Scientific Research Publishing

Journal of Computer and Communications, 2019, 7, 105-115
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Dynamic Access Control Method for SDN

Dexian Chang[1,2], Wanzhong Sun[3], Yingjie Yang[1,2], Tingting Wang[4]

[1]Third Institute of Information Engineering University, Zhengzhou, China
[2]Henan Key Laboratory of Information Security, Zhengzhou, China
[3]Second Institute of Information Engineering University, Zhengzhou, China
[4]The North Sea Fleet of PLA Navy, Qingdao, China
Email: changdexian@126.com

How to cite this paper: Chang, D.X., Sun, W.Z., Yang, Y.J. and Wang, T.T. (2019) A Dynamic Access Control Method for SDN. *Journal of Computer and Communications*, **7**, 105-115.
https://doi.org/10.4236/jcc.2019.710010

Received: September 21, 2019
Accepted: October 9, 2019
Published: October 16, 2019

Abstract

Aiming at the problem that network topology changes frequently in SDN (Software Defined Network) environment and it is difficult to implement fine-grained access control, utilizing the characteristics of SDN transfer control separation and software programming, the ABAC model (Attribute-Based Access Control) is extended by introducing security level, and the security level is defined for the attributes of subject and object to establish the access mapping relationship based on mandatory access rules. At the same time, with secure access path as SDN access control attribute, a dynamic generation method of access control path based on PSO (Particle Swarm Optimization) algorithm is designed to ensure the security of access data flow. The prototype system experiments show that the proposed method takes into account the fine-grained and dynamic requirements of SDN access control, and improves the access security of SDN while ensuring the access efficiency.

Keywords

Access Control, Security, SDN, ABAC, Dynamic

1. Introduction

Software-defined network SDN is designed to effectively solve the problems of complex structure of traditional network forwarding unit and inefficient network management [1] [2]. With the rapid development of Internet services, SDN is now widely used in large mobile network [3]. In order to ensure the security of SDN nodes in the process of accessing resources, it is necessary to implement effective access control. Access control mechanism can ensure that network resources are not illegally used and accessed.

In the process of SDN application, the network node may move continuously, and the access data object may change in real time. At the same time, the node may access and exit continuously, reflecting a strong dynamic. Therefore, SDN-oriented access control mechanism needs to solve dynamic problems such as timely update of access nodes and active adjustment of access rights [4]. Traditional access control models for enclosed environments, such as DAC (Discretionary Access Control), MAC (Mandatory Access Control), RBAC (Role-Based Access Control), require preset node-privilege correspondence and SDN access [5] [6]. The dynamic nature of control makes it necessary to update the preset node-privilege correspondence frequently, which makes the traditional access control model difficult to meet the needs of fine-grained control and dynamic adjustment of access privileges in SDN environment. Attribute-based access control model ABAC [7] does not directly define the authorization relationship between the subject and the object. It uses the attributes between the subject and the object as the basis of authorization decision-making, so as to solve the problems of fine-grained access control in complex network information systems and the dynamic changes of nodes in large-scale networks. Compared with classical access control models such as RBAC and Biba, it is more suitable for the dynamic and scalable requirements of SDN environment.

For this reason, this paper integrates BLP and BIBA mandatory access control mechanism, extends attribute-based access control ABAC model, designs new access control rules for E-ABAC, takes the security level of switching equipment as SDN environment attribute, designs a secure path planning method based on PSO algorithm, and makes full use of SDN flow table update characteristics to ensure data flow security.

2. E-ABAC Model Based on Security Level

BLP and Biba are traditional mandatory access control models, which focus on the protection of confidentiality and integrity, but they are not suitable for new network environments. ABAC models are usually used to solve the access control problems in dynamic scenarios of nodes, but lack of consideration of confidentiality and integrity. In this scheme, ABAC is combined with BLP and Biba models, and the security level definition is introduced to extend ABAC to meet the access control requirements in large-scale distributed network environment.

2.1. E-ABAC (Extended ABAC) Model

In order to effectively combine the hierarchical ideas of BLP and Biba models with the flexibility of ABAC models, the following definitions are given in this paper.

Define1Entity Attribute *EA*. *EA* (*id, value, w*) is a variable used to describe the basic characteristics of entities, including entity attribute identification, entity attribute range, attribute weight. Among them, weights are divided into two categories, $w(c)$ denotes classified weights of confidentiality and $w(i)$ is classified weights of integrity.

Define 2E-ABAC Model. E-ABAC stands for {*SA, OA, EA, PU*} and represents the principal attribute set, the customer attribute set, the environment attribute set and the access priority set, respectively.

Define 3 Attribute Range *V*. *V* denotes the range of values of specific attributes a. Here we quantify its specific values as a set of values Φ. If there are *x* kinds of attributes in common, then the set of attributes is $\Delta = \{a_1, \dots, a_x\}$. Each attribute has its own value space, assuming that the range of value of a_1 is defined as $\theta_{a_1} = \left(v_{a_1}, \dots, v_{a_m}\right)$, then the global attribute value range is defined as $\Gamma = \left(\theta_{a_1}, \dots, \theta_{a_x}\right)$.

For example, if the value range of attribute a_1 is (1, 10], the higher the security level is, the lower the convention is that the value of attribute a_1 approaches the maximum value of 10. In practical application, the range can be defined according to the need to ensure that the attribute values can be calculated.

Define 4 Security Level Values *SLV*. *SLV* = (*C*, *I*), where *C* is the confidentiality value and *I* is the integrity value, then S_{SLV}, O_{SLV} and E_{SLV} represent the subject and object security values respectively.

Rule 1 Weight *w* is computable values, and the sum of attribute weights involved in a single visit is a fixed value.

Rule 2 calculates the security level value by $slv = w \otimes a$, the confidentiality value $slv(c) = w(c) \otimes a$, and the integrity value $slv(i) = w(i) \otimes a$. Considering an entity e, its security value is:

$$slv_s\left(c,i\right) = \begin{cases} c\left(s\right) = slv_0\left(c\right) \oplus slv_1\left(c\right) \oplus \dots \oplus slv_{m-1}\left(c\right) \\ i\left(s\right) = slv_0\left(i\right) \oplus slv_1\left(i\right) \oplus \dots \oplus slv_{m-1}\left(i\right) \end{cases} \tag{1}$$

Among them, denotes an operation mode, which can be used to calculate the values of confidentiality and integrity simply by adding. When the subject accesses the object, the security level of the subject and the object needs to correspond.

2.2. New Access Control Rules

Define 5 Operation Behavior *A*. *S* is the main body set, *O* is the object set, *A* = {*action* | *r, e, w, x*} is the operation behavior set, where *r* means read-only and not write, *e* means write-only and not read, *w* means both read and write, *x* means execution.

Definition 6 *Invoke* indicates that a subject *s* calls an object *o* in some way *y*. *i* (*s*: a_1, ..., a_n) is a collection of all the ways to call object o, *i.e.*

$$\begin{aligned} & Invoke\left(s : x_1, \dots, x_n\right) \\ & = \left\{o \mid o \in O \wedge \left[\left(s, o, x_1\right) \in Invoke \vee \dots \vee \left(s, o, x_n\right) \in Invoke\right]\right\} \end{aligned}$$

According to BLP and Biba mandatory access control rules, when the security level of the subject is exactly the same as that of the object, the subject can read and write the object. However, in E-ABAC, the security level of subject and object is a relatively accurate value, which is a refined representation of the security of subject and object. If BLP and BIBA are used directly, the scope of object that a particular subject can read and write at the same time will be smaller, and there will be almost no qualified access object except the subject. Except for the object created by the object, other objects are not satisfied with the entirely equal security value of the subject. Therefore, this paper presents an access control rule that integrates BLP confidentiality and BIBA integrity model.

Based on the security range, the definition of security level domain can be given, where C_{s+} is the upper limit of the confidentiality value of the access subject, C_{s-} is the lower limit of the confidentiality value of the access subject. Similarly, I_{c+} and I_{c-} are the upper and lower limits of the integrity value of the subject, while C_o and I_o are the current confidentiality and integrity value of the access object.

Inference 1 E-ABAC Confidentiality

$$Invoke(s:a) \Rightarrow \left[\forall o \in Invoke(s:a)\left[c_o(o) \geq c_{s-}(s)\right]\right]$$

$$Invoke(s:w) \Rightarrow \left[\forall o \in Invoke(s:w)\left[c_{s-}(s) \leq c_o(o) \leq c_{s+}(s)\right]\right]$$

$$Invoke(s:r) \Rightarrow \left[\forall o \in Invoke(s:r)\left[c_o(o) \leq c_{s+}(s)\right]\right]$$

That is, when the upper limit of subject confidentiality value is greater than that of object confidentiality value, the subject can read and access the object. When the lower limit of the subject confidentiality range is less than the object confidentiality value, the subject can write access to the object.

Inference 2 E-ABAC Integrity

$$Invoke(s:a) \Rightarrow \left[\forall o \in Invoke(s:a)\left[i_o(o) \leq i_{s+}(s)\right]\right]$$

$$Invoke(s:w) \Rightarrow \left[\forall o \in Invoke(s:w)\left[i_{s-}(s) \leq i_o(o) \leq i_{s+}(s)\right]\right]$$

$$Invoke(s:r) \Rightarrow \left[\forall o \in Invoke(s:r)\left[i_o(o) \geq i_{s-}(s)\right]\right]$$

When the upper limit of the subject integrity range is not less than the object integrity value, the subject can write to the object, and when the lower limit of the subject integrity range is not more than the object integrity value, the subject can read to the object.

According to Reasoning 1 and Reasoning 2, E-ABAC access control rules can be obtained:

$$Invoke(s:a) \Rightarrow \left[\forall o \in Invoke(s:a)\left[c_o(o) \geq c_{s-}(s), i_o(o) \leq i_{s+}(s)\right]\right] \tag{2}$$

$$\begin{gathered} Invoke(s:w) \Rightarrow \\ \left[\forall o \in Invoke(s:w)\left[c_{s-}(s) \leq c_o(o) \leq c_{s+}(s)\right], i_{s-}(s) \leq i_o(o) \leq i_{s+}(s)\right] \end{gathered} \tag{3}$$

$$Invoke(s:r) \Rightarrow \left[\forall o \in Invoke(s:r)\left[c_o(o) \leq c_{s+}(s), i_o(o) \geq i_{s-}(s)\right]\right] \tag{4}$$

It should be noted that when a write operation occurs, if the subject's confidentiality value is higher than the object's, the object's confidentiality value should be increased; if the subject's confidentiality value is lower than the object's confidentiality value, the object's confidentiality value will remain unchanged. If the value of subject integrity is higher than that of object integrity, the value of object integrity remains unchanged. If the value of subject integrity is lower than that of object integrity, the value of object integrity should be reduced. The E-ABAC architecture based on the above model is illustrated below **Figure 1**.

Among them, attribute authority (AA) is responsible for creating and managing the attributes and initial security values of subject, object and environment. Policy enforcement point (PEP) is responsible for requesting access decision and implementation. Policy Decision point (PDP) is responsible for assessing applicable security policies and making authorization decisions.

The security management engine (SM Engine) is responsible for establishing and storing the mapping value of the subject and object attributes. The security value calculation module (SV) is used to calculate and manage the security value of the subject and object, and the results are fed back to PDP, mainly including the mapping set of the subject and object attributes.

In SDN, the access subject is usually the user, the object is usually the service resource, and the access control decision point is the SDN switch. The access control strategy is generated by the SDN controller. At the same time, SDN environment attributes mainly consider the security of forwarding data flow between switches. This attribute authority provides decision support for PDP. Its specific rules in the implementation of access control are detailed in the next section.

3. Security Access Path Planning Method

3.1. Access Control Based on SDN Flow Table

SDN is characterized by the separation of data forwarding and control. Its flow table mechanism provides technical support for the implementation of data forwarding access control. When a SDN user authenticates successfully based on the previous E-ABAC model, data forwarding is required. The SDN controller generates the corresponding flow table based on the access policy of the security level of the host and the object, and sends it to the corresponding SDN switch. All data packets from the user are forwarded according to the access rules.

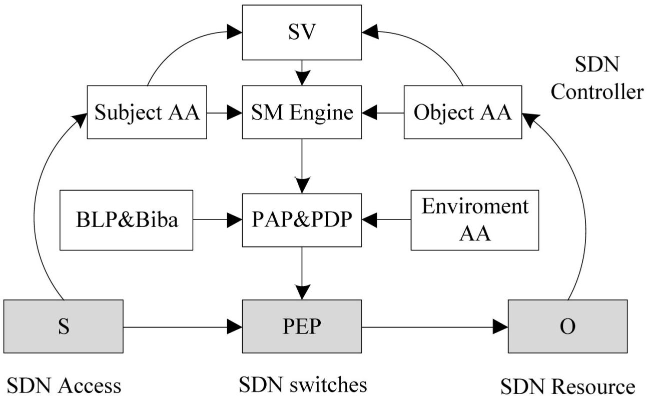

Figure 1. Extended ABAC model in SDN.

Considering the constraints of environment attributes in ABAC model, the forwarding device, SDN switch, is taken as the environment factor in access control. A path planning method oriented to secure access relationship between host and object is designed, which makes the normal access of the host and object data meet the security requirements in the forwarding process and ensures secure access control.

3.2. Secure Path Planning Algorithms

The classical path planning method uses the shortest path algorithm, but in SDN environment with different security levels, the security levels of different subnets or application domains are different, and the corresponding forwarding devices have different sensitivity levels. In order to ensure that the data is not destroyed in the access process, it is necessary to plan the path to increase the safety requirement. In order to obtain the optimal path of multi-level security and multi-switching nodes, this paper uses Particle Swarm Optimization (PSO) [8] algorithm to solve the problem.

Particle Swarm Optimization (PSO) is a modern evolutionary algorithm with concise form, fast convergence and flexible parameter adjustment mechanism, which simulates the foraging behavior of bird clusters in flight. It has been successfully applied to the solution of path search and optimization problems.

In particle swarm optimization, a massless particle *i* can be represented by position vector and velocity vector. Among them,

$$x_i = \left[x_{i,1}, x_{i,2}, \ldots, x_{i,N},\right]^T \in R^N$$
$$v_i = \left[v_{i,1}, v_{i,2}, \ldots, v_{i,N},\right]^T \in R^N, i = 1, 2, \ldots, N$$

N represents the number of particles in the population. Particles in a population update their speed and position in evolution by following formulas:

$$\left.\begin{aligned} & v_i(t+1) = wv_i(t) + c_1r_1(pBest) - x_i(t)) + c_1r_1(gBest - x_i(t)) \\ & x_i(t+1) = x_i(t) + v_i(t+1) \end{aligned}\right\} \quad (5)$$

Among them, *t* represents the number of iterations, $w \ge 0$ represents the inertial compression factor, $c_1, c_2 \ge 0$ represents the acceleration factor, $r_1, r_2 \in [0,1]$ represent the random numbers with uniform distribution, $x_i(t)$ represents the current position of the particle, the individual optimal solution of the first particle and the global optimal solution of the whole population are represented by *pBest* and *gBest*. In the velocity renewal equation of Equation (1), the first part is the inertial motion of particles according to their own velocities, which expresses the trust of particles in their current motion; the second part is the self-recognition part of particles, which expresses the reflection of particles on their own history; and the third part is the social cognition part of particles, which expresses the trust of particles in the group, representing the information sharing and collaboration.

Firstly, the algorithm condition is prepared.

- Definition of variables

A particle swarm δ is formed by all nodes of SDN network (except source host node and target host node) that need path planning. δ consists of m particles, corresponding to switch nodes in the network. The speed and mass definitions of each particle correspond to its attribute values.

- Constraints

The mapping between source and target potentially represents the security level SLV of an access process. When planning a path, the security level of all the exchanges passing through must not be lower than that of $MAP(s,d)$, that is $SLV_{MAP} \geq SLV_{N_i}$, which is used as the inertia parameter w of particles.

- Steps of Algorithms

The specific steps of the algorithm are described as follows:

Step 1 initializes the particle swarm, sets the size of the population N, randomly generates the initial positionx_0 and velocity v_o of each particle, and sets the number $t = 0$ of iterations.

Step 2 uses the championship selection strategy proposed in [9] to compare the current individual extreme value and individual historical optimum value, and uses this method to select the global extreme value $gBest[1]$ and $gBest[2]$ of the population.

Step 3 updates the position and velocity of the population particles according to Formula (1).

Step 4 If the iteration condition is satisfied; it will output the last generation of population individuals, namely Pareto optimal solution; otherwise, it will return to Step 2.

4. Implementation and Evaluation

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your journal for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper.

4.1. Prototype Implementation

Based on the scheme of [4], this paper implements the prototype system and constructs the experimental environment as shown in the following **Figure 2**.

Among them, the authentication server uses FreeRADIUS [10] to realize SV function, all authentications and authorization resources are stored in Mysql database; the Authenticator uses the 802.1× host pad [11] to realize PDP. The controller is based on POX [12], which mainly includes the second layer forwarding and AA, SME modules designed in this paper. It uses OpenFlow 1.3 to cooperate with the switch. The protocol of SSL (Secure Socket Layer) is used to communicate with the discriminator, the analog network of 5 fully connected Openvswitch switches is built based on Mininet [13], communication between

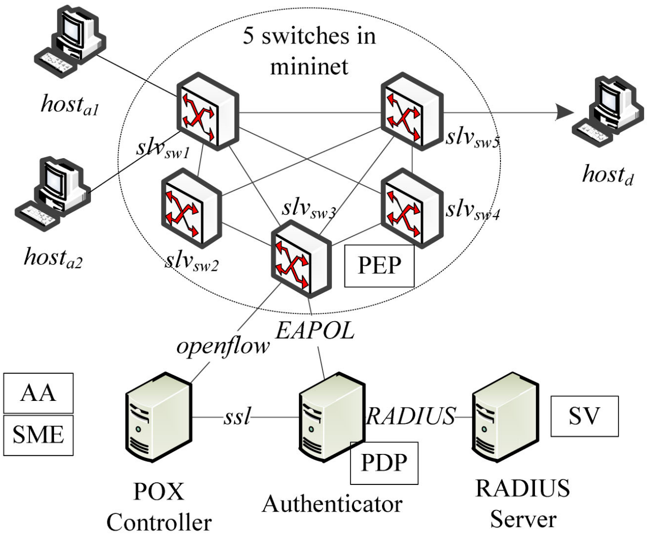

Figure 2. Prototype system experiment Testbed.

PDP and switches are based on EAPOL (Extensible Authentication Protocol over LANs) protocol, and the system supporting wpa is installed and deployed in the access host. There are three physical hosts in use, one for POX controller and one for Mininet, and one for Authenticator and RADIUS. The required physical host has an Intel (R) Core (TM) CPU i5 7500 @3.40 GHz and 64 GB of RAM.

In the experimental network, $host_s$ and $host_d$ are the access subject and the access object respectively. The basic parameters are collected by the controller. According to the decision of Authenticator and RADIUS Server, the final access decision is formed and mapping relationship Map ($host_a$, $host_d$) is generated. The forwarding network consists of five virtual switches with the security level value SLV. As an environmental attribute, the controller uses Particle Swarm Optimization (PSO) algorithm to get the forwarding path and sends it to the switches with the flow tables.

4.2. Experiment and Result Analysis

In this section, through experiments, the security and operational efficiency of the scheme are analyzed and compared.

- Access Control Verification Based on E-ABAC

For $host_{a1}$, $host_{a2}$and $host_d$, the attribute values shown in the following **Table 1** are set according to the application requirements, including OS version (OS), security protection (firewall FW, IPS, etc.) and user password (pwd), attribute security level value, integrity weight ($w(i)$) and machine. The density weight value ($w(c)$ is set by SDN controller according to AA feedback value before implementing access control, and dynamically adjusted according to application requirements. After each adjustment, the security value is recalculated.

According to the formula for calculating the security value of E-ABAC model (1), the security value $slv_s(c,i)$ of confidentiality and integrity of $host_{a1}$ can be obtained as follows,

Table 1. Attributes of entity and the security values.

Entity	Attributes	Value	$w(c)$	$w(i)$
host$_{a1}$	*OS, FW, pwd*	1, 3, 6	0.1, 0.4, 0.6	0.2, 0.3, 0.5
host$_{a2}$	*OS, IPS, pwd*	1, 4, 6	0.1, 0.2, 0.7	0.2, 0.4, 0.4
host$_{d}$ *(FTP/WEB)*	*OS, Anti, pwd*	3, 4, 6	0.3, 0.3, 0.4	0.4, 0.2, 0.4
	OS, FW, pwd	2, 5, 6	0.2, 0.3, 0.5	0.4, 0.3, 0.3

$$slv_{host_{a1}}(c,i)=\begin{cases} c(host_{a1})=1*0.1+3*0.4+6*0.6=4.9 \\ i(host_{a1})=1*0.2+3*0.3+6*0.5=4.1 \end{cases}$$

Similarly, the confidentiality and integrity security values $slv_{host_{a2}}(c,i)$ of *host*$_{a2}$ can be obtained as follows,

$$slv_{host_{a2}}(c,i)=\begin{cases} c(host_{a2})=1*0.1+4*0.2+6*0.7=5.1 \\ i(host_{a2})=1*0.2+4*0.4+6*0.4=4.2 \end{cases}$$

When visiting *host*$_d$ provides FTP and WEB services respectively, the corresponding security values are:

$$slv_{host_{d}}(c,i)_{FTP}=\begin{cases} c(host_{d})=3*0.3+4*0.3+6*0.4=4.5 \\ i(host_{d})=3*0.4+4*0.2+6*0.4=4.4 \end{cases}$$

$$slv_{host_{d}}(c,i)_{WEB}=\begin{cases} c(host_{d})=2*0.2+5*0.3+6*0.5=4.9 \\ i(host_{d})=2*0.4+5*0.3+6*0.3=4.1 \end{cases}$$

According to the E-ABAC model access rule (2) - (4), the access permission of *host*$_{a1}$ and *host*$_{a2}$ to different services of *host*$_d$ is shown in the following **Table 2**. All accessible behaviors are based on access rules (4), such as only *host*$_{a1}$ can read and write WEB services.

From the table, we can see that the E-ABAC model can not only take into account the attribute-based access method, but also effectively implement BLP and Biba mandatory access, which can meet the application requirements of SDN accessing entities with strong mobility and frequent topology updates, and achieve dynamic access control.

- Path Planning Efficiency Based on PSO

In order to verify the efficiency of SDN service access based on E-ABAC model, this paper tests the service response time under different number of concurrent requests (100, 200, 300, 400) and different number of attributes (0, 2, 4, 6). The results are shown in the following **Figure 3**.

The results show that the increasing number of concurrent requests has little effect on the service response time, because the PSO-based path planning algorithm has high efficiency. For a small SDN network composed of five switches, the algorithm can converge quickly and get the optimal path. However, if the number of attributes exceeds 4, the response time will increase greatly. Therefore, when using E-ABAC model to implement access control, it is necessary to consider restricting the number of attributes.

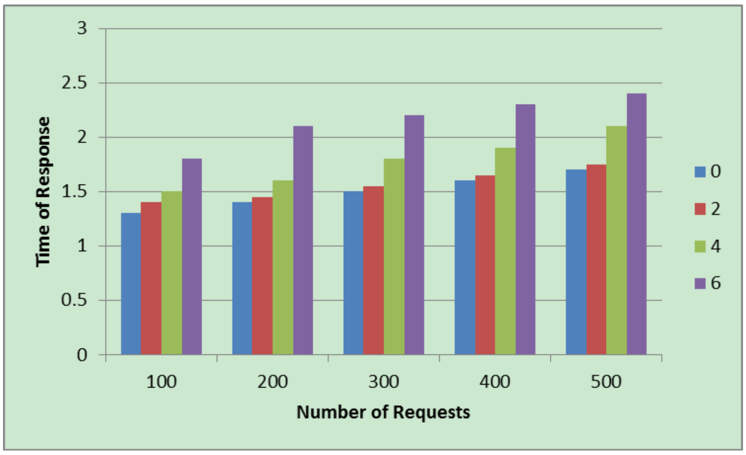

Figure 3. Host response time test.

Table 2. The access relations between $host_a$ and $host_d$.

Map (*s*, *d*)	*slv*(*s*), *slv*(*o*)	*action*
$host_{a1}$-$host_d$ (*FTP*)	*c*(*s*) = 4.9, *i*(*s*) = 4.1 *c*(*o*) = 4.5, *i*(o) = 4.4	*r*
$host_{a1}$-$host_d$ (*WEB*)	*c*(*s*) = 4.9, *i*(*s*) = 4.1 *c*(*o*) = 4.9, *i*(o) = 4.1	*a, w, r*
$host_{a2}$-$host_d$ (*FTP*)	*c*(*s*) = 5.1, *i*(*s*) = 4.2 *c*(*o*) = 4.5, *i*(o) = 4.4	*r*
$host_{a2}$-$host_d$ (*WEB*)	*c*(*s*) = 5.1, *i*(*s*) = 4.2 *c*(*o*) = 4.9, *i*(o) = 4.1	*w*

5. Conclusion

Aiming at the problem that SDN is difficult to implement fine-grained and hierarchical access control, this paper combines the existing mandatory access control mechanism and extends the ABAC model. On the one hand, BLP and Biba models are integrated into ABAC to make access decisions based on security level values, so as to realize flexible and fine-grained access control. On the other hand, SDN switches are regarded as environment attributes. Security path planning based on PSO algorithm ensures the security of access flow. Experiments show that the model can meet the requirements of dynamic access control for SDN, and has little impact on response time. Next we will run our system at hardware switches and improve the implementation feasible for a more practical deployment.

Acknowledgements

The authors would like to thank their colleagues and the anonymous reviewers for their helpful feedback and comments. This work is partly supported by the National High-Tech Research and Development Plan of China (No. 2015AA016006) and National Key Research and Development Program (NO. 2016YFF0204003).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Xia, W.F., Wen, Y.G., Foh, C.H., *et al.* (2015) A Survey on Software-Define Networking*. IEEE Communications Surveys & Tutorials*, **17**, 27-51. https://doi.org/10.1109/comst.2014.2330903

[2] Farhady, H., Lee, H. and Nakao, A. (2015) Software-Defined Networking: A Survey. *Computer Networks*, **81**, 79-96. https://doi.org/10.1016/j.comnet.2015.02.014

[3] Pujolle, G.: (2015) Software Networks Virtualization, SDN, 5G and Security. ISTE Ltd and Wiley, London and New York. https://doi.org/10.1002/9781119005100.ch1

[4] Nife, F. and Kotulski, Z. (2018) New SDN-Oriented Authentication and Access Control Mechanism. *International Conference on Computer Networks.*

[5] Zhang, J., Yun, L.J. and Zhou, Z. (2008) Research of BLP and Biba Dynamic Union Model Based on Check Domain. *International Conference on Machine Learning & Cybernetics*. https://doi.org/10.1109/icmlc.2008.4621044

[6] Kumar, N.V.N. and Shyamasundar, R.K. (2017) A Complete Generative Label Model for Lattice-Based Access Control Models. *International Conference on Software Engineering & Formal Methods.*

[7] Hu, V.C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R. and Scarfone, K. (2014) Guide to Attribute Based Access Control (ABAC) Definition and Considerations. Special Publication 800-162, U.S. Department of Commerce, January. National Institute of Standards and Technology. https://doi.org/10.6028/nist.sp.800-162

[8] Kennedy, J. (1995) Particle Swarm Optimization. *Proc. of* 1995 *IEEE Int. Conf. Neural Networks*, Perth, Australia, 27 November-December 1995.

[9] Hu, W., Yen, G.G. and Zhang, X. (2014) Multiobjective Particle Swarm Optimization Based on Pareto Entropy. *Journal of Software*, **25**, 1025-1050.

[10] Malinen, J. Hostapd: IEEE 802.11 AP, IEEE 802.1x/WPA/WPA2/EAP/RADIUS Authenticator. https://w1.fi/hostapd/

[11] FreeRADIUS, FreeRADIUS Project. https:freeradius.org/

[12] POX Controller, POX Wiki. https://openflow.stanford.edu/display/ONL/POX+Wiki

[13] Neri, G., Morling, R.C.S., Cain, G.D., *et al.* (1984) MININET: A Local Area Network for Real-Time Instrumentation Applications. *Computer Networks*, **8**, 107-131. https://doi.org/10.1016/0376-5075(84)90039-4