International Journal of Advanced Science and Technology
Vol. 29, No. 4, (2020), pp. 6512 - 6519

Integration of Honeypots and Machine Learning in Network Security

Chaitanya D Patil*, Thyagarajamurthy A**
*M.Tech. Student at JSS Science & Technology University(SJCE), Mysuru, Karnataka, India.
**Associate Professor in the Department of Electronics and Communication Engineering, JSS Science & Technology University (SJCE), Mysuru,Karnataka, India.
*cdpgdg@gmail.com **thyagarajamurthy@sjce.ac.in

Abstract

The integration of honeypots and machine learning technologies in network security offers a comprehensive and robust security management framework that enables organizations to secure their information systems from malwares like Trojan Horse, Adware and Spyware. There are a lot of proposed approaches in the area of network security but they still lack handling the newer malwares. System security software like firewall and antivirus fail to detect the malicious software. So the technique is the combination of these strategies allows security analysts to effectively conduct a dynamic analysis of network threats and automate threat management and modeling based on machine learning tools. Such automated analysis and control of breaches enable the company to understand the intent, source, nature, and functionality of the malware. This understanding leads to the development of robust security management frameworks that enhance the resilience and preparedness of the company. Support Vector Machine (SVM) and Decision Tree algorithms are implemented for classification of datasets. This paper examines this integration of honeypots in security network infrastructure and mitigating malicious software. Through this focus, the article offers a robust assessment of machine learning and honeypots in securing business information infrastructure.

Keywords: *Decision tree, honeypot, machine learning, malware analysis, network security, SVM.*

1. Introduction

The evolving threat landscape, the complexity of attack strategies, and the increased connectedness of corporate devices pose significant challenges in effectively securing business systems. Contemporary firms have pervasively connected their information systems, mainly due to the emergence of cloud computing, the Internet of Things, laptops, and smartphone devices. Although these devices expand the corporate network infrastructure, increase efficiency and productivity, they also add new attack surfaces that increase the vulnerability of the organizational systems. In response to these security issues, information security researchers and cybersecurity professionals have developed offensive and defensive approaches to identify and respond to security incidents. While defensive strategies focus on mitigating and containing threats, offensive strategies focus on deceiving the attacker, identifying vulnerabilities, and patching system flaws. One such security approaches are the use of honeypots, which mislead attackers in the target organizational system. The integration of machine learning strategies in honeypots offers a robust security management framework, enabling corporations to effectively manage their information infrastructures, such as securing the corporate network and mitigating denial of service attacks. The amount of malwares also continues to increase and as per the data released by GData there were 5,998,685 malwares in 2014[1]. The largest amount of malware is in Trojan Horse malware while the most dramatically increased malware is in Adware malware. Security framework tools such as antivirus, firewall, and signature-based IDS are known to fail in malware detection process. This is because computer malware is spreading very quickly, and signatures are growing. Besides signature-based protection systems new approaches, viruses or worms used by attackers are hard to detect. Another option is to use honeypot with machine learning when detecting malware. Honeypot can be used as a trap for suspicious programs when studying the software to detect malware by classifying classes [2]. Network protection has become very critical in protecting the organization's information. With tremendous Internet growth, attack cases are increasing daily along with modern method

of attack. One approach to this issue is using the IDS (Intrusion Detection System). One of the techniques used in IDS is Machine Learning. Machine Learning Intrusion Detection technology has been providing high precision and effective detection on novel attacks. In this paper, the performance of the Support Vector Machine(SVM) machine learning algorithm is evaluated [3].

2. Literature Survey

Conventionally, corporate information security focuses on deploying antivirus and antimalware tools that provided defensive security control of the business information systems. In this security approach, the information security officer relies on the signature or heuristic detection strategies adopted by the security solution to identify and mitigate threats facing the company. Although this framework offers robust security management for the known threats, it fails to effectively analyze the nature of the attack, review the risk, and understand the intent of the attacker. Such weaknesses limit the ability of this security management framework to effectively contain new threats, which do not have their signatures or heuristics extracted yet. It is due to these security management flaws that honeypots are developed and deployed at the corporate network. A honeypot is a security resource designed and implemented to lure attackers, enabling them to probe, attack, and compromise the system [4]. The objective of this system is to study attack landscape, analyze the threats, and understand the attack strategy and intention of the hackers. Through this approach, they enable the organization to effectively monitor, analyze, and review security breaches without compromising critical infrastructure, leading to the effective management of such incidents [5]. Thus, a honeypot provides a comprehensive threat analysis framework that enables organizations to contain threats and develop robust security management frameworks effectively. Fig. 1. below highlights a typical honeypot infrastructure.

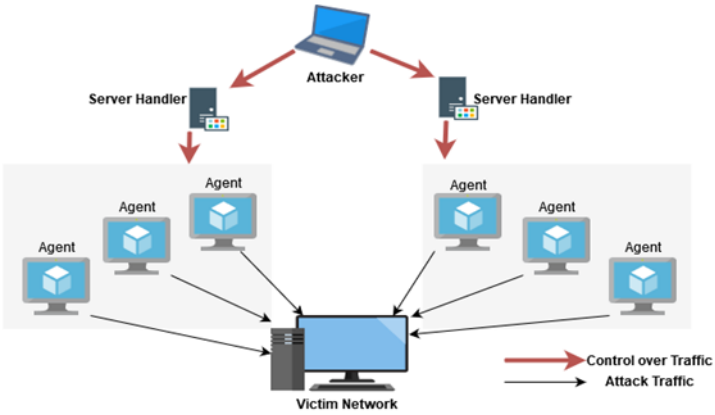

Figure1. A typical honeypot infrastructure

Honeypots play a significant role in information security management due to their ability to capture, analyze, and harden organizational information systems, leading to resilience and preparedness. The organization deploys a LAN (Local Area Network) of honeypots within its information infrastructure to establish a robust threat analysis and management framework. Such LAN is known as a honeynet and optimizes threat analysis and modeling in the company, and is effectively separated from the core network infrastructure. There are two major types of honeypots, which are production and research. A production honeypot comprises of the minimal information infrastructure that is designed to monitor malicious traffic or malware within the corporate network [5]. A honeynet built from such honeypots offers minimum threat analysis within the company. On the other hand, research honeypot provides comprehensive malware analysis and modeling features that enable the organization to review malicious traffic or malware effectively. A honeynet built with research honeypots offers robust threat management features that allow the organization to effectively model, examine and review intrusions, breaches, and malware. After the honeynet is deployed, border routers are configured to redirect traffic to this LAN, providing active logging,

notification, and analysis features [6]. During the logging process, honeynets extract features from malicious traffic or remote malware being deployed in the research honeynet network. The features extracted are documented in the logging system, leading to the effective capturing of the threat attributes.

The design and development of honeypots have undergone significant progress. Initially, honeypots provided minimal threat analysis, with limited feature extraction. However, contemporary honeypots have advanced feature extraction capabilities, high interaction, and are low risk, effectively separating the attacker's traffic from the corporate network [7]. The ability to extract these comprehensive features from the intrusion offers robust malware analysis and threat modeling framework that enhances security management and framework development in the organization. Such honeypots engage the attack sequence, a phenomenon that prolongs hacking activity to capture the motive of the attacker effectively, tools used, strategies implemented, and understand the data being targeted [7]. Although these security systems are extensively automated to capture, review, and analyze malware, there is a need for human intervention in developing the appropriate security framework. In this approach, the security analyst examines the features extracted and uses automated tools to model the attack and understand the threat, enabling him/her to develop a robust security management framework that enhances the resilience of organizational systems. The complexity of distributed attacks and limitations of conventional security strategies raises the need for robust honeypot systems that rely on machine learning frameworks for feature extraction and threat analysis. Traditional security frameworks such as antimalware and antivirus have limited capability of analyzing botnets and distributed network attacks due to their complexity [10]. Although these tools identify the malware based on the known signatures or heuristics, they are incapable of analyzing the malware traffic or its request processing, which limits their ability to examine distributed attacks in large corporate networks effectively. On the other hand, traditional intrusion detection systems are based on heuristics or signature detection frameworks, which limits their ability to detect novel or zero-day distributed attacks [11]. This inability to detect unknown malware or attacks compromises the corporate information infrastructure, exposing it to new attacks or threats. Similarly, these intrusion detection systems depend on static signature analysis, which hinders them from identifying and analyzing variants of the malware. As a result, they are incapable of detecting the malware before initializing the payload, limiting the ability of these security systems to offer real-time threat management features [11].

A typical honeypot system comprises of three major components that collectively allow the organization to secure organizational systems. The first component is the data collection, which captures the traffic and malware directed to the corporate network [12]. The data collection tool obtains the source IP addresses, timestamps, and other basic features obtained during logging. The feature selection component extracts attributes of the malware, providing adequate data for threat analysis [12]. The feature selection component extracts attributes such as attack commands, threat signature, and hash value. The last element is the separation module, which is responsible for threat analysis and decision making based on the underlying threat intelligence. The integration of machine learning systems in honeypots is primarily implemented in the separation component, leading to robust threat identification and analysis. Decision Tree and Support Vector Machines are the standard algorithms used in honeypots [2]. These machine learning algorithms detect and analyze malware based on their features and behavior on organizational systems. Decision Trees and Support Vector Machines are preferred due to their efficiency, accuracy, and pattern identification [2].
The integration of machine learning algorithms in the separation component of the honeypot automates threat detection and analysis, leading to effective threat management. The common automation strategies used in these systems are classification and clustering algorithms, which have various functionality and performance on the honeypot infrastructure. Classification machine learning algorithms are useful in examining existing malware and its variants, leading to robust threat management [12]. However, these threat analysis frameworks have limited capability in detecting new malware or threats. On the other hand, clustering algorithms are useful in analyzing and identifying novel and zero-day malware, leading to robust threat management at the corporate level. The optimized performance of the clustering algorithms makes them the ideal machine learning algorithms for the separation component of the honeypot infrastructure. This use of a clustering algorithm offers a robust dynamic analysis of new malware and threats, enabling

security analysts to review the function calls, parameters, information flow, and instruction trace [13]. The typical implementation of this clustering algorithm-based honeypot is in examining the communication patterns of the intrusion, detecting malicious servers on the corporate network, and analyzing corporate networks to determine a breach [9]. Thus, the integration of machine learning with honeypots offers a robust security management framework, enabling the company to secure its information resources adequately.

3. Challenges in Network Security

The primary security challenge facing corporate networks is the denial of service attack, which seeks to compromise system availability and accessibility. A denial of service attack targets the server resources through request saturation, exceeding the maximum requests the server may handle at an instance [8]. There are instances when denial-of-service attacks are coordinated through botnets. The integration of these attack strategies leads to distributed denial-of-service attacks that emerge from many compromised hosts within the corporate network [9]. This approach leads to a sophisticated flood of requests redirected to potential single-point-of-failure in the corporate network, such as a server or router. Fig. 2 shows the conventional way of how a honeypot isolates the network from benign traffic. The demilitarized zone (DMZ) is created as a perimeter network around the web servers to expose an organization's external services to a usually large network such as the Internet.

Intrusion Prevention System (IPS) analyzes the transmission packets but also stops the packets based on the kind of attacks the system detects which helps in stopping the attack. Intrusion Detection System (IDS) only analyzes the network traffic for specific signatures which might match the known cyberattacks. IDS/IPS flags any matching packets by comparing them with the known signatures of cyber threat databases.

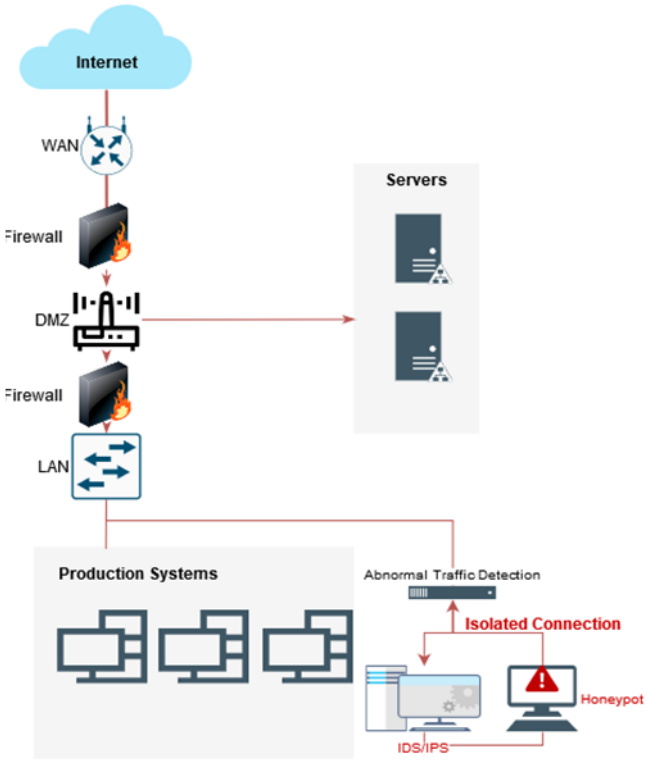

Figure2. A conventional distributed network attack (DDOS)

4. Methodology

Most necessary part of any specified novel approach is to check the feasibility of the approach and access the efficiency of the already available similar solutions. Fig 3. Shows the proposed methodology and process flow for the honeypot based solution.

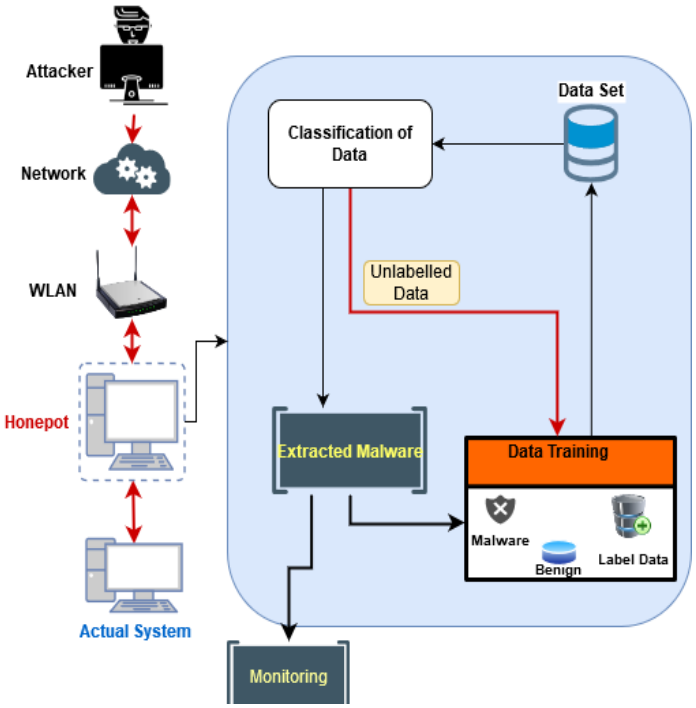

Figure 3. Proposed Architecture

The architecture design of the complete honeypot and machine learning system is delineated. This design consists of network components like wireless routers and honeypots and the actual system. The attacker tries to penetrate the network but due to the implementation of honeypot the actual system is not affected. All the traffic of external network comes directly into the internal network. Honeypot is responsible for capturing the traffic packets that have entered the interior network and stored in a virtual network. The router forwards the packets from the external network into the internal network and then to the honeypot as its intended purpose. The process of classification can be shown in the steps below:

- The classification will we between malicious and benign. As the dataset contains unlabeled data we classify the data into labeled data, extract the malware and predict the unlabeled malware by training the labeled data with the predicted data. This process of labeling unlabeled data is known as pseudo labeling.
- We then take these predictions and label each piece of unlabeled data with the individual output that was predicted for them.
- We then train our model on full dataset which is now consisting of both truly labeled data with the data that was pseudo labeled.
- So as Fig.3. shows the extracted malware is set for data training and monitored by an expert.

For the classification and analysis of data, learning process is done by the malware detection system. Data sets are classified by two algorithms, Decision Tree algorithm and Support Vector Machine (SVM). These algorithms play a role in classification of the dataset. This is more clearly demonstrated in the experimental results.

5. Experimental Results

A. Employed Dataset

Figure 4. Distribution of datasets

Dataset employed here is Endgame Malware BEnchmark for Research (EMBER) [14]. It is a collection features that serve as benchmark dataset analysts. It consists of version 1 features gathered before 2017 and version 2 features calculated before 2018. The dataset includes 9,00,000 training samples (3,00,000 malicious samples, 3,00,000 benign samples, and 300,000 unlabeled samples). Classification based on semi-supervised machine learning can be done if the dataset has some labeled and most of it as unlabeled data. So Fig. 4. shows the simple distribution of all the data samples.

B. Split dataset

Dataset should be split into training, validation and holdout data. So we use data collected from before as a training set and the newer data as a validation set. After that we need to choose an accuracy metric to select the qualified data. This process iterates and refines the model's performance. It finds ways to split the data more accurately to provide the targeted outcomes. This process of splitting and training the model is performed using the Scikit-Learn library [15]. Fig. 5 shows the test size inside a function that indicates the percentage of data for testing. We have split the dataset into 80% training and 20% testing.

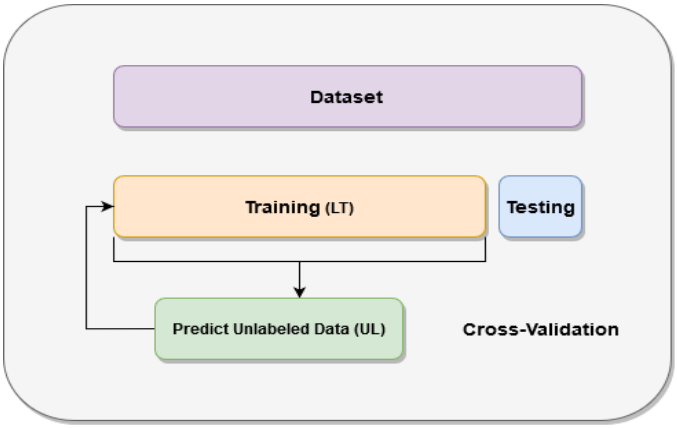

Figure 5. Splitting Dataset into testing and training data

At the beginning the dataset is first split into test data(DT) and training data(TRD). Then the training data is used to train the model and use it to predict the unlabeled data(UL). Now we group the predicted results (P) and TRD and use that to test on the DT.

C. Cross Validation Testing

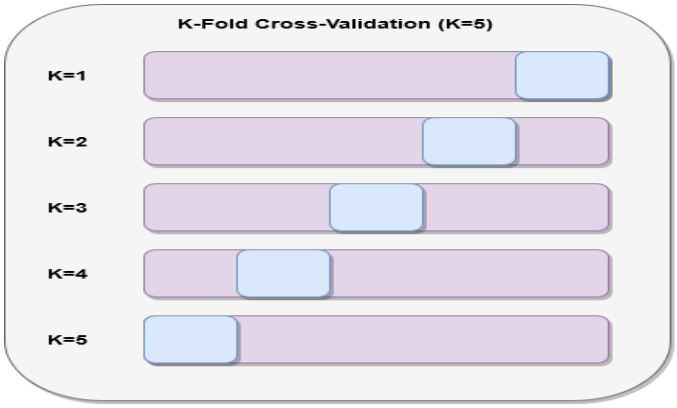

Figure 6. Cross-Validation

Cross-Validation tests are normally used to compare and examine the data classifier algorithms. Here K-Fold cross validation is used where we split the data into k different subsets. Fig. 6. shows the number of cross-validation with k=5. We trained the k-1 subsets and use the last subset as a test data. This is done for each of the folds to finalize the model. Testing data should not include the training data as it will become the cause of over-fitting and also an unfair classifier performance. The number of folds k=5. As datasets are large the computation time becomes a problem and over-fitting can occur so we used a lower K value. As semi-supervised learning is applied for this dataset we used SVM algorithm and built the classifier. The accuracy obtained was 93.84% and if I apply 10 fold the accuracy reduces. But it included some misclassifications. We didn't remove the misclassified data as they provide more information regarding the correct location of decision boundary. The evaluation of measurement of accuracy can be shown by Table. 1. of confusion matrix with four results.

Table. 1. Confusion Matrix

Parameters		Prediction	
		Malicious	Benign
Actual Data	Malicious	True Positive(TP)	False Negative(FN)
	Benign	False Positive(FP)	True Negative(TN)

1. True Positive (TP) which means the malicious file is identified.
2. False Negative (FN) shows that it failed to identify the malicious file.
3. True Negative (TN) proves that the file identified is benign.
4. False Positive (FP) shows that the benign file is identified as a malicious file.

These four results determine the accuracy of the model. The Sensitivity and Specificity can be shown as below.

$$Sensitivity = \frac{TP}{TP+FN}, \qquad Specificity = \frac{TN}{TN+FP}.$$

An important component affecting the execution of semi-supervised learning procedures is the number of labeled examples so it was necessary to look at the classification accuracy of the presented algorithms.

6. Conclusion

The evolving complexity of the threat landscape and inefficiency of conventional security frameworks make machine learning-based honeypots the ideal security management infrastructure in the corporate sector. The proposed architecture for detecting the malicious files using honeypot and semi-supervised machine learning using Decision Tree and SVM algorithm to classify the datasets and split them by 80:20 ratios to provide higher accuracy is provided. Cross-Validation is performed by 5 K-Fold experiments. By the utilization of pseudo labeling we trained on a vastly larger dataset which otherwise may have likely taken many hours of struggle. So through this process we can conclude how this approach makes use of both supervised learning and unsupervised learning with labeled and unlabeled data together giving rise to a practice of semi-supervised learning. Further research and implementation can be done by comparing the accuracy of other classification algorithms and different cross validations suitable for unsupervised machine learning.

References

[1] GData Security, “Malware Report: Half-Year Report July-December 2014,” 2014.
[2] I. M. Matin and B. Rahardjo, “Malware Detection Using Honeypot and Machine Learning,” in *CITSM*, 2019, pp. 1-4.
[3] Kamarularifin Abd Jalil, Muhammad Hilmi Kamarudin, and Mohamad Noorman Masrek, “Comparison of Machine Learning algorithms performance in detecting network intrusion,” in *2010 International Conference on Networking and Information Technology*, Manila, Philippines, 2010, pp. 221–226.
[4] S. Dowling, M. Schukat, and E. Barrett, “Using reinforcement learning to conceal honeypot functionality,” in *ECML-PKDD*, 2018, pp. 341-355.
[5] M. Zuzcak and T. Sochor, “ Behavioral analysis of bot activity in infected systems using honeypots,” in *ICCNCT*, 2017, pp. 118-133.
[6] V. Mehta et al., “Threat prediction using honeypot and machine learning,” in *ICFT-CAKM*, 2015, pp. 278-282.
[7] S. Dowling, M. Schukat, and E. Barrett, “Improving adaptive honeypot functionality with efficient reinforcement learning parameters for automated malware,” *J. of Cyb. Sec. Tech.*, vol. 3, no. 2, pp. 75-91, Apr. 2018.
[8] M. Jonker et al., “Millions of targets under attack: a macroscopic characterization of the DoS ecosystem,” in *Proc of IMC*, 2017, pp. 100-113.
[9] Y. Feng et al., “Feature selection for machine learning-based early detection of distributed cyber-attacks,” in *PiCom*, 2018, pp. 173-180.
[10] C. I. Rene and J. Abdullah, “Malicious code intrusion detection using machine learning and indicators of compromise,” *Int. J. of Comp. Sci. and Info. Sec.*, vol. 15, no. 9, pp. 160-171, Sep. 2017.
[11] C. Moore, “Detecting ransomware with honeypot techniques,” in *CCC*, 2016, pp. 77-81.
[12] J. Gardiner and S. Nagaraja, “On the security of machine learning in malware C & C detection: A survey,” *ACM Comp. Surv.*, vol. 49, no. 3, pp. 1-39, Dec. 2016.
[13] P. D. Ali and T. G. Kumar, “Malware capturing and detection in dionaea honeypot,” in *i-PACT*, 2017, pp. 1-5.
[14] H. Anderson and P. Roth, "EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models”, in ArXiv e-prints. Apr. 2018.
[15] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.