CONCEPTUAL RESEARCH PAPER

COMPUTING TRENDZ Vol VII No. 1,2 , January - December 2017 Print ISSN : 2230-9152, Online ISSN 2456-138X

Detection of Fake Request and Response using Certificates Information and IP Address of SDN

Vivek Srivastava[1*], Lal Babu Yadav[2], Dr. Ravi Shankar Pandey[2]

[1]Department of Computer Science & Engineering, Birla Institute of Technology, Mesra-Ranchi, India

[2]Department of Computer Science & Engineering, Birla Institute of Technology, Mesra-Ranchi, India

ABSTRACT

Data communication in network facilitates access of the content stored at the remote servers using the IP address. In SDN these communications are monitored centrally to be out in network operation. The attackers can change the IP address and disturb the communication to authenticate services or IP address certificate authorities are generated security certificate to the remote servers. Attackers also able to modify the host address of the certificate which again creates failure network operations. These attacks create unnecessary overload at the switches/controller side in SDN.

In this paper, we have proposed a model for stopping the malicious IP's at the switch/controller level to reduce the load of the controller. For finding the actual attacker, In this formal model, we have considered the ascertain of certificate and IP address. We have considered one observer which records the certificate and IP address of the request/responses. These request and response are converted in LTS. These LTSes are merged for finding the malicious attacker location. We have demonstrated our proposal with the examples.

Keywords: SDN, LTS, Security Requirements, Certificate Authority.

Computing Trendz (2017). DOI: 10.21844/cttjetit.v7i1-2.3

Introduction

SDN concept emerges to improve network operations. These network operations are monitored by individual units in the traditional system. SDN integrates responsibility of these network operations at a higher level to provide solutions based on the analysis of all network devices present in the system. This integration increases the network reliability in the ideal case. In general condition, this centralization may decrease network reliability due to attacks either at the controller side or at a different layer of the SDN. The security is major challenges in SDN. These challenges are to provide prompt action in the case of attacks, to identify the categories of network attack and to manage the overhead of resources for security services. The solution of SDN network security should incorporate. The security views of the network attack including excess control policy in anautonomous manner. These security solutions can be provided at a different layer of SDN architecture application layer, northbound interface, control layer, southbound layer, and the data layer. Some examples are FRESCO, VeriFlow, CloudWatcher, etc. During request and response in SDN, a MITM attack is serious issues. The attacker modifies the IP address in the security certificate. In data communication, the IP address and the certificate for the security plays an important role. The certificate authority issues certificates to a different entity, which has one or series of the certificate. The attacker changes the contents of the certificate to disturb the network. The attacker also modifies the IP address by corrupting the routing tables.

In this paper, we have argued that if the malicious

Corresponding Author: Vivek Srivastava, Department of Computer Science & Engineering, Birla Institute of Technology, Mesra-Ranchi, India, e-mail: viveksrivastava@united.ac.in

How to cite this article: Srivastava, V., Yadav, L.B. & Pandey, R.S. (2017). Detection of Fake Request and Response using Certificates Information and IP Address of SDN. Computing Trendz 7(1&2): 13-23

Source of support: Nil

Conflict of interest: None

requests are stopped before the processing by the controller then existing security infrastructure may be improved. We have proposed a formal model for detecting malicious request on the basis of details of certificate and IP address. Using this information we can stop to enter malicious request/response at the entry level of switches/controller. This solution is proposed to enrich security at the data layer.

This paper is organized into five sections: section 1 is for the introduction. Section 2 describes the work associated with the security aspect of SDN. Section 3 illustrates the proposed model. Section 4 contains examples and the last section is used to conclude our proposal.

Related Work

SDN is basically based on centralized control and visibility of a network. The attackers can poison the network to damage the ARP of the networks. These kinds of attack may damage the topology information of a network. This topology information plays an important role in SDN to maintain the information about the network structure. These attacks can be in the category of spoofed IP address, DoS, hijacking, and MITM attacks. S. Hong et al. have developed a software component TopoGuard as a new security extension of SDN controllers, which automatically detects in real time network topology position attack. They have presented a threat model. Each host profile is maintained at the control side to track the mobility. This mobility is capture using by monitoring packet in messages and ingress port id. These host profiles are used for resolving security issues.

Switches are a backbone of the SDN. The malicious attack on switches creates failure in network operations. Po-Wen Chi et al. have proposed attacker models for switches and also suggested detection mechanism. They have categorized the attacks on a switch as a passive attack and active attack. In the passive attack, switch behave normal and in inactive the behavior of switch abnormal. The attacker uses incorrect forwarding, packet manipulating and malicious weight adjusting (the weight is used for selecting the optimal switch during load balancing). Their detection technique based on that if any switch does not use program flow rule for processing the packets. Two detection algorithms have been proposed named forwarding detection and waiting detection, forwarding is used for correct packet forwarding while waiting is used for dispatching process of the packet.

The controller is a dominant part of SDN and which super wises the all network operations centrally the failure of the controller damages the whole network architecture. The threat always tries to attack the controller side. These attacks can be stopped before reaching the controller. The controller safety will improve. S. M. Mousavi et al. haveproposed an attacker detection model on the based on the entropy of destinations IP address, which detects the malicious IP addresses before the controller. They have used the entropy concept which is based on randomness. Entropy is used for finding the probability of occurrence of an event with respect to the total number of events. The destination IP address of incoming packets describes the detection methodology. For storing the information about the IP address and the number of packets received, an extra table is created at the controller side. The entropy is calculated on the basis of the number of packets arrive and the total number of packets. They have used mininet simulator for simulating the result.

The OpenFlow protocol is used for communication in the SDN. The IP address in the switches does not change during mobility. S. Namal et al have proposed OpenFlow Host Identity Protocol (OFHIP) architecture which in reaches OpenFlow protocol for changing the IP address of switches for secures mobility.

J. Kim et al. have proposed a framework for security of SDN which is based on virtualization. They have considered centralized firewall system and DDoS system. Their limitations of the existing system are also discussed in this paper. The objective of the SDN security is explored like prompt action for the new attack, autonomous defense in network attack, network load balance aware resource allocation. They have also listed the research issues like to prevent unauthorized control of switches to save from a single point of failure of the controller and to take automatic serious action attacks on the network.

E. Ahmed et al. [6] have discussed different security solutions for SDN and categorized on the basis of usability of solutions at different layers of SDN. The different domains of the solution are secure design, security audit, security enforcement policy, security enhancement, and security analysis. For secure design FRESCO and FortNox are given, for security audit Verificare is given, for security enforcement policy FLOVER, Perm OF, and VeriFlow have been given, for security enhancement FleXam, CloudWatcher and L-IDS have been given and for security analysis OpenWatch, AVANT-GUARD and Header space analysis have been given for Application layer, control layer, southbound interface, and data layer.

Proposed Approach

The proposed architecture for the security of SDN at the data layer

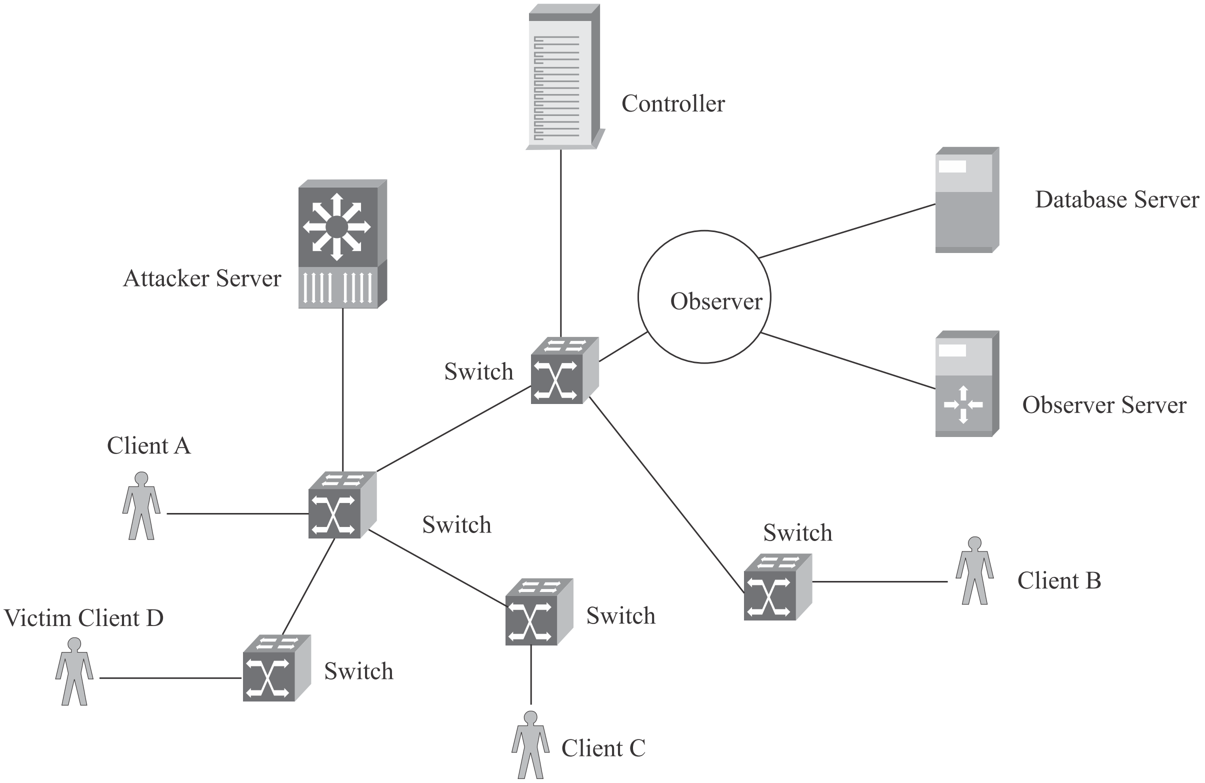

Figure 1: SDN Security architecture with the observer

In the proposed architecture we have used one server which is named as observer server for storing the IP address of the request and responses with the security certificate. We have used one database which is named as IP address and certificate table. The certificate table contains the different vectors as MAC address, IP address and the certificate tree address. Traceroute command is used to obtain the MAC address, IP address and certificate chain of that IP address. In this paper, we have considered the flow table for obtaining the port number of corresponding addresses. The observer server is continuously recorded and computed this information and stored in the database. We have used the proposed algorithm for computing the malicious point and IP address and then MAC to stop the attack before the flow table/controller. We have also proposed a formal model of security in SDN using the information of the IP address and certificate as stored in the proposed table.

Formal Model of Security in SDN

In SDN controller is decided the optimal path for any request and response. The control always is heavily loaded due to many requests and responses are generated through different hosts which are connected with different switches. The unauthorized request and responses may generate additional load on the controller side. These unauthorized requests and responses can be stopped before the controller or switches, which improve the performance of the controller.In this proposal, we have proposed a formal model of security in SDN. The certificate plays an important role in identifying unauthorized access/malicious packets. The IP address and certificate together can be used to detect malicious hosts. We have used a certificate and IP address for identifying a malicious host. A formal model is used to detect the malicious host after merging the history of traces. Our work is refined over a period of time and accuracy for detection increases over a period of time. Certificates play an important role in security. X.509 is a security standard for public key infrastructure (PKI) which is integrated with SSL/TLS protocol suite [2]. This protocol uses server authentication. The server generates a certificate to the client for declaring as security service providers. The certificate contains information hostname, communication peer information, issuer information, and signature. The malicious host modifies hostname information which is present in the subject attribute in certificate and length of chain certificate is very long. The certificate validity is based on the following facts:

- Certificate chain should be complete and must have a root certificate.
- All certificates present in the chain should be within the expiry date.
- All signatures should be verified and the root certificate should be present for checking the validity of the certificate.

Currently, many certificate authorities are present which are authenticating the service providers. In this situation, any service provider may have either one certificate or set of the certificate. These certificates are arranged in the form of tree data structure or forest data structure. If more than one root certificate is available for any service provider then the certificate is arranged in the forest data structure. The root certificates are a self-signed certificate and generated by the same certificate authority. The certificate authority is also capable of the issued certificate to other service providers which are called intermediate certificates and the final certificate or end-user certificate issued by the certificate authority. The intermediate and end-user certificates are generated to overcome the workload on our certificate authority. The root certificates are always stored in the safe area and intermediate certificates are used for normal certification. In the certificate chain, a number of intermediate certificate authority are responsible more attack. One certificate may be issued for a different domain. We have recorded traces T_1, T_2, T_3 …T_n. Where T_i is consists of the IP address and certificate of the tree.

	Certificate tree
IP_1,IP_2,……..IP_n	T_1………T_n
IP_{n+1},……….IP_{n+k}	T_{n+1}……T_{n+k}
IP_{n+k+1}…..IP_{n+k+l}	T_{n+k+1}…..T_{n+k+l}
$IP_{n+k+l+1}$…..$IP_{n+k+l+m}$	$T_{n+k+l+1}$…$T_{n+k+l+m}$

Table 1: Trace record

The certificate tree contains a root node which is for root certificate and intermediate nodes for intermediate certificate and a leaf node for end certificate.

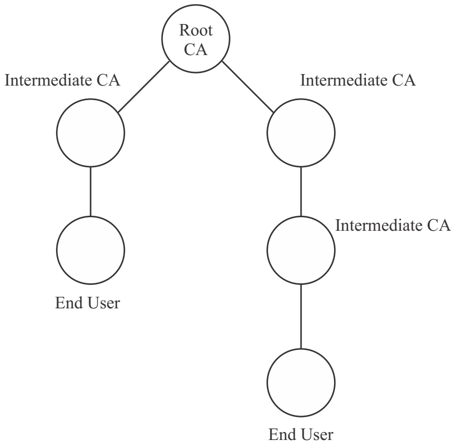

Figure 2: Tree T1

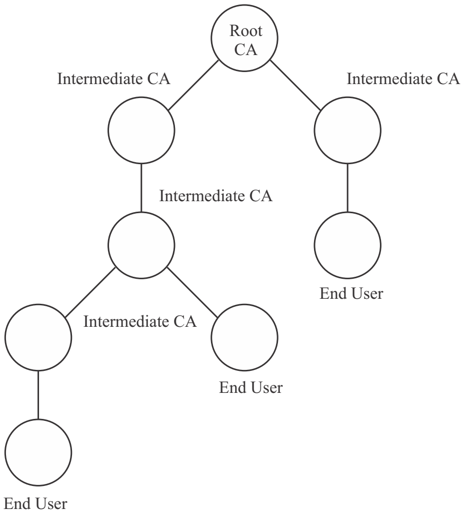

Figure 3: Tree T2

In this formal model, we are considering IP address, MAC address, port number, root certificate, certificate of the tree and attributes of the certificate for finding the unauthorized request and response. Our formal model consists of four tuples < CI, F, where CI is a set of input which consists of (ci, mac_i, pn_i, rc_i, t_i, a_i) is state which stores the state information of certificate state information of IP address, certificate root information, mac information and status of port number.

c takes certificate information and signature. It checks the validity of the signature. takes the certificate tree and the first node. It finds the first node is a root or non-root node. takes trace route and mac number of IP address is return by the IP. Traceroute is taken by and IP address returns the port number. F is a set of flowtables search functions. Each f_i consists of three-tuples certificate state c_i and $c_{i'}$ < c_i, ser, $c_{i'}$>. The ser function finds the security state using the flow table. It has any one value from the set {s,u,un}. S stands for secure, u for un-secure and un for IP address is not present in the flow table. a transition function which takes packet information as input (IP address, certificate state, root status, mac no, port no.) and security information from flow table and tells about whether request and response trusted or untrusted.

ip = IPState = CA × subject → IP address
c = Certificate_state = CA × signature →
true / false
croot = Tree × first node → true / false
mac_no = traceroute × IP → mac
pn = traceroute × IP → pn
= IPState × certificate_state × croot × mac_no ×
pn_no → secure / insecure

The composition of transition function and search result of flow table find the security status of the resulting transition. If IP is present in the flow table then either it is blocked or allowed. The blocked tells security status as un-secure and allow indicates security status as secure. If IP is not present in the flow table then the security status of IP is unknown and depends on the finding of the observer. The rules for security compositions are given below. If both functions ti and fi are secure then resultant security status is secure. If anyone is un-secure then the result is un-secure. If flow table result is unknown then the result depends on the transition function.

i o fi= s.s=s
=s.u=u
=s.un=s
=u.un=u

3.3 Proposed Algorithm

We have proposed an algorithm to find the malicious host on the basis of the history of traceroute of request and response and certificate details in SDN. Our algorithm includes the following steps:

Step 1: The collection of traceroute and certificate

We have stored the information of IP address, MAC address and port number from the trace route of request and response, and also stored the certificate details in the form of tree or forest data structure. This information is stored at each state of LTS. The next state is referred to as a different time stamp. The next state may receive the change in IP address, MAC address, on or in certificate details. In our LTS each state represents different timestamps and the transition function maps the change occurs during the change of states. Our LTS is generated for each request and response at a different time stamp. Any LTS has initial and final state including some intermediate state. The number of states depends upon a number of timestamps.

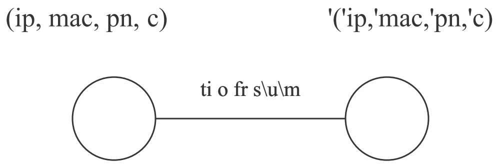

Figure 4: Change of transition state

Step 2: Merging the traces and detect the malicious hosts

In this step, several LTSes are recorded during data transmission of requests and responses which are from the hosts present in the SDN. Thus LTSes can be merged to identify the malicious host on the basis of the history of traceroute and details of certificates. the k-tails algorithm is used for merging the LTSes. The merging stands with the common initial state of different request and responses of the same source and same destination. If any state has a common MAC address, IP address, and certificate details, then the merging of the state can be done.

If the value of IP address, MAC address, pn, and certificate are same then next state is created and then the transition is from source path.

If the next state has a different IP address, different MAC address, different certificate and same port number then one new state is created and a path is different.

If the next state has a different IP address, different MAC addresses different pn and different certificate then malicious transition. In the last case, the packet may be a loss.

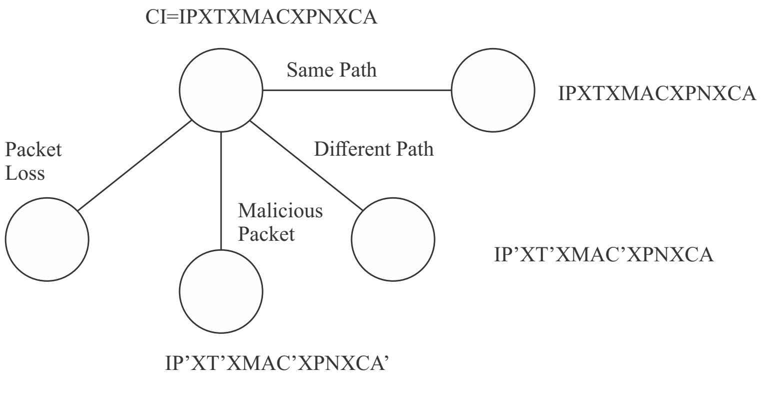

Figure 5: Possible states of the same IP address

Definition of K-path: Any LTS P may consist of two or more states with transitions. Each transition represents any change in state. The K-path represents the K-number of transitions which also represents K-number of hops.

K-equivalent: Two states CI$_1$ and CI$_2$ in LTS P are called K-equivalent if and only if K- paths of state CI$_1$ is same as of state CI$_2$. The security status of each hop/transition in CI$_1$ and CI$_2$ should be same either secure or un-secure. CI$_i$ represents the ith state of the stored value of MAC address, IP address, certificate tree, and pn.

If there is the common value of MAC address, IP address and pn of the state's LTS$_1$ and LTS$_2$, then LTS$_1$ and LTS$_2$ can be merged. The certificate tree T$_1$and T$_2$of both states may be the same or different. If path certificate trees have the same route then tree T$_1$and T$_2$are merged otherwise the merged state has forest T$_1$and T$_2$.

K-tail [7] Algorithm is based on the k-equivalent path. It finds K-equivalent path in two LTSes and considers only one path in resulting LTS and adding all transition in resulting LTS states.

Merged: Find the subtree from tree T$_2$ which child node does not match with the same parent in T$_1$ at the remaining subtree from the child node of T$_2$ at the lost most child of the same parent in T$_1$. Apply in this algorithm recursively to get the merged certificate tree.

Input: LTSes stored in a list data structure
Output: Merged LTSes stored in graph data structure G (V, E)

- Read data (IP, Certificate) from the table stored in the observer server.
- Construct a graph to create a node of the graph.
- Store information of IP, mac, pn, certificate details.
- if the same IP with other mac and same pn

  create another node and connect with the previous node with the new edge if the same certificate is the same

  Store same details
  else
  Merge (Ti, Tj) \\where Ti and Tj certificate trees.

- if the same IP with other mac and pn
  create another node and connect with the previous node with the new edge if the same certificate is the same
  Store same details
  else
  Merge (Ti, Tj)

- Repeat step 3,4 and 5 till all rows of the table present in the observer table.

Examples

In this example, we have considered five traces each trace details are stored in the respective table. This table consists of information state no/ hops which indicate the switch number, IP address, mac address, port number and address of the route certificate. One table stored the record of passes from different hops and one request and response. In a different table, we have recorded information at a different time for the same request and response. The merged graph finds the malicious hosts.

The state no/Hop	IP address	MAC address	Port number	Certificate root
I1	192.168.10.7	78-54-2e-f7-36-8c	80	
I2	103.78.12.1	01-00-5e-00-00-16	80	Ica
I3	115.255.251.220	01-00-5e-00-00-fb	80	Ica
I4	115.255.252.229	01-00-5e-00-00-fc	80	Ica
I5	72.14.218.146	01-00-5e-7f-ff-fa	80	rct

Table 2: IP trace at timestamp t1

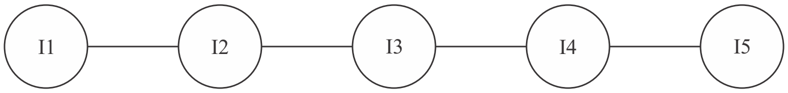

Figure 5: LTS 1

The stateno/Hop	IP address	MAC address	Port number	Certificate root
I1	192.168.10.7	78-54-2e-f7-36-8c	80	
I6	103.78.12.100	01-00-5e-00-00-16	80	Ica
I7	115.255.250.10	01-00-5e-00-00-fb	80	Ica
I4	115.255.252.229	01-00-5e-00-00-fc	80	Ica
I5	72.14.218.146	01-00-5e-7f-ff-fa	80	rct

Table 3: IP trace at timestamp t2

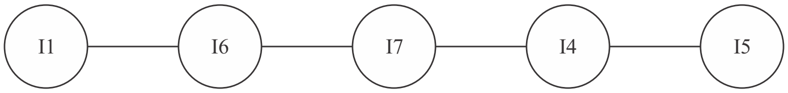

Figure 6: LTS 2

The stateno/Hop	IP address	MAC address	Port number	Certificate root
I8	192.168.10.70	78-54-2e-f7-36-8c	80	
I9	103.78.12.105	01-00-5e-00-00-16	80	Ica
I3	115.255.251.220	01-00-5e-00-00-fb	80	Ica
I4	115.255.252.229	01-00-5e-00-00-fc	80	Ica
I5	72.14.218.146	01-00-5e-7f-ff-fa	80	rct

Table 4: IP trace at timestamp t3

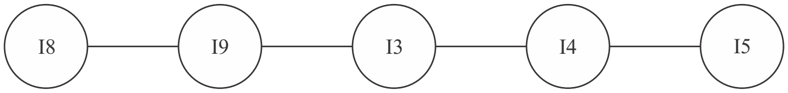

Figure 7: LTS 3

The stateno/Hop	IP address	MAC address	Port number	Certificate root
I8	192.168.10.70	78-54-2e-f7-36-8c	80	
I2	103.78.12.1	01-00-5e-00-00-16	80	Ica
I3	115.255.251.220	01-00-5e-00-00-fb	80	Ica
I10	115.255.252.140	01-00-5e-00-00-fc	80	Ica
I5	72.14.218.146	01-00-5e-7f-ff-fa	80	rct

Table 5: IP trace at timestamp t4

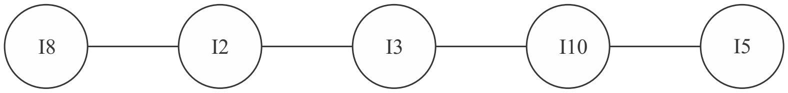

Figure 8: LTS 4

The stateno/Hop	IP address	MAC address	Port number	Certificate root
I1	192.168.10.7	78-54-2e-f7-36-8c	80	
I2	103.78.12.1	01-00-5e-00-00-16	80	Ica
I3	115.255.251.220	01-00-5e-00-00-fb	80	Ica
I4	115.255.252.229	01-00-5e-00-00-fc	80	Ica
I5	72.14.218.146	01-00-5e-7f-ff-fa	80	rct

Table 6: IP trace at timestamp t5

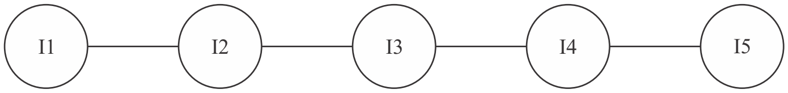

Figure 9: LTS 5

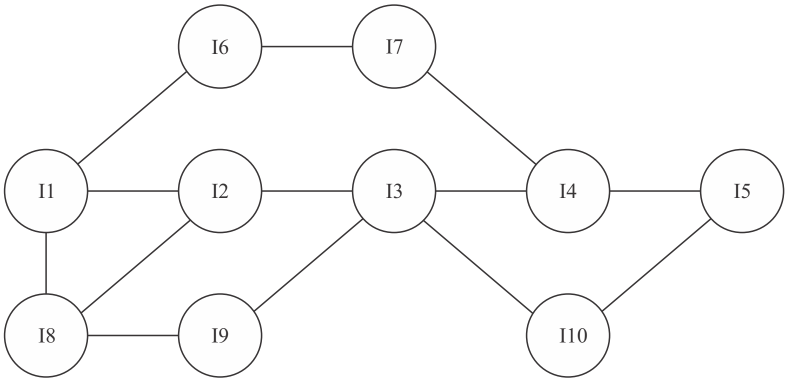

Figure 10: Merged LTSes

5 Conclusion

In this paper, we have proposed a model at the data layer for the security of SDN. In this model, we have used the details of IP address and security certificate to detect the malicious attack. We have used one observer server for recording the IP address and certificate. This information converted into the LTS. In this level LTS security state, mac state, IP state, and port number state are defined for computing the current status. The transition function is used to find the security status of the communication. Our proposed work helps to reduce the load at controller \ switches during the overflow of the attacks. In the future, we will use this information at the switch level for stopping the attack.

References

S. Hong et al. "Poisoning Network Visibility in Software Defined Networks: New Attack and Countermeasures", *NDSS* 2015.

Po-Wen Chi et al. "How to Detect a Compromised SDN Switch" *1st IEEE Conference on Network Softwarization (Net Soft)*, 2015.

S.M. Mousavi et al. "Early Detection of DDoS Attacks against SDN Controllers", *International Conference on Computing, Networking and Communications (ICNC),* 2015.

S. Namal et al. "Enabling Secure Mobility with OpenFlow", *IEEE SDN on Future Networks and Services (SDN4FNS)*, 2013

J. Kim et al. "SDN-based Security Services using Interface to Network Security Functions", *International Conference on Information and Communication Technology Convergence (ICTC)*, 2015.

E. Ahmed et al. Securing Software Defined Networks: Taxonomy, Requirements, and Open Issues", *IEEE Communication Magazine vol. 4*, 2015, pp. 36-44.

A. Biermann et al., "on the synthesis of finite-state machines from samples of their behavior", *IEEE Trans. On Computers, vol. 21*, pp. 592-597, 1972.