International Journal of Safety and Security Engineering
Vol. 12, No. 6, December, 2022, pp. 737-743
Journal homepage: http://iieta.org/journals/ijsse

Spark-Based Network Security Honeypot System: Detailed Performance Analysis

Akshay Mudgal*, Shaveta Bhatia

Manav Rachna International Institute of Research and Studies, Faridabad 121004, India

Corresponding Author Email: toakshaymudgal@gmail.com

https://doi.org/10.18280/ijsse.120610

Received: 1 April 2022
Accepted: 13 November 2022

Keywords:
honeypot system, Apache Spark, network security, cyber security, big data

ABSTRACT

In the contemporary world, network security has been of the biggest importance and acute worry in both individual and institutional wisdom, concurrent with the newly emerging technologies. Firewalls, encryption techniques, intrusion detection systems, and honeypots are just a few of the systems and technologies that have been developed to ensure information security. Systems for safeguarding an organizational environment through various defensive strategies are traditionally developed. "The enemy continues on attacking" is a primarily defensive statement. By empowering an organization to take action, Honeypot demonstrates its importance. An institution can discover, gather, address, and absorb new security policy flaws with the aid of honeypots. This methodology allows an organization's security measures to continuously incorporate new threats and penetration methods. This is the main justification of creating, developing, and using a honeypot. It is a resource that is meant to be taken advantage of and compromised. To evaluate the methodology, a spark-based honeypot strategy has been designed, put into practice, and tested in this study. The suggested study strategy has undergone many days of testing on a campus-based network. The designed system's primary function is to behave as a fully resourced computer or vault to draw intruders with the requirement that they not defend against or respond to intrusions. The studies were carried out using a number of parameters that were designed.

1. INTRODUCTION

A honeypot is a resource whose value and utility is in being attacked or compromised, according to a group of American researchers called Project Honeynet. This definition suggests that a honeypot is a luring system built and anticipated to be examined and potentially abused. Honeypot functions as a genuine and very resourceful vault machine, neither fixing nor retaliating in any way [2]. They give us new, important details regarding the attacks. Since the public's access to the internet was granted in 1993, the number of users has skyrocketed, a wide range of online services and websites of various kinds have seen rapid growth [2, 3]. Parallel to this rapid and radical development in internet technology, more important problems like security are raised. The researcher's community had employed passive protection technology to deal with this issue [3, 4]. This method operates both after and during an assault. However, it did not prove to be effective, therefore active defensive technology gained popularity, giving rise to Honeypot.

Honeypots are generally intended to detect and manage network threats such as intruders and attackers [5].

However, due to its passive defensive methodology, it is sometimes observed that Honeypot consumes a long delay of time in order to effectively solve the issues caused by the intruder's.

Because of the time delay, a Big Data technique known as Apache Spark is used to improve the utilization of network information [5, 6].

Apache Spark is a fantastic big data processing structure built around speed and sophisticated analytics to process large amounts of data in a relatively short period of time [7, 8]. It is an open-source computing framework developed by the UC Berkeley AMP Lab that consists of a general parallel map reducing system with all of the benefits of Hadoop [3, 5, 6] and the construction of resilient distributed datasets (RDD) to provide cluster-oriented computing based on memory, thereby increasing processing speed [8, 9].

2. PROBLEM STATEMENT

A network administrator can better understand an intruder's behavior and operational style by using honeypot systems, which are designed to entice bad actors and allow them to invest their time and resources [10, 11]. However, those designed honeypot systems frequently exhibit opacity and sluggishness for reasons related to insufficient processing power. The concept of deploying them on internet-based websites or in a networking environment becomes rather tough and difficult. By integrating the traditional honeypot system with Apache Spark [3, 12], a big data platform, the proposed solution improves on it. This change essentially enhances the honeypot system's processing capability.

3. DESIGNING HONEYPOT SYSTEM

In order to meet the requirements of the networking environment, the system has been built. As was said previously, a honeypot is a piece of equipment that has value only when it is scanned and attacked [2]. A number of virtual

computers have been set up to show how the network functions while the honeypot system imitates the protected system's web services [13, 14]. The constructed honeypot will be contacted if the inbound services pose a threat. The connection will typically be made with the main protected machinery [15-17]. Even if the established connection is encrypted and the data being transmitted is secure, the data capturing module will still record the user's actions and behavior to stop any intruders from entering the network [18, 19].

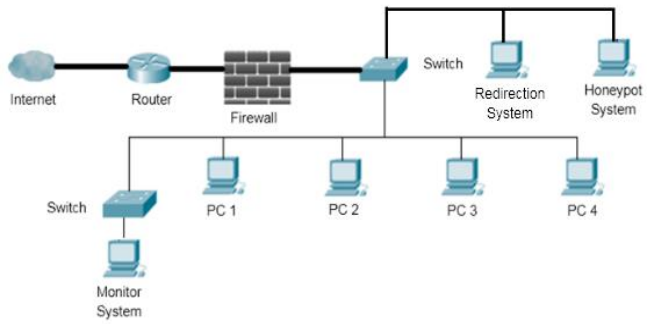

Figure 1. Generalized framework

The suggested system's operational flow is shown in the diagram above. User systems (from PC1 to PC4) have been connected through a switch, displaying the network's usual operational state. If an intrusion attempt is made, the intruder redirection module directs the attempt to the fake machine. A monitoring system has also been added to observe and record all the desired and necessary statistics of the forwarding and redirecting triggers. This monitoring system lessens the load on the developed honeypot system and also helps to study the latencies of attacks and other important information.

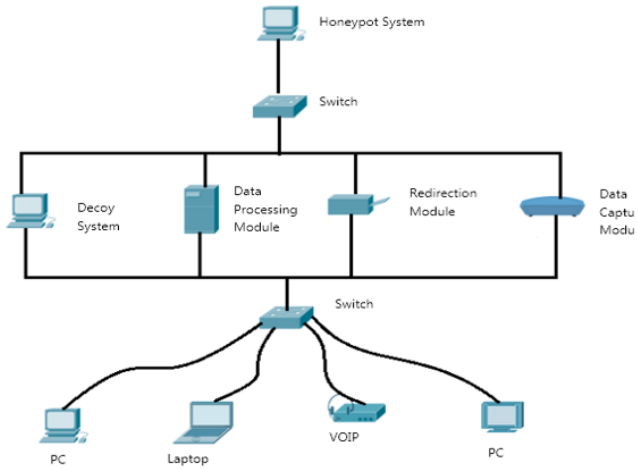

Figure 2. Itemization of honeypot system

The presented (Figure 2) image elucidates the internal portions of the Honeypot system. The underneath four deployed systems depict the lure or trap module, collection of data and processing, a module for redirection, and the last capturing of data. These four sub-systems collectively develop and form the proposed system.

3.1 Configuration details

```
COPY . /app

WORKDIR /app

RUN python3 setup.py sdist bdist_wheel && pip3 install

CMD /bin/bash

FROM ubuntu:20.04

MAINTAINER qeeqbox

ENV DEBIAN_FRONTEND=noninteractive

RUN apt-get update && apt-get install -y syslog-ng

ADD syslog-ng.conf /etc/syslog-ng/syslog-ng.conf

EXPOSE 514/udp

WORKDIR /var/log/syslog-ng/

ENTRYPOINT ["syslog-ng", "-F"]
```

3.2 Configuration system logs

```
{
"logs": "syslog",
"logs_location": "",
"syslog_address": "udp://localhost:415",
"syslog_facility": 3,
"postgres":"//changeme027a088931d22:changeme0f40773
877963@0.0.0.0:9999/chameleon",
"sqlite_file":"/home/p98h0digu248jr/Desktop/db/test.db",
"db_options":["drop"],
"filter": "",
"interface": "",
"honeypots": {
"http": {
"port": 5566,
"username": "test",
"password": "retest"
}
}
}
```

3.3 Calling Apache Spark

```
import sys

from typing import cast, overload, Dict, Iterable, List, Optional, Tuple, TYPE_CHECKING, Union

from py4j.java_gateway import JavaClass, JavaObject

from pyspark import RDD, since
from pyspark.sql.column import _to_seq, _to_java_column, Column
from pyspark.sql.types import StructType
from pyspark.sql import utils
from pyspark.sql.utils import to_str

if TYPE_CHECKING:
from pyspark.sql._typing import OptionalPrimitiveType, ColumnOrName
from pyspark.sql.session import SparkSession
from pyspark.sql.dataframe import DataFrame
from pyspark.sql.streaming import StreamingQuery

__all__ = ["DataFrameReader", "DataFrameWriter"]

PathOrPaths = Union[str, List[str]]
TupleOrListOfString = Union[List[str], Tuple[str, ...]]
```

4. LURING MODULE

To attract invaders and enable attackers to commit a significant amount of time and resources in it, a lure or decoy system is created [20, 21]. This requires a high level of simulation because the honeypot exists and is standing as a result of the lure module. An assailant or invader must not be able to comprehend it easily. The Windows operating system on a virtual machine is the foundation for the decoy system's implementation [22]. To lessen the likelihood of failure or to keep the attacker from determining the difference between the honeypot and the protected system, the configuration of the honeypot is identical to that of the protected system.

5. DATA CAPTURING AND PROCESSING SYSTEM (DCPS)

The data is gathered and further processed for inspection purposes in this module. DCPS has been divided into two sequences [25], one of which is responsible for providing the system's log files for processing and the other of which is in charge of providing web or internet services [23]. The data collection and processing system functions in this configuration and completes the required duties. The user's activities, working patterns, browsing habits, and the amount of time and resources he has invested are evaluated and netted under this section of the Honeypot system to determine whether the individual is an intruder or a potential user [24]. To complete the aforementioned duties, a specific unit of the operating environment is needed. This unit comes with a system running the most recent version of Linux or Red hat, requiring a minimum of 150 Gb of storage and at least 10 Gb of memory. In addition to these configuration-related requirements, the most recent version of Apache Spark is needed for consumption of Scala, Zookeeper, and Hadoop.

6. REROUTING/REDIRECTION/FORWARDING ADD-ON

The Redirection Module is intended to carry out a predefined task in relation to an intruder [2]. When a bad actor engages in damaging and malevolent behavior, the task of this system is to absorb him secretly and transfer or pass him over to the well-designated honeypot. The system must be developed in such a way that the intruder is unaware that it has been changed or assigned from one machine to another. The block diagram below will concisely illustrate the architecture and function of the Redirection Module (Figure 3).

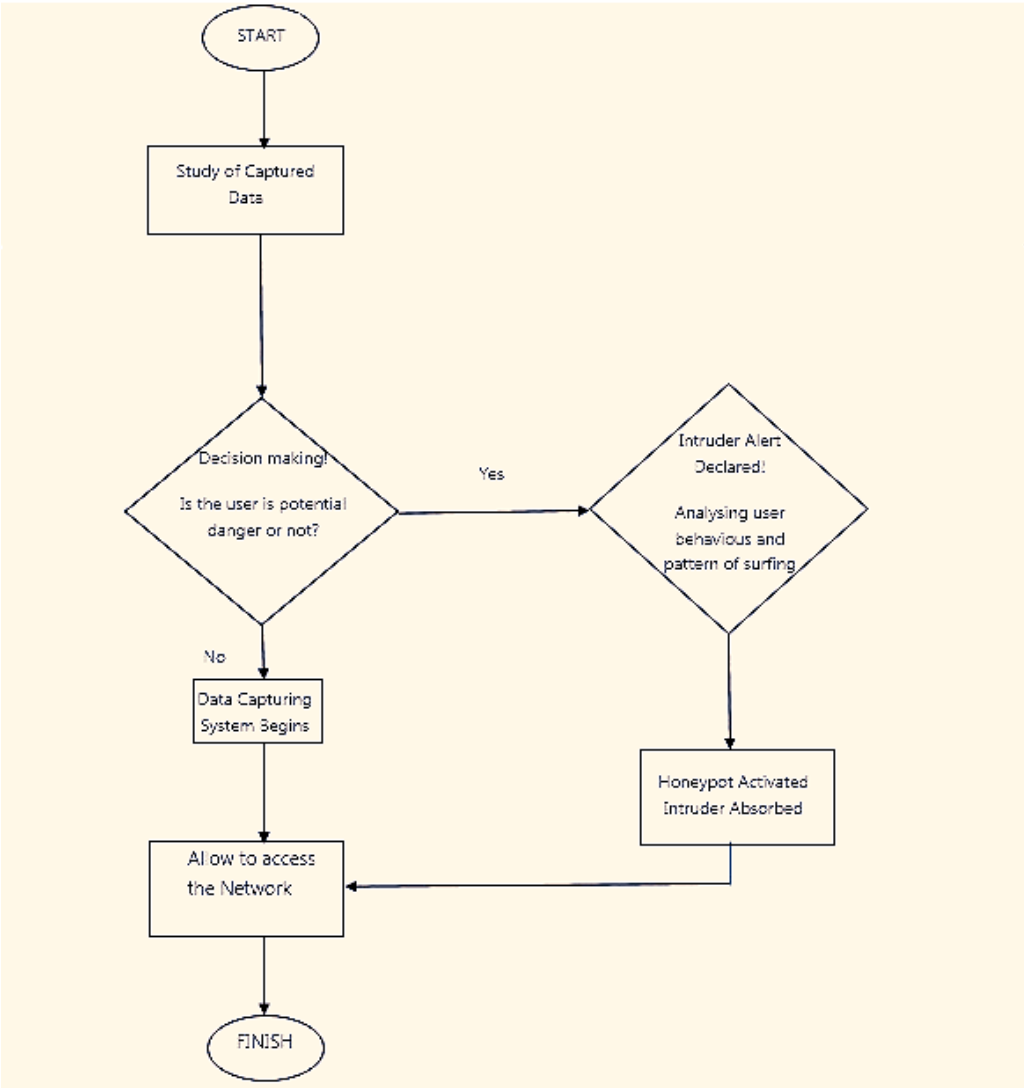

Figure 3. Data flow diagram

7. SIMULATION AND PERFORMANCE ANALYSIS

The constructed system is tested by counting and pushing all of the stress variables during the simulation test, which is a genuine network environment creation test. The Decoy System, the Data Capturing and Processing System, and the Redirection System are the three main stress causes in this situation. In order to get the required findings, three tests are administered. The core honeypot had a first test to examine how it operated and performed. One system was randomly chosen from a pool of four dedicated machines each time to act as an intruder. As depicted in Figure 1, the full networking environment was developed. The honeypot's perfect operation leaves an imprint of its high quality.

As was mentioned prior a Honeypot's value comes from being attacked and compromised, hence this was an important test parameter.

The Data Capturing and Processing System underwent the Second Test. A specific working environment is used in this experiment. Three different forms of data—Non-Infected Data, Semi-Infected Data, and Fully-Infected Data—were injected into the same architecture (Figure 1) by a randomly chosen machine. The designated developed machine's (DCPS) job is to examine and process user behavior. In DCPS, Apache Spark plays a fundamental and important role.

The line chart below, which was created to evaluate the data processing, shows a brief comparison between conventional processing and spark-based data processing.

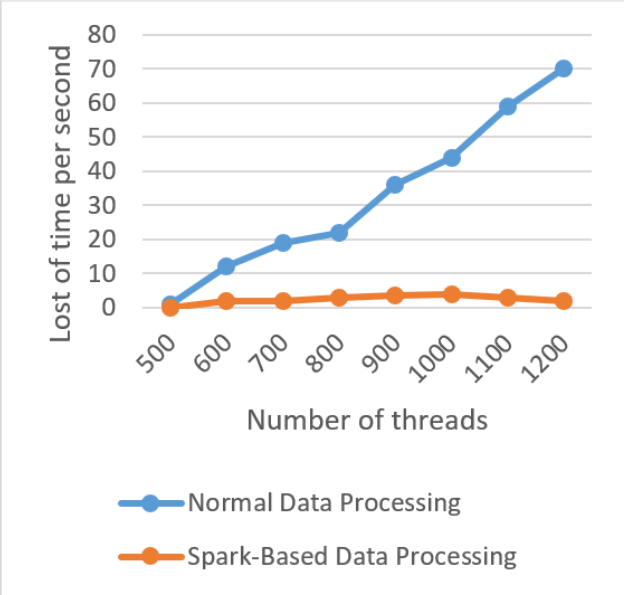

Figure 4. Comparative representation between normal and spark based data processing

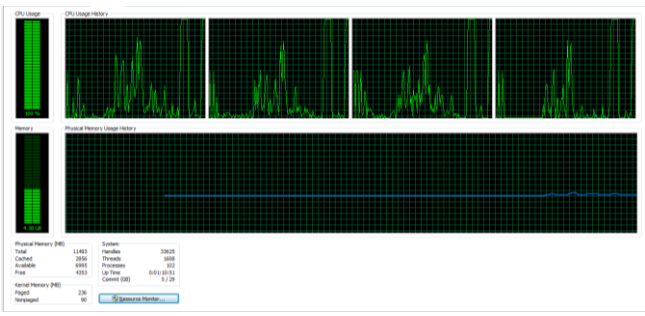

Figure 5. System load capturing without Apache Spark

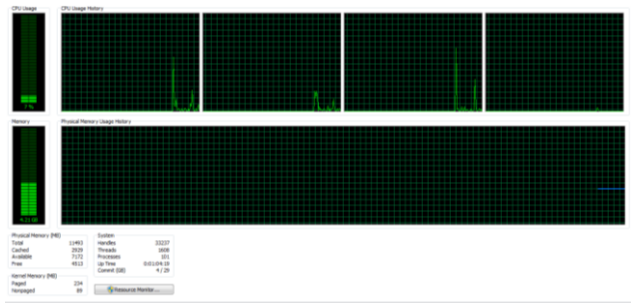

Figure 6. System load capturing with Apache Spark

The line graph (Figure 4) above illustrates the differences between Spark-based and conventional data processing methods, while Figures 5 and 6 show the load test burden by running both systems concurrently on two different machines with the same setup. The honeypot system that runs on Apache has actually seen a significant increase in processing speed.

In this experiment, the system was tested utilizing multiple threads starting at 500 using a simulation of an SNMP/DDoS assault. A distributed Denial of Service attack is a malicious attempt to stymie and disrupt ongoing services or the regular operation of a targeted server, machine, or network by flooding the targeted machine/network or its surrounding infrastructure with Internet traffic and packets. In the preceding scenario, the simulation was carried out, with the number of threads functioning as the flood of data packets to be recorded and processed, and the time period fixed in seconds on the vertical axis to gauge performance. According to this graph, the processing power of the spark-based system is significantly more than the other.

The third test, which is dependent on the users' IP addresses, is the Redirection System Test.

The DCPS-produced processed data is regarded as an input in this experiment. The redirection module activates its mechanism to direct the specific IP address either to the secured network (102.202.113.224) or toward the honeypot, per the DCPS report (102.202.113.223). For example, if the DCPS reports the IP address 192.168.50.10 with the MAC address XX: XX: XX: XX: XX as an intruder, the redirection module smoothly redirects it to the dedicated Honeypot System, whereas if the IP address 192.168.50.11 is reported as a potential user, his activity will still be captured, but he will be allowed to connect to the core network and its surrounding pieces of equipment. This system can associate an IP address with the system MAC address for future reference.

8. CONTRIBUTION TOWARDS SOCIETY

This is a dedicated system designed for active defense and learning how an invader works. It could be a valuable tool for institutional or organizational administrators to trap and capture the latent user's activity (s). This system is intended to fill the void left by older generation honeypots by incorporating and deploying cutting-edge technology.

9. CONCLUSION

Honeypot is a backup technology that serves no purpose when used alone. It is a strategy used in conjunction with other passive defensive strategies such as firewalls and intrusion detection systems to defend and procure the burglar's

methodology of disrupting the computing environment. Honeypots can be used on websites, intranets, LANs, WANs, or anyplace there is a gathering of computing equipment on a network. The incorporation of Big Data technology into the traditional Honeypot system is a requirement of the era, as it can improve system efficiency by offering security and speed at the same time. The capabilities are significantly improved as a result of this change, which meets the needs of the research worthy.

REFERENCES

[1] 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.

[2] Chen, W., Yao, J., Tan, J. (2018). The design and implementation of the honeypot system based on spark. In 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS) Xiamen, China, pp. 545-548. https://doi.org/10.1109/ICITBS.2018.00143

[3] Dowling, S., Schukat, M., Barrett, E. (2019). Using reinforcement learning to conceal honeypot functionality. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Dublin, Ireland, pp. 341-355. https://doi.org/10.1007/978-3-030-10997-4_21

[4] Fan, W., Du, Z., Fernández, D., Villagra, V.A. (2017). Enabling an anatomic view to investigate honeypot systems: A survey. IEEE Systems Journal, 12(4): 3906-3919. https://doi.org/10.1109/JSYST.2017.2762161

[5] Fan, W., Du, Z., Smith-Creasey, M., Fernandez, D. (2019). Honeydoc: An efficient honeypot architecture enabling all-round design. IEEE Journal on Selected Areas in Communications, 37(3): 683-697. https://doi.org/10.1109/JSAC.2019.2894307

[6] Gupta, A., Sharma, L.S. (2020). Performance analysis and comparison of snort on various platforms. International Journal of Computer Information Systems and Industrial Management Applications, 10: 23-32. https://doi.org/10.1109/JPROC.2017.2748421

[7] Irvene, C., Formby, D., Litchfield, S., Beyah, R. (2017). Honeybot: A honeypot for robotic systems. Proceedings of the IEEE, 106(1): 61-70. https://doi.org/10.1109/JPROC.2017.2748421

[8] Luo, T., Xu, Z., Jin, X., Jia, Y., Ouyang, X. (2017). Iotcandyjar: Towards an intelligent-interaction honeypot for iot devices. Black Hat, 1: 1-11.

[9] Moore, C. (2016). Detecting ransomware with honeypot techniques. Proceedings - 2016 Cybersecurity and Cyberforensics Conference, CCC 2016, pp. 77-81. https://doi.org/10.1109/CCC.2016.14.

[10] Naik, N., Jenkins, P., Savage, N., Yang, L. (2021). A computational intelligence enabled honeypot for chasing ghosts in the wires. Complex & Intelligent Systems, 7(1): 477-494. https://doi.org/10.1007/s40747-020-00209-5

[11] Nawrocki, M., Wählisch, M., Schmidt, T.C., Keil, C., Schönfelder, J. (2016). A survey on honeypot software and data analysis. arXiv preprint arXiv:1608.06249.

[12] Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., Rossow, C. (2016). IoTPOT: A novel honeypot for revealing current IoT threats. Journal of Information Processing, 24(3): 522-533. https://doi.org/10.2197/ipsjjip.24.522

[13] Tsikerdekis, M., Zeadally, S., Schlesener, A., Sklavos, N. (2018). Approaches for preventing honeypot detection and compromise. In 2018 Global Information Infrastructure and Networking Symposium (GIIS), Thessaloniki, Greece, pp. 1-6. https://doi.org/10.1109/GIIS.2018.8635603

[14] Saikawa, K., Klyuev, V. (2019). Detection and classification of malicious access using a Dionaea honeypot. In 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2: 844-848. https://doi.org/10.1109/IDAACS.2019.8924340

[15] Sedlar, U., Južnič, L.Š., Volk, M. (2020). An iteratively-improving internet-of-things honeypot experiment. In 2020 International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom) Graz, Austria, pp. 1-6. https://doi.org/10.1109/CoBCom49975.2020.9174014

[16] Sethi, T., Mathew, R. (2021). A study on advancement in honeypot based network security model. In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) Tirunelveli, India, pp. 94-97. https://doi.org/10.1109/ICICV50876.2021.9388412

[17] Thongkanchorn, K., Ngamsuriyaroj, S., Visoottiviseth, V. (2013). Evaluation studies of three intrusion detection systems under various attacks and rule sets. In 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013) Xi'an, China, pp. 1-4. https://doi.org/10.1109/TENCON.2013.6718975

[18] Ullah, I., Mahmoud, Q.H. (2017). A hybrid model for anomaly-based intrusion detection in SCADA networks. In 2017 IEEE International Conference on Big Data (Big Data) Boston, MA, USA, pp. 2160-2167. https://doi.org/10.1109/BigData.2017.8258164

[19] Vishwakarma, R., Jain, A.K. (2019). A honeypot with machine learning based detection framework for defending IoT based botnet DDoS attacks. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 1019-1024. https://doi.org/10.1109/ICOEI.2019.8862720

[20] Wafi, H., Fiade, A., Hakiem, N., Bahaweres, R.B. (2017). Implementation of a modern security systems honeypot honey network on wireless networks. In 2017 International Young Engineers Forum (YEF-ECE), Costa da Caparica, Portugal, pp. 91-96. https://doi.org/10.1109/YEF-ECE.2017.7935647

[21] Wang, K., Tong, M., Yang, D., Liu, Y. (2020). A web-based honeypot in IPv6 to enhance security. Information, 11(9): 440-440. https://doi.org/10.3390/info11090440

[22] Wong, K., Dillabaugh, C., Seddigh, N., Nandy, B. (2017). Enhancing Suricata intrusion detection system for cyber security in SCADA networks. In 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE) Windsor, ON, Canada, pp. 1-5. https://doi.org/10.1109/CCECE.2017.7946818

[23] Xiao, G. (2020). Research on computer network information security based on big data technology. In 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS) Dalian,

China, pp. 325-329. https://doi.org/10.1109/ICAIIS49377.2020.9194896

[24] Zhao, G., Song, J. (2020). Network security model based on active defense and passive defense hybrid strategy. Journal of Intelligent & Fuzzy Systems, 39(6): 8897-8905. https://doi.org/10.3233/JIFS-189287

[25] Zobal, L., Kolář, D., Fujdiak, R. (2019). Current state of honeypots and deception strategies in cybersecurity. In 2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT) Dublin, Ireland, pp. 1-9. https://doi.org/10.1109/ICUMT48472.2019.8970921