PROJECT INFORMATION

Project Title	**A Practical Introduction to Applying Machine Learning to Malicious Traffic Detection**	
Technology Area	**Machine Learning**	
Project Team	**Name of Team Members**	**Main Responsibilities**
	Gaston Carvallo	
	Loyd Rafols	
Keywords (max. 4)	**1. Machine Learning**	**2. Command and Control**
	3. Educational	**4. Network Analysis**

Abstract

Malicious attacks are one of the most damaging and expensive threats organizations face. Machine learning methods have shown great potential toward the problem of an increasing number of variants through there are still a number of challenges to implement these methods in production environments that require further research. Yet, in our experience there seem to be a tendency for students and beginners to treat it as an inscrutable topic reserved for only the selected few. We propose creating a practical introduction to the subject that can illustrate how machine learning works, helping to demystify it and serve as a base for further research and learning.

1. INTRODUCTION

Malware is one of the fastest growing cybersecurity threats that individuals and organization face on a regular basis. The constant evolution of malware proves to be an ongoing issue for solutions that attempt to stop its attacks, as relying on human-based analysis proves to be more infeasible day by day.

The current state of detecting malware through network-based and host-based methods through machine learning is well-developed, but tend to look at each aspect in isolation. With the majority of malware payloads taking place through the network and/or through the compromised host, it is important to consider both types when implementing a malware detection scheme. Information is missed when only considering one aspect, 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.

The number of recent disruptive ransomware attacks is growing significantly (Cook). Symantec claims to have found over 186,000 new ransomware variants in 2018 alone (Symantec, 38). One notable victim of these malware attacks was the UK's National Health Services, which cost them over $100 million in damages (Brunau) because it failed to detect the malware traversing their network and the encryption it did on the infected systems.

In 2020, the average cost for a destructive malware breach was $4.52 million, and the average cost for a ransomware breach was $4.44 million (IBM, 44). As the number of malware variants keeps increasing, detecting malware using machine learning is thought to be one the most promising ways to help with this issues and present significant amount of research in the field. And yet in our experience most students in information security have no practical experience on how theses method work.

We propose the creation of a curriculum that includes practical labs describing the steps we needed to take in order to create a machine learning model to detect malicious traffic. That can serve as a practical introduction the use of machine learning in security.

2. PROJECT OBJECTIVES

The main objective is to create a machine learning model to detect network malicious attacks to illustrate the process to newcomers. We will create our own dataset by generating the malicious samples using the Metasploit Framework. Our goal is to illustrate the practical aspects in applying Machine Learning to security so we will create accompanying material illustrating in detail how the data was gathered, and the model was created and applied.

3. LITERATURE REVIEW

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, 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 (1) instead propose to classify detection methods as host-based and network-based contexts.

In the literature, we noticed one certain common limitation between a majority of our papers. That limitation was that each paper or model was good at detecting one certain characteristic of a malware, however, they were poor at detecting other characteristics, or omit them entirely. There is little overlap in terms of a detecting both network-based and host-based features of malware in a hybrid malware detection model, and thus in a realistic scenario, would excel at detecting one type or family of malware, but would fail at detecting another.

3.1 Network-based Detection

Some malware families require a connection to a command and control server[1] in order to grab data needed for delivering its payload. 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 using subdomains instead of hard coded IPs to prevent defenders from blocking the specific IP or domain used by a family of malware.

Salehi and others (6) studied and showed success in detecting ransomwares based on their use of DGAs for subdomains. 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. Zhang utilized deep learning algorithms to use one-hot encoding[3] for their DGA detection features (Zhang, 464).

Most of the research for detecting DGAs is under the assumption that the traffic is in plain text (Patsakis et al. 2) however there are several protocols being evaluated to offer encrypted DNS services. These approaches are good at detecting network-based feature of malware, but their limitation is around their easiness of tampering by attackers.

Patsakis and others (6) developed indicators of compromise that could distinguish legitimate DNS from those generated by a malware DGA. They identified that DGAs tend to generate domains

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.
[3] A method of encoding non-rankable items into a numeric order (such as colors)

of similar length and therefore the response packets tend to be similar in length, they also noticed that DGAs queries have a cyclical component that is possible to detect through a statistical analysis (Patsakis et al. 4). While these methods might work currently, the behaviors seem to be easily modified by attackers.

Research has also been conducted in detecting malware directly through its network traffic. Zhu and others (1008) proposed a model to detect Remote Access Trojans[4] that looks at the TCP[5] headers, they selected 4 features based on RAT's different traffic pattern. For example, benign applications tend to send as much data as possible as soon as the connection is established, RATs might show what they called early-stage, a period of time where noticeable idle time is present between packets (Zhu et al. 1008). This model has a good baseline for detecting RATs, and can be modified to detect C2 traffic for malware.

Alhawi, and others (5) proposed a model to detect ransomware on Windows machines called NetConverse. They manually selected 13 features from traffic conversations[6] but their model cannot detect ransomware using real-time data.

In contrast, Almashhadani and others (47063) created a working prototype with two network detectors, one packet-based and a second flow-based for the Locky ransomware family. The features were selected both manually and through the WEKA[7] feature selection tool. The features revolved around 3 aspects of the network traffic: a distinguishable use of RST, ACK-flagged[8] packets to terminate connections, its use of POST[9] requests and DGA-generated subdomains. This paper

[4] Also known as RATs - allows an attacker to remotely control a machine over a network or the Internet
[5] Transport Control Protocol - used for transporting data over a network, the internet.
[6] Defined as the bidirectional traffic for a 5-tuple flow (from an source ip:port to a destination ip:port on the same protocol)
[7] Waikato Environment for Knowledge Analysis, an open-source data mining and machine learning tool
[8] Flags used to terminate a TCP connection - stands for Reset, Acknowledge
[9] POST is one the methods used in for HTTP traffic

provides a good basis for network feature-based detection of malware, but does not test other malware families aside from the Locky ransomware family.

In order to obfuscate their presence, some malware variants encrypt their traffic. Premrn explores creating a device capable of detecting encrypted C2 channels using a machine learning model (Premrn, 5). They manually selected 6 features from the connection logs (instead of traffic capture) (Premrn, 54). Their model presented a high False Positive Rate which would make it unsuitable for day-to-day operations, so, they proposed integrating it with some kind of IP whitelisting to reduce false positives (Premrn, 90).

Modi (6) also explored detecting malware through encrypted traffic, instead of just using connection statistics they also selected features related to the TLS[10] hand-shake and the certificate used (Modi, 35). They propose to increase the model efficiency by adding an additional detector of DGAs (Modi, 68). Overall, their model is limited in capability because it can only classify if the sample is ransomware or not, and it cannot perform multiclass classification to attribute the ransomware to a specific family or as a general malware.

In summary, the approaches we reviewed have the limitation in that they do not account for host-based features, so if malware was to exist on a host that does not communicate with an external host, this approach would be ineffective.

3.2 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. 144929). Host-based methods are also harder to evade, while attackers can and do change malware behavior

[10] Transport Layer Security - a protocol that encrypts internet traffic

to obfuscate their presence, ultimately there are action the malware need to perform to accomplish its objective which cannot be hidden (Almashhadani et al. 47057).

Arabo and others (291) proposed a system to detect ransomware that used two detection modules: One that uses machine learning and the other based on manually configured thresholds. The machine learning features were selected around the malware resource usage (CPU, RAM and disk access). Their machine learning model was only partially successful in detecting the ransomware (Arabo et al. 294), as it does not consider if the malware was not particularly resource intensive, or used other resources such as networking.

Bae, Lee and Im (3) explored using machine learning to detect ransomware through Windows Native API[11] invocation sequences when a file is executed (Bae et al. 4). They proposed a classification model called Class Frequency - Non-Class Frequency (CF-NCF). This classification model focuses around how many times something shows up in a certain class (benign, malware and ransomware), instead of the traditional Term Frequency - Inverse Document Frequency that looks how many times the term shows up in a document (Bae et al. 4). This approach is limited as it does not utilize other API function calls for malware detection, as well as not utilizing network-based features on the host for C2 detection.

In comparison to the previous paper, Hirano and Kobayashi (1) proposed a framework to detect ransomware that collects I/O requests through a hypervisor[12] instead of the OS to make the framework portable. They selected 5 features related to the read/write characteristic of the encryption process ransomware use (Hirano and Kobayashi, 4). This enables usage with any operating system instead of just Windows exclusively in the previous paper.

[11] Application Programming Interface - a means for software to allow interaction with itself through predefined functions or tasks.
[12] A means of running one or several virtual computers on one or more physical computers

Some researchers select their features manually, according to their knowledge of the dataset and the malware behavior, others however use automated tools to extract numerous raw features from the system and then use an automated algorithm (heuristics), such the Chi-squared test method[13] and fine tune the final feature set used by their machine learning model.

For example, Sethi and others (1) put forward a framework where raw features are extracted from a sandbox's report when a file is executed and then the chi-squared test is used to select features for the detection model, they create two models, the first one classifies the executable in benign/malware, when a malware is detected a second model classify the malware family (Sethi et al. 3). This approach to host-based malware detection provides a good framework for the host-based component of our hybrid malware detection model, and future work can be done using other malware families and network-based features.

Shhadat and others (918) looked at the impact the heuristics can have in the model accuracy. They expanded on the work of Chumachenko that used a similar framework than Sethi of extracting features from a sandbox. Shhadat et al. (919) used a different heuristic to select the features (cross-validation). While the models that used decision-tree and Random forest saw no significant change, models using Naïve Bayes saw significant improvement (Shhadat et al. 922).

Jethva (6) suggested a hybrid host-based malware detection model. Their solution has two detectors, one based on a ML model using heuristics (chi-squared test) to narrow the features (Jethva, 43); and the other based on the combination of file entropy (encrypted files show higher entropy) and the presence of file signatures (magic numbers) to help distinguish benign compressed files that also show high entropy (Jethva, 34).

[13] A test used to determine the differences between a theoretical model and actual data, in this case used to refine the accuracy of the model

The lack of labelling datasets may prove to be a limitation on research datasets. Noorbehbahani and Saberi (24) looked into the use of semi-supervised methods. They used 5 supervised heuristics to extract the feature set and then used semi-supervised classifiers to identify ransomware (Noorbehbahani and Saberi, 25). The main limitation of this approach is that the utilized unsupervised feature selection accuracy was very poor, which makes it unfeasible to use by itself and would improve by implementing it in a hybrid malware detection scheme.

In summary, the literature around host-based detection mainly suggests that there is a limitation with most approaches in that only host-based features are considered, whereas the detection rate would improve if network-based features were also implemented.

4. DESCRIPTION OF THE PROPOSED WORK

In order to accomplish our project, we plan on breaking it down in 4 different objectives (see table 1), these are: creating the dataset, creating the Machine Learning model, creating the educational material, and creating the final report.

4.1 Approach, tasks and phases:

The creation of the sample data is further breakdown in two phases (see table 2). Deploying the testbed (see figure 1) and generating and capturing the malicious and benign traffic.

The creation of a machine learning model to detect malicious traffic will consists of 3 phases (see table 2): Selecting the features, training and validation, and evaluating the model. it should be able to do a binary classification (benign/malicious). The objective of this model is to be used as an example and to help illustrate educational objectives. It is not expected that the model should be able to run in a production environment, or to offer improvements in detection over existing models.

The creation of a curriculum designed to introduce detection through machine learning. This curriculum will cover 4 topics, gathering data, feature selection, training and validating a ML model and evaluation an ML. They each will include a theorical background and a practical lab to reinforce the content. They are not meant to satisfy requirements for any accreditation or certification, and only serve as a practical introduction to the topic.

TABLE 1: APPROACH UTILIZED FOR ACHIEVING OBJECTIVES

Objective	Approach of achieving the objective
Create dataset	Experiments, automated testing
Create Machine Learning model	Data analysis
Create educational material	Labs, PowerPoint presentations
Create final report	Report writing

TABLE 2: MAPPING OF PHASES AND TASKS TO ACHIEVE OBJECTIVES

Objectives	Phases	Tasks
Create dataset	Deploy Testbed	• Deploy Kali workstation • Deploy windows and Linux metasploitable 3 • Deploy Security Onion • Configure Mirror port • Validate connectivity
Create dataset	Create Dataset	• Capture benign traffic • Capture malicious traffic • Clean and label data
Create Machine Learning model	Feature Selection	• Pre-process and normalize dataset • Perform feature selection through a filter method • Perform feature selection through a wrapper method
Create Machine Learning model	Machine Learning Application	• Train and validate ML applying Random Forest • Train and validate ML applying J48 Decision Tree • Train and validate ML applying Support Vector Machine • Train and validate ML applying Naive Bayes
Create Machine Learning model	Model evaluation	• Create confusion matrix • Create ROC chart • Create classification report
Create educational material	PowerPoint Presentation	• Create presentation for data gathering • Create presentation for feature selection • Create presentation for training and validation • Create presentation for model evaluation
Create educational material	Labs	• Create tutorial for gathering data • Create tutorial for feature selection • Create tutorial for training and validation • Create tutorial for model evaluation
Create final report	Final Report Writing final report	• Write abstract and introduction • Write Literature Review and Background • Write Methodology and Experiments • Write Discussion and Conclusion

4.2 Research methodology

A virtual testbed (see figure 1) will be created to perform the experiments, all machines will be run as virtual machines using VMware. It will consist of one Kali workstation to perform malicious attack through the Metasploit framework, two victim machines one windows and one Linux and a monitor station running Security Onion. The Security Onion host will monitor the traffic through a mirror port on the switch.

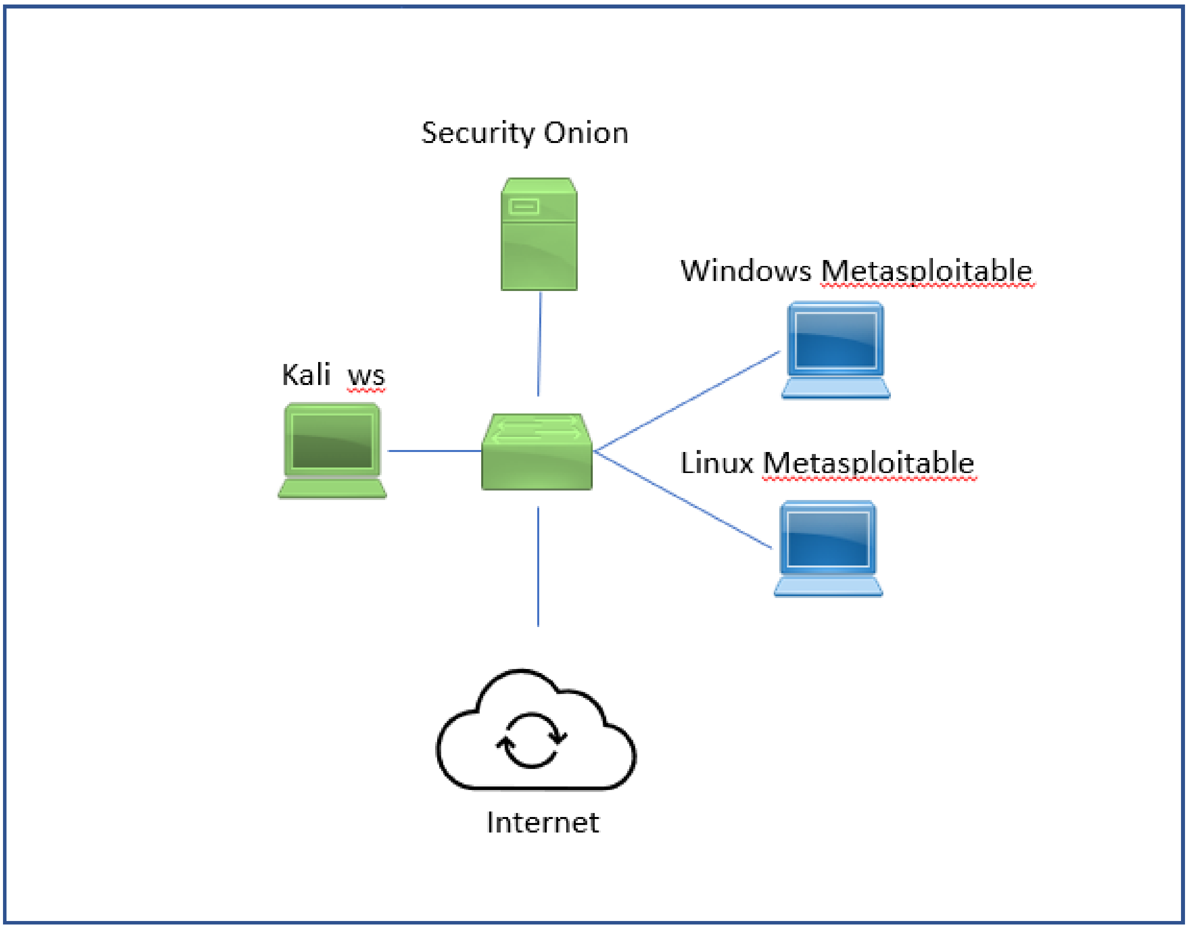

Figure 1: Testbed

The dataset will be created by generating 50 samples of malicious traffic and 50 benign samples, each sample will consist of a network capture (PCAP file) of 15 minutes of user activity. The malicious activity will be created by using Metasploit's payloads on the target machine and conducting activities through the reverse shell. Benign traffic will be performed by visiting legitimate websites selected using Alexa top 100 domains or similar method. The capture will be done in Security Onion through its Zeek logs.

The plan is to use the scikit-learn library to perform the data analysis however if needed, alternative tools such WEKA will be used. At least 2 different algorithms will be used for the feature selection in order to compare the results and illustrate how they work. If possible, one will be a filter method and the other one a wrapper one to maximize the contrasting possibilities between the two type of methods.

For the classification engine, at least 4 algorithms will be used. Based on our research we have initially selected the J48 decision tree, Random Forest, Support Vector Machine and Naive Bayes, these were among the most common ones we observed researchers using. The model will only perform binary classification i.e., malicious/benign traffic.

The classifiers will be evaluated using a confusion matrix, the receiver operating characteristics curve (ROC) and the classification report. These are standards and are used universally when evaluating machine learning models.

5. REFERENCES

Alhawi, Omar MK, et al. "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, et al. "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, et al. "Detecting Ransomware Using Process Behavior Analysis." *Procedia Computer Science 168* (2020): 289-296.

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

Bae, Seong Il, et al. "Ransomware detection using machine learning algorithms." *Concurrency and Computation: Practice and Experience* 32, no. 18 (2020): e5422. doi: 10.1002/cpe.5422

Berrueta, Eduardo, et al. "A survey on detection techniques for cryptographic ransomware*." IEEE Access 7* (2019): 144925-144944. doi: 10.1109/ACCESS.2019.2945839.

Brunau, Chris. “Ransomware News: WannaCry Attack Costs NHS Over $100 Million”. Datto, last modified October 18, 2018. www.datto.com/uk/blog/ransomware-news-wannacry-attack-costs-nhs-over-100-million

Chumachenko, Kateryna. *Machine learning methods for malware detection and classification*. Bachelor’s Thesis, 2017, South-Eastern Finland University of Applied Sciences (Xamk).

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

Hirano, Manabu and R. Kobayashi, "Machine Learning Based Ransomware Detection Using Storage Access Patterns Obtained From Live-forensic Hypervisor,*" 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, Granada, Spain, 2019, pp. 1-6, doi: 10.1109/IOTSMS48152.2019.8939214.

IBM. "Cost Of A Data Breach Report 2020." IBM Security, IBM Corporation, last modified July 2020, www.ibm.com/security/digital-assets/cost-data-breach-report/

Jethva, Brijesh. *A new ransomware detection scheme based on tracking file signature and file entropy*. Master's Thesis, 2019, University of Victoria, Department of Electrical and Computer Engineering.

Manzano, Carlos, et al. "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.

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

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

Patsakis, Constantinos, et al. "Encrypted and covert DNS queries for botnets: Challenges and countermeasures." *Computers & Security 88.* 2020 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.

Salehi, Saeid, et al. "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.

Sethi, Kamalakanta, et al. "A novel machine learning based malware detection and classification framework." *In 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, 2019, pp. 1-4. IEEE, doi:10.1109/CyberSecPODS.2019.8885196.

Shhadat, Ihab, et al. "The Use of Machine Learning Techniques to Advance the Detection and Classification of Unknown Malware.*" Procedia Computer Science 170*, 2020, pp 917-922.

Symantec. "Internet Security Threat Report Volume 24, February 2019". ISTR, Symantec Corporation, last modified 2019. docs.broadcom.com/doc/istr-24-2019-en

Zhang, Yihang. "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., et al. "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, 2018, pp. 1007-1010, doi: 10.1109/ICSESS.2018.8663903.