3rd International Conference on Computational Sciences and Technologies

Detection Of Server-Side DHCP DoS And Spoofing Attack Using Machine Learning Techniques

Hina Lilaram
Computer Systems Department
Mehran UET
Jamshoro, Pakistan
hinalilaram@gmail.com

Sarmad Talpur
Computer Systems Department
Mehran UET
Jamshoro, Pakistan
sarmadtalpur60@gmail.com

Ali Murtaza Bozdar
Computer Systems Department
Mehran UET
Jamshoro, Pakistan
murtazabozdar5@gmail.com

Shameel Syed
Computer Systems Department
Mehran UET
Jamshoro, Pakistan
shameel_uddin@yahoo.com

Faheem Khuhawar
Telecommunication Department
Mehran UET
Jamshoro, Pakistan
faheem.khuhawar@faculty.muet.edu.pk

***Abstract*— The main aim of Network Intrusion Detection Systems (NIDS) is to efficiently and accurately detect anomalous activities occurring in the network. Machine Learning techniques have now been induced in NIDS. The researchers have used publicly available datasets to develop efficient NIDS. However, the datasets used in existing studies are insufficient as they do not include the most widely used protocols, including DHCP, which plays a vital role in network architecture. The Dynamic Host Configuration Protocol (DHCP) is used in a network to dynamically allocate IP addresses and other important network configuration parameters. The main contribution of this paper is to develop a DHCP-specific dataset that contains the most critical characteristics for detecting two forms of DHCP attacks: DHCP starvation and DHCP spoofing. The dataset has been evaluated using Random Forest (RF) and the K-Nearest Neighbors (KNN) algorithm. The RF algorithm reached its highest accuracy of 0.98 and precision value of 1 on increasing the number of trees to 100 trees whiles the KNN algorithm resulted in the near-perfect precision and accuracy value of 1 on decreasing the number of neighbors to 14 neighbors. The KNN classifier results in the best accuracy and performance values as compared to the RF classifier.**

Keywords— DHCP, DHCP Starvation, NIDS, IDS, Network Security.

I. INTRODUCTION

Due to the exponential growth in the network size, there is an increase in the devices connected to the network, such as mobile devices, telephones, and IoT sensors. Each of these devices requires an Internet Protocol (IP) [1]. In comparison with small-scale networks, the task of manually assigning IP addresses becomes extremely tedious and time taking in a large-scale network. Therefore, Dynamic Host Configuration Protocol (DHCP) is used to automatically assign IP addresses when requested by any host.

As there is no security or authentication mechanism available by default within a DHCP server, this protocol is vulnerable to Denial of Service (DoS) attack, classically known as DHCP starvation attack which is considered as one of the most harmful attacks as it can lead to other major security attacks that can compromise an organization's data and cause enormous damage to its confidentiality. It is also vulnerable to spoofing attacks by implanting DHCP rogue servers within the network environment.

Numerous studies assess machine learning algorithms to develop efficient NIDS, this study focuses on the Decision Tree algorithm. Machine learning "classification" algorithms are used due to the reason that the ultimate aim of an IDS is to detect network anomalies as "normal" from an "abnormal" traffic behavior which is a classification problem [2]. Additionally, the dataset is of small scale and focused on classifying DHCP starvation and spoofing attacks.

The major problem with existing datasets is that they do not contain certain protocols such as DHCP or may not incorporate all attacks related to that specific protocol. Moreover, additional features and characteristics are provided in existing datasets that do not play any part in the network attack. Analysis of machine learning classifiers on publicly available datasets neglects Dynamic Host Configuration Protocol (DHCP). Attacks under DHCP are present in real-world networks.

This study emulates a network environment. The network traffic is utilized to produce a dataset, which is subsequently used to train and test machine learning algorithms for classification.

The main contributions of our research are summarized as follows:

1. A novel dataset based on the DHCP protocol is developed.
2. The dataset includes attacks that are not available in other publicly available datasets.
3. Train and test machine learning classifier to assess its accuracy on the newly generated DHCP-specific dataset.

The rest of the paper is organized as follows, section 2 provides a brief background of DHCP and DHCP attacks. section 3. provides an overview of past machine learning approaches for NIDS on publicly available datasets. section 4. covers the generation of the new DHCP protocol-based dataset and its analysis against various machine learning algorithms. Finally, the results of the tests and experiments are discussed in section 5.

ISBN # 978-969-23372-2-9

II. BACKGROUND

A. *Dynamic Host Configuration Protocol (DHCP)*

DHCP protocol was developed to dynamically assign network information including IP address, subnet mask, default gateway, and DNS information to a specific client. The working principle of DHCP is based on User Datagram Protocol (UDP). For DHCP server traffic port 67 is used as UDP port number and for DHCP client traffic port 68 is used [3]. Figure 1 depicts the primary parameters used in a DHCP packet. The features involved in the attack and dataset are discussed below:

- **Transaction Identifier (xid):** DHCP client assigns a random 4-octet field that is used for conversation between the DHCP client and DHCP server.
- **Seconds (secs):** A DHCPDISCOVER message with elapsed seconds is included in a 2-octet field generated by the DHCP client to request binding network information or request a renewal process.
- **Client Hardware Address (chaddr):** This field contains DHCP client MAC address of 16-octets.
- **DHCP Options:** This is a variable-length field that contains parameters such as:
  1. Option 3: An ordered list of IP addresses specified by the DHCP Router
  2. Option 12: specifies the DHCP client name.
  3. Option 53: Contains DHCP message type.
  4. Option 61: Specifies client-identifier that is used to identify and assign IP to a legitimate client.

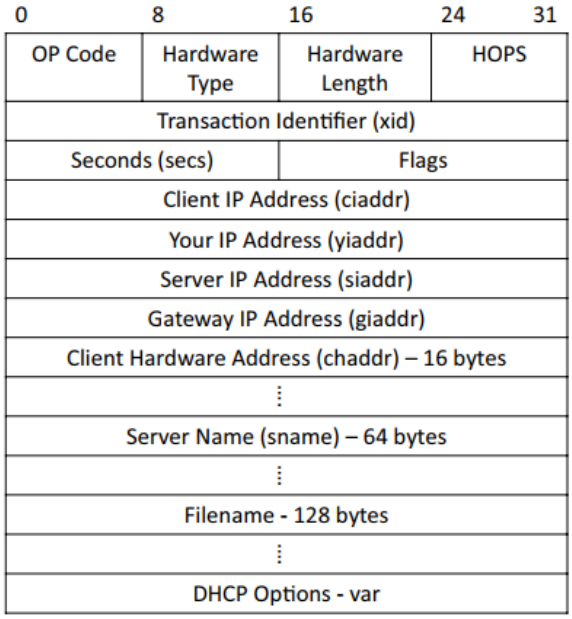

Figure 1: DHCP Packet Parameter

The IP address binding process is achieved through a process called D.O.R.A that consists of four fundamental steps described below:

1. DHCP DISCOVER: A DHCP client sends a broadcast message that looks for a DHCP server that can grant an IP.
2. DHCP OFFER: This message is generated by the DHCP server in response to a DHCP discover message that informs the client about the availability of the DHCP server and offers an IP. This is a unicast message; it is only sent to a designated client who initiated the discover message.
3. DHCP REQUEST: This broadcast message is generated by a client which requests the DHCP server to assign the offered IP to it.
4. DHCP ACKNOWLEDGMENT:

The acknowledgment message is sent from the DHCP server to the DHCP client to ensure that the target IP has been assigned to the legitimate host.

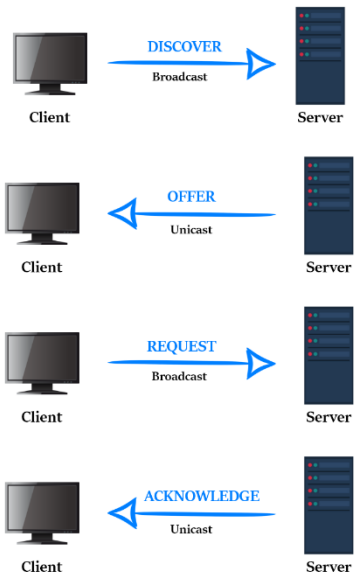

Figure 2: DHCP D.O.R.A Process

B. *DHCP Attacks*

1) *Classic DHCP Starvation Attack*

A starvation attack operates by exhausting the DHCP server from its pool of available IP addresses. Once an IP has been assigned to a specific host the DHCP server marks it "unavailable". A malicious host may continuously broadcast DHCP Discover requests to make the DHCP server mark all available IPs as assigned, this eventually results in exhaustion of available IP addresses which results in the DHCP server running out of all available IPs, and therefore, the DHCP server is no longer able to serve any authentic host requests afterward. With minimum bandwidth requirement, one can launch a DoS attack. DHCP starvation attacks can be launched using tools such as DHCPig, dhcpstarv, Dstar, DHCPwn, Yersinia, Hyenae, and Ettercap [4]. A malicious client sends a large amount of DHCPDISCOVER messages to the DHCP server using different MAC addresses. To handle these requests, the server assigns all its available IPs and gets into a state of starvation, the server is then no longer able to assign IP addresses to legitimate hosts. Hence, a successful starvation attack has been performed.

ISBN # 978-969-23372-2-9

2) DHCP Starvation Attack by Changing CHADDR

An attacker may continuously send a large number of DHCP message packets with different Client Hardware Address (chaddr) field while keeping the Media Access Control (MAC) address constant. As the DHCP server only verifies the source MAC address in the header frame, this process results in DHCP assigning all its IPs to a malicious host until it exhausts itself.

3) DHCP Starvation due to Server-Side Conflict

A DHCP starvation attack caused by server-side conflict is proposed in the study [5] which is executed according to the following mentioned steps:

1. In a wired network, the host sends a DHCP to discover a message as soon as it boots up. On the contrary, in a wireless network, the host first identifies the DHCP server through the association with an Access Point (AP).
2. When a DHCPDISCOVER message is received, the DHCP server probes the network to verify if the target IP is already in use by the host, then, it assigns an IP address from its pool.
3. The DHCP server may use an ARP request or an ICMP request for probing, depending on the DHCP server vendor.
4. According to the request, the malicious host sends an ARP response if an ARP probing request is received or it sends an ICMP response in case an ICMP request is received. This is to inform the DHCP server that the IP address is already assigned to a legitimate host.
5. The DHCP server on receiving the fake response marks the target IP as “assigned” and provides the client with another ARP or ICMP request with a new IP address from its pool of IP addresses. The malicious host again sends a fake response and the process continues in this loop until the DHCP server is left with no IPs to serve.
6. As the entire attack is based on ARP or ICMP response, therefore, it goes undetected by port security and DHCP snooping security features.

4) DHCP Spoofing Attack

To perform a DHCP spoofing attack a rogue DHCP server is implanted in the network. This rogue server has all the configurations of a legitimate DHCP server and it acts to provide DHCP service along with the actual DHCP server. The rogue server serves the incoming DHCP messages and responds with malicious network parameters. This creates a Man-In-The-Middle (MITM) attack due to which the actual DHCP server is not able to provide valid IP addresses to legitimate clients.

III. LITERATURE REVIEW

Several researchers have contributed to the development of benchmark datasets and machine learning classifiers to detect anomalous network activities, their findings and shortcomings are listed below:

A. Challenges in Detecting DHCP Attacks

A comprehensive overview of machine learning and deep learning approaches has been presented in the existing studies that provide a detailed review of machine learning and deep learning techniques for developing network intrusion detection systems. The study highlights shortcomings of the datasets used and the complex deep learning models that pose a challenge to build an efficient NIDS:

1. Deep learning-based algorithms were used by researchers to extract features and reduce the time for training of large datasets whereas, Machine learning classification algorithms like Support Vector Machine, Decision Tree, K-Means Clustering were utilized for the classification of small-scale datasets. This study reveals that the performance of IDS is reduced due to the complex structure of DL models and the integration of DL-based IDS is highly expensive.

2. Unavailability of up-to-date datasets that are not able to detect novel attacks for modern networks including the zero-day attack. An updated dataset for efficient IDS is required that integrates both the older and newer attacks which will allow the machine learning and deep learning models to detect frequent and relevant patterns.

3. Lower accuracy is observed while training the models due to the imbalance in the dataset features. In comparison to the "Attack" and "Non-Attack" classes, excess of the "Attack" class is observed in the available NIDS datasets.

4. Poor performance in a real-world network environment: When models are trained on benchmark datasets that do not contain real-world attacks, the results are unrealistic and poor.

Comparative analysis of Machine Learning classifiers like Random Forest, Decision Tree, Extreme Learning Machine, and Support Vector Machine is proposed in [6] However, due to the use of a GPU-integrated testbed, the research was conducted on an outdated dataset, NSL-KDD, which failed to produce promising results for real-time networks.

A study in [7] proposes an efficient self-learning NIDS using Support Vector Machine and sparse autoencoder. However, this research is also conducted on NSL-KDD.

Distributed abnormal behavior approach was used by [8] where DBN is used to identify features, ensemble SVM is used to combine them, and voting is used to forecast the outcome. However, because of its complexity, training takes a long time, which is increased even more for deeper layers.

B. Existing Prevention Approaches

The existing work focuses on preventing DHCP attacks using authentication as compared to detection. Apart from the most broadly used mitigation techniques of port security and DHCP snooping.

1. The machine learning-based detection framework proposed in [9] works on detecting DHCP and ARP messages to train, test, and successively classify their

ISBN # 978-969-23372-2-9

specific attack while neglecting the other DHCP attacks. Moreover, the attack proposed was not compared against other DHCP attacks.

2. An improved detection method was proposed in the study [10] that sends ICMP ECHO messages to all assigned IP addresses. In case the client is unable to respond due to a firewall or maliciously leased IP, an ARP request is sent. If the client can respond to the ARP request, then it is considered a legitimate client whereas a no response is considered as the IP being leased to an attacker. This methodology is unable to detect real-time attacks and may consider clients with static IP addresses as malicious.
3. A different approach was proposed in [11] that helped in identifying the originating port and switch but was unable to prevent the attack. It operated on enabling the Relay Agent Information "DHCP Option 82" that inserts the network port details in the header frame of a DHCP packet.
4. An effective presentation method used by various vendors including Cisco and Juniper uses port security which allows only a specific number of MAC addresses on a switch port, on exceeding this limit the port either drops traffics or goes into a shutdown state. However, this method is not effective to detect attacks with static MAC and different CHADDR addresses.
5. "DHCP snooping" is another effective rouge server attack mitigating technique that configures DHCP server port as "trusted" and DHCP client port as "untrusted" [12]. This drops traffic coming from the client port. This method eliminated network traffic and drops packets with MAC and CHADDR header mismatch.
6. Another study in [13] proposed that DHCP snooping is unable to detect spoofing attacks outside the server broadcast domain.

IV. METHODOLOGY

The methodology section is divided into three phases: the first section discussed emulated network topology; the second phase provides details on how the dataset was generated and finally, the third phase elaborates machine learning classification analysis on the data collected.

A. Experimentation

An emulated testbed topology is generated to conduct experiments. Raw network traffic is collected using Wireshark packet capturing software. The data collected from Wireshark is then converted into a DHCP-specific dataset. The experiment is conducted to test the server-side of the DHCP attack. During the experiment DHCP Denial of Service and DHCP Starvation attacks were conducted. The topology of the network includes a DHCP server, equipped with a pool of 255 available IP addresses. This DHCP server is connected to a core network switch which acts as a relay switch to five individual network switches. Each network switch uses a port mirroring technique to mirror network traffic to the core switch where the IDS is connected and Wireshark is integrated to collect network traffic. Each network switch has100 host computers with one attacker in each network as is displayed in figure 3. All five networks are assigned an individual subnet of 10.0.0.0/24, 10.0.1.0/24, 10.0.2.0/24, 10.0.3.0/24 and 10.0.4.0/24 respectively. The first IP address of each network is assigned to the DHCP server.

For each experiment, the environment was emulated four times. Initially, a DHCP spoofing attack was performed followed by a DHCP starvation attack having variable 'chaddr' value but same MAC address at layer 2. Afterward, a server-side DHCP attack was performed. Then features from each experiment were extracted and merged.

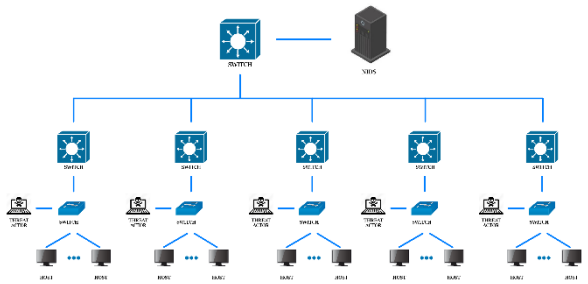

Figure 3: Network Topology

Figure 4: Wireshark Packet Capture

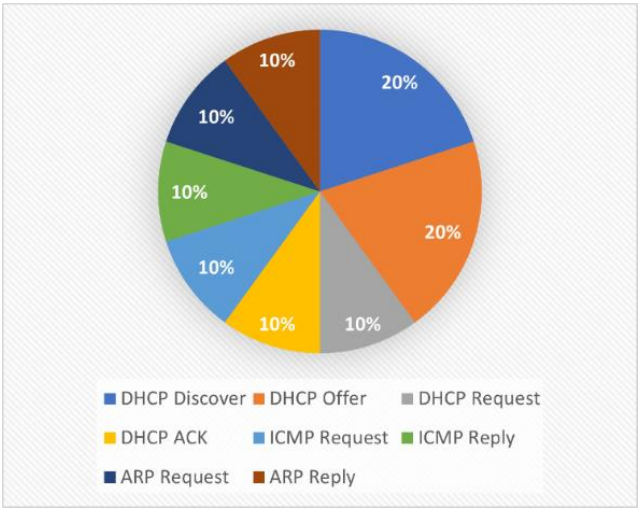

Figure 5: Total Protocol Message Types

ISBN # 978-969-23372-2-9

Table 1: Dataset Feature List

Feature	Data	Description
l2_source	MAC Address	Source of Layer 2
l2_destination	MAC Address	Destination of Layer 2
l3_source	IP Address	Source of Layer 3
l3_destination	IP Address	Destination of Layer 3
protocol	DHCP, ICMP, ACK,	Network protocols
message_info	ARP Request, ICMP Request, ARP Reply, ICMP Reply, DHCP Request, DHCP Offer, DHCP Discover, DHCP Ack	Protocol message type
chaddr	MAC address	Client hardware address in DHCP packet data

B. DHCP Based Dataset Generation

There are hundreds of features available in one D.O.R.A. process when extracting from Wireshark. However, not all of them are important for the detection of DHCP-specific attacks. After careful observation and analysis, the features considered in the research are mentioned in Table 1 as key features for our dataset. The dataset has been developed synthetically which incorporates DHCP, ARP, and ICMP messages as key features. Data points with the class label of 'Attack' and 'Non-Attack' are depicted in figure 6 while the total number of message types is shown in figure 5.

C. Training and testing machine learning classifier

To test the dataset's credibility, Random Forest and KNN classifiers were trained using RStudio. The evaluated metrics may differ depending on the architecture of the various networks.

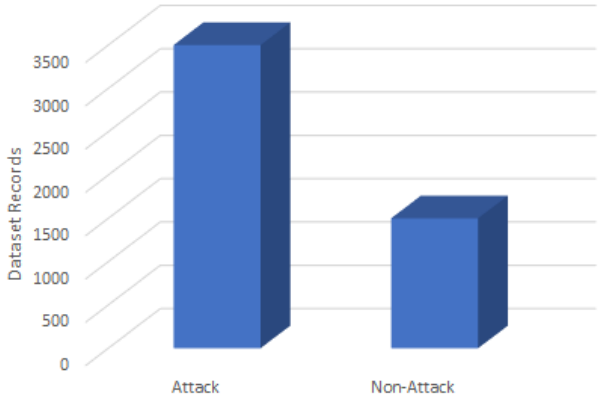

Figure 6: Frequency of Attacked and Non-Attacked Classes

Both the models were trained on 70% of the data whereas, the other 30% was used for testing purposes. Different results were obtained by changing one parameter in the training set of the models. For Random Forest, the 'ntree' (number. of trees) parameter was changed while for KNN the 'k' (number of neighbors) parameter was changed. Four graphs are obtained for each model with the following results: Accuracy, Precision, Recall, and f1-score. The graphs show variation in the aforementioned results with changing parameters in the model training set.

1) Random Forest classifier Results

Results obtained on increasing number of trees from a range of 50 to 500 trees in RF algorithm, depicted in Figure 7, are described below:

a. Accuracy: On an increasing number of trees the accuracy of RF increases dramatically from 95% to 97.5% and becomes quite constant after 100 trees, fluctuating between 97% to 98%.
b. F1-score: Based on precision and recall the highest f1 score is recorded to be 0.96 on 100 trees.
c. Precision: The RF precision graph shows a constant precision of 100% with an increasing number of trees.
d. Recall: The recall graph shows similar results as the accuracy and f1 score graph resulting in the highest value of 0.92 on increasing the number of trees to 100.

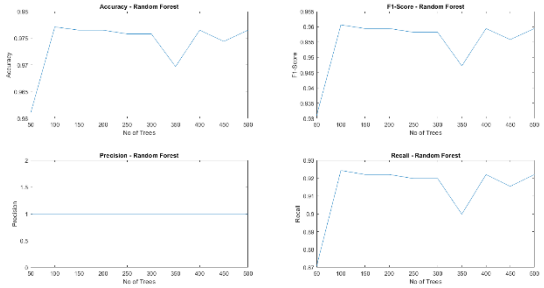

Figure 7: RF Results

2) K-Nearest Neighbors (KNN) classifier Results

Various results obtained on decreasing number of neighbors initially with 50 neighbors in KNN algorithm, depicted in Figure 8, are described below:

a. Accuracy: The accuracy of KNN reaches near 100% on decreasing the number of neighbors to 14.
b. F1-score: The highest recorded value of f1-score is 0.5 at 17 neighbors.
c. Precision: The precision of the KNN algorithm between the range of 50 to 1 neighbors flatulates between 0.99 to 1.
d. Recall: Displaying similar trends as the KNN accuracy graph, the recall reaches its maximum value of 1 at 14 neighbors

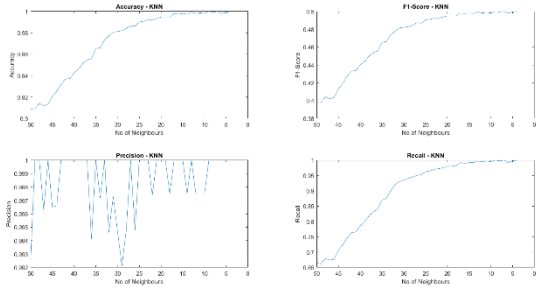

Figure 8: KNN Results

The performance of the RF algorithm is highest at 100 trees and for KNN it is recorded to be highest at 14 neighbors. The detailed analysis of the parameter values is depicted in Table 2.

ISBN # 978-969-23372-2-9

Table 2: Performance Analysis of RF and KNN Classifiers

Classifier	Accuracy	F1-score	Precision	Recall	ntree/K
RF	0.98	0.96	1	0.92	100
KNN	1	0.5	1	1	14

V. CONCLUSION

In this paper, the drawback for current research was first discussed due to the lack of protocol-specific datasets. Afterward, the main contribution of this paper includes the preparation of a DHCP-specific dataset explaining the need for features concerning their explanation, covering two different types of DHCP starvation attack and DHCP spoofing attack. The dataset contains 1500 benign traffic and 3500 attack labels. Random Forest and KNN are applied on the generated dataset where different values of Accuracy, F1 score, Precision, and Recall are calculated by changing the number of trees in Random Forest and the number of neighbors in KNN. The KNN classifier outperforms RF with near-perfect accuracy and precision of 1.

VI. REFERENCES

[1] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, pp. 1–29, 2021, doi: 10.1002/ett.4150.

[2] M. S. Alsahli, M. M. Almasri, M. Al-Akhras, A. I. Al-Issa, and M. Alawairdhi, "Evaluation of Machine Learning Algorithms for Intrusion Detection System in WSN," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 5, pp. 617–626, 2021, doi: 10.14569/IJACSA.2021.0120574.

[3] M. Aldaoud, D. Al-Abri, A. Al Maashri, and F. Kausar, "DHCP attacking tools: an analysis," *J. Comput. Virol. Hacking Tech.*, vol. 17, no. 2, pp. 119–129, 2021, doi: 10.1007/s11416-020-00374-8.

[4] S. Steinke, "Dynamic Host Configuration Protocol," *Netw. Tutor.*, no. March 1997, pp. 184–187, 2020, doi: 10.1201/9781482280876-45.

[5] N. Tripathi and N. Hubballi, "Exploiting DHCP Server-side IP Address Conflict Detection: A DHCP Starvation Attack," *Int. Symp. Adv. Networks Telecommun. Syst. ANTS*, vol. 2016-February, pp. 4–6, 2016.

[6] S. Naseer *et al.*, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, no. 8, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.

[7] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep Learning Approach Combining Sparse Autoencoder with SVM for Network Intrusion Detection," *IEEE Access*, vol. 6, no. c, pp. 52843–52856, 2018, doi: 10.1109/ACCESS.2018.2869577.

[8] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, "Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark," *IEEE Access*, vol. 6, no. c, pp. 59657–59671, 2018, doi: 10.1109/ACCESS.2018.2875045.

[9] N. Tripathi and N. Hubballi, "Detecting stealth DHCP starvation attack using machine learning approach," *J. Comput. Virol. Hacking Tech.*, vol. 14, no. 3, pp. 233–244, 2018, doi: 10.1007/s11416-017-0310-x.

[10] M. Yaibuates and R. Chaisricharoen, "A Combination of ICMP and ARP for DHCP Malicious Attack Identification," *2020 Jt. Int. Conf. Digit. Arts, Media Technol. with ECTI North. Sect. Conf. Electr. Electron. Comput. Telecommun. Eng. ECTI DAMT NCON 2020*, pp. 15–19, 2020, doi: 10.1109/ECTIDAMTNCON48261.2020.9090760.

[11] A. Jeklin, "DHCP Relay Agent Information Option," no. July, pp. 1–23, 2016.

[12] C. D. Snooping and I. P. S. Guard, "Configuring DHCP Snooping and IP Source Guard," *Database*, vol. 1, no. 20, pp. 1–16.

[13] S. Akashi and Y. Tong, "Classification of DHCP spoofing and effectiveness of DHCP snooping," *Proc. Int. Conf. Adv. Comput. Technol. Inf. Sci. Commun. CTISC 2019*, no. Ctisc, pp. 233–238, 2019, doi: 10.5220/0008099002330238.

ISBN # 978-969-23372-2-9