An Action-based Access Control Model Implementation for MLS Information Systems

Li Fenghua[1], Shi Guozhen[1], Zhang Jinghui[2], Li Li[1]
Dept. of Electronic Engineering Beijing Electronic Science and Technology Institute, Beijing,100070, China
School of Computer Science and Technology, Xidian University Xi'an , 710071, China
e-mail: jj880306@163.com

***Abstract*—Environmental states retrieving is the major issue in the implementation of action-based access control model. In this paper, the environmental states retrieving mechanism based on ABAC (Action-Based Access Control) was proposed. Then, the architecture of network location detector was presented, and the method to get the logical address based on DHCP and the method to get the physical address based on SNMP and MIB were introduced. By proposing the item, security association request, and its producing procedure, the trusted software and hardware monitoring mechanism was presented. Finally, the method to obtain the temporal states was described.**

Keywords- Access Control, Action, Environmental States, Timestamp

I. INTRODUCTION

MLS (Multi-level Security) information system allows information about different sensitivities (classifications) to be stored, and allows users having different clearances, authorizations[1]. As a generalized approach to the security management, access control is used to protect certain data or resources that only authorized users can get access to.

As a traditional access control model, Role-Based Access Control model[2] is widely used. In the basis of the separation of user and role, only the effect of role on the permissions distribution was considered in RBAC. In fact, the environmental states and the temporal states when the user gets access to data resources also determine the assignment of permissions. In [3,4], the concepts of environmental states, temporal states and the term "action" based on the affection of role were introduced, and ABAC, the administrative action and the administrative model for ABAC were also proposed. In [5], the access control mechanism based on ABAC for collaborative information systems was introduced, and the security association producing procedure was described. Also, to exchange the security properties among the user, the Action server and the Resources server, a secure authentication protocol was proposed and its security was proven under the universally composable model. In [6], the security architecture of ABAC for Web services is proposed. In the architecture, the Action server manages the action information, the Domain server determines the security rank of request resources, and the Resource server storing the resources with different security ranks responses the request from the user.

The environmental state is an important factor to access control policy in MLS information system. Although the access control mechanism for collaborative information systems was introduced in [5], the specific method to get environmental states and temporal states was not described. In view of the importance of environmental states in the MLS information system, a new approach to obtain the environmental state based on ABAC was presented, which gets network location relying on network location detector, and gets software and hardware information relying on trusted software and hardware monitoring mechanism. Meanwhile, timestamp service provides temporal states needed for ABAC.

II. ENVIRONMENTAL STATES RETRIEVING

In ABAC, the general form of the environmental state set E is [EN|EL|EH|ES][3,4], where EN refers to the network logical locations including MAC address and IP address, EL specifies the network physical locations where the access event takes place, EH refers to the hardware information, and ES refers to the software information. As shown in Figure1, Step1 describes the process of network location detecting，which is introduced in Section 3. The procedure of the client "retrieving user ID, passwords, software information, hardware information, MAC address and the Resource server address to create security association request" is introduced in Step2.1~Step2.12. The procedure of the Action server "retrieving environmental states and temporal states after receiving security association request" is introduced in Step3.1~Step3.11. The communication process between both interactive sides is described in Section 4.

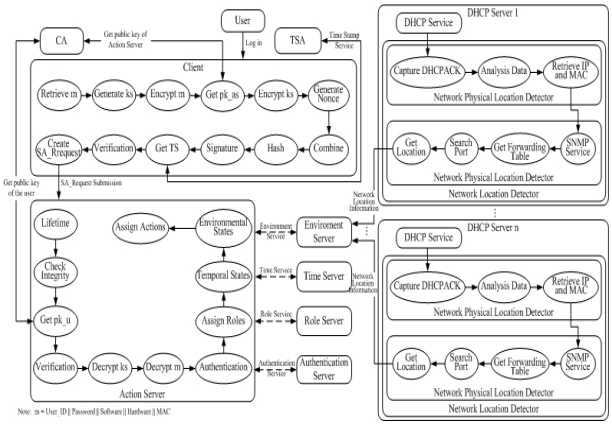

Figure 1. Environmental states retrieving

III. NETWORK LOCATION DETECTING

To provide network logical and physical location for the Environment server, network location detector, the architecture of which has been described in Figure 1, is composed of network logical location detector and network physical location detector.

Network logical location detector is implemented based on DHCP(Dynamic Host Configuration Protocol) and WinPcap(Windows Packet Capture). The DHCP server will distribute dynamic IP address for it when a host logs in MLS information system.Then, the host receives the DHCP server's response, that is, DHCPACK message. Therefore, responding to the host with DHCPACK message shows the host logging in the MLS information system. Network logical location detector captures DHCPACK messages, analysises protocal and processes data to retrieve network logical location information, which is then recorded in fields yiaaddr and chaddr.

Network Physical Location Detector is implemented based on SNMP (Simple Network Management Protocol). Network logical location detector obtains the host's network logical location information, and sends it to network physical location detector. Network Physical Location Detector sends SNMP messages to switches, and gets the port connecting to the destination host by the forwarding table and routing table. Network physical location would be obtained after selecting in LocationInfo Table, which records the device location in the MLS information system. Finally, the network location information is sent to the Environment server, which updates NetworkLocationInflo Table stored in MySQL. Table I shows the structure of NetworkLocationInfo.

TABLE I. STURCTURE OF NETWORKLOCATIONINFLO

Fields	Data Type	Annotation
MAC	VARCHAR(45)	Primary Key. Uniquely identify the host.
IP	VARCHAR(45)	Record the host's IP address.
Location	VARCHAR(45)	Record the host's network physical location.

IV. TRUST SOFTWARE AND HARDWARE MONITORING

The client retrieves the software and hardware information and sends them to the Action server the first time a user tries to get access to resources in the MLS information system. Therefore, attacks during message sending should not be neglected, such as content leakage, garnish, replay attack, ect.. Aiming at these security problems, the platform embedded TPM is chosen and the communication mechanism described in 4.1 is applied. An enhanced architecture of TPM introduced in [7] is used in this paper, which adds a new I/O component.

A. Trusted software and hardware monitoring

Trusted software and hardware monitoring provides the software and hardware information of users logging in MLS information system. Figure 1 has described the communication mechanism of both interactive sides.

The Action server gets environmental states and temporal states by the following interactions the first time a user tries to get access to some resources. The structure of SA_Reuqest is introduced in 4.2.

Step 1 Network location detector sends the network location information to the Enviroment server after detecting his network logical and physical location when a user logs in MLS information system. Enviroment server updates NetworkLocationInfo table.

Step 2 SA_Request is sent to the Action server.

Step 2.1 The software and hardware information of the platform and MAC address are retrieved, which are packaged with user_ID, Password, and Resource_Server_Address.

Step 2.2 The session key ks, is generated by the key generator of TPM.

Step 2.3 The results of Step 2.1 are encrypted based on AES by the session key generated in step 2.2.

Step 2.4 The user gets access to the Action server to require the public key of the Action server. Then CA distributes the public key pk_as to the requestor using X.509 certification.

Step 2.5 The session key ks is encrypted based on RSA by pk_as.

Step 2.6 The random number generator of TPM generates random numbers noted as nonce.

Step 2.7 Random numbers, Resource_Server_Address, Encrypted_ks, Encrypted_field, and Life_time are combined.

Step 2.8 The result of Step 2.7 is hashed by SHA-1 engine of TPM.

Step 2.9 The hashed value is signed by user's pravite key sk_u.

Step 2.10 The result of Step 2.9 is sent to TSA for the timestamp. TSA generates the timestamp file and sends it to the user.

Step 2.11 The user verifies the timestamp file after receiving. If it is valid, goto Step 2.12; Otherwise, return failed.

Step 2.12 The client creates the SA_Request and sends it to the Action server.

Step 3 The Action server gets SA_Request and assigns action to the user by the Time server, the Role server and the Environment server.

Step 3.1 The Action server checks the Lifetime field. If it is valid, goto Step 3.2; Otherwise, return failed.

Step 3.2 The Action server computes the hashed value of the random number, Resource_Server_Address, Encryption_field and Lifetime field by SHA-1 engine of TPM, and compares the result with the Hash field. If the hashed value is identical, goto Step 3.3; Otherwise, return failed.

Step 3.3 The Action server sends request to CA to get the public key of the user, then CA distributes the public key pk_u to the Action server using X.509 certification.

Step 3.4 The Action server verifies the Digital_sign field by pk_u. If signature is verified, goto Step 3.5; Otherwise, return failed.

Step 3.5 The Action server decrypts the session key by its private key sk_as. If success, goto Step 3.6; Otherwise, return failed.

Step 3.6 The Action server decrypts Encryption_field based on AES by the session key ks.

Step 3.7 The Action server compares the user ID, passwords with the ones stored in the Authentication server. If they are identical, goto Step3.8; Otherwise, return failed.

Step 3.8 The Action server assigns the user to matching roles with the help of the Role server.

Step 3.9 The Action server retrieves temporal states from timestamp field with the help of the Time server.

Step 3.10 The Action server retrieves Environmental states with the help of Environment server.

Step 3.10.1 The Environment server retrieves user's MAC address from SA_Request.

Steo 3.10.2 The Environment server selects the results of Step 3.10.1 from NetworkLoncationInfo stored in MySQL in order to get the network location information.

Step 3.10.3 The Environment server retrieves software information and hardware information, with which the result of Step 3.10.2 is combined to create the environmental states.

Step 3.11 The Action server assigns the user to matching actions according to the role, environmental states and temporal states.

B. Security Association Request

The user sends security association request, the structure of which is shown in Figure 2, to the Action server for SA[7] the first time he tries to get access to resources.

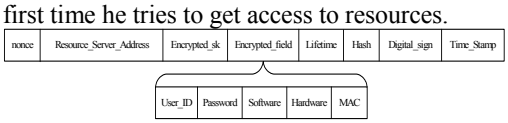

Figure 2. Structure of the SA_Request

The client computes these following values to create SA_Request.

Encrypted_field = ENCks(User_ID || Password || Software || Hardware || MAC)

Encrypted_ks = ENCpk_as(ks)

Hash = HASH (nonce || Resource_Server_Address || Encrypted_as || Encrypted_field || Lifetime)

Digital_Sign = SIGNsk_u(HASH)

The Action server computes these following values when it receives SA_Reauest.

$HASH(nonce \| RMSAddr \| Enc \| Lifetime) \stackrel{?}{=} Hash$

$VERIFYpk\_u(Sign) \stackrel{?}{=} Digital\_Sign$

DECsk_as(Encrypted_ks)

DECks(Encrypted_field)

Suppose that ks is the session key, sk_as is the private key of the Action server, pk_as is the public key of the Action server, sk_u is the private key of a certain user, pk_u is the public key of a certain user, the function to encrypt information m using the key k is ENCk(m), the function to decrypt information c using the key k is DECk(c), the function to sign information m using the key k is SIGNk(m), VERIFYk(m) is the function to verify information m using the key k is SIGNk(m), the hash function is HASH, x||y denotes the concatenation of x and y.

C. Timestamp service

Timestamp service provides an unchangeable and identifiable time mark for data to prevent repudiation afterwards. Meanwhile, a timestamp server gets temporal states from a timestamp.

TSA provides users with a timestamp service[8]. The user submits hashed value needed to add the timestamp to TSA. After receiving the application, TSA will verifies the legality of the request according to the format of the timestamp. If being legitimate, TSA will fill in correct timestamp format and sign by its private key. And then, TSA presents the new timestamp to the user. Receiving the new timestamp, the user verifies it by certification of the TSA , and makes comparison between the hashed value to be added a timestamp and that in the timestamp file. If consistent, it shows that the data was created without changing during the application of timestamp. Otherwise, the data could have been changed or the application could have failed. If so, the user will report such abnormal situation to the TSA immediately.

V. CONCLUSION

ABAC model provides safety guarantee for resources in MLS information system. In this paper, in order to implement ABAC, the environmental states retrieving mechanism is presented. To obtain the network location information, the architecture of network location detector is introduced. To obtain the platform information, the trusted software and hardware monitoring mechanism is proposed. Finally, the approach to get the temporal states was described.

ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China (No.60633020, No.60702059), the Key Program of Scientific and Technology Research of Ministry of Education (No.209156), the Beijing Natural Science Foundation (No.4102056), the Major Science and Technology Project of Press and Publication – Research and Development Project on Digital Rights Protection (No. GXTC-CZ-1015004/05), and the Foundation of Information Security Key Laboratory of Beijing Electronic Science and Technology Institute (No.YZDJ0807).

REFERENCES

[1] The Future of Multi-Level Secure (MLS) Information Systems [DB/OL], http://csrc.nist.gov /nissc/1998/proceedings/panelF3.pdf, 1998.

[2] Ferraiolo D F, Sandhu R, Gavrila S, et al. Proposed NIST Standard for Role-Based Access Control [J] ACM Transactions on Information and System Security, 2001, 08, 4(3): 224-274.

[3] LI Feng-hua, WANG Wei, MA Jian-feng and SangJae Moon. Action-Based Access Control Model[J]. Chinese Journal of Electronics, 2008, 07, 17(3): 396 - 401.

[4] LI Feng-hua, WANG Wei, MA Jian-feng, LIANG Xiao-yan. Action-Based Access Control Model and Administration of Action [J]. Acta Electronia Sinica(in Chinese), 2008, 10, 36(10): 1881 - 1890.

[5] LI Feng-hua, WANG Wei, MA Jian-feng, LIU Hong-yue. Access control model and its application for collaborative information systems[J]. Journal on Communications(in Chinese), 2008, 09, 29(9): 116 - 123.

[6] LI Feng-hua, WANG Wei, MA Jian-feng and SU Hao-xin. Action-Based Access Control for Web Services [J]. Journal of Information Assurance and Security, 2010, (5): 162 - 170.

[7] LI Feng-hua, WANG Wei, MA Jianfeng and Zhenguo Ding. Enhanced Architecture of TPM[A]. In Proceedings of the 9th International Conference for Young Computer Scientists(ICYCS'08)[C], IEEE Computer Society, Zhang Jia Jie, Hunan, China. 2008, 11. 1532 - 1537.

[8] GB/T 20520 – 2006 Information security technology – Public key infrastructure – Time stamp specification[S]. 2006.