Design of Secure System Architecture Model for Active Network*

XIA Zheng-you, ZHANG Shi-yong

(*Department of Computer and Information Technology, Fudan University, Shanghai* 200433, *China*)

E-mail: xiazhengyou@sina.com

http://www.fudan.cn

Received November 29, 2001; accepted February 4, 2002

Abstract: In this paper, the assumption model and the threat model of active network security system are introduced. A secure system architecture model based on these models and security requirement is presented. Definition of secure system architecture model includes authentication, authorization, integrity and encryption. To protect the integrity of the contents of active packet, the encryption and the digital signatures can be employed and the authorization mechanisms or policies are defined and enforced to provide controlled access to the active node resources.

Key words: active network; security architecture; model; design

Active networks[1] provide a programmable platform on which network services can be defined or altered by injecting code or other information into the nodes of the network. This paradigm offers a number of potential advantages, including the ability to develop and deploy new network protocols and services quickly, and the ability to customize services to meet the different needs of the different classes of users.

Since the concept of the active network was put forward in 1996, the current active network research focuses on the support of flexible, dynamically changing[2], fine-grained quality of service. Similar to traditional network security, it is crucial thing for active networks to protect its security. Active network security presents significant security challenges. There are a few research security features that exploit active networking. Despite significant energy devoted to security research in active networks[3~12], the issues of the security are by no means solved. Only some security requirements and challenges in active networks are described. This paper attempts to present a security architecture model that is based on the security requirements in active networks, threat model, assumption model, and challenges of the meeting those requirements. Definition of secure system architecture includes authentication, authorization, integrity and encryption. To protect the integrity of the contents of the active packet, encryption and digital signatures can be employees; authorization mechanisms or policies are defined and enforced to provide controlled access to the active node resources

1 Background and Related Work

In this section, it provides two parts. One provides background for active network. The other describes the related work for secure architecture for active network.

The DARPA active network community has defined architecture for an active network node (ANN)[12,13]. That

---

* **XIA Zheng-you** received his M.S. degree from Nanjing University of Science and Technology in 1999. Now He is a Ph.D. candidate at the Fudan University. His interests are network security and management. **ZHANG Shi-yong** is a professor at the Fudan University. His interests are network protocol and security.

depicts a node as comprising a Node OS and one or more Execution Environments. The Execution Environments (EEs) provide a programming interface or virtual machine that can be programmed or controlled by the active packets. A node operating system manages the resources such as memory regions, CPU cycles and link bandwidth, and multiplexes packets among multiple execution environments (EEs) running on the node.

The objectives of current Node OS interface[14] are to support fast network packets forwarding and fine-grained quality of service. It defines the following five primary abstractions of system resources:

- Thread pool: computation resource.
- Memory pool: memory resource.
- Channel: communication resource, including not only network bandwidth, but also CPU cycle and memory space.
- File system: persistent storage resource
- Domain: the domain is the primary abstraction for thread pool, memory pool, and channel and file system. A domain may create a sub-domain at any time. The principal assigned to a domain is established at creation time and governs the activities of the domain thereafter. The node begins operation with one or more domains assigned to EEs. An EE is permitted to start sub-domain with more explicit packet filters when and as to it wishes.

A security working group[3,11] only gives some requirement and general security architecture; Lindell[4] proposes a protocol of the Hop-by-Hop and attempts to solve message authentication and integrity, however, he doesn't describe secure system architecture. In others thesis[7~10], requirement and challenge are described. Campbell[4] describes security policy framework in Active Bone Network, Campbell[6] present application level security architecture based agent. In this paper, we attempt to present a security architecture that is suggested by our solution and relate that architecture to our implementation. In order to design a flexible and simple secure architecture prototype. The system architecture is based on the security requirements in active networks, threat model, assumption model, and challenges of the meeting those requirements. Definition of secure system architecture model includes authentication, authorization, integrity, and encryption. The model is described by Entry-level security and execute-level security description.

2 Assumption Model

Research for active network security is based on its components. So it is necessary to make security assumption for these components. Security assumption model consists of Node OS assumption, EE assumption, and active code/active packet assumption.

2.1 Node OS security assumption

- Node OS provides API to EEs
- Node OS establishes channels /domain, assigns resources to channel/flows and controls usage
- Node OS start EEs as a channel
- Any channel/domain can start sub-channels/domain with a portion of their resources

2.2 EE security assumption

- Multiple EE in a Node, but it is small number. EEs are installed, replaces, and terminated dynamically. It is not infrequent to change EE and number of EE
- EE can share services and resources. Node OS API must provide for inter-EE calls, creation of shared state, provision for EE policy governance of inter-EE calls and sharing
- EE provide their API to the code in active packets
- EE have services and resources to protect

• Active packet's code (Active code) runs inside EE. Active code is not Node Os level object using EE library.

2.3 Active Packet/Active Code security assumption

• Active codes share services and resources EE must provide for inter-active code calls, creation of share state, and provision for active code policy governance over calls and sharing.

• Active code can change EE state, and then it will change Node state. Active codes include leaving itself behind for other active code to use.

• Packet can be modified by Node, EE or Active Code

3 Threat Model

We briefly describe a simple threat model, which can be used to evaluate the effectiveness of our proposed solution to the active networks security problem. There are four parts that have hoped to protect them in a node. The threat relation of these parts is shown in Fig.1. The first is the active node itself, the second is the execution environment, the third is the sender of packet, and the fourth is the active code/active packet that is executing in the node.

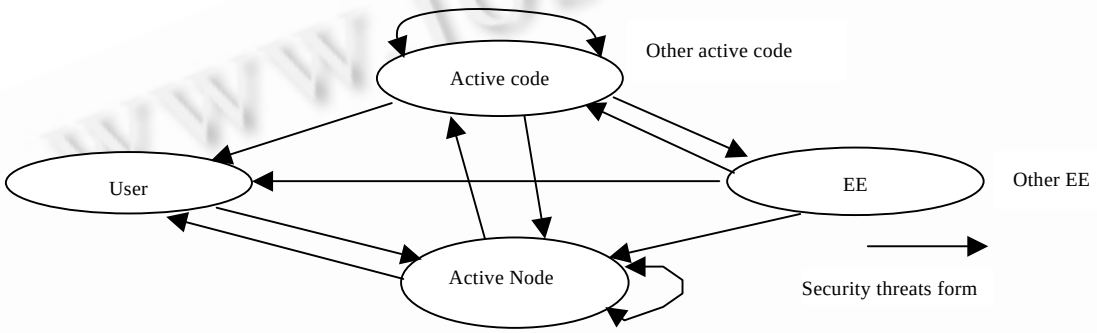

Fig.1 Security threats model

3.1 Node's viewpoint for threat

The node hopes to protect its resource against unauthorized usage, protect the availability of its services, protect the integrity of the state that will allow it to continue to offer services, and protect its state against unauthorized exposure. The node can feel these threats coming from the execution environment (because it may consume resource, or modify the node state, or view sensitive node statement) from the sender of the packet (who may consume resources, or modify the node state, or view sensitive node statement while executing in the EE).

3.2 EE's viewpoint for threat

The EE can feel these threats coming from other EEs, from the senders of packets, and from the active code it hosts, because others EEs may consume resources of active node that should allocated to EE. At the same time, packets may consume resource of EE.

3.3 The sender's viewpoint for threat

The sender of the active packet hopes to protect the data being transmitted in the packet: ensure the integrity and confidentiality of the data in the packet and ensure other attributes of the packet not represented by the bits in the packet such as the latency through the network. The sender of the packet feels threats directly to the data in the packet from other active code in the node, from the execution environment and from the node itself.

3.4 Active code/active packet's viewpoint for threat

This procession that active code /active packet can operate is described as following:

- Active code may request access to the node, for processing and forwarding.
- A packet may request access to an active code, for servicing (modification, forwarding,).
- An active code may request access to a packet, in order to service that packet. This implies that the active code can identify the packets it wishes to services.
- A node may wish to access a packet in order to process it, delete it, modify it, etc.
- A node may wish to access an active code to install it, to retrieve it, to modify it, to terminate it, etc.

Active code/packet may be able to create state that can be shared data and packet payload from unauthorized exposure or modification, to protect its services from unauthorized use, and to protect its resources against unauthorized usage, the active code sees these threats arising from packets, from other active code, from the EE and from the node. It can protect itself against and other active code. But it cannot protect itself against the EE and node on which it is executing. It can only protect itself by ensuring that it does not forward itself to un-trust worthy nodes or EEs.

4 Security Requirements

From the description of the above section, one can see that there are many entities in an active network, which hope to get protection. The end user at the source and destination, the active node itself, the execution environments and the active code or domain all have security concerns. So the basic requirements for a secure active network are the following.

4.1 Authentication

Authentication is an action of securely identifying the user that is requesting a particular service or resource within an ANN. Authentication is crucial since security critical decisions depend on it, for example, when active packet arrives active node, the active node must ensure that active packet is trusted; When active code is executing in EE, EE must ensure that the active code is trusted, etc.

4.2 Cryptography

Senders of packet don't want to exposure their message to others. They make extensive use of cryptography to achieve security. Cryptography enables the development of protocol to protect information and prove identity.

4.3 Authorization

The conception of authorization is access control that governs who may access what and what restrictions are to be placed on those accesses. For active network there are a plenty of accesses that may occur during the lifetime of a packet inside a node. It is essential that active network security architectures model must consider these to prevent unauthorized access. For example, when active code want to access node resource or EE resource, the active code must get authorization from active node or EE.

4.4 Policy

Policy prevents users that have been granted privileges to perform certain action from abusing those privileges. Policy is always integrated with authorization and authentication.

4.5 Integrity

Integrity of an active network system can only be ensured if the underlying mechanism for providing authorizations prevents any misleading modifications to the system components. The integrity is divided into two

kinds; they are active packet integrity and active code integrity. Their integrity must also be assured by granting the permission to modify them only to those authorized.

5 Secure System Architecture

In this section, in the first part, we analyze security requirements and give meeting security requirements. In the second part, secure architecture model is presented.

5.1 Meeting security requirements

5.1.1 Authentication

Authentication is a process of verifying an identity of an entity. Usually it is based on public key cryptography that means that each entity presenting a request to active network node has to have a public/private key pair and a public key certificate. So two function components must exist and cooperate in every ANN in order to provide authentication. They are crypto engine and authentication engine.

5.1.2 Cryptography

The choice of hop-hop[4] cryptographic techniques is a challenge in active network. Symmetric techniques could be used. However, this requires that every node on the path must be completely trust. It doesn't provide non-repudiation. Asymmetric techniques require only that the originator be trusted and can provide non-repudiation. But, if the packet were changed in a node, Digital signature would fail. Active Network Security Groups give packet format[4] (see Fig.2). The packet payload is divided into areas. Symmetric key cryptography is used in an area that contains varying data. Digital signature is used in other area that contains the code and static data. In order to support cryptography, crypto engine, key database, and key manager must be existed in our secure system architecture model.

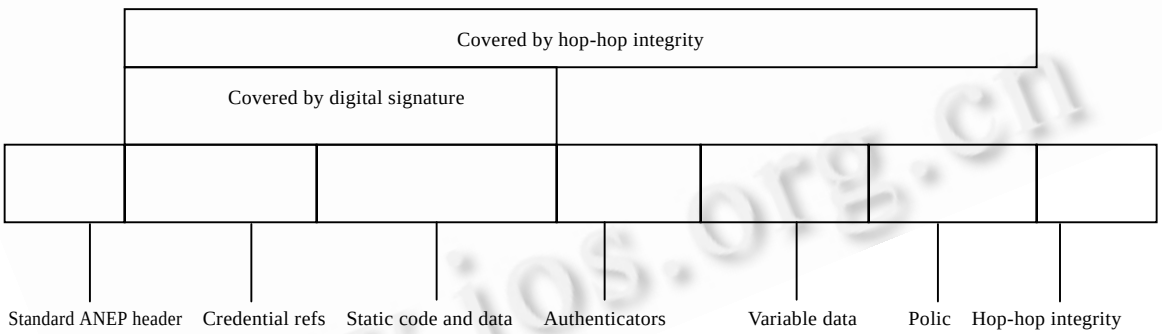

Fig.2 ANEP packet format

5.1.3 Authorization

Authorization is a very important problem in any secure environment. It is an area that governs authenticated entities may access what and what activities they can perform on a resource. There are three major properties of authorization. It is: defining the access restriction types, the components to provide these restrictions and how to deploy these restrictions in the system.

In active network, when Active code arrives at an active node, it will present its credentials, interface and policy. The interface is then provided to the enforcement engine to mediate accesses, which governed by the policy. The credentials are stored in a credentials database for reference when the active code's authorization is questioned. In order to provide policy-based authorization, the five-function components must exist in every ANN: Authorization engine; policy database; policy manager; credentials database; credentials manager.

5.1.4 Policy

It is necessary for policy in a secure system model. According to Ref.[11], enforcement includes the following steps.

- Intercept request for access to an object
- Extract security context
- ANN checks the request against authorization decision
- Evaluate policy

In order to provide policy, three function components must exist in every ANN enforcement engine: enforcement engine; mechanisms for enforcement of hardware resource usage; mechanisms for intercepting function calls.

5.1.5 Integrity

Active code and packet can be tampered with while in transit over the network: malicious user can modify, replay, or even forge packets and code. It is important to note, that in general contents of an active packet comprise a static part, which is not changed while in transiting, dynamic part, which can legally change within the active network node. Thus end-to-end integrity doesn't suffice in Active network. It has to be augmented with hop-by-hop integrity[4], which leads us to the following prerequisite for integrity:

- User has a public key pair and a valid PK certificate
- Each active network node maintains secret key with each of his neighbors

So each packet should contain digital signature of static part of packet and MAC of dynamic part of packet. Packet Format[4] is shown in Fig.2. When active network node receives a packet, it checks contents of the packet against its MAC and digital signature values and validity of the anti-replay value. In order to provide active packet/code integrity, their function must exist in every active network node. They are integrity engine, crypto engine aw well as authentication engine. It also follows that these general mechanisms are required for active packet/code integrity: secure active network node key exchange mechanism, digital signature mechanism, symmetric encryption, asymmetric encryption mechanism required by specific replay protection scheme and hash function.

5.2 Secure system architecture

We refer to Refs.[3,4,11,14,15] and secure system architecture model is shown in Fig.3. We describe the system architecture model through two procession. When a packet arrives an active network node, when an active code tries to execute some action within an active network node.

This secure system architecture model includes the following components (see Fig.3).

- Crypto Engine: it performs the actual cryptographic operation, such as symmetric encryption/decryption, Asymmetric encryption/decryption and hashing. It is used by other components in the security subsystem.
- Key manager (key mgr): key management provides key generation, retrieval, exchange, and agreement, etc.
- Key database (key DB): it stores various encryption keys.
- Integer engine: it contains active code and packet integrity. It depends on integrity protection data contained within an active packet and on crypto engine to do the necessary cryptographic operation.
- Authentication engine: it verifies the authenticity of active packets and depends on authentication data contained within an active packet and on crypto engine to do the necessary cryptographic operation.
- Authorization engine: it make decision if a given request to execute specific action or to access particular object within an active network node is authorized or not.
- Policy database (key DB): store policies.

• Policy manager (Key mgr): creating policies, editing policy, retrieval policy, etc.

• Credential database (Cred DB): store users credentials, such as public key certificates and attribute certificates.

• Credential manager (Cred mgr): when asked by authorization engine, search credential DB and return all credentials. It includes crating, editing credentials, and downloads credentials from an external credential repository.

• Enforcement engine: intercept request for access to an object. Active network node checks the request against authorization decision.

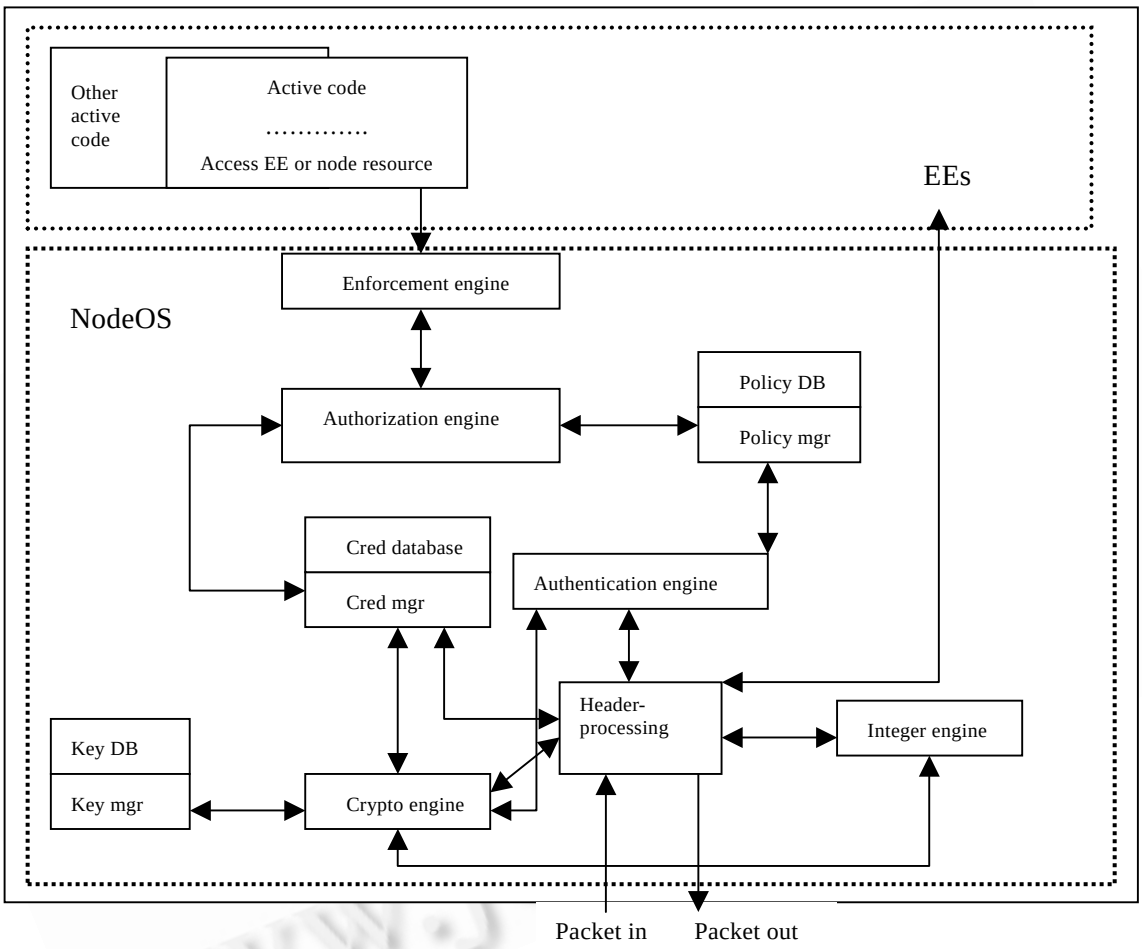

Fig.3 Secure system architecture

When an active packet arrives an active network node, operation of active node security system should be as following steps:

• An active packet arrives at active network node.

• The active network node's header-processing module parse packet.

• The header processing extract information from the packet, the information includes: Digital signature that is used for authentication and integrity, etc.; MAC values, which are used for integrity protection; public key certificate that is used for checking digital signature; Attribute credentials, which are used for authorization.

• The credentials are verified (possibly many credentials exist and possibly a recursive process to verify a chain of credentials). If any credentials fail the verification, the packet processing continues, using those credentials

that succeed.

• Check packet integrity. This operation is executed by Integrity engine. Integrity engine checks packet integrity in following steps: First, it requires crypto engine that performs all cryptographic calculations, to decrypt the integrity token. The token is a MAC value; crypto engine request an appropriate key from key manager. The key manager receives the request, then, fetches this key from key database and gives the key to crypto engine. This encryption process returns a hash of the packet as it was seen by its sender; Second, integrity engine asks crypto engine to calculate hash of the packet; The third step is to compare this hash against the decrypted token. If they are equal, integrity of the packet can be assumed right and processing continue. If these two values differ, integrity check has failed and the packet is dropped.

• Check code integrity. This operation is executed by integrity engine. Integrity engine checks code integrity in three steps. The First it asks crypto engine to decrypt the integrity token. In this case, the token is a digital signature. This decryption process returns a hash of the code that was seen by the code provider. The Second, it asks crypto engine to calculate hash of the code. The last step is to compare this hash against the decrypted token. If they are equal, integrity of the code can be assumed right and processing continue. If these two values differ, integrity check has failed and the packet is dropped.

• Authentication check. The authentication engine sends policy ID to policy manager to verify whether it exists in policy database. The policy manager retrieve in policy database. If it cannot find it, it means error, else it passes authentication checking.

• Security checks result return header processing. The header-processing module makes other processing for the packet. At last the packet is delivered to EE

• END

These steps depict a sequence of security operations that are performed for every packet that arrives at ANN. These security checks are aimed at detecting anything suspicious about this particular packet and, if so, discarding it.

Once an active packet has successfully passed checking, active code(s) can execute and perform operation within an active network node. When active codes access EE resource or NodeOS resource, they have to send request to enforcement engine. The following steps check the request:

• Active codes send request. Information

• Enforcement engine receive the request information, then it will send authorization request information to authorization engine.

• According to request information, authorization asks policy from policy manager.

• Policy manger fetches policy from policy database and gives it to authorization engine.

• Authorization receives the policy, then it send verify credentials request to credential manager.

• Credential manager fetches the credentials form credential database and send credentials verification result to authorization engine.

• Authorization engine check it and make decision. Authorization decision is based on the following information: request information (action, object name, object ID); local security policies; credentials associated with particular object ID; other values.

• Authorization engine send authorization result to enforcement engine.

• Enforcement engine judge by the authorization result.

• END

6 Conclusion and Other Problems

We have developed a secure system architecture model. The system architecture model is based on the security

requirements in active networks, threat model, assumption model, and challenges of the meeting those requirements. Definition of secure system architecture model includes authentication, authorization, integrity and encryption. In order to protect the integrity of the contents of the active packet, encryption and digital signatures can be employed; authorization mechanisms or policies are defined and enforced to provide controlled access to the active node resources. With the development of the active network, the secure system architecture model for active networks may be changed and some flaw would be found in our secure system architecture model.

Though authentication, authorization and other methods are adopted to solve the security of active network, these algorithms will consume plenty of resource of active network. Per active packet will be processed when it arrives the active network node. It must provide one methods to resolve the problem. We are studying one kind of security protocol for active network. We hope to use the protocol and the architecture model to resolve security of active network in the future time.

References:

[1] Tennenhouse, D., Wetherall, D. Towards an active network architecture. In: Proceedings of the Multimedia Computing and Networking 1996. San Jose, CA, 1996.

[2] O'Malley, S.W., Peterson, L.L. A dynamic network architecture. ACM Transactions on Computer Systems, 1992,10(2):110~143.

[3] A Security Working Group. 2000. http://www.choices.cs.uiuc.edu/Security/seraphim/May2000/SecurityArchitecture.pdf.

[4] Lindell, B. Active networks protocol specification for hop-by-hop message authentication and integrity. April 2000. http://www.isi.edu/abone/Documents/Ossec.txt.

[5] Campbell, R.H. Liu, Zhao-yu. Dynamic interoperable security architecture for active network. IEEE OPENARCH 2000, Israel, March 2000. 32~41.

[6] Liu, Zhao-yu, Naldurg, P. Agent based architecture for supporting application level security. In: Proceedings of the DARPA Information Survivability Conference and Exposition. Hilton Head Island, 2000. 129~143.

[7] Liu, Zhao-yu, Campbell, R.H. Securing the node of active networks. In: Hariri, S., Lee, C., eds. Active Middleware Services. Boston, MA: Kluwer Academic Publishers, September 2000.

[8] Alexander, D.S. Safely and security of programmable network infrastructures. IEEE Communication Magazine, 1998,36(10): 84~92.

[9] Alexander, D.S. Security in active networks. LNCS 1603, 1999.

[10] Smith, J.M. Activating networks: a progress report. Computer, 1999,32(4):32~41.

[11] A Security Working Group. Security architecture for active nets. 1998. ftp://ftp.tislabs.com/pub/activenets/secrarch2.ps.

[12] Tennenhouse, D.L. A survey of active network research. IEEE Communications Magazine, 1997,35(1):80~86.

[13] A Node OS Working Group. NodeOS interface specification. 2000. http://www.cs.princeton.edu/nsg/papers/nodeos.ps

[14] Cavert, K.L. Architecture framework for active networks. http://www.cc.gatech.edu/projects/canes/papers/arch1-0.ps.gz.

[15] Initial Active Network and Active Node Architecture. 2002. http://www.ist-fain.org/deliverables/del2/d2.pdf.

主动网络安全结构模型设计

夏正友, 张世永

(复旦大学 计算机信息技术系,上海 200433)

摘要: 介绍了主动网络安全系统的假设模型和威胁模型.基于上述模型和主动网络的安全需要提出了一种安全系统结构模型.该安全模型包括授权、认证、完整性检查和加密等.使用加密和数字签名方法来保护主动网络报文的完整性,使用授权和政策来阻止非法访问以及主动节点的资源请求和行为.

关键词: 主动网络;安全结构;模型;设计

中图法分类号: TP393 文献标识码: A