Hindawi
Security and Communication Networks
Volume 2020, Article ID 8865474, 9 pages
https://doi.org/10.1155/2020/8865474

Research Article

A Smart Agent Design for Cyber Security Based on Honeypot and Machine Learning

Nadiya El Kamel,[1] **Mohamed Eddabbah**,[2] **Youssef Lmoumen**,[1] **and Raja Touahni**[1]

[1]*Laboratoire des Systèmes de Télécommunication et Ingénierie de la Décision (LASTID), Département de Physique, Faculté des Sciences, Université Ibn Tofail, Kenitra, Morocco*
[2]*LABTIC Laboratory ENSA, Abdelmalek Essaadi University Tangier, Tangier, Morocco*

Correspondence should be addressed to Nadiya El Kamel; nadiya.elkamel@uit.ac.ma

Received 17 May 2020; Accepted 16 July 2020; Published 7 August 2020

Academic Editor: Sajjad Shaukat

The development of Internet and social media contributes to multiplying the data produced on the Internet and the connected nodes, but the default installation and the configuration of variety of software systems represent some security holes and shortcomings, while the majority of Internet users have not really set up safety awareness, leading to huge security risks. With the development of network attack techniques, every host on the Internet has become the target of attacks. Therefore, the network information security cannot be ignored as a problem. To deal with 0-day and future attacks, the honeypot technique can be used not only passively as an information system, but also to reinforce the traditional defense systems against future attacks. In this paper, we present an introduction of machine learning and honeypot systems, and based on these technologies, we design a smart agent for cyber-attack prevention and prediction.

1. Introduction

According to GDATA [1], the number of new attacks increases exponentially, each year, millions of attacks are detected (Figure 1), which involve more sophisticated and automatic analysis tools, since traditional tools are limited in the case of a huge quantity of information or when it is about new kinds of attacks. In fact, the main disadvantages of expert-based analysis are time consumption and the difficulty of classifying attacks [2].

The knowledge of opponent motivations, objectives, and techniques used to gain unauthorized access to the systems is the key not only to stop and protect systems from attacks but also to learn and predict new attacks that can hit our systems. Honeypots technology was deployed since 1992 [3], as a powerful information system, which consists of monitoring, detecting, and analyzing malicious activities, it is used to complement the traditional strategies such as intrusion detection systems (IDS) and log files, which are ineffective due to the huge quantity of information, false alarms, and the inability of detecting new attacks [4]. The honeypot is a security resource implemented for being probed, attacked, or compromised [4, 5], it was proposed to automatically consider any interaction detected as a malicious activity, while the administrator network uses the reports generated by the malicious source, to learn about the identity, motivations, and techniques used by the intruder to infiltrate the system.

The purpose of this paper is to show, firstly, the strength of using machine learning and honeypots, as solutions for the cyber security purpose, through some related works and by introducing these technologies. The second purpose of this work is to discuss a cyber security solution based on honeypot and machine learning techniques. Our main objective is to design an intelligent agent for predicting new attack profiles by analyzing, automatically, the gathered data via the honeypot, using a combination of machine learning algorithms. The objective of the algorithms combination is to represent the data with a lot of accuracy and build an efficient predictive agent for cyber security, especially for the future and 0-day attacks prediction.

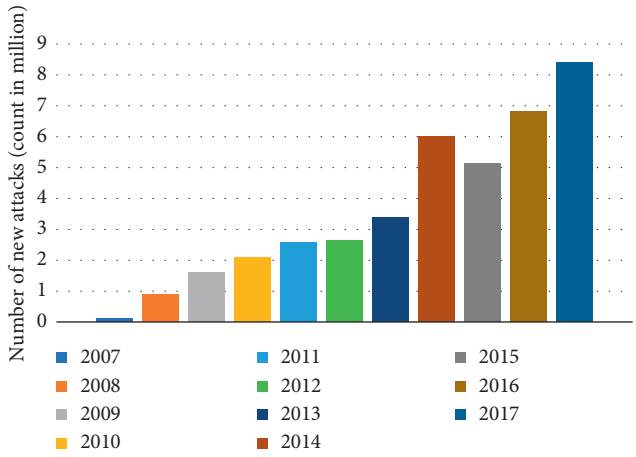

Figure 1: New attacks evolution.

The rest of this paper is organized as follows: in Section 2, we discuss some related work, Sections 3 and 4 are devoted to the introduction of machine learning and honeypot technologies for cyber security, and in Section 5, we discuss the proposed predictive design.

2. Related Work

Anomalies characterization receives a lot of attention; all seek to protect themselves against fraudulent use of their data or against malicious intrusions into computer systems. A lot of security solutions were proposed in the last decade, but results still present some limitations [2], and the most recent works are based on machine learning algorithms to model anomalies activities using data collected by information technologies such as honeypots.

The authors in [6] propose an intelligent honeypot which improves IoTs devices' security, based on machine learning. In order to store each device response, an IoT-scanner was proposed to probe accessible IoT devices on the Internet and scan the Internet for each malicious interaction, and a model called IoTLearner was trained to be used by the intelligent honeypot that can optimize a model to reply attackers.

The authors in [2] propose an autonomous method for attacks characterization, based on unsupervised anomalies learning, using the collected information by honeypots. This approach is based on clustering techniques such as density-based clustering, subspace clustering, and evidence accumulation for classifying flow ensembles in traffic classes. The advantage of this method is that it does not require a training phase.

The authors in [7] propose an automatic classification of social spam-based machine learning (e.g., SVM), for network communities such as Facebook and MySpace using a social honeypot to gather information about malicious profiles.

The authors in [8] propose a linkage defense system [9] based honeypot to overcome the limitations of the traditional tools. The linkage technique will ensure management and communication between the honeypot and components of the defense system, constructing a linkage management module based on SNMP protocol for network management. In order to overcome the problem of new attacks, the system is centroid honeypot for treating suspicious flows arrived from the traditional defense system, and the decision to block or not will depend on the state of the honeypot. If the honeypot is damaged, then the correspondent intruder will be blocked by the firewall [8].

3. Machine Learning

A computer is not smart; it performs tasks described in a program form, as orders of what to do and how to do it, and this is called traditional programming. While writing a traditional program, the decision is made directly into the program. Machine learning is a subarea of artificial intelligence [10] that aims to give computers the opportunity to learn; its techniques allow understanding the structure of the data and integrating them into models that can be understood and used to solve complex problems in real-life situations, and its techniques represent an efficient tool to address the significant challenges posted by the big data [11].

Machine learning has turned the concept of traditional programming around (Figure 2), by training models and making them able to learn and make decisions without being explicitly programmed. The job of a machine learning model is to output predictions based on mathematical hypothesis functions, and while passing data, this hypothesis maps from input variables to outputs or to find structures or clusters in them.

A machine learning model is designed in two phases: the first is to estimate the model from the available data, by executing practical tasks such as animal recognition in pictures, speech translation, or participating in autonomous vehicles driving, this is called the training phase, and it is generally performed before the practical use of the model. The second is the production phase, meaning the phase of passing new data to obtain the result corresponding to the desired task. According to the information available during the learning phase, learning is qualified in different ways; if the data is labelled, then the learning would be supervised, in a more general case, and without labels, the learning is unsupervised.

Depending on the nature of the problem, there are different approaches that vary depending on the type and volume of data. Machine learning categories are divided into supervised, unsupervised, and reinforcement learning [12]. In this section, we discuss briefly each category.

The supervised learning [11, 12] consists of learning prediction functions from a database of pairs input-output or from labelled examples. The supervised learning problems can be divided into two categories, namely, regression and classification problems [13]. In regression problems, results are output within continuous values by mapping input variables to some continuous functions. In contrast, classification problems allow to output results within discrete values. There are several supervised algorithms widely used such as linear regression (LR), support vector machine (SVM), and decision trees (DT). For the unsupervised learning [11, 12], the data have no labels, and we only receive

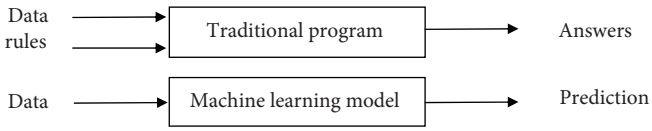

Figure 2: Comparison between machine learning and traditional programming.

raw observations of random variables. Therefore, the training algorithm applies in this case, to find similarities and distinctions within the data and group together all those which have common characteristics [12]. For example, an epidemiologist would like to try to bring out, in a large set of liver cancer victims, an explanatory hypothesis to this illness. The computer could differentiate some groups, which the epidemiologist would, then, associate with various explanatory factors, geographical origins, genetics, consumption habits or practices, and exposure to various potentially or actually toxic agents (heavy metals and toxins such as aflatoxin). Another approach of semisupervised learning [12] was proposed; it consists of a combination between a set of labelled and not labelled data; it is a good compromise between the two "supervised" and "unsupervised" types of learning, it allows to analyze a large number of data without the need to label them all, and takes advantage of both types mentioned. The last type of learning is the reinforcement learning [3, 11, 12], in which an agent interacts iteratively with the environment to learn and to improve its comportment simultaneously. Reinforcement learning differs from the first type of learning in the way that, in supervised learning, the training data are labelled, so the model is trained with the correct answer itself, whereas in reinforcement learning, there are no labels; instead, the agent learns from its experience to performs the given task [3].

The next part is devoted to discuss some machine learning examples.

3.1. Linear Regression. LR (Linear regression) [14] is used to estimate a linear hypothesis function between the output and the input variables, as regression or a classification tool [15], and it is written as follows:

$$h_\theta(X) = \theta_0 + \theta_1 \cdot X_1 + \ldots + \theta_n \cdot X_n, \tag{1}$$

where h_θ is the hypothesis function, X_i represents input variables, and θ_{ii} represents weights of the hypothesis function.

Weights represent the parameters of the hypothesis function. For estimating weight values, the first step is to calculate the error between the estimated result $(\widehat{y})$ and the expected result (y) using the cost function. The mean squared error (MSE) is the most widely used cost function, and it is written as follows:

$$J = \frac{1}{N} \sum_{i=1}^{N} \left(\widehat{y}\left(x^{(i)}\right) - y^{(i)}\right)^2. \tag{2}$$

The second step is to apply the gradient descent algorithm [16], which represents the most important part in linear regression, is the simplest algorithm to apply, and it allows to estimate the optimal weight values by minimizing the cost function. The gradient descent is an iterative algorithm which updates the weights at each iteration to minimize the cost function by setting a threshold value.

3.2. K-Means Algorithm. *K*-means [17] is one of the most widely used clustering algorithms; it is an iterative algorithm, where the first step is to randomly initialize cluster centroids [17] and start moving the centroids until every point is allocated to the nearest cluster, keeping the centroids as small as possible. The most widely used clustering criterion is the sum of the squared Euclidean distances between each data point $(x_{1,}x_2, \ldots, x_N)$ and the centroid m_k, and this criterion is called clustering error *E*.

$$E\left(m_{1,}m_2, \ldots, m_M\right) = \sum_{1}^{N} \sum_{1}^{M} \left\|x_i - m_k\right\|^2. \tag{3}$$

The K-means algorithm finds locally optimal solutions with respect to the clustering error. It is a fast-iterative algorithm that has been used in many clustering applications [18], but it is still having a major disadvantage which backs to its sensitivity to initial positions of the cluster centers and the number of clusters [17, 19]. Therefore, in order to obtain near optimal solutions using the k-means algorithm, several runs must be scheduled differing in the initial positions of the cluster centers. To overcome this problem, different approaches have been proposed such as the Global K-means approach [20].

3.3. Decision Trees. A decision tree is a very simple model representing a set of choices in a graphic form of a tree [21], and it is a hierarchical representation of the data structure in the form of decision sequences (tests) for classes or results prediction. Given several characteristics, the decision begins with one of these characteristics, and if that is not enough, another one is used and so on. Each individual (or observation), must be assigned to a class which is described by a set of variables tested in the tree's nodes. Tests are performed in internal nodes, and decisions are made in leaf nodes [22]. DT (decision trees) can be used for both classification and regression problems. There are several automatic algorithms for building decision trees: ID3, C4.5, and CART.

ID3 starts with placing all the learning examples in the root node. Then, each node is cut to one of the remaining attributes (which has not been tested yet). The choice of this attribute is made through a measure of homogeneity; it uses the entropy function and the gain of information according to an attribute to decide which is the best characteristic [21].

ID3 shows some limitations when it is about continuous characteristics and a large number of feature values [22]. C4.5 is an extension of ID3; it was proposed in 1993 by Ross Quinlan as an amelioration of ID3, to support continuous attributes and missing values, and it is based on the pruning technique to reduce the prediction error rate [22]. The authors in [22] proposed a comparative study of decision trees ID3 and C4.5, they compared the execution time and

the accuracy in function of the dataset size, and the results show that C4.5 is similar to ID3 in terms of accuracy, but it is more effective than ID3 in the execution time.

Machine learning is used for a wide spectrum of applications, and it represents an efficient analysis method for smart cities, especially in intelligent transportation systems (ITS) and smart grids, due to the large amount of generated data by control systems, information and communication technologies, and advanced sensors. In ITS, different machine learning methods such as deep neural network (DNN) and deep reinforcement learning (DRL) were proposed for monitoring and estimating real-time traffic flow data, estimating the possibility of accidents, trajectory design, and cyber physical security [12]. Smart grid networks use computer technologies to optimize the production, distribution, consumption, and possibly, the storage of energy in order to better coordinate all the meshes of the electrical network, from producers to the final consumer. A lot of proposed works are based on machine learning algorithms to analyze heterogeneous data arrived from different sources, and power grid control, a Deep Long Short-Term Memory (DLSTM) model, was proposed to forecast the price and demand for electricity for a day and week ahead [12].

4. Honeypot

Information security policy generally aims to set up mechanisms to guarantee services in terms of integrity, confidentiality, authentication, identification, availability, and access control. Attacks are based on tools that scan the entire networks looking for vulnerable systems [23]; hence, the originality of the honeypot lies in the fact that the system is voluntarily presented as a weak source able to hold the attention of attackers [5]. The general purpose of honeypots is to make the intruder believe that he can take control of a real production machine, which will allow the administrator to observe the means of compromising the attackers, to guard against new attacks, and give him more time to react.

Honeypots are very flexible and exist in different forms. Most works classify honeypots within two ways: the first classification consists in classifying honeypots according to the interactions they allow, and a second classification categorizes them according to their usefulness [5, 24]. In our classification, we will be interested in presenting the advantages and disadvantages of each class of Honeypots and giving some examples.

Low-interaction Honeypots [25] are limited to the degree of emulation offered by the honeypot; hence, the interaction between the attacker and honeypot system is low. These honeypots offer few privileges to the intruder who will have a limited scope. For example, the Honeypot can emulate an FTP (File Transfer Protocol) service on port 21, but emulate only the login command or one other command. The advantage of low-interaction Honeypots is their simplicity; they are easier to implement and manage and pose little risk since the attacker is limited. But, this type records limited information and can only monitor known activities, and emulated services cannot do a lot. It is, therefore, easier for an attacker to detect a low-interaction honeypot. Tools such as KFSENSOR (refer to the next section) are examples of low-interaction honeypots. Medium-interaction honeypots [26] are also discussed in some works; this type of honeypots allows the intruder to get little more access than low-interaction honeypots, offers better simulation services, and enables logging of more advanced attacks, but it requires more time to implement and certain level of expertise. High-interaction honeypots [27] involve the use of real operating systems and real applications, and this is not about emulation; we provide the attacker with something real. The risks are numerous since our machine is intended to be compromised [24]. One of the advantages of this solution is that it is possible to obtain a lot of information because attackers have access to real systems. Hence, high-interaction honeypots allow examining all behaviors and methods, as well as the tools used by the attacker, and take knowledge if the attack is new or not. An example of high-interaction honeypots is the honeynet. On the other hand, honeypots classification is performed according to the specification of utilization; one type is production oriented, the other is research oriented. Honeypots are used in production to protect a company (prevention and detection) and help to find solutions against attacks. Low-interacting honeypots are often used for production [24]. They can be used for research, to collect information about attackers, tools, and motivations, in order to predict future attacks or to provide judicial elements.

4.1. Honeypot Deployment. Deployment of honeypots depends on whether the decoy system is intended to monitor external or internal attacks to the organization's network; hence, it can be installed in front of the firewall, in a demilitarized zone (DMZ), or behind the firewall [5].

The main advantage of choosing the first position (Figure 3) is that there is no change to be made to the firewall filtering rules which protects the internal network, and the location does not introduce new risks for machines on the internal network, but it does not detect attacks carried out from inside the network since, generally, the outgoing flows are blocked by the firewall. The second position is to place the honeypot in a demilitarized zone, and the advantage of the DMZ is that it provides public servers on the Internet isolated from the internal network. This DMZ can be used for production servers (Figure 4) or dedicated to honeypot, while the firewall only allows incoming flows to pass through to the DMZ for available services. This makes it possible to analyze only attempted attacks for the services in question. For the last position (Figure 5) and in case the decoy system is used to detect external attacks, it can induce greater risks of vulnerabilities since once compromised, the decoy system can be used by the attacker to launch other attacks on the internal network. But, this position allows detecting attacks from internal users of the organization to internal services or detecting a bad firewall configuration [5].

4.2. KFSensor. This tool is very simple to set up and does not require a lot of resources systems, and it is based on intrusion detection systems [28]. KFSensor is a server listening

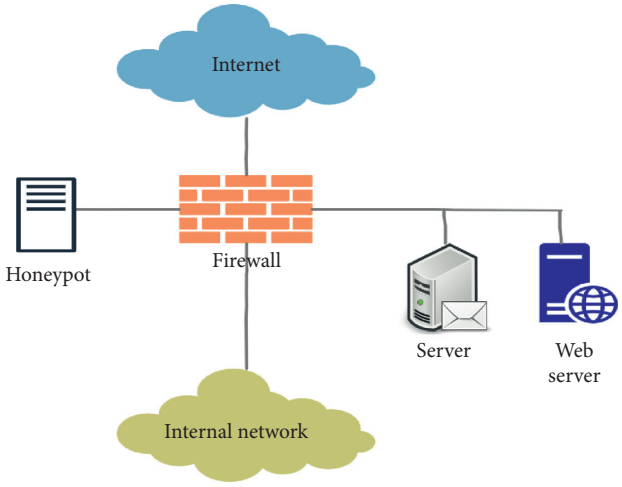

Figure 3: Honeypot deployed independently.

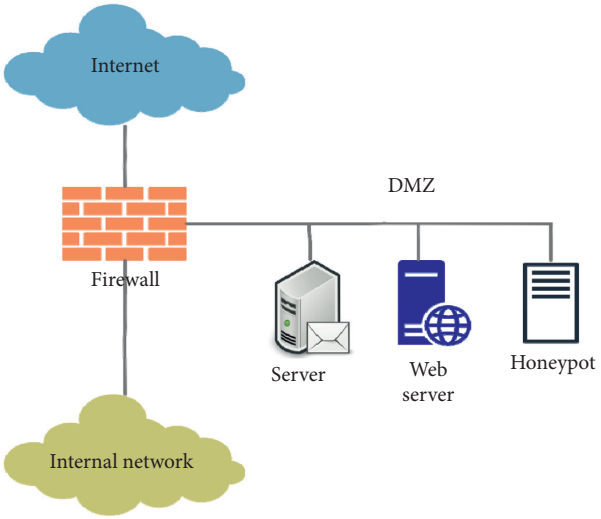

Figure 4: Honeypot deployed in a DMZ.

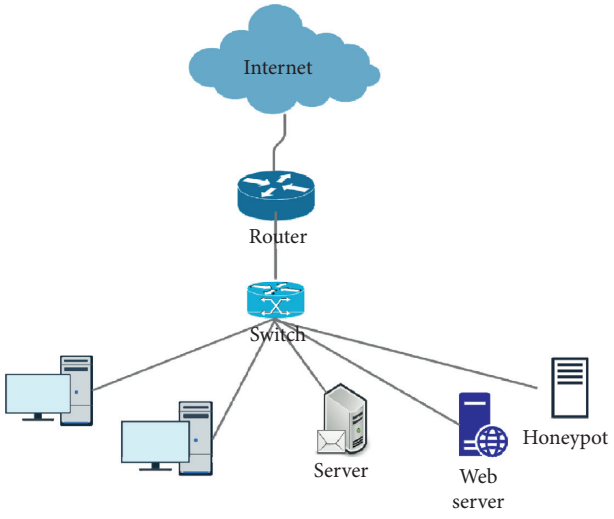

Figure 5: Honeypot deployed in the LAN.

for connections to the network input and monitoring the different port numbers [28], which has a standby user interface (monitoring user interface).

KFSensor professional monitor attacks on every TCP and UDP port, as well as detects ICMP [29] or ping messages. When KFSensor passes new connection information and the data received from visitors to the signature engine, for comparing its data with each signature rule stored in its signature base, and if a match is found, the signature ID is stored with the event in the event log.

4.3. Netfacade. Netfacade allows to simulate a network of vulnerable hosts (honeynet) using the redirect unused IP addresses on a range of addresses to vulnerable services [28].

Netfacade provides a medium level of interaction, it is able to emulate services such as FTP, SendMail, IMAP, HTTP, NFS, and SSH and various operating systems such as Linux, Solaris, and Windows NT.

4.4. Specter. Specter can emulate some network services, as well as some systems of the most common operations. It is classified between as a medium-interaction honeypot and emulates various services such as FTP, POP3, and HTTP [30]. Specter is usable on operating systems based on Windows NT, 2000 or XP [28].

4.5. CurrPorts. CurrPort displays the list of all opened TCP/IP and UDP ports on the local network, with the process that opened the port, and allows to close unwanted TCP connections, kill the process that opened the ports, save the TCP/UDP ports information to HTML file, XML file, or to tab-delimited text file.

5. Smart Agent for Cyber Security Attacks Prevention and a Prediction-Based Machine Learning and Honeypot System

Companies have invested a great deal on time and money in manual networks reconfiguration, in order to protect information systems from infiltration. It is well known that the locks break and the keys can be copied; therefore, it is an illusion to think that a lock and a key represent perfect security. So, the real challenge in terms of cyber security is to accept the probability of an imminent attack and to understand what is really going on within complex information systems. Traditional security tools such as IDS, Firewalls, and IPS can protect systems against simple attacks that use the same tools and tactics repeatedly. They are implemented independently; hence, there is no contact between them to block intrusion detected in an IDS by the firewall [8], for example, they represent a passive solution when it is about 0-day attacks.

The proposed solution is based on honeypot systems and ML techniques combination, as tools for gathering information, analysis, and threat predictions, in order to ensure the security of companies' networks. The implementation of honeypots depends on the services offered to customers by the production company's servers. In Figure 6,

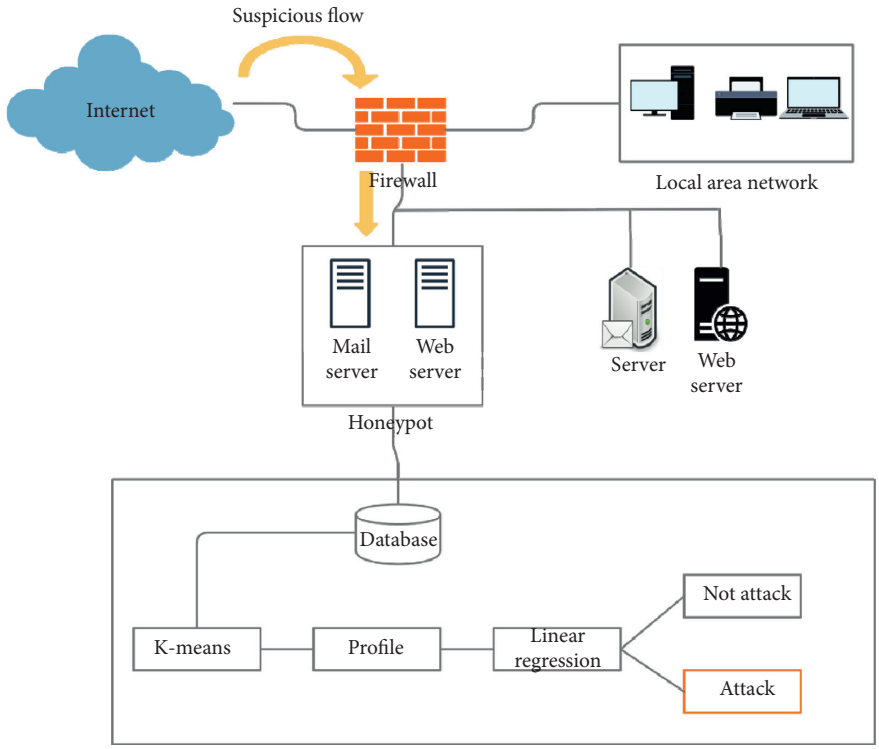

Figure 6: Suspicious flow path.

it is shown that a company deploys tree servers, a web server for its site web hosting, an FTP server for files transfer, and a mail server. In order to monitor suspicious profiles through the services provided by this company, in the same honeypot server, three virtual machines can be implemented, and each one is configured to emulate one of the previous services with the secure shell (SSH) module to allow remote access to any of the virtual machines.

This implementation allows detecting suspicious profile patterns on services, for predicting attacker profiles based on machine learning analysis. Decisions will allow reconfiguring the security policy (e.g., Firewall) in order to block the attackers. Hence, the firewall should be configured in a way to redirect suspicious flow to the honeypots in order to gather information about the application and the transport layers. The collected data will be submitted to a combination of algorithms. A clustering technique will be used to cluster the data into homogenous classes and create the user profile. The profile will, then, be classified into an attacker or nonattacker profile based on another classification algorithm.

In each interaction with a suspicious flow, the honeypot stores the collected information vector V_i^{user}(IP, logging, and packet length) in a database, in order to construct a user profile. Based on K-means algorithm, the data will be clustered into homogenous classes for creating a profile. Linear regression will be used for modeling each class, to give more significant and homogenous presentation to the data (Figures 7 and 8).

Let there be a vector V_i^{user} with n elements of information collected by the honeypot system; the V_i^{user} data will be adapted into qualitative N_{cd} and quantitative data N_{cd}. The qualitative data such as the IP address will be stocked directly in the user profile, while the quantitative quantities are clustered into homogenous groups using K-means $C_j[K]$. Each produced class will be represented by a linear model $CL_j[f]$ using linear regression. The qualitative data and the proposed quantitative modeling form the suspicious profile, the decision (decision stage) is made by projecting the suspicious profile on attacker profiles using distance metric between the trained models (learning stage) and the suspicious models.

In the learning phase (Algorithm 1), three parameters are required: the number of clusters, the initialization of centroids, and parameters of the linear function. The higher the precision of initialization and calculation of these parameters, the higher the precision of the fitting model.

The algorithm returns the following information (Algorithm 2): profile creation and classification based on the attackers model. In the Euclidean sense, $HCL_j[f]$ is the closest vector to $HCL_j[f]$, the decision is made based on projections of the new profile on the hackers profiles.

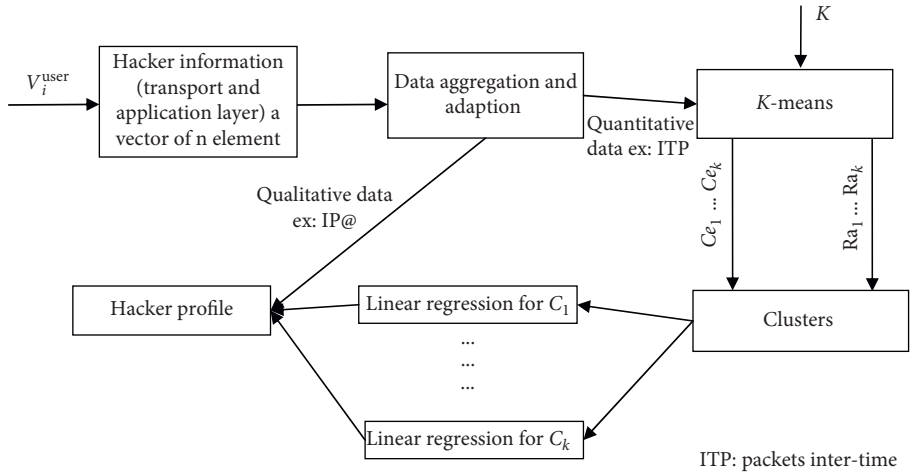

Figure 7: Learning stage.

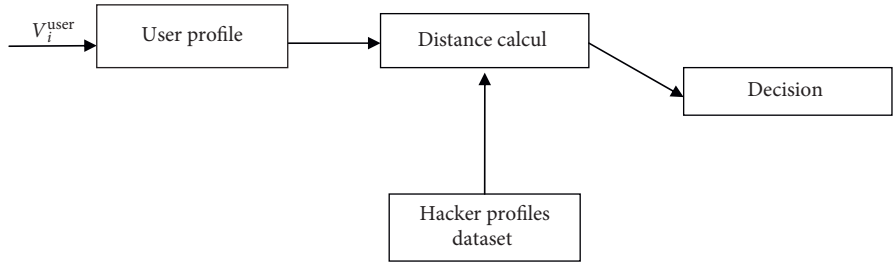

Figure 8: Decision stage.

INPUT
K//number of clusters
Vpn = ($\mathbf{V}_1^{\mathbf{user}}$, $\mathbf{V}_2^{\mathbf{user}}$,...,$\mathbf{V}_n^{\mathbf{user}}$)//Hacker information array (vector of vectors)
START
$\mathbf{M}_{\mathbf{ab}}$ = **Q1A**(**Vpn**)//qualitative adaptation
$\mathbf{N}_{\mathrm{cd}}$ = **Q2A**(**Vpn**)//quantitative adaptation
f = **c**−1//linear regression order = space dimension−1
for $j = 1$; $i < c$; j++//For each row of N_{cd}
(Cj [K], Rj [K]) = K-means ($\mathbf{N}_{id}$, K)//creation of clusters center et radius
for $j = 1$; $i < c$; j++
CLj[f] = Linear Regression (Cj [K], Rj [K], $\mathbf{N}_{id}$)//Linear Regression of every clusters
OUTPUT//Hacker profile
$\mathbf{M}_{\mathbf{ab}}$//qualitative adaptation
CLc[f]//linear regression coefficients

Algorithm 1: Learning.

6. Conclusions

In this paper, we have presented an introduction of machine learning and honeypot as solutions for cyber security. We also presented an efficient algorithm which returns two important information, one for profile creation and the other for classifying this profile. In fact, the specific solution based on honeypot and the combination of machine learning algorithms forms a solid modeling and predictive system for suspicious profile recognition and classification. Hence, it represents an integrated efficient system for cyber security to deal with future and 0-day attacks. Our next work

INPUT
K//number of clusters
$\mathbf{Vpn} = (\mathbf{V_1}^{\mathbf{user}}, \mathbf{V_2}^{\mathbf{user}}....,\mathbf{V_n}^{\mathbf{user}})$//user information array (vector of vectors)
$(\mathbf{HM_{ab}}, \mathbf{HCLc[f]})$ Hackers profiles
START
$\mathbf{UM_{ab}} = \mathbf{Q1A}(\mathbf{Vpn})$//qualitative adaptation
$\mathbf{N}_{cd} = \mathbf{Q2A}(\mathbf{Vpn})$//quantitative adaptation
$\mathbf{f} = \mathbf{c}-1$//linear regression order = space dimension−1
for $j = 1$; $i < c$; j++//For each row of N_{cd}
$(Cj\ [K], Rj\ [K]) = K$-means $(\mathbf{N}_{id}, K)$//creation of clusters center et radius
for $j = 1$; $i < c$; j++
$\mathbf{UCLj[f]}$ = Linear Regression $(Cj\ [K], Rj\ [K], \mathbf{N}_{id})$//Linear Regression of every clusters
OUTPUT
Distance($HCLc[f]$, $UCLj[f]$)
Isequal(HM_{ab}, UM_{ab})

Algorithm 2: Decision.

will be devoted to implement the smart agent in a real environment in order to evaluate and test its performances.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] GData, *Malware Numbers*, http://www.gdatasoftware.com, 2017.

[2] P. Owezarski, "Unsupervised classification and characterization of honeypot attacks," in *Proceedings of 10th International Conference on Network and Service Management (CNSM) and Workshop*, pp. 10–18, Rio de Janeiro, Brazil, November 2014.

[3] S. Dowling, M. Schukat, and E. Barrett, "Improving adaptive honeypot functionality with efficient reinforcement learning parameters for automated malware," *Journal of Cyber Security Technology*, vol. 2, no. 2, pp. 75–91, 2018.

[4] I. M. M. Matin and B. Rahardjo, "Malware detection using honeypot and machine learning," in *Proceedings of 2019 7th International Conference on Cyber and IT Service Management (CITSM)*, Bandung Institute of Technology, Bandung, Indonesia, pp. 1–4, November 2019.

[5] L. Spitzner, *Honeypots: Tracking Hackers*, Addison-Wesley, Clemson, SC, USA, 2003.

[6] T. Luo, Z. Xu, X. Jin, Y. Jia, and X. Ouyang, "Iotcandyjar: towards an intelligent-interaction honeypot for iot devices," in *Proceedings of the Black Hat*, Las Vegas, NV, USA, 2017.

[7] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots+ machine learning," in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR'10*, pp. 435–442, The ACM Digital Library, New York; NY, USA, July 2010.

[8] G. Feng, C. Zhang, and Q. Zhang, *A Design of Linkage Security Defense System Based on Honeypot: Trustworthy Computing and Services*, Springer, Berlin, Heidelberg, Germany, 2014.

[9] L.-j. Li and H. Peng, "A defense model study based on IDS and firewall linkage," in *Proceedings of 2010 International Conference of Information Science and Management Engineering*, pp. 91–94, IEEE, Xi'an, China, August 2010.

[10] Y. LeCun, "L'apprentissage profond, une révolution en intelligence artificielle," *La lettre du Collège de France*, vol. 41, p. 13, 2016.

[11] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, p. 67, 2016.

[12] Z. Ullah, F. Al-Turjman, L. Mostarda, and R. Gagliardi, "Applications of artificial intelligence and machine learning in smart cities," *Computer Communications*, vol. 154, pp. 313–323, 2020.

[13] J. H. Lee, J. Shin, and M. J. Realff, "Machine learning: overview of the recent progresses and implications for the process systems engineering field," *Computers & Chemical Engineering*, vol. 114, pp. 111–121, 2018.

[14] G. A. Seber and A. J. Lee, *Linear Regression Analysis*, John Wiley & Sons, Hoboken, NJ, USA, 2012.

[15] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112, 2010.

[16] M. Schleich, D. Olteanu, and R. Ciucanu, "Learning linear regression models over factorized joins," in *Proceedings of SIGMOD '16: Proceedings of the 2016 International Conference on Management of Data*, ACM, San Francisco, CA, USA, pp. 3–18, July 2016.

[17] T. M. Kodinariya and P. R. Makwana, "Review on determining number of cluster in K-means clustering," *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[18] M. Eddabbah, M. Moussaoui, and Y. Laaziz, "A smart architecture design for health remote monitoring systems and heterogeneous wireless sensor network technologies: a machine learning breathlessness prediction prototype," *International Journal of Intelligent Enterprise*, vol. 6, no. 2–4, pp. 293–310, 2019.

[19] S. Ray and R. H. Turi, "Determination of number of clusters in k-means clustering and application in colour image

segmentation," in *Proceedings of 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99)*, Narosa Publishing House, New Delhi India, pp. 137–143, December 1999.

[20] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[21] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[22] B. Hssina, A. Merbouha, H. Ezzikouri, and M. Erritali, "A comparative study of decision tree ID3 and C4. 5," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 2, pp. 13–19, 2014.

[23] H. J. Wang, C. Guo, D. R. Simon, and A. Zugenmaier, "Shield: vulnerability-driven network filters for preventing known vulnerability exploits," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols*, pp. 193–204, The ACM Digital Library, New York; NY, USA, August 2004.

[24] K. Sadasivam, B. Samudrala, and T. A. Yang, "Design of network security projects using honeypots," *Journal of Computing Sciences in Colleges*, vol. 20, no. 4, pp. 282–293, 2005.

[25] P. S. Negi, A. Garg, and R. Lal, "Intrusion detection and prevention using honeypot network for cloud security," in *Proceedings of 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 129–132, IEEE, Noida, India, Janauary 2020.

[26] D. Fraunholz, F. Pohl, and H. D. Schotten, "Towards basic design principles for high-and medium-interaction honeypots," in *Proceedings of 16th European Conference on Cyber Warfare and Security*, p. 120, Dublin, Ireland, June 2017.

[27] H. Wang and B. Wu, "SDN-based hybrid honeypot for attack capture," in *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1602–1606, Chengdu, China, March 2019.

[28] F. Pouget and M. Dacier, "White paper: honeypot, honeynet: a comparative survey," Rep. RR-03-082, Eurecom, Biot, France, 2003.

[29] N. Naik and P. Jenkins, "A fuzzy approach for detecting and defending against spoofing attacks on low interaction honeypots," in *Proceedings of 2018 21st International Conference on Information Fusion (FUSION)*, pp. 904–910, IEEE, Cambridge, CA, UK, July 2018.

[30] B. Nagpal, N. Singh, N. Chauhan, and P. Sharma, "Catch: comparison and analysis of tools covering honeypots," in *Proceedings of 2015 International Conference on Advances in Computer Engineering and Applicationspp*, pp. 783–786, IEEE, Ghaziabad, India, March 2015.