ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20210523121614
Preliminary communication

Real-Time Cyber Attack Detection Over HoneyPi Using Machine Learning

Birkan ALHAN, Serkan GÖNEN, Gökçe KARACAYILMAZ, Mehmet Ali BARIŞKAN, Ercan Nurcan YILMAZ*

Abstract: The rapid transition of all areas of our lives to the digital environment has kept people away from their intertwined social lives and made them dependent on the isolated cyber environment. This dependency has led to increased cyber threats and, subsequently, cyber-attacks nationally or internationally. Due to the high cost of cybersecurity systems and the expert nature of these systems' management, the cybersecurity component has been mostly ignored, especially in small and medium-sized organizations. In this context, a holistic cybersecurity architecture is designed in which fully open source and free software and hardware-based Raspberry Pi devices with low-cost embedded operating systems are used as a honeypot. In addition, the architectural structure has an integrated, flexible, and easily configurable end-to-end security approach. It is suitable for different platforms by creating end-user screens with personalized software for network security guards and system administrators.

Keywords: Artificial Intelligence; Cyber Security; Honeypot; Internet of Things; LSTM; Naive Bayes

1 INTRODUCTION

Advances in the developing science and technology field have caused our lives to move to the digital environment. With this process, all our activities, from health to education, from banking to shopping, including our personal data, have become carried out through information systems, especially the Internet. Especially with the current Covid-19 pandemic, all our lives, including our social life, have become dependent on the cyber environment. Therefore, together with factors that increase our quality of life, such as efficiency, productivity, and speed, the importance of cyber security stands out as much as other factors. Compared to the investments we make in software and hardware that positively increase the functioning of our systems, such as efficiency, organizations are very reluctant and stubborn in investing in cyber security systems because they think that cyber security imposes an additional burden on the system, requires high costs, and is a method out of our habits.

At the end of the study, an integrated, flexible, and easily configurable end-to-end security approach consisting entirely of open source code components has been designed. In the architecture realized with a holistic approach, cyber security software and hardware are used to achieve privacy, integrity, and accessibility, which are the three pillars of cyber security. In this process, end-user screens (web, desktop, and mobile) have been developed and made available for different platforms with personalized software for network security guards and system admins.

In this context, network security has been ensured with the architecture implemented in the study. A design was created to track the attackers by configuring a honeypot, which works like a real server, for security purposes against possible attacks. Artificial intelligence integration has been provided in order to label and interpret these honeypot accesses as an attack or not. These interpreted data are shown on the end-user screens in real-time.

A fully open-source, very low-cost, holistic intrusion detection and warning system has been designed for small and medium-sized organizations with the project implemented. In this context, it has been evaluated that security systems will provide significant contributions as it is an architecture that can be preferred by institutions that cannot invest in security systems. The components in the topology used in the study are brand-model-independent. They are designed with the flexibility to be built on other brand model systems.

In the ongoing parts of the study, related works are mentioned in the 2nd section. The Testbed designed within the scope of the study is introduced in the 3rd section respectively. In the 4th section, the HoneyPi system is discussed. Recording the traffic coming to honeypot over the network in the cloud environment is specified. Artificial Intelligence analysis of the network traffic coming to HoneyPi, which is the focus of the study, is examined in the 5th section. In the 6th section, the process of transmitting data via a secure environment (encryption applications), which Artificial Intelligence evaluates as harmful, is examined. The study is completed with the 7th section, the conclusion section.

2 RELATED WORKS

Research on honeypots, developed for the first time in the 80s, has grown tremendously in the last decade. The use of honeypots to solve different security problems within the scope of cyber security analysis has become widespread [1, 2]. Since the 1990's honeypots for catching and canalizing attackers to an enclosed part of cyber systems.[3] Although one of the main usage applications of honeypots is intrusion detection [4-7], there are not enough studies on honeypots' methods or models. So far, all honeypot systems are that any interaction with a honeypot is most likely an unauthorized or malicious activity. The increasing number of cybersecurity attacks and the level of complexity that is constantly improving have created the need for in-depth investigation of attacks. Studies reveal a significant change in the direction of attacks called targeted attacks [8]. Compared to traditional malware that spreads independently of the target, these attacks, also known as APT (Advanced Persistent Threats) [9], contain specially created exploit vectors and, in some cases, require direct human interaction with extensive systems. Thus, it is vital to analyze the traffic to honeypots and detect malicious packets at the earliest time. Among the events that occur in honeypots are purposeful and malicious attacks, malware spread, and some are accidental harmless activities of legitimate users.

In this context, the important thing is to correctly analyze harmful and harmless traffic and inform security experts by generating warnings only for harmful ones. In recent years, many honeypot-based automatic signature creation systems have been installed. However, the most significant disadvantage of these systems is false positives [10]. Due to the continuous false alarms that occur after a particular use, there is a loss of attention that may occur to the experts.

Machine learning algorithm and SSH (secure shell) platform of Kippohoneypot were used by Pauna et al. [11]. Fraunholz et al. proposed a self-adaptive honeypot system using game theory as the basis for deciding actions to be taken while interacting with attackers. In the system, by using a modified kernel module (LKM-Loadable Kernel Module) on the Linux platform, interaction with the attacker over the SSH service was provided [12]. In another study by Pauna et al., a self-adaptive SSH honeypot system using a Deep Q-Learning algorithm was proposed to decide how to interact with attackers [13]. El Kamel et al. presented research on the strategic choice of various honeypot configurations to accommodate network attacks [14]. Tian et al. analyzed the interactions between attackers and security professionals in the Smart Grid environment. The study showed the Bayesian Nash equilibrium in the honeypot game that derives the most appropriate strategies for both parties [15]. However, these studies' most significant deficiency is that they are based on the monitoring and analysis of some opened ports. This situation does not reflect the working structure of natural systems.

In this study, honeypot deployed in a holistic security architecture designed as completely open-source has been configured as depicted in Fig. 1 to detect intruders and malicious information systems on the network. All honeypot configurations in the architecture are configured by considering the system settings used by default in a real system. The basic approach of our system is to use an artificial intelligence algorithm simultaneously to collect data from a honeypot and detect intruders and/or malicious packets. In addition, the data is transferred to the database in the cloud environment, where attackers cannot access it so that the attackers do not detect honeypot and damage the collected data. In this way, even if the artificial intelligence algorithm is considered harmless, the data for post-event analysis is stored in the cloud environment within limits determined by the law. In addition, databases in which critical corporate information is stored within the network are deployed in a separate DMZ (DemilitarizedZone). Access to this DMZ is only possible with a username and password authorized through a VPN. As a result, unauthorized access to the corporate network from outside is restricted.

Our contributions in this study are as follows:

- A completely open-source and low-cost integrated cyber security architecture has been designed and implemented.
- The need for detection models for honeypots has been addressed.
- Algorithm types of Naive Bayes, LSTM, CNN, and KNN detection models for artificial intelligence in the holistic security architecture where honeypot is located have been examined.
- Real attacks have been applied to the system to analyze the detection model.
- Components in the topology are independent of brand and model. Thanks to its flexible structure, it can be adapted to other brands and models.

3 TESTBED

Open-source software has been used in the network architecture designed for the test environment created in the study, as depicted in Fig. 1. In hardware security systems, both the cost has been reduced by using embedded systems, and the system has been enabled to work faster. Each package that logs into the system comes to the pfSense firewall. This open-source code is used as a firewall, free of charge, and can be configured according to user demands. Depending on whether there is a request to be directed to the VPN connection in the content of each incoming packet, the packet is directed to the corporate internal network or HoneyPi. Packets routed to the internal network go to subnets that are logically separated from each other via different switching devices by checking their authorization status. For example, an authorized user working in the finance department who wants to connect to the corporate network from outside the institution will be able to access the database where financial transactions are made by entering her /his authorized username and password via VPN (OpenVPN). However, since the same user is not authorized, he/she will not be able to access the database containing personnel information. In this way, a precaution will be taken against attacks on information disclosure such as Man-in-the-middle, eavesdropping.

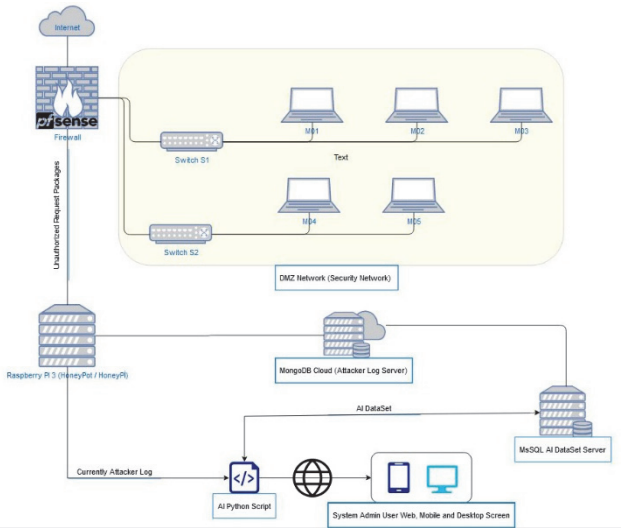

Figure 1 Network topology

Traffic packages belonging to other people who want to access the institution from outside but do not have a corporate username and password will be directed to HoneyPi via firewall. Packages coming on HoneyPi will be processed simultaneously in an artificial intelligence algorithm. Their logs will be transferred to Cloud DB to be stored on MongoDB with the created script. The DMZ is a physical or logical subnet that contains an organization's external services. It exposes these services to a larger insecure network (usually the Internet). The purpose of DMZ is to add an additional layer of security to an

organization's Local Area Network (LAN). An outside attacker only has access to equipment inside the DMZ rather than any other part of the network. In this way, complete protection is provided within the DMZ.

Requests for access to the unauthorized system that do not have an authorized user name and password for accessing the corporate network have been directed to HoneyPi via firewall. In this way, packages coming on HoneyPi will be labelled as packets without attack or attack, thanks to the algorithm that will be explained in detail in the artificial intelligence section and recorded on the MsSQL database. The artificial intelligence scripts use the tagged datasets on the MsSQL database to check whether the log records occurring instantaneously are attacked or not. Then, by writing to the MsSQL database with Token-based AES crypto web services, information is transferred to the end-user (system admin, network name, etc.) on mobile, desktop, and web screens using the same service ends.

Network traffic packets labelled as attacks will be sent as a warning to security staff. Network traffic packets that are not intrusive will be recorded within the legal limits, which is 16 MB, on MongoDB for post-event investigation purposes in case of any false-negative situation. The logs of all packages coming to HoneyPi are instantly transferred to AI Python Script. In this way, detecting the existence of HoneyPi and changing the logs in it by the attackers will be prevented.

The flow diagram of the operations performed within the scope of the topology is depicted in Fig. 2.

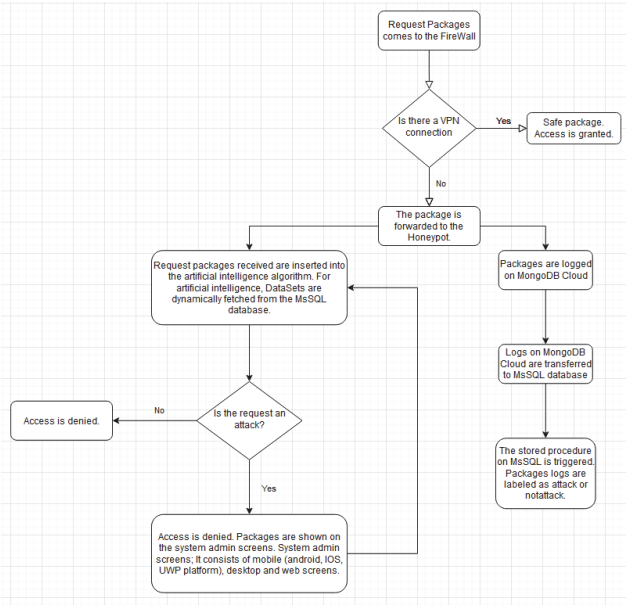

Figure 2 Attack Analysis Flowchart

4 SECURE NETWORK APPROACH

In the network architecture realized in the study, a DMZ has been used in the internal network. This DMZ has been created in a structure that is accessed via VPN and has been divided into subnets according to their authorization levels. DMZ is not a physical structure but a logical approach. In this context, DMZ installation is created as a result of the configuration of firewalls by integrating them with each other in various ways. A properly configured firewall (in cases where 0-day vulnerabilities and DoS (Denial of Service) attacks are ignored) is expected to provide nearly 100 percent security and protect the DMZ.

For this reason, it is called the DMZ as a non-military, unarmed zone. This study used the DMZ installation approach with a single-layer firewall. In this way, a secure internal network has been created, and the security of critical devices has been ensured.

An open-source pfSense firewall has been installed on the network. While configuring the firewall, only network packets coming through VPN are allowed. It is aimed for employees to connect with OpenVPN and to tunnel to the DMZ network behind the firewall. External access to the company network is not allowed. However, company employees are allowed to access servers located in the DMZ by connecting via a VPN. In this context, if the attacker does not connect via VPN and scans the network, it will encounter the firewall device and subsequently the honeypot device. Therefore, it is nearly impossible to reach the DMZ behind the firewall.

As a security measure against the attacker's scanning and operating system analysis, the operating system information has been manipulated as depicted in Fig. 3, Fig. 4 thus preventing the attacker's access to the basic operating system information. As seen in Fig. 3, the operating system Linux appears to be between 4.15 and 5.6 as a result of the ZenMAP scan.

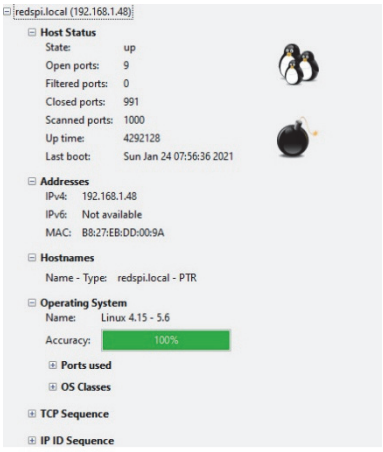

Figure 3 Operating System Information Manipulation (Linux)

However, in Fig. 4, the operating system was changed to Playstation 5 using the IP Personality approach.The main purpose of this change is to keep the attackers away from the real system by hiding the honeypot system. The reason for replacing it with Playstation in this study is that it can be easily converted to any system. In this way, a system design has been created that attracted the attention of attackers. The second image is made in the form of a game console to show that it can be changed. The device's name appears to be a different server by changing the device's fingerprint number. However, in the system prototype (live environments of organizations), the operating system name has been changed, and it has been designed as Windows Server 2008, which has many vulnerabilities. In this way, if the attacker wants to scan the network devices, they will see the HoneyPi device as a standard server and try to access it.

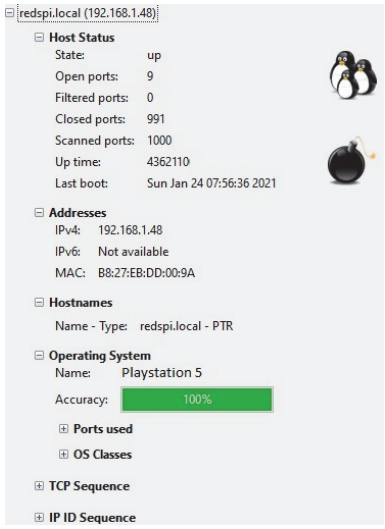

Figure 4 Operating System Information Manipulation (Playstation)

5 HONEYPOT AND LOG SYSTEM

HoneyPi has been used as a honeypot to protect the critical systems in the study from attackers. For the designed measure to work effectively, the honeypot has to be designed as a real system and behave like a real system. An effective and functionally configured honeypot on the system provides important evidence for legal processes after the attack phase. It will keep the attacker away from the real network and collect information about the attacker without revealing it to the attacker. In this context, functional honeypots play an important role in the network.

To embody the scenario applied in the study, when an attacker who wants to infiltrate a system detects an IP address that is not protected by any security system, tries to infiltrate this IP address instead of the addresses behind other security systems. When she/he infiltrates the IP address with deliberately left vulnerable, she/he finds system packets (dummy data) especially stored inside create a real system impression for the attacker. In the meantime, collecting critical information (mac address, IP address, location, computer ID, etc.) about the attacker as soon as the attacker performs a port scan as the first step will be good evidence for detecting and blocking the legal process and similar attack packets afterward. In this context, honeypots are examples that embody the concept of "phishing attacks (spear, whaling, etc.)" of the new popular technology, which is important in the net.

Unauthorized connections (not accessing via VPN) on the firewall are transferred to the Honeypot device on the network. As a honeypot, an open-source and free HoneyPi honeypot has been run on Raspberry Pi 3 in order to use the advantages of embedded systems, thus benefiting from its saving costs. All requests received on HoneyPi have been logged and transferred to be stored in the cloud database. In addition, HoneyPi is designed like a real server; it is not understood as a Honeypot. In this context, the ports of the Honeypot device (webmail, FTP (File Transfer Protocol), TELNET, SSH, etc.) required for communication like a real server have been left open. As indicated in Fig. 5 on HoneyPi, it is aimed to leave FTP (21), SSH (22), SMTP (25), HTTP (80), POP3 (110), HTTPS (443), RPC (135), VNC (5009) ports open to provide a realistic server view for HoneyPi.

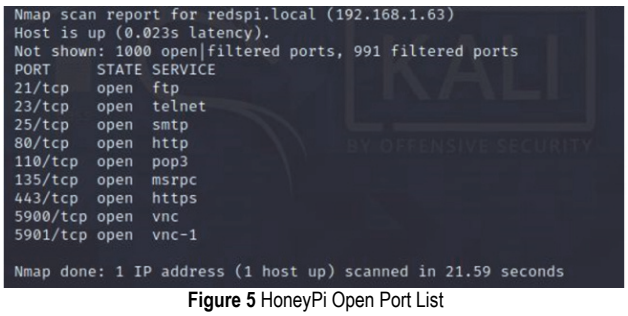

Figure 5 HoneyPi Open Port List

Incoming requests to ports left open on HoneyPi are stored in MongoDB Cloud. At this stage, pseudo-static packets are sent to respond to the requests sent by the attacker. As depicted in Fig. 6, the data logged on the MongoDB cloud contain the IP Address, Port Number, Service Type, Access Time, and information from which port of the client/attacker the request packet came out.

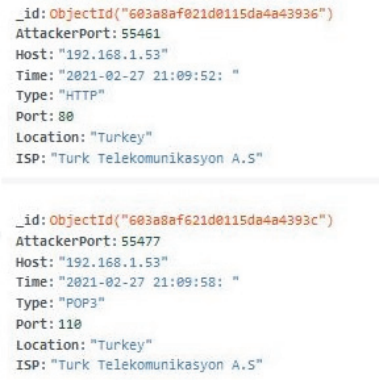

Figure 6 Cloud Logs of the MongoDB

IP Geolocation information and Internet provider company information has also been kept for the attacker's IP address spoofing. As shown in Fig. 7, the attacks attempted to be carried out using the BOGON network have been prevented by this means. The configuration for blocking traffic packets from BOGON networks has been defined on the pfSense firewall.

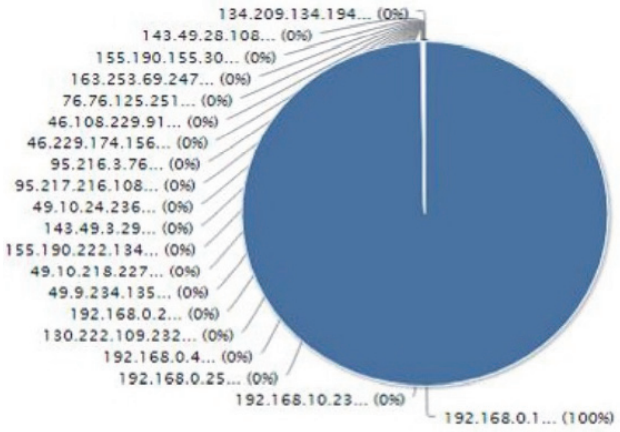

Figure 7 Attack Carried out from BOGON Network

6 ARTIFICIAL INTELLIGENCE MODULE

Artificial intelligence has been used to analyze packets that do not come via VPN and are routed to a honeypot. In this context, datasets for artificial intelligence have been customized as described in Fig. 8. In the datasets prepared

in this process, 70% of the logs tagged on the MsSQL database have been used as training sets and 30% as control sets. During the creation of Datasets, the log records received over MongoDB are transferred to an MsSQL Server located on the cloud with a script running continuously on HoneyPi. The transfer is labelled as an attack (1) or not attack (0) with a Stored procedure run on MSSQL. For this labelling, the calculation is made on the data in minutes as time definition. When the dataset is transferred to MSSQL, the minute difference between the log dates is considered the end time in this calculation. As indicated in Fig. 9, a time definition (DIFF) column appears on a minute basis. Request logs made at the same time interval with the script running on HoneyPi are considered attacks.

	LogID	Host	AttackerPort	AttackPort	AttackType	LogTime
1	1	192.168.1.53	53272	23	TELNET	2021-02-27 20:36:11.000
2	2	192.168.1.53	53275	21	FTP	2021-02-27 20:36:21.000
3	3	192.168.1.53	55013	5901	VNC	2021-02-27 20:36:28.000
4	4	192.168.1.53	55459	443	SSL	2021-02-27 21:09:48.000
5	5	192.168.1.53	55461	80	HTTP	2021-02-27 21:09:52.000
6	6	192.168.1.53	55465	21	FTP	2021-02-27 21:09:54.000
7	7	192.168.1.53	55473	135	SSL	2021-02-27 21:09:56.000
8	8	192.168.1.53	55477	110	POP3	2021-02-27 21:09:58.000
9	9	192.168.1.53	55502	25	SMTP	2021-02-27 21:10:00.000
10	10	192.168.1.53	55504	443	SSL	2021-02-27 21:10:05.000
11	11	192.168.1.53	55854	80	HTTP	2021-02-27 21:10:07.000
12	12	192.168.1.53	56541	5901	VNC	2021-02-27 21:10:09.000
13	13	192.168.1.53	55893	80	HTTP	2021-02-27 21:10:11.000
14	14	192.168.1.53	56115	80	HTTP	2021-02-27 21:10:13.000
15	15	192.168.1.53	56144	80	HTTP	2021-02-27 21:10:15.000
16	16	192.168.1.53	56275	80	HTTP	2021-02-27 21:10:19.000
17	17	192.168.1.53	56878	80	HTTP	2021-02-27 21:10:20.000
18	18	192.168.1.53	57308	80	HTTP	2021-02-27 21:10:22.000
19	19	192.168.1.53	57476	443	SSL	2021-02-27 21:14:03.000

Figure 8 MsSQL records streaming over MongoDB

	Host	AttackerPort	AttackPort	AttackType	LogTime	DIFF
1	192.168.1.53	53272	23	TELNET	2021-02-27 20:36:11.000	15893
2	192.168.1.53	53275	21	FTP	2021-02-27 20:36:21.000	15893
3	192.168.1.53	55013	5901	VNC	2021-02-27 20:36:28.000	15893
4	192.168.1.53	55459	443	SSL	2021-02-27 21:09:48.000	15860
5	192.168.1.53	55461	80	HTTP	2021-02-27 21:09:52.000	15860
6	192.168.1.53	55465	21	FTP	2021-02-27 21:09:54.000	15860
7	192.168.1.53	55473	135	SSL	2021-02-27 21:09:56.000	15860
8	192.168.1.53	55477	110	POP3	2021-02-27 21:09:58.000	15860
9	192.168.1.53	55502	25	SMTP	2021-02-27 21:10:00.000	15859
10	192.168.1.53	55504	443	SSL	2021-02-27 21:10:05.000	15859
11	192.168.1.53	55854	80	HTTP	2021-02-27 21:10:07.000	15859
12	192.168.1.53	56541	5901	VNC	2021-02-27 21:10:09.000	15859
13	192.168.1.53	55893	80	HTTP	2021-02-27 21:10:11.000	15859
14	192.168.1.53	56115	80	HTTP	2021-02-27 21:10:13.000	15859
15	192.168.1.53	56144	80	HTTP	2021-02-27 21:10:15.000	15859

Figure 9 Time Definition (DIFF) View

Fig. 10 contains the definition of Stored procedure. The stored procedure image where the view image filtered by minutes is processed.

After running the Stored procedure, the dataset with time definition is grouped. Logs that have multiple packet access to different ports in a fairly short period of time and the same port number of the client device from which the request packets arrive are labelled as an attack. On the other hand, requests from a single port at the same time interval or request packets that originate from different ports of the client devices at the same time interval are not considered attacks. For example, based on the basics of the project design, the company employee made the VPN connection, but after a certain time, it dropped from the system due to the timeout. If the attacker wants to access again in this process, the firewall will log as unauthorized access. Considering that multiple applications on the employee's computer are trying to access the same server, there will be more than one log for the same service in the log records. However, since the port numbers of the client (working) computer will be different, the logs in this state are not attacks. These requests also will be logged, even if these are not attack logs.

```
USE [PiLog]
GO
/****** Object:  StoredProcedure [dbo].[InsertMLDataSet]    Script Date: 10.03.2021 21:32:20 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[InsertMLDataSet]
AS
truncate table PiAttackLogMLDataSet
insert into PiAttackLogMLDataSet select Host, Count(*) as DIFF,
CASE
    WHEN Count(*) > 2 THEN 1
    ELSE 0
END AS AttackLAbel
from LogDeviation group by DIFF,Host
```

Figure 10 Procedure stored in the system

As depicted in Fig. 11, a total of 1.971.790 logs have been sent by HoneyPi to be evaluated in the artificial intelligence algorithm. However, these logs must be labelled as an attack or not attack in order to feed the artificial intelligence. In Fig. 10, the logs have been processed by the stored procedure, and a total of 1,092,317 log records have been sent to the artificial intelligence as a DataSet simultaneously, as a result of the logs that have multiple packet access to different ports between the same time and the port number of the client device from which the request packets sent are evaluated as an attack.

Results | Messages

	DataId	Host	Port	DIFF	RequestCount	ISP	Location
1	1	192.168.1.74	5901	7616	2	Türk Telekomünikasyon A.Ş	Turkey
2	2	192.168.1.49	80	1331	1	Türk Telekomünikasyon A.Ş	Turkey
3	3	192.168.1.55	21	5742	1	Türk Telekomünikasyon A.Ş	Turkey
4	4	192.168.1.42	5901	6822	2	Türk Telekomünikasyon A.Ş	Turkey
5	5	192.168.1.4	110	4624	1	Türk Telekomünikasyon A.Ş	Turkey
6	6	192.168.1.55	21	7540	1	Türk Telekomünikasyon A.Ş	Turkey
7	7	192.168.1.70	21	9120	2	Türk Telekomünikasyon A.Ş	Turkey
8	8	192.168.1.24	110	7148	2	Türk Telekomünikasyon A.Ş	Turkey
9	9	192.168.1.54	110	2402	1	Türk Telekomünikasyon A.Ş	Turkey
10	10	192.168.1.44	23	716	2	Türk Telekomünikasyon A.Ş	Turkey

localhost\SQLEXPRESS (15.0 ... | DESKTOP-8JB67OC\Reds (64) | PiLog | 00:00:17 | 1.971.790 rows

Figure 11 Logs Evaluated by Artificial Intelligence

Libraries belonging to 4 different algorithms have been used as artificial intelligence algorithms. Six different infrastructures have been used in total. In this process, LSTM, KNN (2 different neighbourhood relations functions), Artificial Neural Networks (CNN), and Naive Bayes (2 sub-branches) algorithms have been tested over real attack logs.

First, the Naive Bayes algorithm is used. The reason for using the Naive Bayes algorithm is that the Naive Bayes algorithm extracts a high degree of accuracy with a small amount of data. In addition, the Naive Bayes algorithm does not work learning. It calculates the probability directly on the datasets, so its cost is low in performance on devices with small memory and limited memory, such as the Raspberry Pi. When the three main sub-branches of Navi Bayes algorithms are examined, the first branch, GaussianNB, is used if the data is to be predicted or the column is continuous (such as natural, decimal). BernoulliNB, the second branch, is used if the data to be predicted or column is binary (such as Yes / No, Smoking / Not Smoking). MultinomialNB, the third branch, is used if the data guess or column is nominal (int numbers). In this

context, Multinominal and Bernoulli algorithms, which are sub-branches of Naive Bayes, are suitable within this study's scope.

Second, the LSTM algorithm is used. Machine learning never stops in the LSTM algorithm because it provides continuous and sequential learning on the DataSets it processes. It is possible to compare this situation to recursive functions. Thanks to the continuous learning and continuity of the processed data, the accuracy rates are higher. For the LSTM algorithm, the epoch number has been entered as 30, and long-term and repetitive learning has been provided. As a result of the LSTM algorithm, the RMSE (The Root Mean Squared Error) value has been determined as 0.4, and the RMSPE (The Root Mean Squared Power Error) value as 0.18, and the following equation has been solved by performing operations on this RMSPE value. As a result of the equation, an accuracy rate of 81.83 percent has been obtained.

Below is the equation:

$$RMSE = \sqrt{\frac{\sum\left(y_{pred} - y_{ref}\right)^2}{N}} \tag{1}$$

$$RMSPE = \sqrt{\frac{100}{N}\frac{\sum\left(y_{pred} - y_{ref}\right)^2}{y_{ref}}} \tag{2}$$

$$Accuracy = \left(1 - RMSPE\right)\cdot 100 \tag{3}$$

Thirdly, the KNN algorithm is used. The distance of the new data to be included in the data set is calculated according to the existing data and the k number of close neighbours. In this way, it is aimed to interpret the new attack / not attack data to be entered into artificial intelligence. The KNN algorithm has been used because of its fast analysis and success in graphical interpretation.

Using the two neighbourhood functions mentioned above, a 96.05% success rate has been obtained for the Euclidean function. In contrast, the Minkowski function has obtained a 98.375% success rate.

The KERAS library has been used as the fourth artificial neural network algorithm (CNN). More than 20 artificial neural cells have been used in the model created. In addition, neural networks have been run for 15 epochs and have been run based on the closest values. In this way, the most successful value among the long-lasting and clearer results has been taken as the basis.

The KERAS library has been used as the fourth artificial neural network algorithm (CNN). More than 20 artificial neural cells have been used in the model created. In addition, neural networks have been run for 15 epochs and have been run based on the closest values. In this way, the most successful value among the long-lasting and clearer results has been taken as the basis. As depicted in Fig. 12, a success rate of 81.27 percent was achieved as a result of the KERAS library.

Artificial intelligence algorithms and accuracy values applied to data transferred from HoneyPie in the study are as given in Tab. 1.

Table 1 Success rates for artificial intelligence algorithms.

Artificial Intelligence Algorithm	Total Data Count	Success Rate
KNN Minkowski Neighborhood Relationship	1.092.317	98.37%
KNN Euclidean Neighborhood Relationship	1.092.317	96.05%
MultinominalNB	1.092.317	85.19%
LSTM	1.092.317	81.83%
KERAS (CNN)	1.092.317	81.27%
BernoulliNB	1.092.317	80,19%

The algorithm with the highest accuracy has been the KNN Minkowski Neighborhood Relationship with approximately 98.37 percent. The reference model created is interpreted as an attack or not an attack by processing the last MongoDB log record with the script running on HoneyPi on artificial intelligence.

7 END-USER INTERFACES AND SECURITY

User interfaces are designed considering that the end-user is an employee or server administrator. In this context, all data on MongoDB and MsSQL can be viewed on mobile, web, and desktop screens. Web services are used at this stage. Web services are designed on token-based. However, considering the vulnerabilities of token-based services against the man in the middle attack, tokens are encrypted with the AES (Advanced Encryption Standart) encryption algorithm at the services endpoint and entry point points and the client machine's service connections. In this way, in the case of ARP (Address Resolution Protocol) poisoning that may occur on the network, it is prevented from being used even if the tokens belonging to the users are accessed. All data are transmitted over the network in encrypted form with AES-256 encryption. Due to the use of AES encryption, it is not feasible to crack with today's technology. User Account information is used as a key for encryption. If the user information is known, the traffic packets are accessible with ARP poisoning. However, suppose the user information is known. In that case, user awareness comes into play at this point, as the database servers can be accessed directly by an attacker. The screenshot where Token-based services are transferred with AES encrypted is depicted in Fig. 13. The screenshot shows the unencrypted token and the token encrypted with AES-256.

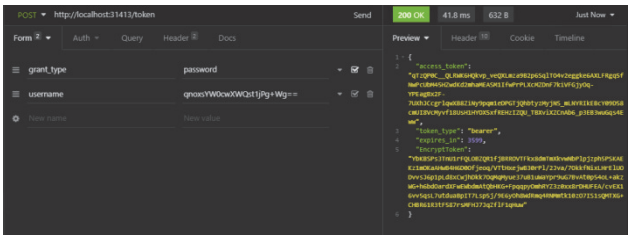

Figure 13 Secure Transfer of Data Regarding Information Disclosure

For the end-user, data flows over the database with AES encrypted token-based APIs on the web, desktop, and mobile screens. C# is used for all screens. Windows Forms is used for desktop screens. It is designed using ASP.NET MVC (Model/View/Controller) for web screens, as indicated in Fig. 14.

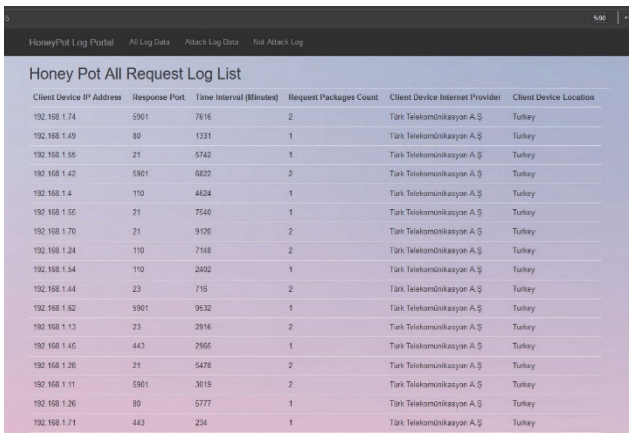

Figure 14 Web Screens

It is designed to work with cross-platform Xamarin for mobile screens for iOS, Android, and Universal Windows Platform (UWP).

Besides, as depicted in Fig. 15 for security staff, it has been ensured that reports for attack analysis are published on the Azure Cloud by using the Microsoft Power BI reporting tool. It is possible to access Azure Cloud from mobile and web screens by integrating web screens and mobile screens with Azure Cloud. System configurations are made on the database. However, it can be defined through the configuration settings on the desktop screens. In this way, these reports, which are understandable, simple, and functional, can be accessed anywhere. A sample report screen regarding the attacks carried out on the network is shown in Fig. 15. The tool has a flexible structure where desired components can be added in the following processes.

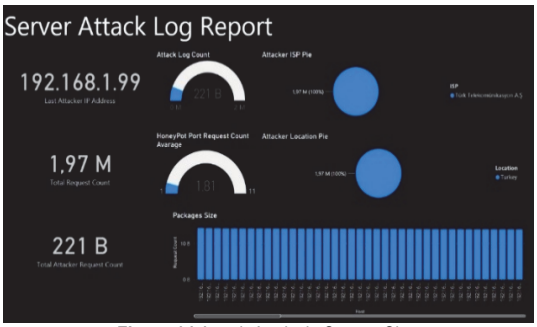

Figure 14 Attack Analysis Screen Shot

The comparison table regarding the techniques used throughout the study and the security measures currently used is given in Tab. 2.

Table 2 The comparison of the techniques used throughout the study and the security measures.

Institution Type / Configuration	Large Scale	Medium Scale	Small Scale
Firewall Configuration	Yes	Yes	Partially
Log Tracking	Yes	Partially	No
Artificial Intelligence Integration	Partially	No	No
Segmentation	Yes	Partially	No
DMZ Configuration	Yes	Partially	Partially
Mobile/Web Interface Integration	Partially	Partially	No
Secure Data Flow	Partially	No	No

8 CONCLUSION

Cyberattacks carried out today, where dependence on the digital environment is increasing, are now prepared in a much more complex and specialized structure. These attack tools and payloads used for attack can be accessed quite easily and relatively cheaply. Therefore, combating cyberattacks has become quite complex and challenging. One of the critical methods used in combating cyberattacks is to perform a practical attack analysis by deploying honeypots on systems. However, to benefit from the existing honeypot systems effectively, their place in the architecture is designed, the analysis of the packages passing over it with high accuracy, and the safe storage of the analyzed traffic packets are crucial.

Within the scope of the study, a secure network approach has been tried to be provided for small and medium-sized organizations in terms of cyber security. For this purpose, the pfSense firewall has been used as an open-source and free firewall. With this firewall, a DMZ is created on the network, and all unauthorized requests from outside are directed to HoneyPi. Access to the authorized zone can only be achieved with a VPN connection.

While the data on the user screens comes on the web services, a double layer architecture is designed for the web services in order not to be exploited by the man in the middle (MitM) attacks. The web services are designed to flow in a token-based manner and encrypted with the given AES. Token-based services provide an advanced Authentication Integration. In this way, the interpretation of data by unauthorized users has been prevented. In addition, in order to prevent an authorized user from becoming weak in case of ARP poisoning, all packets flowing in the client-server architecture where web services are used as an intermediate layer are transmitted encrypted with AES.

Since HoneyPi logs all incoming unauthorized connections (non-VPN connection), these logs must be labelled as to whether it is an attack or not. The attack logs must be displayed for the end-user. For example, if a person working in the company whose VPN connection is disconnected tries to connect to the network, this package will be logged. However, it is not an attack. Artificial intelligence has been integrated as an intermediate layer to separate legal and attack packets coming to HoneyPi. Two different Cloud databases work integrated to meet the dataset needs for artificial intelligence. In this way, data that is attacked by the end-user is selected and interpreted. HoneyPi has logged a total of 1,971,790 logs. However, these logs must be labeled as to whether it is an attack or not and simultaneously feed the artificial intelligence with feedback. When the logs are grouped according to having multiple packet access to different ports in a relatively short period and the same port number of the client device from which the request packets arrive is labelled as an attack, the number of logs calculated 1.092.317. Therefore 1.092.317 logs have been sent to artificial intelligence as a DataSet.

Libraries of 4 different algorithms have been used as artificial intelligence algorithms, and six different infrastructures were used in total. In this process, LSTM, KNN (2 different neighborhood functions), Artificial Neural Networks (CNN), and Naive Bayes (2 sub-

branches) algorithms have been tested on real attack logs. The highest accuracy value among them is LSTM, with approximately 96 percent.

Within the scope of the study, a secure network approach has been provided, incoming requests have been filtered by the specified rules and logged as not attack or attack. At this stage, end-user screens are designed for system administrators or security personnel to access these logs. These screens are designed for three different mobile, web, and desktop applications. Applications designed for mobile are suitable for Android, IOS, and UWP platforms. In this way, it can be published on Google Play, App Store, and Microsoft Store. Web screens have been designed as a response, and thus, they are compatible with browser access from mobile devices. It is aimed to access attack logs and non-attack logs via mobile and Web screens. On the other hand, desktop applications are designed only for system configurations. However, it also shows the log records.

As a result, in the study, an end-to-end secure network design was created by considering the cyber security needs for small and medium-sized organizations. In addition, an integrated approach was provided with artificial intelligence and end-user interfaces in order to report and inform network security officers against possible attacks. The architectural brand model created is independent and flexible. It can be implemented by applying the same steps to other brands and models. It has been evaluated that the study will make significant contributions within the scope of cyber security architecture.

9 REFERENCES

[1] Cabaj, K., & Gawkowski, P. (2015). Honeypot systems in practice. *Przegląd Elektrotechniczny*, *91*(2), 63-67. https://doi.org/10.15199/48.2015.02.16

[2] Gutierrez, M. & Kiekintveld, C. (2020). Online Learning Methods for Controlling Dynamic Cyber Deception Strategies. *Adaptive Autonomous Secure Cyber Systems*, 231-251. https://doi.org/10.1007/978-3-030-33432-1_11

[3] Litchfield, S., Formby, D., Rogers, J., Meliopoulos, S., & Beyah, R. (2016). Rethinking the Honeypot for Cyber-Physical Systems. *IEEE Internet Computing*, *20*(5), 9-17. https://doi.org/10.1109/MIC.2016.103

[4] Pandire, P. A. & Gaikwad, V. B. (2018, July). Attack detection in cloud virtual environment and prevention using honeypot. *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, 515-520. https://doi.org/10.1109/ICIRCA.2018.8597359

[5] Sharma, S. & Kaul, A. (2018). A survey on Intrusion Detection Systems and Honeypot-based proactive security mechanisms in VANETs and VANET Cloud. *Vehicular communications*, *12*, 138-164. https://doi.org/10.1016/j.vehcom.2018.04.005

[6] Du, M. & Wang, K. (2019). An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in the industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, *16*(1), 648-657. https://doi.org/10.1109/TII.2019.2917912

[7] Baykara, M. & Das, R. (2018). A novel honeypot-based security approach for real-time intrusion detection and prevention systems. *Journal of Information Security and Applications*, *41*, 103-116. https://doi.org/10.1016/j.jisa.2018.06.004

[8] Tounsi, W. & Rais, H. (2018). A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & Security*, *72*, 212-233. https://doi.org/10.1016/j.cose.2017.09.001

[9] Truong, T. C., Zelinka, I., Plucar, J., Čandík, M., & Šulc, V. (2020). Artificial intelligence and cybersecurity: Past, presence, and future. *Artificial Intelligence and Evolutionary Computations in Engineering Systems,* 351-363. https://doi.org/10.1007/978-981-15-0199-9_30

[10] Zobal, L., Kolář, D., & Fujdiak, R. (2019). Current State of Honeypots and Deception Strategies in Cybersecurity. *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 1-9. https://doi.org/10.1109/ICUMT48472.2019.8970921

[11] Pauna, A. & Bica, I. (2014). "RASSH - Reinforced adaptive SSH honeypot". *2014 10th International Conference on Communications (COMM), Bucharest*, 2014, 1-6. https://doi.org/10.1109/ICComm.2014.6866707

[12] Fraunholz, D., Pohl, F., & Schotten, H. D. (2017). Towards basic design principles for high-and medium-interaction honeypots. *Proceedings of 16th European Conference on Cyber Warfare and Security*, 120.

[13] Pauna, A., Iacob, A. C., & Bica, I. (2018). Qrassh-a self-adaptive sshhoneypot driven by q-learning. *2018 international conference on communications (COMM)*, 441-446. https://doi.org/10.1109/ICComm.2018.8484261

[14] El Kamel, N., Eddabbah, M., Lmoumen, Y., & Touahni, R. (2020). A Smart Agent Design for Cyber Security Based on Honeypot and Machine Learning. *Security and Communication Networks*, *2020*. https://doi.org/10.1155/2020/8865474

[15] Tian, W., Ji, X. P., Liu, W., Zhai, J., Liu, G., Dai, Y., & Huang, S. (2019). Honeypot game-theoretical model for defending against APT attacks with limited resources in cyber-physical systems. *ETRI Journal*, *41*(5), 585-598. https://doi.org/10.4218/etrij.2019-0152

Contact information:

Birkan ALHAN
Faculty of Engineering and Architecture, Istanbul Gelisim University,
İstanbul, Turkey
E-mail: birkanalhan@gmail.com

Serkan GÖNEN, Assist. Prof. Dr.
Faculty of Engineering and Architecture, Istanbul Gelisim University,
İstanbul, Turkey
E-mail: sgonen@gelisim.edu.tr

Gökçe KARACAYILMAZ
Forensic Sciences, Hacettepe University,
Ankara, Turkey
E-mail: gkaracayilmaz@gmail.com

Mehmet Ali BARIŞKAN
Faculty of Engineering and Architecture, Istanbul Gelisim University,
İstanbul, Turkey
E-mail: mabariskan@gelisim.edu.tr

Ercan Nurcan YILMAZ, Full Professor
(Corresponding author)
Gazi University, Faculty of Technology,
Ankara, Turkey
E-mail: enyilmaz@gazi.edu.tr