ISSN: 0974-5823 Vol. 7 No. 1 January, 2022

International Journal of Mechanical Engineering

Security against ARP Spoofing Attacks using Convolution Neural Networks

Govindavaram Madhusri
Assistant Professor, Dept. of Master of Computer Science. University College For Women, Kakatiya University, Warangal

Dr.Chakunta Venkata Guru Rao
Dean, School of Computer Science and Engineering & AI, SR University, Warangal

Abstract

There are several hazards and cybersecurity problems associated with ubiquitous computing, regardless its conveniences. To develop highly secured and enhanced ubiquitous environments, safety issues must be taken into account in the ubiquitous computing. You can determine your network card's IP and physical addresses using this mechanism called Address Resolution Protocol (ARP). It is intended to function in many kinds of situations. An attacker may mimic another host via ARP spoofing or obtain crucial information because of the lack of security features in its architecture. Detection of ARP spoofing attacks using a Convolutional Neural Network CNN is proposed in this study. Tracing routing is a method for determining network route changes. There is no need to change the ARP protocol in order to implement the suggested solution.

Introduction

The ARP protocol is one of the most fundamental yet crucial LAN communication technologies. A host's MAC address may be determined by its IP address via the ARP protocol. Sending an ARP request packet (broadcast) over the network accomplishes this. ARP reply packets from the affected host now include its MAC address (unicast). Gratuitous ARP packets may be used to broadcast the host's own MAC address under certain circumstances. All hosts have an ARP cache that stores all dynamic and static addressing mappings learned from the networks or specified by the administration. After a certain amount of time, the dynamic entries are removed from the database. An ARP request must be sent again if a host wishes to interact with a peer that has expired from the cache. It's impossible for the static entries to become old.
The ARP protocol does not have any states. Even if a host does not explicitly request an ARP response, it will nonetheless store the response in its cache. Most operating systems will overwrite an older ARP reply packet, even if it still has an unexpired dynamic ARP entry in the ARP cache.. As there is no means to authenticate the peer, all hosts just store the ARP answers they get without verifying them. ARP spoofing is a symptom of the underlying issue.
In order to mimic another host on the network, one must forge ARP packets. If you're using the most basic type of ARP spoofing, you'll see an attacker sending the victim bogus ARP answers. The duration between faked answers is substantially less than the operating system's ARP cache entry timeout period. This will prevent the victim host from making an ARP request for the attacker's target host. In the next section, existing detection and mitigation methods are briefly discussed.

Presently used Detecting and Mitigation Methods

Monitoring systems for ARP spoofing will be described successively.
S-ARP Protocol
An alternative to the ARP protocol has been suggested in [10]. Despite the fact that the S-ARP protocol is a long-term solution to ARP spoofing, the most significant limitation is that we must make modifications to the network stacks of all hosts. As a result, this is not particularly sustainable, as a stacks update throughout all computer systems is that both suppliers and consumers would not be pleased about. Although the authors of the research have argued that the extra cost of cryptographic computations is not considerable, we still have to deal with the additional expense of DSA.

Static MAC Entries

In order to prevent spoofing, it is necessary to add stable MAC addresses to every host for all other hosts. However, this is not a scalable solution and maintaining all these entries is a full-time task. Introducing mobile hosts like laptops into the network might lead to a disastrous failure of this strategy. In addition, it is well-known that some operating systems will replace static ARP entries if they receive Gratuitous ARP packets (GARP).

Kernal Based Patches

Attempts to defend individual hosts against ARP spoofing have been undertaken via kernel-based modifications like Anticap and Antidote. When a host ARP cache is updated by an ARP reply that contains a different IP address than the one that currently exists in the cache, Anticap prevents this. ARP protocol specification prohibits superfluous ARP responses, hence this is a violation of the ARP protocol. As soon as Antidote receives an ARP reply, it checks to see whether that response's MAC address matches the previously cached one. MAC addresses that have been previously learned are not updated if the previously learned MAC is still active.

There are two ways to get around this, both of which require that the ARP entry stored in the cache be the correct one. This creates a race between the perpetrator and the victim, which is a very dangerous scenario. MAC addresses may be blacklisted if an attacker gets his faked ARP entry into the hosts cache before they can from the actual host. The only way to reverse this is by administrative action. Consequently, we may infer that these technologies may not be able to identify ARP spoofing because of incorrect learning.

Passive Detection

It's possible to use PD to build a MAC address-to-IP address database by listening to the ARP requests and answers on the network. An ARP spoofing assault is ongoing if we see a change in any of these mappings in the future ARP transmission. ARPWATCH is the most widely used tool in this category.

There is a significant latency among learning the addresses mappings and detecting an attack using the passive technique. Once a detection programme has been running for a long period, it will be able to learn the faked answers in its IP to MAC address mapping database. The inconsistency will be identified and an alert raised only once the victim begins interacting with another host. This extra time may have given the assailant time to flee. The network administrator would also have to manually undo a faked entry learnt in the preceding case. Before beginning the programme, either manually input the necessary address mappings or build an attack free learning traffic. It's impossible for either of them to be scalable or mobile. Mobile hosts, such as laptops brought in by customers or visitors to a business, would be an appropriate illustration of this. This sluggish learning curve makes it impractical to deploy passive tools on a big network (1000+ hosts) and expect them to immediately detect assaults.

Unlike the active strategies, the passive ones do not have any intelligence and just search for a mismatch between ARP traffic and their database tables. This means that it is impossible to tell whether or not an ARP spoofing attempt is responsible for a newly observed address map-ping or whether or not the previously learned one was genuinely faked. To a reasonable degree of accuracy during a genuine assault, our method can identify the true MAC to IP address mappings.

Aside from being unreliable, the passive learning method is also dangerous. When ARP traffic is detected from them, a new address mapping is learned. For example, the production of random MAC and IP addresses per second in an ARP cache table overflow attempt will only result in the discovery of new stations rather than the reporting of attacks. One of the difficulties with previous ARP spoofing detection methods is that they all have flaws. ARP spoofing may be detected using an active strategy. In order to verify the legitimacy of ARP traffic, we send out ARP requests and TCP SYN packets. Attacks are detected more quickly and reliably using this strategy than passive approaches. In the case of a genuine attack, it can also identify the true mapping of MAC addresses to IP addresses to a reasonable degree of precision. In the next sections, a detailed explanation of the method is provided. Neural Networks (NNs) are designed to recognize patters through set of algorithms are modelled after the human brain (Fatayer et al., 2019). They interpret data through kind of clustering raw input, machine perception or labelling. Neural Networks can recognize only numerical patterns therefore any real-world data such as images, text and sounds must be pre-processed to numerical patterns before being processed by neural networks.

Neural networks incorporate two fundamental components which are (Hanif et al., 2019):

- Neurons (Nodes): They gets input values.
- Connections (weights): Connections holds weights which are adjusted during training neural networks.
- Neural networks are constructed from three different types of layers which are:
- Input: This layer receives external data.
- Hidden: This layer transforms the input data into output.
- Output: This layer provides the output.

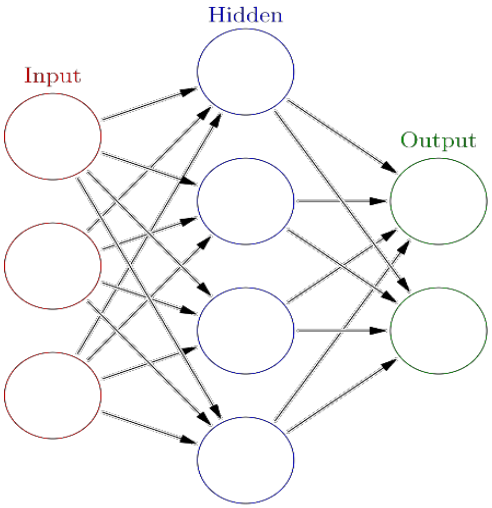

Figure 1 Neural Network

In this paper, a method that uses convolution neural network is proposed to detect ARP-Spoofing. This neural network can be trained to analyze TCP, UDP, ARP traffic in the network and alert when ARP- Attacks is detected.

Related Work

Several studies on network traffic analysis were recently conducted. Additionally, there are statistical approaches that have been increasingly popular over the last several years in addition to the more classic way based on the port number. (Shafi et al. 2018).

Lateef et al. (2019) explained that the traditional traffic analysis on IDS is based on port number, pattern matching, and finite state machine. This approach has several challenges in accuracy and detecting unknown attacks. To solve these problems researchers explored the use of other statistical and machine learning approaches in traffic analysis.

The use of Autoregressive Integrated Moving Average (ARIMA) statistical model in predicting traffic in networks and flag abnormal status was investigated by Alghamdi et al. (2019). Based on the discovery of trends that adversely affect the effectiveness of time series analysis, they have developed their own model. They found that ARIMA may assist decision-makers better manage traffic congestion by collecting and anticipating anomalous status.

An ARIMA model to predict the number of packets in the next minutes was created by Nezhad and colleagues (2016). They also developed a set of criteria to categorize normal and attack traffic based on the rise in the ratio of packets to source IP addresses during attack periods. For traffic classification, their suggested method had a 99.5 percent success rate.

There is an IoT traffic categorization method that may be employed with the combination of deep learning models created by Lopez-Martin et al. (2017). Research and education network RedIRIS provided them with the data they needed. Recurrent neural networks paired with convolutional neural networks provide the greatest results for networks categorization, according to the researchers.

Li et al. (2019) presented a method to detect malicious HTTP traffic on mobile networks. They utilized conventional neural networks to extract spatial characteristics in traffic. They created asandboxtogeneratedtrafficfortestingtheirmethod.Theyachievedmorethan99.4%accuracy in actualnetwork traffic.

Srinath et al, [11] developed an ARP spoofing detectionand prevention scheme by employing a centralized server. In this scheme, the drawback of voting based model is considered and in response a table based process is done by the centralized server which collects all the IP-MAC pairs of every node and maintains a legitimate host table. Based on this table of hosts, the malicious nodes performing ARP attacks are detected and prevented. However, one limitation of this scheme is that even unauthenticated host can also be registered in the network, which means the detection process at registration is not effective. Nam et al, [24] proposed a new collaborative framework for detecting the MITM attacks in the co-existing wired and wireless nodes. This approach utilizes the fair voting concept for achieving uniform transmission without loss and the neighboring nodes decide whether malicious activities arise from a node through this voting process. This approach also improves the voting fairness through filtering of the voting reply messages and determining key voting related parameters. It provides ARPattacks. In this system, the client/server model is equipped with IP-MAC table to detect and prevent ARP spoofing.

H. Ding et al, [12] presented ARP attack detection Bayes- based algorithm using the famous Bayesian theory. This detection algorithm calculates the host attacker probability using a prediction representation to determine the normal and abnormal features of the host nodes. However this detection algorithm has the drawback of wrong detection when the attack occurs less frequently and the subsequent features gets added to the normal host features. Sakhawat et al, [10] presented an agent based scheme for detection of ARP cache poisoning for detecting the MITM and DoS attacks. This approach has been installed over switch LAN system to detect and avert the insider malicious users. The detection accuracy of this scheme is high while there are only minimum effects on the system performance.

Prabadevi and Jeyanthi, [4] presented a security approach for protecting against the ARP attacks in large scale data centre networks. Similarly, Prabhadevi and Jayanthi, [5] also developed Time Stamp and Counter based approach (TSCBA) for detecting the ARP cache poisoning attacks. TSCBA utilizes packet analyzer and cross layer checker for pre-processing and inspection of Ethernet header and ARP header MAC addresses. Then these addresses are checked with ARP table to detect abnormalities and the time

stamp is generated along with alert messages to broadcast the entire system. This detection process is highly accurate and secured but the only drawback of using TSCBA is its cost expensive. Prabadevi and Jeyanthi, [3] also developed a similar approach for detecting and protecting against the ARP sniffing attacks through comparison of genuine IP- MAC pairs in ARP tables and Ethernet headers. These methodologies are dependent on the ARP table and Ethernet packet headers but when the packet formats are altered to resemble past session normal hosts, the detection process become complex.

Hong et al, [21] presented a protection model against theARP attacks through AES and RSA data encryption schemes. This method averts the attackers by authenticating the data and does not require expensive equipment or protocol modifications. However this model is lighter application and only used under constrained environments. Likewise, Singh et al, [23] presented two-phase validation system used for detecting the ARP attacks and averting them. This model works on the basis of validation of the new binding acknowledged by each host for sensing pair of ICMP packets to old and new binding of ARP cache while new hosts are validated through the ARP packets from the claiming hosts. This model also prevents the flooding attacks but its limitation is the complexity in validating a new host. Based on the insights from these models in literature, the problem of detecting the presence of existing attacker hosts of past misbehaviours is considered to be resolved in this research using BSVR-ARP scheme.

Convolution Neural Network

Neural networks are types of algorithms created as an inspiration for biological neural networks. Initially, the idea was to create an artificial system that would function the way the human brain works. The basis of neural networks are neurons that are interconnected depending on the type of network.

Usually, neural networks consist of layers where each layer consists of multiple neurons. Neural networks that have at least one hidden layer, the layer that is neither input nor output, are called deep neural networks. From that name comes a class of machine learning known as deep learning, where the main focus is deep neural networks.

There are many types of neural networks, but roughly, they fall into three main classes:

- Feed forward NN
- CNN
- RNN

The main difference between them is the type of neurons that form them and how the information flows through the network. In this article, we'll describe only the convolutional class of NN networks.

The CNN networks are a type of ANN networks, which is a ML technique. They've been around for a while but have recently gained more exposure because of their success in image recognition. A CNN is a powerful tool that we can use to process any kind of data where we can apply the convolution operation.

The success of convolutional neural nets is largely attributed to the fact that they can process large amounts of data such as images, videos, and text. Primarily, we can use them to classify images, localize objects, and extract features from the image such as edges or corners. They're typically composed of one or more hidden layers, each of which contains a set of learnable filters called neurons. As we've mentioned, these networks use convolution that is defined as:

$$g(x,y) = w * f(x,y) = \sum_{s=s_{min}}^{s_{max}} \sum_{t=t_{min}}^{t_{max}} w(s,t) f(x+s, y+t), \quad (1)$$

where f(x, y) is the input image and w is the filter or kernel. More intuitively, we can imagine this process looking at the illustration below:

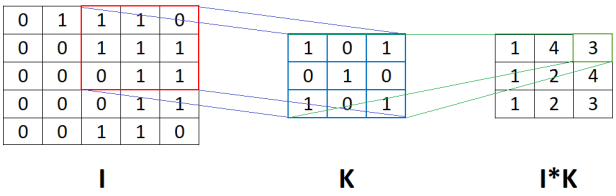

Figure 2 Convolution Neural Network

In the image above, we can see the matrix I to which we apply convolution with the filter K. It means that the filter K goes through the whole matrix I and element-wise multiplication is applied between the corresponding elements of the matrix I and the filter K. After that, we sum the result of this element-wise multiplication into one number.the ReLU activation function is being used after the convolutional layer. After that, often follows the pooling layer that applies filters in the same way as the convolutional layer but only calculating the maximal or average item instead of convolution. In the image below, we can see the example of convolutional layer, ReLU, and max pooling:

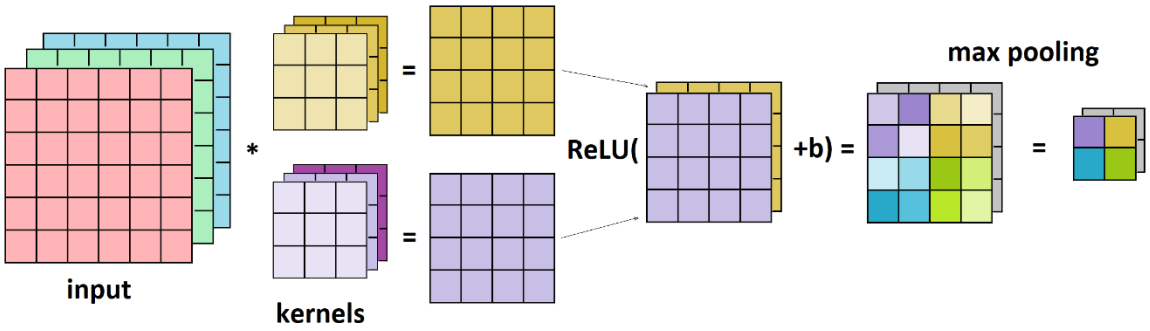

Figure 3 Layers of Convolution Neural Network

Performance Evaluation

Simulation Environment:

The suggested CNN technique is tested in MATLAB sample simulation software. The model most similar to CNN is the RTNSS which utilizes routing trace as the only source to detect the attacker host. Proposed CNN framework utilizes the ARP cache data along with the routing trace to detect the attacker based on attacker probability prediction. The experimental environment for proposed method is represented in table 1. Depending on these criteria, numerous hosts may be added to the network, which can be chosen based on the user's needs.

Table 1 Simulation Environment

OS	Windows10,64 bit
Processor	Intelcorei3 34703.2GHz
RAM	8GBDDR3
Storage	500GBIntel SSD
Networkbandwidth	1Gbps
Simulationtool	MATLABv.2019
Simulationtime	120 seconds
Networkarea	1000x1000m
Packetsize	80bytes

Proposed CNN scheme is evaluated and compared with that of the schemes of BVSR and RTNSS for spoofing detection. As said above, the CNN and the other schemes are tested in a framework similar to that of RTNSS. The detection schemes are compared in terms of detection time, attack accuracy, detection error, false detection probability and packet drop rate.

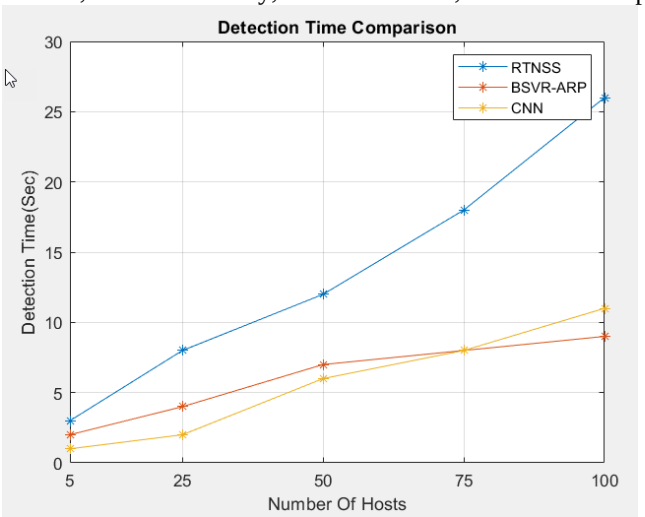

Fig 4. Detection time comparisons

The detection time is the time taken by the proposed CNN to analyze and detect the presence of attacker in the network. The graph presented in Fig. 4 shows the detection time taken by the proposed CNN and other existing ARP attack detection schemes. It can be seen that the proposed CNN scheme takes more time to detect the ARP attacks than the BVSR-ARP and less than the RTNSS.

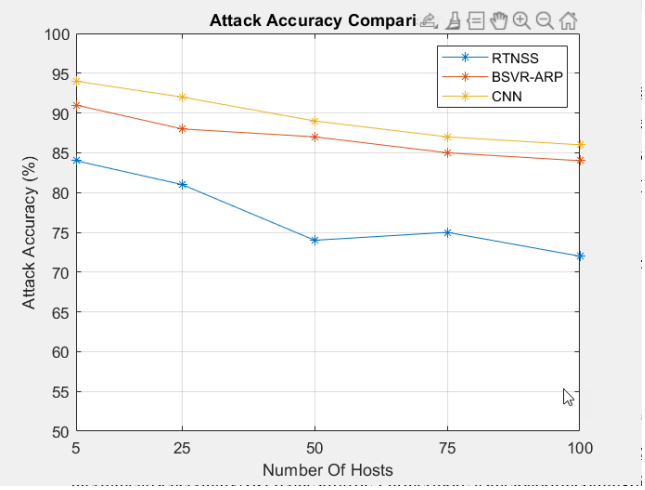

Figure 5 Attack Accuracy Comparision

Fig. 5 illustrates comparisons of the detection techniques in terms of the accuracy of assault detection. For this comparison, four attackers have been initialized when the number of hosts nodes is greater than 5 while 2 attackers are initialized for 5 host nodes. In this evaluation, the proposed model has detected higher percentage of attacks than the other compared schemes.Compared withBVSR and RTNSS in 100 host scenario, the attack detection accuracy in CNN is increased by around 8%.

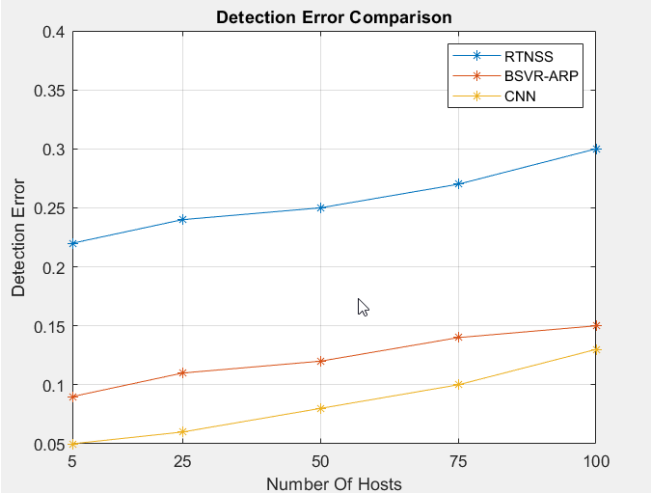

Figure 6 Detection Error Rate

Attack detection error of the detection schemes is compared in the graph shown in Fig. 6. It is a known fact that due to increased host number, the traffic increases and the detection modules fluctuate to determine the normal and attack features based on ARP packets. Yet, the proposed BSVR-ARP has minimal effects of traffic in the overall performance that can be understood by this comparison of detection error. The Proposed CNN scheme has very less detection error which is mostly near the negligible range than the other detection schemes.

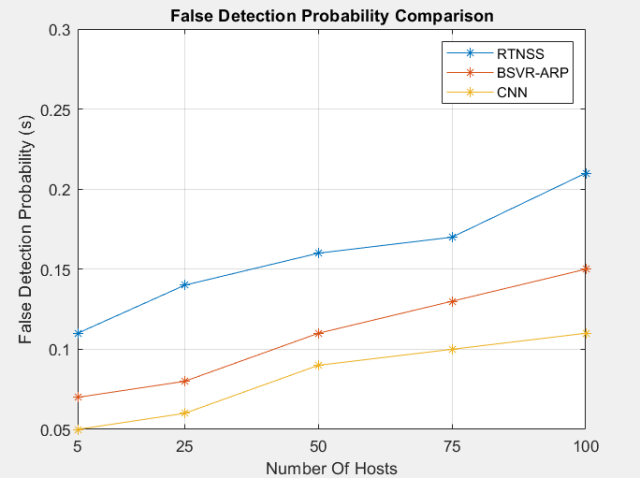

Figure 7 False Detection Probability Comparison

The comparison of detection systems in terms of false detection probability is shown in Fig. 7. False detection probability of a detection scheme is mostly dependent on the independent features that do not have any link with either of normal or attacker host. Literally, false detection is achieved when the number of ARP packets from a host increases due to various factors in such a manner, the detection scheme fails to identify the differing feature between attacker and normal host. In the above figure, it can be seen that theProposed CNN has less false detection probability than the other models.

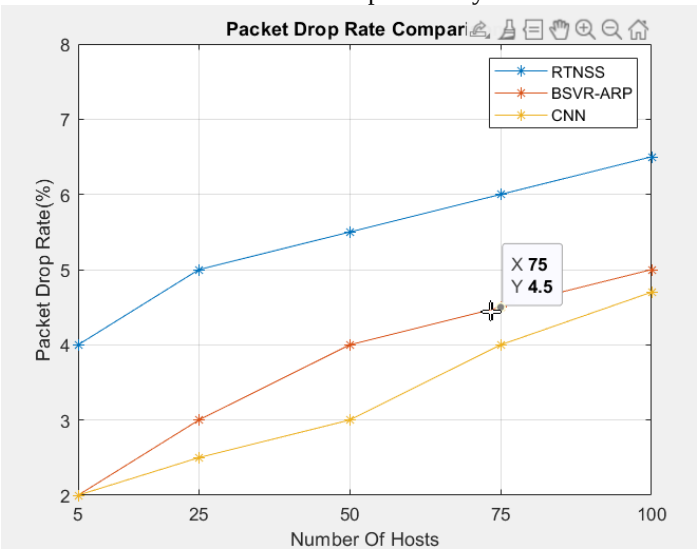

Figure 8. PDR (Packet Drop Rate) Comparisons

Figure 8 depicts the comparison results of the detection schemes in terms of PDR. Packet drop rate is directly dependent on number of host ARP packets and traffic congestion. When the no.of hosts in a network increases the number of request and response ARP packets increases and causes traffic congestion and delay. This is the ideal situation for packet loss as well as vulnerable attacks. The above figure shows that the proposed CNN has less packet drop rate than the other schemes which is mainly due to prior attacker detection and uninterrupted host to server communication.

Conclusion

An active method for detecting ARP spoofing was presented in this work. The suggested CNN algorithm is proved to be more quicker, more intelligent, and more scalable than passive detection methods. During a real assault, our method also identifies the right MAC to IP address mapping. This technique can still identify ARP spoofing in the presence of a modified stack, but it will not be able to deduce the right address. Our active ARP spoofing detection system requires just a little amount of time between learning new addresses and detecting spoofing.

References

1. A. Chonka, Y. Xiang, W. Zhou and A. Bonti, “Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks,” Journal of Network and Computer Applications, vol. 34, no. 4, pp.1097-1107, 2011.
2. A. M. AbdelSalam, W. S. Elkilani and K. M. Amin, “An automated approach for preventing ARP spoofing attack using static ARP entries,” International Journal of Advanced Computer Science and Applications, vol. 5, no. 1, pp. 105-112, 2014.

3. B. Prabadevi and N. Jeyanthi, "A framework to mitigate ARP Sniffingattacks by Cache Poisoning," International Journal of AdvancedIntelligence Paradigms, vol. 10, no. 1 -2, pp. 146-159, 2018.
4. B. Prabadevi and N. Jeyanthi, "Security Solution for ARP Cache Poisoning Attacks in Large Data Centre Networks," Cybernetics and Information Technologies, vol. 17, no. 4, pp. 69 -86, 2017.
5. B. Prabadevi and N. Jeyanthi, "TSCBA-A Mitigation System for ARP Cache Poisoning Attacks," Cybernetics and Information Technologies, vol. 18, no. 4, pp. 75-93, 2018.
6. C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel and M. Rajarajan, "A survey of intrusion detection techniques in cloud," Journal of network and computer applications, vol. 36, no. 1, pp. 42-57, 2013.
7. D. Bruschi, A.Ornaghi and E. Rosti, "S-ARP: a secure address resolution protocol," In Proceedings 19th Annual Computer Security Applications Conference, IEEE, pp. 66-74, 2003.
8. D. Gruss, C. Maurice and S. Mangard, "Rowhammer. js: A remote software-induced fault attack in javascript. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, Cham, pp. 300-321, 2016.
9. D. Moon, J. D. Lee, Y. S. Jeong and J. H. Park, "RTNSS: a routing trace-based network security system for preventing ARP spoofing attacks," The Journal of Supercomputing, vol. 72, no. 5, pp. 1740-1756,2016.
10. D. Sakhawat, A. N. Khan, M. Aslam and A. T. Chronopoulos, "Agent - based ARP cache poisoning detection in switched LAN environments," IET Networks, vol. 8, no. 1, pp. 67 -73, 2018.
11. D. Srinath, S. Panimalar, A. J. Simla and J. Deepa, "Detection andPrevention of ARP spoofing using Centralized Server," InternationalJournal of Computer Applications, vol. 113, no. 19, pp. 26-30, 2015.
12. H. Ma, H. Ding, Y. Yang, Z. Mi, J. Y. Yang and Z. Xiong, "Bayes- based ARP attack detection algorithm for cloud centers," Tsinghua Science and Technology, vol. 21, no. 1, pp. 17-28, 2016.
13. H. S. Kang, J. H. Son and C. S. Hong, "Defense technique against spoofing attacks using reliable ARP table in cloud computing environment," In 2015 17th Asia -Pacific Network Operations and Management Symposium (APNOMS), IEEE, pp. 592-595, 2015.
14. K. Ren, C. Wang and Q. Wang, "Security challenges for the publiccloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69 -73, 2012.
15. M. Conti, N. Dragoni and V. Lesyk, "A survey of man in the middle attacks," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2027-2051, 2016.
16. M. S. Song, J. D. Lee, Y. S. Jeong, H. Y. Jeong and J. H. Park, "DS - ARP: a new detection scheme for ARP spoofing attacks based on routing trace for ubiquitous environments," The Scientific World Journal, vol. 2014, 2014.
17. M. T. Khorshed, A. S. Ali and S. A. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing,"
18. P. Limmaneewichid and W. Lilakiatsakun, "P-ARP: A novel enhanced authentication scheme for securing ARP," In Proc. 2011 Int. Conf. on Telecommunication Technology and Applications, pp. 83-87, 2011.
19. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation computer systems, vol. 25, no. 6, pp. 599-616, 2009.
20. S. Hijazi and M. S. Obaidat, "A New Detection and Prevention System for ARP Attacks Using Static Entry," IEEE Systems Journal, pp. 1-7,2018.
21. S Hong, M.Oh and S. Lee, "Design and implementation of an efficient defense mechanism against ARP spoofing attacks using AES and RSA," Mathematical and Computer Modelling, vol. 58, no. 1 -2, pp. 254-260, 2013.
22. S. Khurana, "A security approach to prevent ARP poisoning and defensive tools," International Journal of Computer and Communication System Engineering, vol. 2, no. 3, pp. 431-437, 2015.
23. S. Singh, D. Singh and A. M. Tripathi, "Two-Phase Validation Scheme for Detection and Prevention of ARP Cache Poisoning," In Progress in Advanced Computing and Intelligent Engineering, Springer, Singapore, pp. 303-315, 2019.
24. S. Y. Nam, S.Djuraev and M. Park, "Collaborative approach tomitigating ARP poisoning-based Man-in-the-Middle attacks,"Computer Networks, vol. 57, no. 18, pp. 3866-3884, 2013.
25. W. Chu, S. S. Keerthi and C. J. Ong, "Bayesian support vector regression using a unified loss functions," IEEE transactions on neural networks, vol. 15, no. 1, pp. 29 -44, 2004.
26. Z. Trabelsi and W. El-Hajj, "ARP spoofing: a comparative study for education purposes," In 2009 Information Security Curriculum Development Conference, ACM, pp. 60-66, 2009.
27. Z. Trabelsi and W. El-Hajj, "On investigating ARP spoofing security solutions," International Journal of Internet Protocol Technology, vol.5, no. 1, pp. 92, 2010.