The Proposal of Security Management Architecture using Programmable Networks Technology

Myung-eun Kim*, Dong-il Seo**, Sang-ho Lee***

* Network Security Research Dept, Electronics and Telecommunications Research Institute, Korea
(Tel : +82-42-860-5303; E-mail: mekim@etri.re.kr)
** Network Security Research Dept, Electronics and Telecommunications Research Institute, Korea
(Tel : +82-42-860-3814; E-mail: bluesea@etri.re.kr)
***Computer Science Dept., Chung-Buk University, Korea
(Tel : +82-43-276-2253; E-mail: shlee@chungbuk.ac.kr)

Abstract: In this paper, we proposed security management architecture that combines programmable network technology and policy based network management technology to manage efficiently heterogeneous security systems. By using proposed security management architecture, a security administrator can manage heterogeneous security systems using security policy, which is automatically translated into a programmable security policy and executed on programmable middleware of security system. In addition, programmable middleware that has the features of programmable network can reduce excessive management traffic. We showed that the programmable middleware could reduce the load of management traffic by comparing processing time between the proposed architecture and PBNM architecture.

Keywords: Policy based network management (PBNM), programmable middleware, security management architecture, and security policy

1. INTRODUCTION

With the development of new network technology, many users can get useful information rapidly through the Internet as well as enjoy its various benefits. However, negative effects such as personal information leakage, system infringement, and network intrusion are also increasing gradually. In the 90s, it was not considered a large social problem because the damages of cyber attack (e.g. virus, hacking, etc) were limited to personal computers. As the Internet expands to connect innumerable networks and computers, the techniques of cyber attack are becoming more complex and bold, so that the damage scale of cyber attacks continues to expand. In these days, cyber attacks are considered an important social problem.

Security technologies are also advancing in response to these cyber attacks. The trends of current security technologies are summarized as follows [1]:

- High speed and large capacity security systems
- Integration of security system and network device using standard interfaces
- Applying policy based management techniques for managing security systems centrally using policy.
- Development of active and aggressive response techniques to cyber attack.
- Integration of various security technologies such as intrusion detection, anti-virus, VPN, etc

Nowadays various security systems, which are satisfied with the changing trends of security requirements, have been producing. Therefore, security administrator has difficulty in controlling heterogeneous security systems. Therefore, security management technology, which can manage security systems that are produced by different company, is a topic receiving a great deal of attention in the field of security management [2].

In this paper, we proposed security management architecture that can control efficiently heterogeneous security systems using security policy and programmable middleware that can reduce the load of management traffic. This paper is organized as follows: Section 2 presents the problems of integrated security management systems and the approach of policy based network management using programmable network technologies. Section 3 presents the proposed security management architecture using programmable middleware. In section 4, we verify the performance of the proposed architecture compared with policy based network management. Conclusions are given in section 5.

2. RELATED WORKS

In this section, the trends in integrated security management technologies and policy based network management using programmable network technologies are introduced.

2.1 Integrated Security Management Technologies

Nowadays, security systems have been developed to satisfy changing security requirements and integrated security management has emerged for the efficient management of heterogeneous security systems. Integrated security management technology has advantage of establishing and controlling global security policies, making secure domain by sharing security management information, guaranteeing autonomous security management, and assuring flexibility and scalability

In next paragraph, the typical integrated security management technologies are presented: OPSEC (Open Platform for Security) researched by Check Point Software Technologies, Inc. and Active Security [3] by Network Associates, Inc.

2.1.1OPSEC

OPSEC is the framework proposed by Check Point Software Technologies, Inc. More than 320 security vendors (e.g. Symantec, Axent, RSA security, VeriSign, IBM, Novell, etc) are participating in it as partners. The goal of OPSEC is autonomous security management through interoperability among the security systems of other vendors with priority given to firewalls. OPSEC offers secure networking through the compatibility of Firewall-1 and VPN-1 based on Secure Virtual Network (SVN) architecture. In early stage, OPSEC is to provide enterprise security infrastructure by integrating IDS, Certificate Authority (CA), and so on.

To support consistent and efficient security management, OPSEC provides the protocols for exchange of information and mutual interoperability between different security systems such as Content Vectoring Protocols (CVP) and Suspicious

Activity Monitoring Protocols (SAMP), and the applications for convenience of management such as Log Export API (LEA) and User-to-Address Mapping (UAM). However, OPSEC has shortcomings in that every security system has to support the protocols and the applications, and limitation that is possible to integrate only the security system with the interface provided by OPSEC.

2.1.2 Active Security

Active Security is also an autonomous centralized security management framework proposed by Network Associates, Inc to interoperate its own security products. Currently, some products of NAI such support Active Security environments. Active Security consists of sensors, arbiters, actors, and an event orchestrator. Sensors are responsible for gathering multiple dimensions of information. Also, they monitor network attacks and detect viruses. Arbiters make their decisions based on the information collected by sensors. Actors cope with security threats according to the decision of event orchestrator. An event orchestrator decides methods of response according to each security problem, and sends the decided response to actors. Active Security has the same architectural limitations as OPSEC, which can only interoperate NAI's own security products.

2.2 Programmable Network Technologies

Programmable networks [4] can process packets using store-compute-forward operations through additional computing ability whereas existing routers use store-and-forward operations. Existing routers do not have computing ability, so they have to process all jobs at end systems. On the other hand, programmable nodes have computing power to do intermediate processing using programmable packets. Therefore, it has better performance than existing end-to-end methods. It provides flexible and fast deployment of new network services, and flexibility, security, and manageability.

Briefly, we consider the FAIN (Future Active IP Networks) project [5] of IST (Information Society Technologies) and the PBNM related project [6] of OKI Electric Industry Co. that apply programmable network technology to policy based network management (PBNM).

2.2.1 FAIN

The FAIN project of IST developed and validated an open, flexible, programmable, and dependable network architecture based on active node concepts. Policy Based Active Network Element Management (PBANEM) that is a part of FAIN applies active network technology to PBNM. It follows the standards of both the Internet Engineering Task Force (IETF) and the Distributed Management Task Force (DMTF).

PBANEM consists of a Policy Decision Point (PDP) and Policy Enforcement Points (PEPs). In the existing PBNM, if there are several different types of PEP, PDP have to create all kinds of policies for all PEPs. However, PBANEM can enforce a meta-policy to different types of PEPs because meta-policy is interpreted into all kinds of policy using policy interpreter. PBANEM uses active network technologies to make policy distribution and enforcement more efficient than existing PBNM does.

2.2.2 Application of Active Networking to Policy Networking of OKI

OKI researched the application of active packets to support quality of service (QoS) in IP networks. At first, OKI applied PBNM to IP networks, but existing management protocols of SNMP (Simple Network Management Protocol), COPS (Common Open policy Service), and CLI (Command Line Interface) generated excessive management traffic. Therefore, these protocols increase control traffic and provide poor scalability, and so, active network technologies have been applied to solve these problems of existing PBNMs.

In this project, the proposed architecture consists of a manager and active nodes that are network devices. Both the manager and network devices use a system called the Active Program Execution System (APES), in which active programs are executed and forwarded to other APES. Active programs can route autonomously and traverse all nodes that should be controlled. This method reduces control traffic and overhead of network nodes and offers customizability, scalability, and flexibility to applied networks.

3. SECURITY MANAGEMENT ARCHITECTURE USING PROGRAMMABLE MIDDLEWARE

It is very hard for a security management system to manage heterogeneous security systems, which are produced by different companies. Therefore, the existing security management systems can usually manage security systems that are produced by the same company. To reduce these difficulties of management, It is one of solutions to deploy security systems that are produced by the same company on the network. However, as the size of the network is increasing, it is difficult to deploy the same company's products on the whole network. Compatibility among security systems has been researched, but the problem of cooperation and management of different companies' products still remains.

In this paper, we proposed a security management architecture to solve problems as stated above. Fig 3.1 represents the proposed security management architecture.

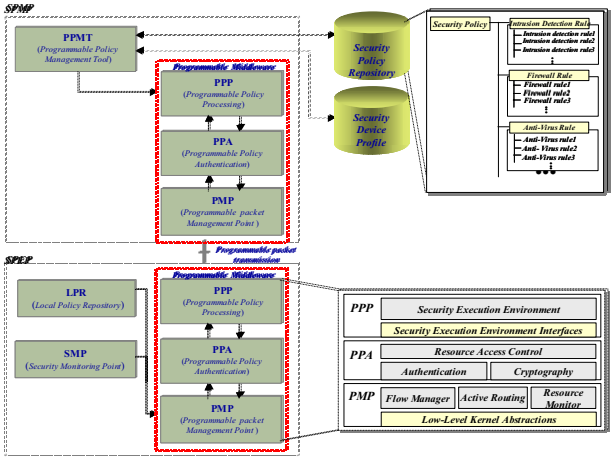


Fig 3.1 the proposed security management architecture

The proposed architecture consists of two parts: SPMP (Security Policy Management Part) and SPEP (Security Policy Enforcement Part). SPMP creates security policies, stores them in a security policy repository, and distributes them to the managed security systems. SPEP executes security policies, after it checks for conflict between the new security policy and the old security policy that has been stored in the LPR. More detailed functions of each component will be explained in the following paragraph.

3.1 PPM

PPM manages security policies, that is, it creates security policies for each security system and stores them in a security policy repository. The security policy is the abstracted configurations and commands of security systems such as intrusion detection rules, intrusion-blocking rules, and anti-virus rules, etc. An administrator doesn't have difficulty in managing heterogeneous security systems because security policy, which is an abstracted format of configuration and command, is automatically executed on programmable middleware which is embedded in security system. These security policies are classified into several groups, for example security policies for IDS, security policies for firewalls, security policies for anti-virus systems, and so on.

Fig 3.2 represents security policy for firewalls, which is represented in two formats, abstracted format and programmable format. When the PPM distributes a security policy, the PPM searches a security policy repository and then translates the retrieved security policy into a programmable security policy, which is composed of various execution commands for all security systems using information of profile database. If there is not an identical security policy, the PPM creates a new security policy. PPM sends the translated programmable security policy into programmable middleware.

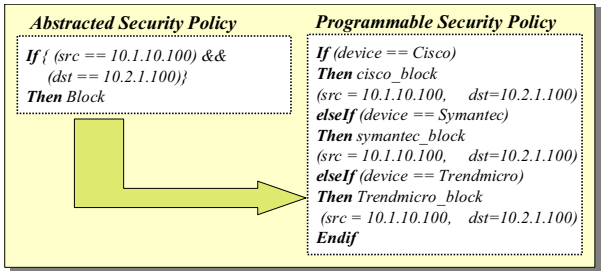


Fig 3.2 the example of security policy

3.2 Programmable Middleware

A programmable middleware consists of PPP (Programmable Policy Processing), PPA (Programmable Policy Authentication), and PMP (Programmable packet Management Point). PMP processes a programmable packet and PPA performs authentication and cryptography for the security of a packet. PPP executes a programmable security policy. Fig 3.4 represents the architecture of a programmable middleware. The detailed functions of each module will be explained in the followings.

3.2.1 PPP

PPP is a programmable policy processing point. It receives a programmable security policy from PPM and puts it in the payload of a packet. Also, PPP receives a programmable security policy from PPA and executes a programmable security policy. The security policy is executed in a security execution environment and other execution environments are not provided in this proposed architecture.

3.2.2 PPA

PPA is a programmable policy authentication point that ciphers a packet using an appropriate cryptography algorithm, for example 3DES, according to the performance of the programmable middleware. PPA certifies the sender and the sending network of a packet and permits access to resources by access authorization.

3.2.3 PMP

PMP is a programmable packet processing point and changes an IP packet from PPA to a programmable packet. PMP also changes a programmable packet from SPMP to an IP packet conversely and then sends it to PPA. The flow manager module manages scheduling for a programmable packet and the active routing module performs communication between security systems. The resource monitor module manages all resources and distributes pertinently.

3.3 LPR

LPR stores a security policy from SPMP temporarily. If the security system requires the same security policy that had been enforced or a neighbor security system requires the retransmission of a security policy because of transmission failure, the security policy in LPR is reused.

3.4 SMP

SMP monitors whether the security policy is properly performed in SPEP and checks whether a new security policy violates the existing security policy. SMP sends an alert message to the policy server, if a conflict occurs.

3.5 Interfaces

The ANEP (Active Network Encapsulation Protocol) packet format is used as a programmable packet format. The programmable packet designed by [7] for secure communications in programmable networks.

The proposed architecture uses transmission technology in [8] that is based on IP and UDP protocol and guarantees reliability. In this paper, ANEP header [9] is attached after the IP header and the UDP header. In the proposed transmission technology, the old options of the ANEP header including Source Identification, Destination Identification, Integrity Checksum, and N/N (Non-Negotiation) Authentication are used, and 4 new options are added. The new options are explained as follows:

- Credential: Verifies the user using X.509 format credential, which creates the programmable packet and authenticates the programmable node
- Signature: Validates the received programmable packet
- In-line policy: Authorizes the data or program of the received programmable packet
- Hop-hop Integrity: Validates authentication, messages, integrity, etc. of the sender that creates the packet as well as intermediate hops

We assume the execution environment proposed in [8] where programmable nodes are located and programmable packets are transmitted:

- The senders and the receivers of programmable packets are programmable nodes including programmable application that uses the Active Packet Transmission Engine (APTE)
- Programmable applications use the 'Loose Source Route' option of the Source Route Options [10]

We use the OSPFv2 Opaque LSA (Link State Advertisement) Option [11] proposed by ALCATEL for programmable nodes to discover the locations and characteristics of each other. We modify following values of the opaque LSA option:

- Opaque Type: Use 128 that is used for experimental usage in IANA
- Opaque ID: Use 0 to recognize opaque information as EE (Execution Environment) ID
- Opaque Information: Use the EE ID of the sender

programmable node.

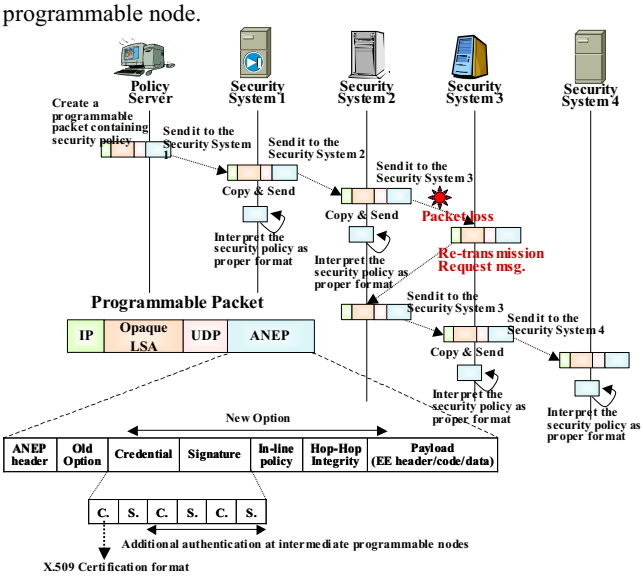


Fig 3.3 Packet flow of a programmable packet containing security policy

Fig 3.3 shows the packet flow of a programmable packet containing security policy that is created in the policy server. The security system copies the programmable packet as soon as it receives it, and then sends it to a next security system while removing the header of the copy packet and interpreting the ANEP part into a format that security system can understand. These processes are repeated until the programmable packet is delivered to all security systems. It finds the next security system using the opaque LSA option.
The reliable transmission of a programmable packet in the proposed architecture is also shown in Fig 3.3. A programmable packet verifies the sequence number to check packet loss and to guarantee reliable transmission. The begin number of the sequence number and the number of transmission packets are contained in the first packet of a sending message. If a receiver recognizes the packet loss, it sends a retransmission request packet to a nearby security system that is a programmable node. A retransmission request packet includes the sequence number of the lost packet. A nearby security system sends the requested programmable packet to the requester using the stored copy of programmable packet as soon as it receives a retransmission request packet.

3.6 The Scenario of proposed architecture

In this paragraph, we will explain a deployment scenario using the proposed management architecture. First, we will present the scenario of security policy distribution for various security systems that are produced by different companies and then describe a scenario of intrusion isolation through cooperation of security systems.

3.6.1 A Scenario of Security Policy Distribution

Fig 3.4 represents the distribution of a security policy to heterogeneous security systems that are produced by different company.
PPM in the policy server creates a security policy and stores it in a security policy repository. PPM retrieves information of security systems from a profile database and changes the abstracted security policy to a programmable security policy that can be executed in a security system. PPP in a programmable middleware puts the programmable security policy in the payload of packet. SPA ciphers the packet and appends the packet header for authentication. PMP changes an IP packet to a programmable packet and then sends it to a security system.
Security system 1 changes the programmable packet to an IP packet in PMP and SPA certificates whether the packet is valid or not. SPA deciphers the packet, if the packet is valid. PPP executes the deciphered programmable security policy and sends the packet to the next destination at the same time. In this manner, security policy is sent and executed from the security system 1 to security system 4. After security policy is enforced to security system 4, security system 4 send complete message to policy server.

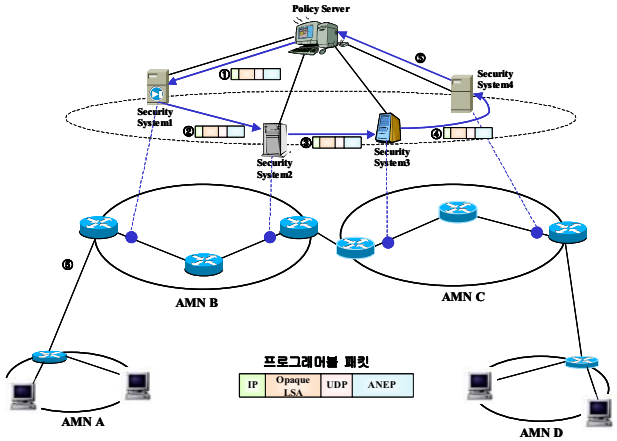


Fig3.4 A scenario of security policy distribution

3.6.2 A Scenario of Intrusion Isolation

Fig 3.5 represents the scenario of intrusion detection and attack response by blocking harmful traffic using the proposed security management architecture. We will explain the scenario step by step.

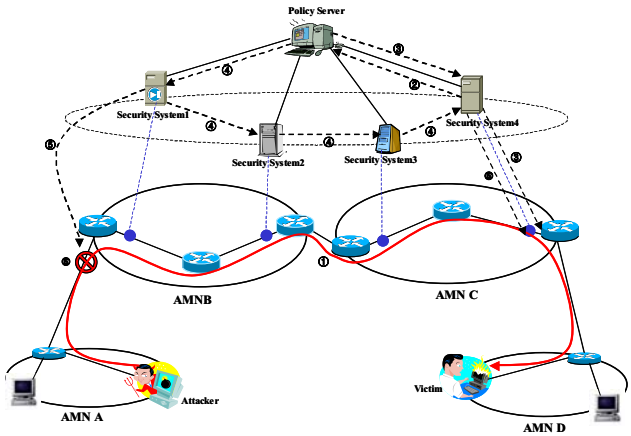


Fig 3.5 A Scenario of Intrusion isolation

(Step 1) An attacker in AMN A attacks a victim in AMN D.
(Step 2) Security system 4 detects harmful traffic and sends alert messages to the policy server.
(Step 3) The policy server analyzes alert messages and sends a blocking policy to security system 4
(Step 4) To block the source of harmful traffics, the policy server sends a blocking policy to a security system and the blocking policy in the format of a programmable packet goes round all security systems and only the security system that is

located in the source network of harmful traffics executes a blocking policy.
(Step 4-1) The policy server creates a blocking policy and changes the programmable security policy through the information in the profile database. After that, the policy server sends to security system 1.
(Step 4-2) Security system 1 copies the programmable security policy and sends it to security system 2.
(Step 4-3) Security system 2 copies the programmable security policy and sends it to security system 3.
(Step 4-4) Security system 3 copies the programmable security policy and sends it to security system 4.
(Step 5) At the same time as active routing, each security system checks whether the source network of harmful traffics are their own network and then decides whether to execute the security policy.
(Step 5-1) Security systems 2,3,4 do not execute the security policy because their own network is not the source network of harmful traffic.
(Step 5-2) Security system 1 detects that the harmful traffic has occurred.
(Step 5-3) Security system 1 executes a programmable security policy.
(Step 6) The harmful traffic is dropped, and the blocking policy that is executed in security system 4 is removed.

4. PERFORMANCE VERIFICATION

In this paragraph, when the policy server manages several security systems that are produced by different companies, we will verify the performance of the proposed architecture by computing the processing time for policy creation and policy enforcement in comparison with PBNM architecture.

4.1 Performance Verification on PBNM based Architecture

In the PBNM architecture, a server communicates with a client using COPS, which is based on TCP, and a server sends a security policy as a PIB (Policy Information Base) format to a client. A server keeps connections with a client and must check if the connection is alive or not. After a server checks whether the connection is alive, it sends a security policy to a client. A client that receives a security policy sends a 'report' message to a server.
Fig 4.1 represents the distributing process of a security policy in the PBNM architecture. We assume that an administrator can put a security policy, that is, command of security systems that are produced by various companies

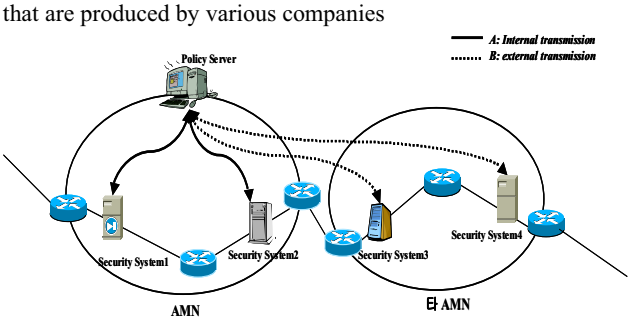


Fig 4.1 A PBNM based architecture

Let T1 be the processing time for distribution and execution of a security policy. T1 can be derived as described below and the parameters used in all equations are represented in table 4.1. Let A be the processing time during which a policy server distributes a security policy to security systems in the same AMN (Autonomous Management Network), and let B be the processing time during which a policy server distributes a security policy to security systems in other AMN (Autonomous Management Network).
We can compute T1 is sum of A and B as follows:

$$T1 = \{A + B*(Nm-1)\}*Nrt \quad (1)$$

A means the sum of processing time such as time for creating a security policy, sending to all security systems in the same AMN, and executing the security policy. A is represented as follows by table 4.1:

$$A = (Tpc+Tpt+Tin*2+Gs + Tpp)*Ns \quad (2)$$

B means the sum of processing time such as time for creating the security policy, sending to all security systems in other AMN, executing the security policy. B is represented as follows by table 4.1:

$$B = (Tpc+Tpt+Tout*2+Gs + Tpp)*Ns \quad (3)$$

Table 4.1 the parameters for enforcing security policy

Name	Description	Type	Value
T_{in}	Transmission delay time of policy/message in the same AMN	Integer	10msec
T_{out}	Transmission delay time of policy/message between AMNs	Integer	100msec
T_{pt}	Translation time of policy in a policy server	Integer	20msec
T_{pp}	Processing time of policy in security system	Integer	30msec
T_{pc}	Creation time of policy in a policy server	Integer	10msec
N_m	Count of managed AMN	Variable	1->10
N_s	Average count of security systems per a AMN	Variable	1->10
G_s	Processing delay time of a packet in security system	Integer	1msec
N_r	Count of repeated transmission	Variable	-
N_{rt}	Average count of retransmission	Variable	-
F_h	Probability of transmission failure	Integer	2%
F_t	Target probability of transmission failure	Integer	5%
S_h	Probability of transmission success	Integer	98%

By equation (2),(3), equation (4) is derived.

$$T1 = [\ \{\ (Tpc+Tpt+Tin*2+Gs + Tpp)*Ns\ \} + \{\ (Tpc+Tpt+Tout*2+Gs + Tpp)*Ns\ \} * (Nm-1)\ ] * Nrt \quad (4)$$

Let Nrt be the average count of retransmission. Nrt can be derived as 1/(1-Fh). We assume the probability of transmission failure, Fh, is 5%. Therefore, T1 is computed by equation (5).

$$T1 = [\ \{\ (Tpc+Tpt+Tin*2+Gs + Tpp)*Ns\ \} + \{\ (Tpc+Tpt+Tout*2+Gs + Tpp)*Ns\ \} * (Nm-1)\ ] * 1/(1-Fh) \quad (5)$$

4.2 Performance Verification of Proposed Architecture

Fig 4.2 represents the distribution of a security policy to security systems in the proposed architecture. In the proposed architecture, a policy server creates a programmable security policy, which is executable on each security system and sends to security systems that are located in the same AMN. After that, a security system transfers a programmable security policy to security systems in all managed AMN and each security system executes it.

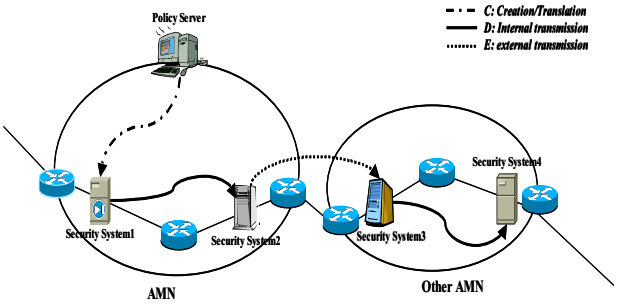


Fig 4.2 the proposed architecture

Let T2 be the processing time during which a programmable security policy is sent to all security systems and executed. We assume the probability of transmission success, Fh is 98% and the target probability of transmission failure, Ft is 5%. If Nm is a count of AMN, T2 is represented as follows:

$T2 = \{ C + D*Nm + E \} * Nr$ (6)

C means the sum of processing time such as time for creating a security policy, translating a security policy to a programmable security policy, executing a programmable security policy. C is derived as follows:

$C = Tpc + Tpt + Tin$ (7)

D means the processing time to transfer and execute a security policy in the same AMN. D is derived as follows:

$D = (Ns * Tpp) + (Ns-1)*(Tin + Gs)$ (8)

E means the processing time to transfer a security policy between AMNs. E is derived as follows:

$E = Tout * (Nm-1)$ (9)

By equation (7),(8),(9), T2 is represented as follows:

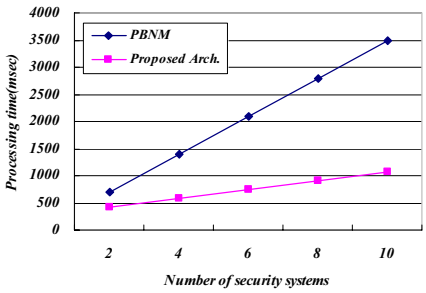
$T2 = [\{ Tpc + Tpt + Tin \} + \{ (Ns * Tpp) + (Ns-1)*(Tin + Gs) \} * Nm + \{ Tout * (Nm-1) \}] * Nr$ (10)

In equation (10), Nr is the count of repeated transmission and we can compute Nr, which is the condition that Ft is larger than Fh is satisfied.

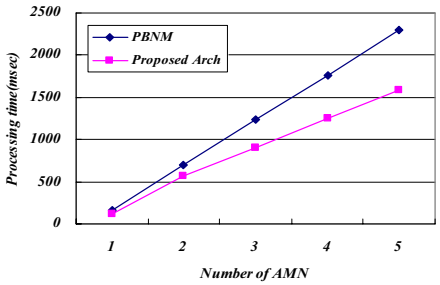
$Nr (Ns, Fh, Nm) = (Ns*Nm-1) * Fh , Fh (Nr, Sh) = (1 - Sh) * Nr$

By equation (5), (10), we can compute T1 and T2. Fig 4.3 illustrates the value of T1 and T2, if Nm = 2, Ns is increased by 2 from 2 and Fig 4.4 illustrates the value of T1 and T1, if Ns = 2, Nm is increased by 1 from 1.

Fig 4.3(a) shows that the proposed architecture in comparison to PBNM architecture improves the performance of processing policies/messages and the processing time is reduced about 170%. Fig 4.3(b) shows that, in the proposed architecture, the processing time is reduced about 33%.



(a) The value of T1 and T2, , if Nm = 2, Ns is increasing



(b) The value of T1 and T1, if Ns = 2, Nm is increasing

Fig 4.3 comparisons T1 with T2

5. CONCLUSION

In this paper, we proposed a security management architecture using programmable middleware. The security management architecture enables an administrator to manage heterogeneous security systems and programmable middleware can reduce excessive management traffic. We showed that the programmable middleware could reduce the load of management traffic by comparing processing time between the proposed architecture and PBNM architecture.

REFERENCES

[1] "Conceptual model description of Active Security System for Next Generation Network V1.0," *Information Security Research Div., ETRI,* Jun. 2002.

[2] Y. Choi, "White paper: The compass of information security – ESM Introduction," *proceeding of 1st workshop on cyber terrors*, 2002

[3] Gerhard Eschelbeck, "Active Security- A proactive approach for computer security systems," *Journal of Network and Computer Applications 2000,* pp.109-130, 2000.

[4] Alex Galis, et al., "A Flexible IP Active Networks Architecture," *Proceedings of International Workshop on Active Networks*, Oct. 2000.

[5] Alex Galis, et al., "Policy-Based Network Management for Active Networks," *IEEE ICT 2001 Conference proceedings*, June 2001.

[6] Kei Kato, et al., "Application of Active Networking to policy networking," *OKI*, Japan.

[7] Jiyong Lim, "Design of security enforcement engine for active nodes in active networks," *The International Conference on Information Networking (ICOIN) 2003*, vol.1, 2003.

[8] Kijoon Chae, et al., "Final report: A Study on the optimized modeling of the sensor communication architecture," *Ewha Womans Univ., Funded by ETRI in Korea,* 2002.

[9] D. Scott Alexander, et al., "Active Network Encapsulation Protocol (ANEP)," *http://www.cis.upenn.edu/switchware/ANEP/docs/ANEP.txt*, 1997.

[10] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6)," *IETF RFC 2460,* 1998.

[11] D. Galand, O. Marce, "Active Router Information in Routing Protocols," *IETF Internet Draft*, 2000.