KeAi CHINESE ROOTS GLOBAL IMPACT

International Journal of Innovation Studies

MITIGATING ARP SPOOFING ATTACKS USING BAYESIAN SUPPORT VECTOR REGRESSION AND STOCHASTIC MARKOVIAN GAME MODEL

Dr.C.Divya
Assistant Professor, Department of Computer,
Science Dr.G.R.Damodaran College of Science,
Coimbatore -641014
divya.c.sekar15@gmail.com

Abstract
The reliability of computer networks highly depends on the performance of Address Resolution Protocol (ARP) protocols which are used to discover the MAC address in the cache table for the related IP address. This process is considered the gateway for many attacks. The attacks are detected and prevented using many methods. However, it does not use the past session information of host to detect attacks, resulting in false detection. This limitation is effectively solved by Bayesian Support Vector Regression-ARP (BSVR-ARP) and Stochastic Markovian game model ARP (SM-ARP). This paper presented a comparative study of these two methods in detecting and preventing attack hosts. The BSVR-ARP model uses the transmission error and configuration changes to detect the malicious host. In addition, it uses the past session's information in attack detection. Whereas the SM-ARP model uses the previous session information of the inactive host in attack detection and attacks during packet forwarding to effectively predict the attack host. The performance analysis of the two models shows that SM-ARP has better performance than BSVR-ARP.

Key Words: Address Resolution Protocol, ARP Spoofing, MAC address, IP address, Stochastic Markov game model, Bayesian Support Vector Regression.

Introduction

The rapid development in internet usage leads to advancement in the networking structure and increasing availability. The major aspects of internet usage are communication and information retrieval. However, security-related issues on the internet degrade the satisfaction of internet users [1]. ARP protocols are the gateway of attacks. The ARP protocols are the communication protocols that discover the Media Access Control (MAC) related to the IP address. While exchanging the MAC address between the hosts, ARP lacks authentication. The intruders use this gap to damage the system or create traffic using ARP spoofing [2]. The IP-MAC relation introduces ARP spoofing attacks such as Denial of Service (DoS) and Man-in-the-Middle (MITM) attacks [3]. Therefore, IP/MAC addresses are managed for secure internet usage.

Many static and dynamic approaches are developed to solve ARP spoofing attacks in wireless or wired LAN networks [4]. However, it is complex to develop prevention techniques for ARP spoofing because it requires modifications in the structure of the network and the tools required for it are more expensive [5]. So attack prevention measures must be taken with reduced modifications in the network structure. The model is constructed based on the Bayesian Support Vector Regression-ARP (BSVR-ARP) technique, which considers the information of alteration in host configuration and errors during transmission to detect the ARP spoofing attack [6]. Based on the prediction, the Software Denied Network (SDN) controller is used for recovery, which may discard the host from the network or repair

the host to include it in the network. This model solves the problem of false detection and prevailing attack of the prior session. But recently, attackers have used the packet forwarding relay strategy and inactive hosts of prior session to initialize the attack, which is considered a major threat and difficult to solve. The BSVR-ARP model does not solve the problem caused due to packet forward relay strategy. It monitors the active hosts of current and previous session to detect the attack hosts, reducing the attack detection performance.

The problem of detecting the attack host is solved using the Stochastic Markovian game model ARP (SM-ARP) [7]. This model monitors packet forwarding strategy to detect the attack. Furthermore, information on the inactive host of the previous session is utilized to detect and prevent the network from attacking the host. The limitation of detecting ARP spoofing in the BSVR-ARP model could be solved using the SM-ARP. This paper presents the comparative study of BSVR-ARP and SM-ARP models in detecting ARP spoofing attacks. In comparison, the performance of SM-ARP is higher compared to the BSVR-ARP model.

RELATED WORK

Many methods have been developed in the recent years for detecting and preventing against the ARP spoofing and related attacks. Moon et al. [8] developed routing trace-based network security system (RTNSS) for averting ARP spoofing attacks while also reducing the impact of server-agent misconfiguration. This model increased the ARP attack detection probability by 20%, but it has shortcomings in terms of past session attacks and packet forwarding attack strategies. Tchendji et al. [9] suggested an efficient Bayes security-based protocol (EBSP) to prevent the ARP caches from ARP spoofing attacks. In this approach, Software Denied Network (SDN) based virtual networks are secured from ARP spoofing attacks utilizing an efficient Bayes algorithm. It has two models, ARP controller and ARP virtual machine (VM), which is taken to detect and prevent ARP spoofing attacks. The developed protocol is executed using Mininet 2.2.2 simulation tool. This model overcomes the limitations of Bayesian Support Vector Regression (BSVR) and Naive Bayes (NB) in detecting the attacks. While analyzing the result, the EBSP model takes a longer detection time compared to BSVR and NB. Sowah et al. [10] developed an intrusion detection model for Mobile-Ad-Hoc Networks using Artificial Neural Networks (ANN). This approach recognizes man-in-the-middle (MITM) attacks by implementing ANN. This model is executed utilizing the NS2 simulation tool and noted that it has an attack detection accuracy of 88.235%. However, this model is particularly used to predict the known type of attacks. Manhas et al. [11] preferred the Deep Q learning approach to construct the MITM attack detection model. In this approach, malicious activities in the network are identified using Deep Q learning. This model is tested with the attack dataset containing 40% test data and 60% training data. However, this model is complex to train.

Li et al. [12] detected MITM attacks in communication-based train control systems utilizing Long Short-Term Memory (LSTM) and SVM. In this approach, Edge intelligence is implemented to improve the computing ability of the system. Then cross-layer defence strategy is executed using LSTM and SVM models, which merge the attack detection probabilities from operation log files and the control parameter sequences. Next, the Bayesian defence model is constructed to work against MITM attacks. In addition, the resource allocation for communication is implemented using the Asynchronous Advantage Actor-Critic algorithm. This method is evaluated using the data collected from the Beijing subway Yanfang line. However, this model not detects complex and advanced types of attacks. Kiran et al. [13] contrasted the performance of SVM, NB, Decision Tree (DT) and Adaboost techniques in detecting the MITM attack. This model categorized the Sensor480 dataset as normal or malicious utilizing the classification models. The experimental result shows that SVM and Adaboost models gained an accuracy of 0.9895. However, the presented model not detects novel attack types. Kponyo et al. [14] compare the performance of linear classification models such as Support Vector Classifier (SVC), Gaussian NB, linear SVC, logistic regression (LR), Random Forest (RF), K-Nearest Neighbour (KNN), Decision Tree (DT) and Gradient boosting in detecting the MITM attack. In this approach, ARP analysis is performed to identify the MITM attack. The dataset with 5300 feature vectors is evaluated, and it observed that linear SVC and Gaussian

NB outperform other models. However, the compared model takes a long time to detect the attacks. Ahuja et al. [15] proposed a hybrid model based on LSTM and CNN to detect ARP spoofing attacks. This model collects log features to categorize normal data from malicious data. This model is evaluated using a network traffic dataset and achieved an attack detection accuracy of 99.73%. However, this model has high computational complexity. The methods in literature show that the need for highly accurate detection model due to the significant impact of the ARP attacks. The extensive analysis has shown that the open challenges such as new attack strategies and complexity in analysis have greatly paved the way for designing new and advanced detection models.

ARCHITECTURE OF ARP SPOOFING DETECTION MODEL

Figure 1 shows the overall architecture of the routing trace based ARP model in which the two models, BSVR-ARP and SM-ARP are implemented. This model comprises of agents and a server; each has a separate management system. The agent is employed to protect from ARP spoofing, whereas the server is employed to manage the agent's information and make decisions in determining the attack hosts. The modules included in the agent are agent manager, protection and detection modules. The attack detection process is carried out in the agent detection module. In this module, the attack host detection algorithms BSVR algorithm and SM-ARP are implemented to detect and prevent from attack host. This module incorporates the host recovery process and routing trace-based attack verification. This detection module compares the IP-MAC pairs from routing trace packets utilizing the prediction probability on the global ARP cache table and ARP cache list. The obtained information is passed to the agent manager to confirm the occurrence of the attack. After notifying the agent protection module to alter the ARP type, it initiates the recovery procedure. In addition, command forwarding, communicating and self-status checking are the tasks carried out agent manager with the server manager.

The server module incorporates an agent controller, server protection, server manager and server monitor. The server monitor performs packet monitoring, monitoring of the hosts' MAC address, whitelisting of the attacker node and checking for duplication. Next, the agent controller maintains the information of agents such as network configuration, control and prevention from threats. The decision-making process on attackers and initiation of host removal from the network is carried out by server protection. Agent and server manager do similar tasks, but server manager does prominent tasks compared to agent manager. It manages the network by collecting information on server protection modules, hosts and agents. In addition, the server manager maintains the ARP cache entry repository, log and network policies. The interaction between the agent and server is carried out utilizing this module. The comparative study of attack detection mechanisms such as BSVR-ARP and SM-ARP implemented in agent detection, server protection and agent protection modules is done in this paper. In addition, this paper explains the SDN controller-based attack verification and recovery process in all the modules.

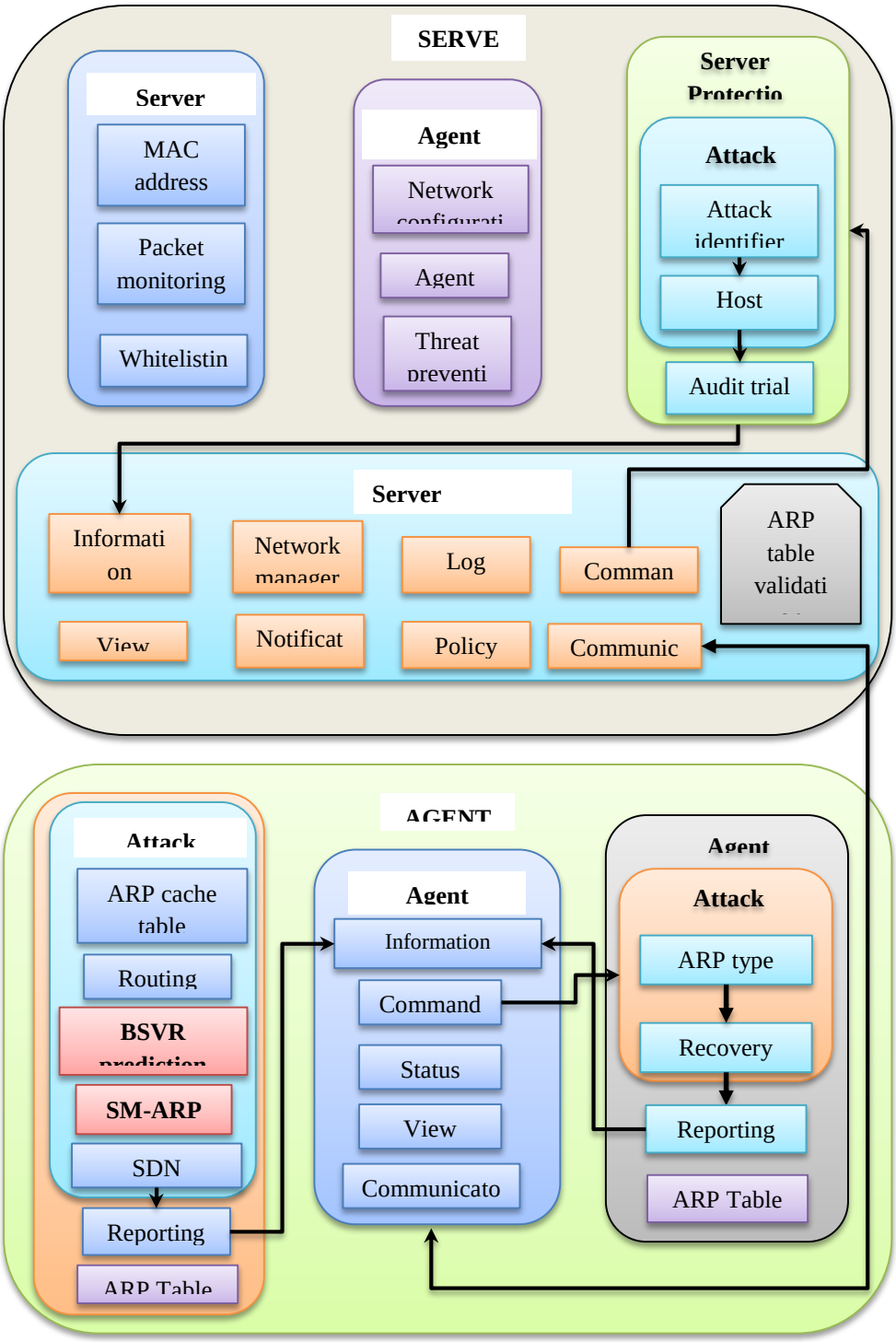

Figure.1.ARP Spoofing Prevention model with BSVR and SM-ARP methods

ARP Spoofing Attack Detection based on BSVR probability prediction

The mapped IP-MAC address in the cache is used to detect the ARP attacks, but it does not detect the attacks from past sessions. This drawback is due to the lack of system cooperation in fetching the information from the previous session. Typically, there are two types of ARP frames: response packets and requests. The ARP response packet stores the IP of the responding host and the IP-MAC address of the requesting host. The global ARP cache stores the information of each response packet, and it is used

to determine the attack features. The drawback of assessing the previous information is first solved to detect the attacks effectively, and it is done with Software Defined Network controllers. This model collects the ARP packets for the creation of a global knowledge base that could be used to detect the attack. The problem is based on transmission errors, and modifications in the host configuration alter the IP-MAC mapping cache, resulting in false detection. These limitations are overcome by BSVR based probability prediction approach, which can predict each host's attack feature. Similar models are created for spoofing detection, but it considers the frequent attack patterns as normal features, leading to false detection.

The attacks are detected by discovering the mismatches in IP-MAC or duplication in IP-MAC addresses utilizing the network information of each host and global ARP cache. The configuration of subsequent hosts is determined to detect attacks, and the network structure has to be modeled in the initial stage. Assume $N = \{N_i, 1 \leq i \leq m\}$ which specifies network hosts placed in the network containing m features of the host. Both the normal and attackers are included in the host set. The initial probability of host H_j, to be an attacker is fixed as $P(H_j)$ and the non-attacker as $P(\overline{\overline{H_j}})$ by considering past knowledge. The total probability is one, as per the probability theory. It is expressed as,

$$P(H_j) + P(\overline{\overline{H_j}}) = 1 \quad (1)$$

Here, $P(H_j)$ representing the attacker host $P(\overline{\overline{H_j}})$ denotes the normal host, and the sum of the attacker and the normal host is always one. The host feature N_x is identified by SDN, and then Bayesian theory is utilized to compute the posterior probability $P(H_j|N_x)$. It is expressed as,

$$P(H_j|N_x) = \frac{P(H_j)P(N_x|H_j)}{P(H_j)P(N_x|H_j)+P(\overline{\overline{H_j}})P(N_x|\overline{\overline{H_j}})} \quad (2)$$

$$P(\overline{\overline{H_j}}|N_x) = \frac{P(\overline{\overline{H_j}})P(N_x|\overline{\overline{H_j}})}{P(H_j)P(N_x|H_j)+P(\overline{\overline{H_j}})P(N_x|\overline{\overline{H_j}})} \quad (3)$$

Here, N_x represents the feature of the host, and it could be normal or an attack. Based on the value of $P(H_j|N_x)$ SDN classifies the host as the attacker. The presence of an attack is confirmed when the value of $P(H_j|N_x)$ is high. But consideration of this value alone might result in a false prediction. Therefore, equation (1) has to be modelled with respect to Eq. (2) and (3). This limitation is resolved by monitoring the subsequent host features and maintaining the feature list. The feature list is constructed with $n + a$ features $FS = \{FS_i, 1 \leq i \leq n + a;\ FS_i \in N\}$, the attack and normal features are denoted as a and n. The posterior probability is simplified iteratively using BSVR, and it verifies the host H_j as normal or attack the host. The posterior probability FS_i could be modelled as

$$P(FS|N_x) = \frac{P(FS)P(N_x|FS)}{P(N_x)} \quad (4)$$

Where FS denotes the feature of the hyper parameter of Support Vector Regression utilized for formulation. These features before distribution are required to calculate the $P(FS)$ probability value. The flat distribution is considered to estimate $P(FS)$, which is insensitive to FS and for the validation of $P(N_x|FS)$, different values are allocated. $P(N_x|FS)$ value could be derived by applying integral over the host H_j in f, the sampling function with the assistance of Taylor expansion. It is expressed as,

$$P(N_x|FS) = \int_0^{n+a} P(N_x|f, FS)\ P(f, FS)df \quad (5)$$

Equation (5) is rewritten by enabling the prior probability. It is expressed as,

$$P(N_x|FS) = Z_f^{-1}Z_{FS}^{-1} \int_0^{n+a} \exp\big(-FS(f)\big).df \quad (6)$$

Here, Z indicates the transferred frame's noise function. The probability of FS_i is extended while considering the Taylor expansion of $FS(f)$. It is written as,

$$P(N_x|FS_i) = Z_f^{-1}Z_{FS}^{-1} \int_0^{n+a} \exp(-FS_i(f) \dots \dots FS_{n+a}(f)).df \quad (7)$$

The probability of attack prediction is modelled in Eq. (8) based on the probability values of BSVR hyper parameters.

$$P(H_j|FS_i \dots \dots FS_{n+a}) + P(\overline{\overline{H_j}}|FS_i \dots \dots FS_{n+a}) =$$

$$\frac{P(H_j)P(FS_i \dots\dots FS_{n+a}|H_j)}{P(H_j)P(FS_i \dots\dots FS_{n+a}|H_j) + P(\overline{H_j})P(FS_i \dots\dots FS_{n+a}|\overline{H_j})} + \frac{P(\overline{H_j})P(FS_i \dots\dots FS_{n+a}|\overline{H_j})}{P(H_j)P(FS_i \dots\dots FS_{n+a}|H_j)+P(\overline{H_j})P(FS_i \dots\dots FS_{n+a}|\overline{H_j})} \quad (8)$$

Equation (8) is simplified for forming the BSVR final iterative probability by maintaining the conditional probability based on host configuration and feature list.

$$P(H_j^i|FS_i(N_x)) = \frac{P(H_j^i)P(FS_i(N_x)|H_j^i)}{P(H_j^i)P(FS_i(N_x)|H_j^i)+P(\overline{H_j^i})P(FS_i(N_x)|\overline{H_j^i})} \quad (9)$$

In equation (9) limit is applied to determine the host H_j is either an attacker or a non-attacker. The attacker host is identified effectively using the BSVR prediction technique, and the verification process is done to prevent negligible errors. During verification, $P(H_j^i)$ at each stage is refreshed by the SDN controller, which confirms the end of the previous session. The session's information is transferred to permanent memory only if the previous session is logged out. The host H_j^i is considered an attack if it satisfies the following condition.

$$P(H_j^i) > P_t, \qquad 0 < P_t < 1 \quad (10)$$

Where, P_t denotes threshold value, and it plays an important role in the probability of exclusion and false detection. Based on the step sizeμ, expected attackersα , the value of P_t is set between $P_{min} < P_t < 1$. The number of attackers A_n in a defined time t is monitored with minimum probability, which is represented as P_{min}. The value of P_{min} is determined when the probability of non-attacker is close to 0. The threshold value is considered to fix the value ofP_{min}. If P_{min} is less than the value of the threshold, then the new value of P_{min} is considered, whereas the value of P_{min} is closer to 1, then 1 is taken as the threshold. The threshold value could be fixed using $P_t + (\alpha - A_n)\mu$, which resolves the errors caused due to changes in host configuration and transmission errors. The threshold value of P_t is reduced when $A_n > \alpha$ and in the case of $A_n < \alpha$ the value of P_t is increased, which enhances the accuracy of attack detection and discards the misdetection. The SDN controller separates the host from the network once the attackers are verified and start the recovery process to the affected host. The network replaces the attack host with the new host or uses new routes to function. During the recovery process, the less damaged hosts are recovered, and they could be included in the network, or else it is not used. This model did not recognize the attack initiated using the packet forward relay strategy and the idle hosts. This limitation degrades the attack detection performance.

SM-ARP scheme of ARP Spoofing Attack detection and protection

The limitation of BSVR-ARP is resolved by modeling the attack behaviour using the SM-ARP model. This approach detects the attacks commenced by idle host and packet forward relay strategy. Game theory models are implemented when two or more players aim to achieve the same goal. The mathematical analysis for detecting the best strategy for every player to obtain the goal is provided by this model. The SM-ARP model uses the number of source host packets and relay node buffers to determine the game. Here the players taken are malicious nodes and legitimate nodes. The fair play rulers are imposed utilizing the stationary Markov Model, and the game is modelled as a multiplayer game. Equal opportunity is given to both the nodes of attack and genuine hosts. The penalty is given to the player with unfair and offensive play results. This idea is applied in the network; here, the packet loss is considered an offense against the player and that player is identified as the attacker. The player who has an unfair offense will initiate the packet loss. This process forges the actual destination address of IP-MAC. Hence the attacker nodes detected by SM-ARP are discarded from the network immediately to ensure packet transmission reliability.

The system model incorporates three nodes: source, relay and destination node. The source node transmits the packets to the destination node directly or with the assistance of relay nodes. The relay node considers the queue to reject or accept the packets and transfers the previously received packets or its packets. In this work, for simplicity, two players are taken, which is the source and relay node of equal buffer size. The source node incorporates a single buffer for the storage of created packets. The relay

node includes two buffers: an internal buffer and a forward buffer. The received packets from the source are stored using a forward buffer, whereas own generated packets are stored in internal buffers. The relay and source node's packet generation rate is represented as G_r and G_s. The buffers are empty, and the stationary Markov process is taken for modelling the system at the beginning of every new session. The state of the Markov process is the identifier of each buffer's occupancy state, which has two possibilities: occupied or empty. This is valued as N^{N+1} states, and it is equivalent to $2^3 = 8$ states having N number of players. The modeling of the game is done by using the criteria that every player in the game knows the present state of other nodes and their possible actions. This situation allows tackling the action of every player by others, but the selection of action is not direct. It averts the attacker without any complications. The suggested game model is $(N, \{S_x\}, \{U_x\}), x \in N$, when the game has N set of players with the strategy S. Here $\{S_x\}$ represents the player$x$'s strategy with the subsets $\{s_x \in S_x\}$, $\{U_x\}$ denotes the utility function of x-th player. The strategy set outlines the behaviour pattern of each player. Considering this, the proposed solution attains Nash equilibrium if all the players in the game have the best possible strategies and equal opportunities. If the following condition is satisfied, the Nash equilibrium S* is obtained by the strategy x

$$\forall_x \in N, \forall_{s_x} \in S_x, \quad U_x(s*_x, s*_{-x}) \geq U_x(s_x, s*_{-x}) \tag{11}$$

Here s_x represents the strategies used by other players than x. The player in the game tries to achieve the goal without cooperating with other players, and fair play is considered to get the details of the attacker. $n \times n$ Transition matrix is included in the Markov game, and it is represented as T having n number of states. The element of T that is p_{ij} is used to define the likelihood of shifting the state i from j with the time slot n+1. The function of the current state of the game q^n is used to define the action sets A^n. Then the game's next state is procured from q^n in the present time slot and the action set. The transition function is created as $T(q^{n+1}|q^n, A^n)$ which is utilized to determine the probability from q^n to q^{n+1} the next state. By utilizing this function, x-th player's immediate utility is computed, which is accounted as the function of the current state and the game action set. It is represented as, $U_x^n = f(A^n, q^n)$. The payoff is considered the players' ultimate goal in any game. The proposed model uses probability distribution ($P_{q^0}^S$) of the game over the action set and the game's state in the entire game to compute the average payoff for the player x of using any strategy S and q^0 the initial state function. It is represented as

$$U_x(S, q^0) = \lim \frac{1}{T} EP_{q^0}^S [\sum_{n=1}^{T} U_x^n(A^n, q^n)\,] \tag{12}$$

Here, $EP_{q^0}^S[\,]$ denotes $P_{q^0}^S$ factor's expectation operation value.

When the strategy action set depends only on the game's current state rather than the entire game, then the proposed strategy is taken as stationary. The probability distribution when the stationary condition is expressed as $\Pi(\delta)$ with $\Pi(\delta) = \Pi(\delta) \times \mathrm{T}(\delta)$. Here $\times$ denotes the multiplication operation of the matrix, $\mathrm{T}(\delta)$ refers to the state transition matrix. For the state k, xth player's stationary utility function is represented as

$$U_x(\delta) = \sum_{q_k \in Q} \Pi_k(\delta) E[U_x(q_k, \delta)] \tag{13}$$

Here, $E[U_x(q_k, \delta)]$ denotes xth player's expected utility function in the state k. Equation (13) could procure the Nash equilibrium for the q^0 initial state.

$$\forall_x \in N, \forall_{s_x} \in S_x, \quad U_x(q^0, \delta_x^*, \delta_{-x}^*) \geq U_x(q^0, \delta_x, \delta_{-x}^*) \tag{14}$$

While applying this concept to the simplified two-player game model enables flexible or reliable relay methods. The game's state is represented as $\{Q_S, Q_R, Q_F\}$, the elements in the set specify the source, relay internal and forward buffer's packet count. The probability distribution of the possible set of actions is used to define each stationary player's mixed strategy in the Markov game model. The source model's strategy space is expressed as (p_{sd}, p_{sr}, p_{sw}) where p_{sd} represent the probability of packet transmission from source to destination, p_{sr} denotes the probability of transmitting from source to relay, p_{sw} refers to the waiting packets at the source. Likewise, the relay node's strategy space is modelled as $(p_{rd}, p_f, p_{rw}, p_{ac}, p_r)$ here, p_r denotes the probability of rejecting the source packets, p_f indicates the

probability of forwarding source packets, p_{rd} represent the probability of packet transmission from relay to destination, p_{rw} indicates the probability of packets waiting at the relay, p_{ac} refers to the probability of accepting the packets of the source node. The state transition probability is determined based on the mixed strategy function.

The possibility of attackers is determined by utilizing the expected payoff of the players as well as the rewards of the players. The delivery reward R^d is received by each node and the forwarding reward is R^f is received by the relay. The transmission cost is denoted as C, and the cost for transmission of packets from source to relay is denoted as C_{sr}^t. Similarly, the transmission cost between the nodes is formulated. C^w is used to represent the waiting or delay cost that occurs due to the collision of attackers. C^r denotes the cost of relay delay. The utility functions of source and relay nodes are calculated using Eq. (13) based on the probability, reward and cost functions. It is expressed as,

$$U_1(\delta) = \Pi_2(\delta) \times \{(R^d)\} + \Pi_4(\delta) \times \{(R^d)\} + \Pi_5(\delta) \times \{p_{sd}(R^d - C_{sd}^t) + p_{sr}.p_{ac}(-R^f - C_{sr}^t - C^r) + p_{sr}(1-p_{ac})(-C^w - C_{sr}^t) + (1-p_{sr}-p_{sd})(-C^w)\} + \Pi_6(\delta) \times \{(R^d - C^w)\} + \Pi_7(\delta) \times \{p_{sd}(1-p_{rd})(R^f - C_{sd}^t) + p_{sd}.p_{rd}(-C^w - C_{sd}^t) + p_{sr}(1-p_{rd}).p_{ac}(-R^d - C_{sr}^t - C^w) + p_{sr}(1-p_{ac}+p_{rd}.p_{ac})(-C^w - C_{sr}^t) + (1-p_{sr}-p_{sd})(-C^w)\} + \Pi_8(\delta) \times \{(R^d - C^w)\} \quad (15)$$

$$U_2(\delta) = \Pi_2(\delta) \times \{(-C_{rd}^t)\} + \Pi_3(\delta) \times \{p_{rd}(R^d - C_{rd}^t) + (1-p_{rd})(-C^w)\} + \Pi_4(\delta) \times \{(-C_{rd}^t - C^w)\} + \Pi_5(\delta) \times \{p_{sr}.p_{ac}(R^f)\} + \Pi_6(\delta) \times \{(-C_{rd}^t)\} + \Pi_7(\delta) \times \{p_{rd}(1-p_{sr}-p_{sd})(R^d - C_{rd}^t) + p_{rd}(p_{sr}+p_{sd})(-C^w - C_{rd}^t) + (1-p_{rd}).p_{sr}.p_{ac}(R^f - C^w) + (1-p_{rd})(1-p_{sr}.p_{ac})(-C^w)\} + \Pi_8(\delta) \times \{(C_{rd}^t - C^w)\} \quad (16)$$

The utility function for the flexible relay method is obtained by adding the probability of forwarding source packets p_f to Eq. (15) and (16).

Likewise, overall packet failure or drop probability likelihood could be altered by the probability of bit error (p_b). The source and relay node's probability of loss could be formulated as,

$$\mathrm{p}_e(s|d) = p(\bar{s}, r|d) + p(\bar{s}, \bar{r}|d) \quad (17)$$

$$\mathrm{p}_e(r|d) = p(s, \bar{r}|d) + p(\bar{s}, \bar{r}|d) \quad (18)$$

Here $p(s, r|d)$, $p(\bar{s}, r|d)$, $p(s, \bar{r}|d)$ and $p(\bar{s}, \bar{r}|d)$ are the probability of both the packets being received successfully, i.e., source packets, relay packets, and source and relay packets failure in the attacker collision process, which leads to the malicious or unfair entry. These probabilities are applied to Eq. (15) and (16) to get the utility function for the method performed adaptively during the collision. In this stage, both the flexible and reliable relay methods are carried out adaptively. Therefore, malicious nodes in the packet forwarding process are detected effectively, and the attacker's presence is verified utilizing the SDN controller. The detection of attacks initialized using pack forwarding relay strategy in ARP spoofing problem is effectively done by the SM-ARP model. This model overcomes the limitation of attack detection in the BSVR-ARP model.

EXPERIMENTS AND RESULTS

The SM-ARP and BSVR-ARP models are evaluated in a prototype simulation model developed utilizing MATLAB (v.2013a). The configuration for simulating the proposed model is given in table 1. The 100 hosts are enabled in the initial setup and could be decreased or increased as per the requirement.

Table.1. Simulation environment

OS	Windows 8, 32bit
Processor	Intel core i5 3470 3.2 GHz
RAM	4GB DDR3
Storage	500GB Intel SSD SC2CT060A3 ATA device
Network bandwidth	1 Gbps
Simulation tool	MATLAB v.2013a

Simulation time	120 seconds
Network area	1000x1000 m
Packet size	80 bytes
Maximum No. of hosts	100

Performance Evaluation

The performance of the SM-ARP model is compared against the performance of the BSVR-ARP model in terms of detection error, attack detection accuracy, false detection probability, packet drop rate, and detection rate and detection time. The information from the previous communication sessions is considered to detect the attack effectively.

Table.1. Performance comparison of BSVR-ARP and SM-ARP

Number of hosts	Detection Error (%)		Detection Accuracy (%)		Detection Time (s)		FDP		PDR (%)	
	BSVR-ARP	SM-ARP	BSVR-ARP	SM-ARP	BSVR-ARP	SM-ARP	BSVR-ARP	SM-ARP	BSVR-ARP	SM-ARP
5	0.15	0.05	91	95	2	1	0.07	0.05	3	2
25	0.27	0.075	89	90	2.75	2.25	0.08	0.07	3.75	3
50	0.31	0.098	86	89	3.85	3.1	0.17	0.09	4.01	3.75
100	0.33	0.11	80	85	4.75	4	0.19	0.15	4.75	3.98

Table 1 presents the performance of BSVR-ARP and SM-ARP models. The number of hosts increases during simulation to predict the attack detection performance. The detection error achieved by BSVR-ARP is 0.33%, and SM-ARP is 0.11 for 100 hosts. The SM-ARP model attained an attack detection accuracy of 85% for 100 hosts, which is 5% higher than the BSVR-ARP model. The FDP in 100 hosts is 0.19 for BSVR-ARP and 0.15 for SM-ARP. Similarly, the value obtained for PDR on 100 hosts is 3.98% in SM-ARP and 4.75% for BSVR-ARP. The detection time increases with the number of hosts, but it is comparatively lower in SM-ARP.

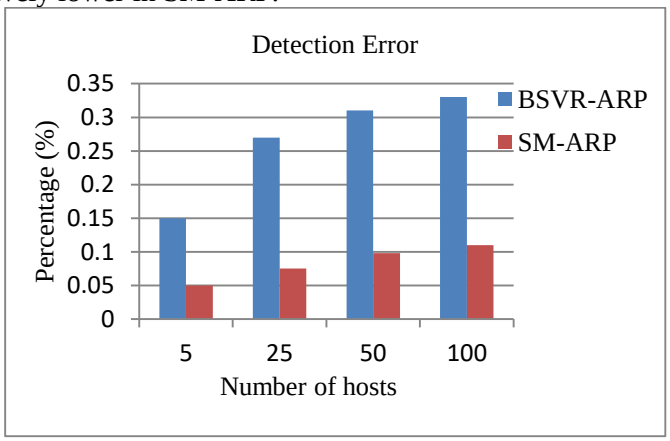

Figure.2. Comparison of detection error of BSVR-ARP and SM-ARP

Figure.2 displays the detection error of SM-ARP and BSVR-ARP models while examining the 5, 25, 50 and 100 hosts. The result shows that SM-ARP has less detection error compared to BSVR-ARP.

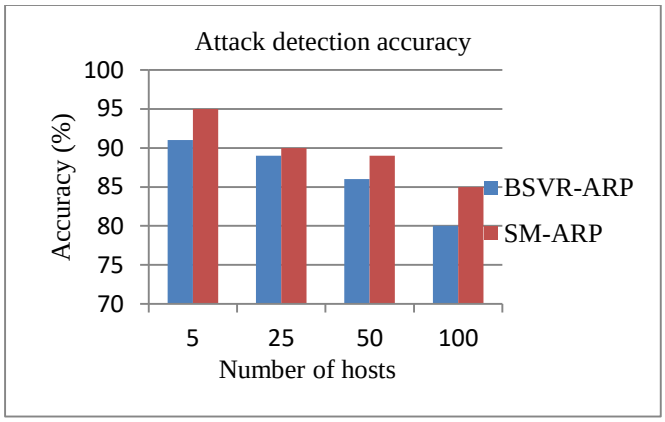

Figure.3. Attack detection accuracy of BSVR-ARP and SM-ARP

Figure.3 illustrates the attack detection accuracy of BSVR-ARP and SM-ARP. The attack detection accuracy is reduced when the count of the host is increased. The detection accuracy of SM-ARP is higher than BSVR-ARP.

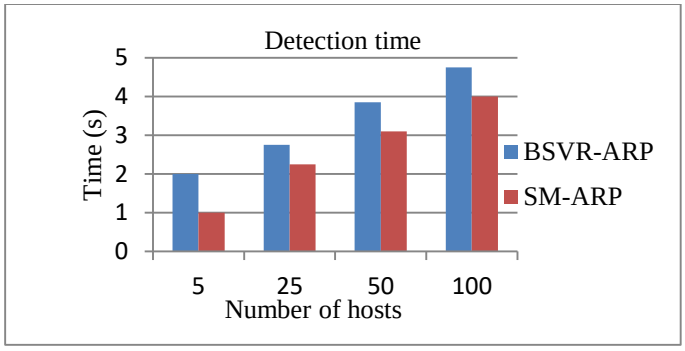

Figure.4. Detection time of BSVR-ARP and SM-ARP

Figure.4 shows the time taken to detect the attack. The detection time of SM-ARP is lower than BSVR-ARP.

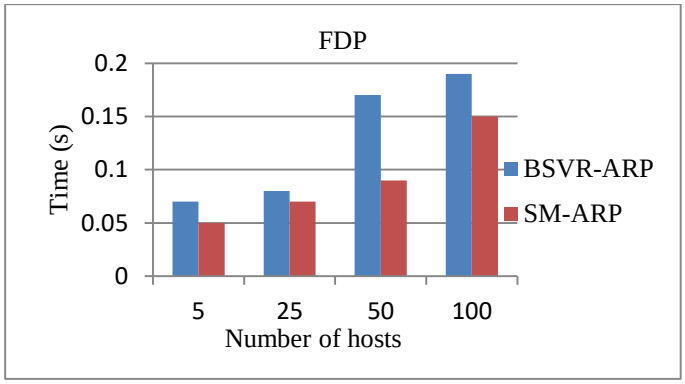

Figure.5. FDP of BSVR-ARP and SM-ARP

Figure.5 displays the FDP of BSVR-ARP and SM-ARP. The SM-ARP model has less FDP value compared to BSVR-ARP.

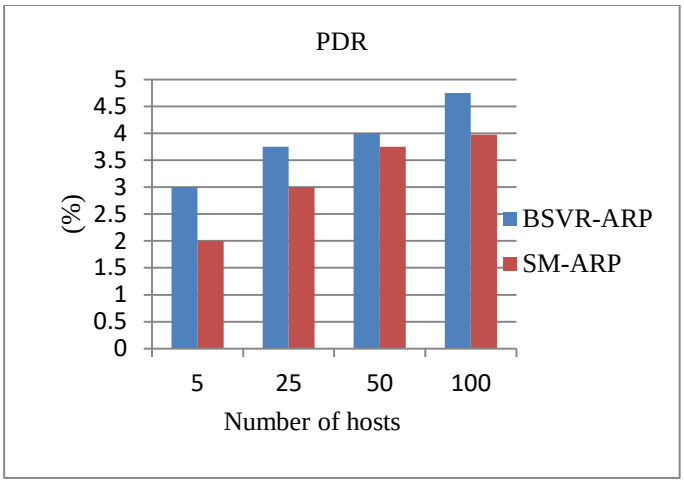

Figure.6. PDR of BSVR-ARP and SM-ARP

Figure.6 represents the PDR of BSVR-ARP and SM-ARP models. The value of PDR in SM-ARP is less compared to the SM-ARP model. But the value of PDR is proportional to the number of hosts.

CONCLUSION

This paper presented the comparative study of BSVR-ARP and SM-ARP attack detection mechanisms. BSVR-ARP model uses the information of the ARP cache table, routing trace, and log to detect the attack, but it does not detect the attack caused while forwarding the packets. This might lead to false detection, and this drawback is resolved by utilizing the SM-ARP model. This model detects the attacks that originated during the packet forwarding process. The attack detection performance is increased in SM-ARP compared to BSVR-ARP.

REFERENCES

1. Hijazi, S., & Obaidat, M. S. (2019). Address resolution protocol spoofing attacks and security approaches: A survey. *Security and Privacy*, *2*(1), e49.
2. Trabelsi, Z., & El-Hajj, W. (2010). On investigating ARP spoofing security solutions. *International Journal of Internet Protocol Technology*, *5*(1), 92.
3. Nam, S. Y., Kim, D., & Kim, J. (2010). Enhanced ARP: preventing ARP poisoning-based man-in-the-middle attacks. *IEEE communications letters*, *14*(2), 187-189.
4. Liu, Y., Dong, K., Dong, L., & Li, B. (2008). Research of the ARP spoofing principle and a defensive algorithm. *International Journal of Communications*, *4*.
5. Usmani, M., Anwar, M., Farooq, K., Ahmed, G., & Siddiqui, S. (2022). Predicting ARP spoofing with Machine Learning. In *2022 International Conference on Emerging Trends in Smart Technologies (ICETST)* (pp. 1-6). IEEE.
6. Divya, C., & Christopher, X. (2019). Security against ARP spoofing attacks using Bayesian support vector regression. *Int J Innov Technol Explor Eng (IJITEE)*, *8*, 2278-3075.
7. Divya, C., & Christopher, X. (2019). SM-ARP: Stochastic Markovian Game Model for Packet Forwarding Based ARP Spoofing Attacks Detection. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9(2), 2310-2318.
8. Moon, D., Lee, J. D., Jeong, Y. S., & Park, J. H. (2016). RTNSS: a routing trace-based network security system for preventing ARP spoofing attacks. *The Journal of Supercomputing*, *72*(5), 1740-1756.
9. Tchendji, V. K., Mvah, F., Djamegni, C. T., & Yankam, Y. F. (2021). E2BaSeP: Efficient Bayes-based security protocol against ARP spoofing attacks in SDN architectures. *Journal of Hardware and Systems Security*, *5*(1), 58-74.
10. Sowah, R. A., Ofori-Amanfo, K. B., Mills, G. A., & Koumadi, K. M. (2019). Detection and prevention of man-in-the-middle spoofing attacks in MANETs using predictive techniques in Artificial Neural Networks (ANN). *Journal of Computer Networks and Communications*, *2019*.

11. Manhas, S., Taterh, S., & Singh, D. (2020). Deep Q learning-based mitigation of man-in-the-middle attack over secure sockets layer websites. *Modern Physics Letters B*, *34*(32), 2050366.
12. Li, Y., Zhu, L., Wang, H., Yu, F. R., & Liu, S. (2020). A cross-layer defense scheme for edge intelligence-enabled CBTC systems against MitM attacks. *IEEE Transactions on Intelligent Transportation Systems*, *22*(4), 2286-2298.
13. Kiran, K. S., Devisetty, R. K., Kalyan, N. P., Mukundini, K., & Karthi, R. (2020). Building an intrusion detection system for IoT environment using machine learning techniques. *Procedia Computer Science*, *171*, 2372-2379.
14. Kponyo, J. J., Agyemang, J. O., & Klogo, G. S. (2020). Detecting end-point (EP) man-in-the-middle (MITM) attack based on ARP analysis: A machine learning approach. *International Journal of Communication Networks and Information Security*, *12*(3), 384-388.
15. Ahuja, N., Singal, G., Mukhopadhyay, D., & Nehra, A. (2022). Ascertain the efficient machine learning approach to detect different ARP attacks. *Computers and Electrical Engineering*, *99*, 107757.