficient and useful for real time analysis. In their results the Random Forest method had an accuracy of 95.7%.

Ghafir and others also presented an approach for detecting of botnet C2 traffic that is capable of real time detection (Ghafir et al, 2018). In their proposed system called BotDet, which is divided in two stages. They developed four modules for detection of different approaches used in malicious C2 communications. Additionally, to reduce the rate of false positives, they also designed a framework for correlation which balances false positives and true positive rates.

In order to obfuscate their presence, some malware variants and families encrypt their traffic. Jakob Premrn explores creating a device capable of detecting encrypted C2 channels using a machine learning model (Premrn, 2020, 5). Due to the difference in size of the samples of legitimate and malicious traffic, a problem in machine learning known as imbalanced dataset was present. Premrn had to use the synthetic minority oversampling technique (SMOTE) to mitigate the effects of the imbalanced dataset. The model presented a high False Positive Rate which would make it unsuitable for day-to-day operations. They proposed future work by integrating it with cyber threat intelligence or some kind of whitelisting traffic (Premrn, 2020,

[3] POST is one the methods used in for HTTP traffic
[4] Also known as RATs - allows an attacker to remotely administratively control a machine over a network or the Internet
[5] Transport Control Protocol - used for transporting data over a network, the internet.

90). Jaimin Modi explored detecting ransomware encrypted traffic by using certificate-based fields of the network connections captures (certificates, signers, and dates) as the features in the machine learning classifier model (Modi, 2019). The model can only perform a binary classification (is the sample ransomware or not), and it cannot perform multiclass classification to attribute the ransomware to a specific family (Modi, 2019, 68). They propose to increase the model efficiency by adding an additional detector of DGAs.

Xia and others (2020) proposed a Network-Assisted Approach (NAA) for detecting ransomware. This is a hybrid solution that contains both local detection and network-level detection methods in a Linux environment. If the local component detects suspicious behaviour it suspends the process and creates an event, the user then classifies it as either normal or anomalous, if normal the NAA resumes the suspended process, otherwise it launches the network component. The NAA then query other hosts and creates a report on how many hosts have reported anomalous behaviour. The user then makes a final decision on whether or not ransomware exists on a host.

Host-based Detection

While most ransomware families need to contact their C2 server, about a third do not require C2 traffic, in such cases detecting it through network traffic is not viable (Berrueta et al, 2019). In this case, host-based detection serves as an important component to detecting malware on a system.

Arabo and others (2020) proposed a hybrid system to detect ransomware that used a machine learning module and one based on manually configured thresholds. While the system was somewhat successful, the machine learning module highest accuracy was only around 70%,

and the ransomware would always crash the local processes that the detector was running on. To function, the local component would send a log with the information to a remote host (Arabo et al, 2020, 294).

One particular feature of host-based detection is through the file-based features, such as API invocations for encrypting a file, or its characteristics such as file signatures. Bae, Lee and Im explored using machine learning to detect and classify using Windows Native API[6] invocation sequences, more specifically those relating to file management and manipulation as their features (Bae, Lee and Im 2018, 4). They propose a classification model, called Class Frequency - Non-Class Frequency (CF-NCF), which focuses around how many times something shows up in a certain class, instead of the traditional Term Frequency - Inverse Document Frequency that looks how many times the term show up in a document. They split the classes based on their category, that is ransomware, malware or benign classes. In conclusion, their model had an accuracy of 98.65% (Bae, Lee and Im, 2018, 4-5). Jevtha explored using file entropy and signatures as part of a ransomware detection scheme. In the paper, they explain that file entropy is not a good means of detecting ransomware, as legitimate applications such as MS Office and 7-zip have a high degree of entropy through compression and the algorithm would have a difficult time differentiating between those and ransomware (Jevtha, 2014, 41). Thus, they propose using file signatures as an additional feature to consider when training the algorithm, as file signatures are always a constant element when ransomware encrypts a file. (Jevtha, 2014, 43).

[6] Application Programming Interface - a means for software to allow interaction with itself through predefined functions or tasks.

Conclusions

In conclusion, we reviewed the state of the art within the topics of malware detection in network-based and host-based environment using machine learning. In a network-base environment, the state of the art revolves around detecting malware through its communication with C2 servers, primarily using DNS subdomains generated by a DGA. In our review, we identified a gap within only focusing on network features on malware. As we identified in the host-based detection section, a third of malware does not actually communicate over the network. In a host-based environment, the state of the art revolves around utilizing detectors to detect malicious processes, and through file-based features such as invocation of file API function calls or file signatures. However, we observed limited development through using other functions, such as cryptography functions, or opening network connections on a host. We believe there is a gap in knowledge surrounding combining network-based and host-based features into a hybrid malware detection scheme using machine learning, and is our focus going forward.

Bibliography

Abrams, Lawrence, 2021. "CD Projekt's Stolen Source Code Allegedly Sold By Ransomware Gang". Bleepingcomputer, Last modified 2021. https://www.bleepingcomputer.com/news/security/cd-projekts-stolen-source-code-allegedly-sold-by-ransomware-gang/.

Alhawi, Omar MK, James Baldwin, and Ali Dehghantanha. 2018. "Leveraging machine learning techniques for windows ransomware network traffic detection." *Cyber Threat Intelligence*, pp. 93-106. Springer, Cham, doi.org/10.1007/978-3-319-73951-9_5

Almashhadani, Ahmad, Mustafa Kaiiali, Sakir Sezer and Philip O'Kane, "A Multi-Classifier Network-Based Crypto Ransomware Detection System: A Case Study of Locky Ransomware", *IEEE Access, vol. 7*, pp. 47053-47067, 2019, doi: 10.1109/ACCESS.2019.2907485

Arabo, Abdullahi, Remi Dijoux, Timothee Poulain, and Gregoire Chevalier. "Detecting Ransomware Using Process Behavior Analysis." *Procedia Computer Science 168* (2020): 289-296.

Aslan, Omar and Refik Samet, 2020. "A Comprehensive Review on Malware Detection Approaches," *IEEE Access*, vol. 8, pp. 6249-6271, doi: 10.1109/ACCESS.2019.2963724.

Babaagba, Kehinde Oluwatoyin, and Samuel Olumide Adesanya. "A study on the effect of feature selection on malware analysis using machine learning." In *Proceedings of the 2019 8th international conference on educational and information technology*, pp. 51-55. 2019.

Bae, Seong Il, Gyu Bin Lee, and Eul Gyu Im. 2018. "Ransomware detection using machine learning algorithms." *Concurrency and Computation: Practice and Experience* 32, no. 18 (2020): e5422. doi: 10.1002/cpe.5422

Berrueta, Eduardo, Daniel Morato, Eduardo Magana, and Mikel Izal. "A survey on detection techniques for cryptographic ransomware." *IEEE Access 7* (2019): 144925-144944. doi: 10.1109/ACCESS.2019.2945839.

Cook, Sam. 2021. "Malware Statistics In 2021: Frequency, Impact, Cost & More". Comparitech, Last modified 2021. https://www.comparitech.com/antivirus/malware-statistics-facts/.

Das, Rishabh and Thomas. H. Morris, 2017. "Machine Learning and Cyber Security," *2017 International Conference on Computer, Electrical & Communication Engineering (ICCECE),* Kolkata, pp. 1-7, doi: 10.1109/ICCECE.2017.8526232.

Ghafir, I., V. Prenosil, M. Hammoudeh, T. Baker, S. Jabbar, S. Khalid, and S. Jaf,. 2018. "BotDet: A System for Real Time Botnet Command and Control Traffic Detection." *IEEE Access*: vol. 6, pp. 38947-38958. doi: 10.1109/ACCESS.2018.2846740.2.

Gümüşbaş, Dilara, Tulay Yıldırım, Angelo Genovese, and Fabio Scotti. 2020 "A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems." *IEEE Systems Journal*, doi: 10.1109/JSYST.2020.2992966.

Jethva, Brijesh. "A new ransomware detection scheme based on tracking file signature and file entropy." PhD diss., University of Victoria, Department of Electrical and Computer Engineering, 2019.

Liu, Xinbo, Jiliang Zhang, Yaping Lin and He Li, "ATMPA: Attacking Machine Learning-based Malware Visualization Detection Methods via Adversarial Examples," *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*, Phoenix, AZ, USA, 2019, pp. 1-10, doi: 10.1145/3326285.3329073.

Manzano, Carlos, Claudio Meneses, and Paul Leger. "An Empirical Comparison of Supervised Algorithms for Ransomware Identification on Network Traffic." *39th International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1-7. IEEE, 2020.

Martins, Muno, Jose M. Cruz, Tiago Cruz and Pedro H. Abreu, "Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review*," in IEEE Access, vol. 8*, pp. 35403-35419, 2020, doi: 10.1109/ACCESS.2020.2974752.

Modi, Jaimin, 2019. "*Detecting Ransomware in Encrypted Network Traffic Using Machine Learning*", Master's thesis, University of Victoria.

Noorbehbahani, Fakhroddin, and Mohammad Saberi. "Ransomware Detection with Semi-Supervised Learning." In *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 024-029. IEEE, 2020.

Patsakis, Constantinos, Fran Casino, and Vasilios Katos, 2020. "Encrypted and covert DNS queries for botnets: Challenges and countermeasures." *Computers & Security 88.* doi.org/10.1016/j.cose.2019.101614.

Premrn, Jakob, 2020. "*Analysis of command and control connections using machine learning algorithms*." Master's thesis, University of Ljubljana, Faculty of Electrical Engineering.

Ren, Fangli, Zhengwei Jiang, Xuren Wang, and Jian Liu. "A DGA domain names detection modeling method based on integrating an attention mechanism and deep neural network." *Cybersecurity 3, no. 1* (2020): 1-13.doi: 10.1186/s42400-020-00046-6

Salehi, Saeid, Hamid Reza Shahriari, Mohammad Mehdi Ahmadian and Ladan Tazik, "A Novel Approach for Detecting DGA-based Ransomwares," *2018 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*, Tehran, Iran, 2018, pp. 1-7, doi: 10.1109/ISCISC.2018.8546941.

Suciu, Octavian, Scott E. Coull and Jeffrey Johns, "Exploring Adversarial Examples in Malware Detection," *2019 IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, USA, 2019, pp. 8-14, doi: 10.1109/SPW.2019.00015.

Xia, T., Y. Sun, S. Zhu, Z. Rasheed, and K. Shafique. "Toward A Network-Assisted Approach for Effective Ransomware Detection." EAI Endorsed Transactions on Security and Safety: Online First, January 2021. doi: 10.4108/eai.28-1-2021.168506.2

Zhang, Yihang 2020."Automatic Algorithmically Generated Domain Detection with Deep Learning Methods," *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, Shenyang, China, 2020, pp. 463-469, doi: 10.1109/AUTEEE50969.2020.9315559.

Zhu, H., Z. Wu, J. Tian, Z. Tian, H. Qiao, X. Li, and S. Chen. 2018. "A Network Behavior Analysis Method to Detect Reverse Remote Access Trojan." *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, pp. 1007-1010, doi: 10.1109/ICSESS.2018.8663903.