Access Control List Mediation System for Large-Scale Network

Kanghee Lee*, Zhefan Jiang*, Sangok Kim*, Sangwook Kim*, Sunkyung Kim**
**Dept. Computer Science, Kyungpook National University, South Korea*
***Dept. Computer and Information Engineering, Daegu University, South Korea*
{khlee, zfjiang, sokim, swkim}@cs.knu.ac.kr, skkim@daegu.ac.kr

Abstract

Large-Scale network such as the Internet consist of a large number of sub-networks. Since each sub-network has management privileges and policies individually, it is difficult to respond to harmful traffic such as warm viruses that affect entire networks. In this paper, we propose a high-level policy language and a middle-level data structure. They were named Triton language and Common Access Management Form (CAMF). It enables the administrator to authorize a policy effectively and rapidly. Large-scale networks have a number of administrators so that policy collision can occur. Access control list mediation system selects or adapts the most important policy among colliding policies through a policy importance valuation.

1. Introduction

Most of the Internet infrastructures were designed more to withstand physical failures such as broken wires or computers rather than harmful traffic launched by legal network users. The rapid growth of the Internet, however, coupled with its cost-effective capability to move data across geographically dispersed heterogeneous information systems, has made it a virtual breeding ground [1].

It is particularly difficult for each network administrator to individually respond to harmful traffic that influences large-scale network. An architecture that allows a network administrator to manage entire networks simultaneously is, therefore, necessary.

In this paper, we propose a high-level language that specifies the security management policy for large-scale network and a mechanism for rapidly delivering such a policy to the entire network. A high-level policy language (Triton) allows a domain administrator to manage large-scale networks consisting of heterogeneous devices from an abstract point of view. And a middle-level data structure (CAMF) is the compile result of high-level policy language that using by mediation system. This system enables the administrator to enforce a security policy onto a large number of sub-networks and lower layer devices simultaneously.

2. Access Control List Mediation System Architecture

A large-scale network consists of a lot of sub-networks, and each sub-network includes smaller sub-networks. The Internet structure is hierarchical, like a tree structure [2]. To manage a large-scale network such as the Internet, we must consider such network structure from a hierarchical point of view.

To perform the large-scale security management effectively, a security policy should be sent to each sub-network simultaneously. This is done because of the characteristics of traffic such as the worm virus, which has recently done much damage to the Internet. The extent of damage is not restricted with in a sub-network but expands into the entire network. To respond to the harmful traffic and minimize the damage, a domain administrator must control the entire net-work. Harmful traffic such as the worm virus spreads rapidly so the control must also be more rapid. The control is performed by sending a policy to each lower layer device. In this process, it is necessary for a domain administrator to simultaneously enforce the policy upon the entire network. In a large-scale network, the policy should be of a high-level language because a domain administrator is not expected to be aware of all network information separately.

Since each lower layer device is controlled by various ACL, the policy for the entire network may become innumerable. The relationship with the network structure and the traffic form is low because ACL is a simple control rule. To integrate various ACL devices and represent the semantic relationship with a network structure and traffic form, it is necessary to

Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'05)
0-7695-2405-2/05 $20.00 © 2005 IEEE

use a high-level language that specifies abstract policies for network management.

2.1 Hierarchical Domain Structure

A large-scale network such as the Internet has a hierarchical structure as shown in Figure 1. Each sub-network consists of smaller sub-networks that have the privileges to enforce security policies to lower layer devices [3].

In this paper, the domain is defined as a unit that can establish and enforce a security policy. Domains may be used to model the hierarchical structure of the system, which also enables policy specification and logically-centralized system administration.

As shown in Figure 1, a domain in a large-scale network is an abstraction of sub-networks [4]. It has different structures and characteristics. Although the policy has the same meaning, the context of policy may be different in each domain. The policy of each domain can influence the policy of other domains. It needs the basic information for device, traffic and network service to specify a policy.

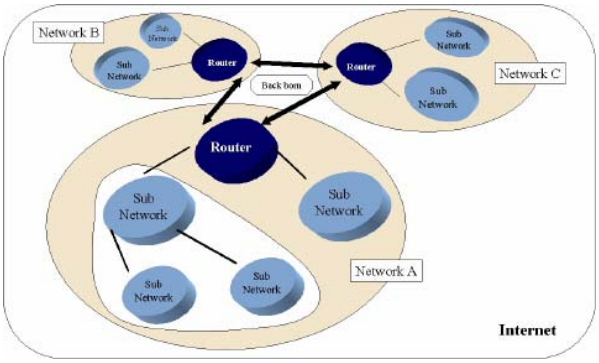

Fig 1. The Structure of Domain

In a large-scale network, a security policy is authorized by a higher domain administrator, who must have information about all lower domains. However, a higher domain administrator cannot know all the information. The forms of policies may also be changed a little even though they have same semantic.

2.2 High-Level Security Policy

Generally the traffic control form in network devices uses network packet information such as the IP address, port number and protocol. The control of traffic in the domain requires details such as its architecture, service type and administrative policy [5]. We cannot, however, disregard the details when making a policy from the network management point of view. This is because large amounts of domain details and particular policies are a little different depending on each domain.

Therefore, large scale network policies provide framework, take a side view of the abstract, and convert abstractive policies into specific policies for practical applications. To authorize a policy such as the above, an abstractive approach for the entire network is necessary. The common attributes of abstracts are the meanings and structures of each domain consisting of a large-scale network.

The high-level policy language proposed in this paper is able to describe traffic information in the abstractive aspect and approach the network information of each domain in an intuitive method. It is able to write and analyze a policy rapidly as an approaching general grammar structure for programming language. The following is an example of the high-level language:

```
Policy SamplePolicy triggered by EVENT_ALERT
{
    Range R1 = [ x:IP | "10.1.1.5" <= x <= "10.1.1.20" ];

    Incoming {
        For ( "DomainA", "DomainB", "DomainC" )
        {
                if(src_addrinR1&&dst_port==8080)
                {
                        deny(3,essential);
                }
        }
    }
}
```

The policy is named SamplePolicy and is triggered by the alter event. R1 defines the range of policy enforcement and represents the IP address range between “10.1.1.5” and “10.1.1.20” The policy is enforced in “DomainA”, “DomainB” and “DomainC” to deny an incoming packet with 8080 as a destination port. The parameters of the denial statement are for policy mediation. ‘3’ represents a priority while ‘essential’ represents an essential policy. As shown in above example, a high-level language uses intuitional keywords such as ‘srcaddr’ and ‘dstaddr’ to represent detail conditions. These keywords generalize control forms of network devices and allow a domain administrator to authorize a policy effectively. The syntax of the high-level language is similar to the general programming language syntax. It uses a block for efficient policy structure.

2.3 CAMF(Common Access Management Form)

High-level language enables the enforcement of a policy without detail information about each domain. It is, however, difficult to enforce a policy on lower layer domains directly because the ACL for controlling the lower layer device is very concrete and varied

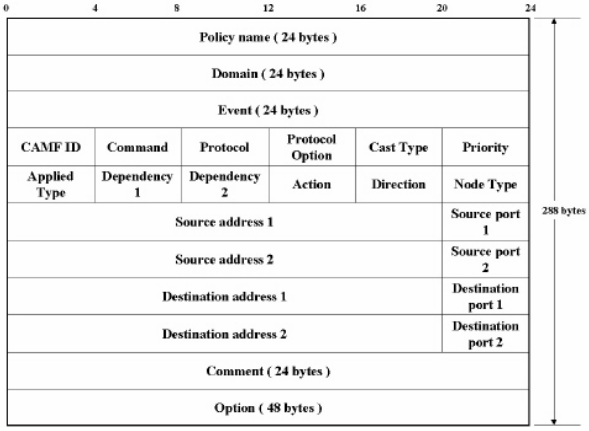

Fig 2. The Structure of CAMF

The CAMF is a mediator between the high-level language and the ACL of the lower layer device. The CAMF makes it easy to translate a high-level language to ACL. The CAMF should include information on various forms of lower layer devices. The Figure 2 shows the structure of the CAMF:

A CAMF includes the basic policy information, ACL information for a lower layer device, and the policy mediation information. A policy has a unique name and is shown as a list.

3. Access Control List Mediation Mechanism

A Security mechanism for a large-scale network divides the entire network into higher domain and lower domain. Since the administrator of each domain can individually authorize a policy, policies of higher domains and lower domains may collide with each other. When policy collision occurs, one of the colliding policies must be selected while the rest should be discarded.

3.1 Policy Collision

When many policies that cannot be executed together in the same area are applied simultaneously, policy collision occurs. Figure 3 represents a typical example of policy collision.

This is a situation of executing a policy to deny a packet with destination port 8080 in some domain. An acceptant policy accepts a packet that with destination port 8080 delivered from a higher domain. In this case, we simultaneously apply two impossible policies which are denial and acceptance of destination port 8080 in the same area. This is where policy collision occurs.

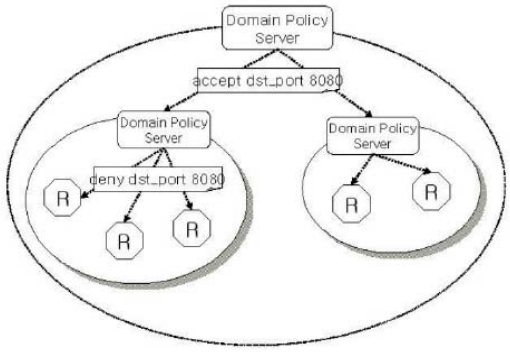

Fig 3. The Situation of Policy Collision

3.2 Policy Mediation

Policy mediation is a process used to select one policy among policies that collide. The selected policy must semantically be most important. The process can be done using the following methods:

First, a domain policy server enforces a higher domain policy above all. This method, however, discolors the meaning of the hierarchical domain layer structure because the highest domain has absolute privilege in the entire network.

Second, a domain policy server enforces the most recent policy. This method preserves the meaning of the hierarchical domain layer structure because the privileges of the higher and lower domains are treated equally in the entire network. However, this method raises a problem because it does not recognize the priority of the higher domain policy.

Lastly, a domain administrator personally selects a policy among colliding polices. In this case, the domain administrator's intension is reflected. However, the speed of policy enforcement may become slower. This method is, therefore, not appropriate when responding to harmful traffic, such as the worm virus, which propagates rapidly.

Consequently, policy mediation in a large-scale network sufficiently reflects the domain administrator's intention. The policy of a higher domain must be enforced as preferentially as possible. Privilege for policy enforcement of each domain should be treated equally, though.

Policy mediation requires policy importance valuation. The core factor for policy importance valuation is the domain administrator's intention. The policy should, therefore, include information that reflects domain administrator's intention [6].

In this paper, policy importance is presented by priority. A domain administrator invests on a policy with priority according to the provided criterion.

When a domain administrator authorizes a policy, a policy group is defined with 'policy' as a keyword of the high-level language, which invests the CAMF list with the same policy name. However, each CAMF may have a different priority according to the domain administrator's intention even though each CAMF is included in the same policy group. In a process of policy mediation, a unit for comparison between policies is not the CAMF list, but a single CAMF. The following is an example of policy mediation according to policy priority:

In Figure 4, there are three CAMF lists that are classified as A, B, C. A and B are currently applying policies while C is a new policy. The policy priority of each CAMF is shown in parentheses.

A and B do not collide with each other because they passed through policy mediation. However, A and C may collide with each other, and B and C may collide with each other. Suppose that policy collision occurs in the following:

CAMF A3 and C1 collide with each other and B4 and C3 collide with each other. One between A3 and C1 must be selected in this case and what ever is left must be discarded. B4 and C3 should be dealt with in the same manner. The policy mediation is basically performed by priority. Therefore, in A3-C1, A3 is selected and C1 is discarded. In B4-C3, B4 is discarded and C3 is selected. The final policy is shown in Figure 4.

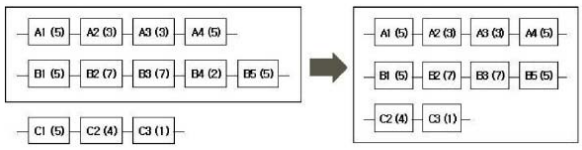

Fig 4. The Policy Mediation Process

The core criterion for policy mediation is a negotiation between the higher domain and lower domain. Basically, the privilege of the higher domain is higher than the privilege of the lower domain. However they should promise a range of privileges before a process of policy mediation. In this paper, such promise is called a domain negotiation.

The privilege of a domain administrator is the range of priority that a domain administrator can vest a policy with. The items included in the domain negotiation information are the following:

- Entire range
- Direction (incoming packet/outgoing packet)
- Protocol (IP/ICMP/IGMP/TCP/UDP) / item IP address range
- TCP/UDP port range / item lower layer device type (router/firewall/web server)

The higher and lower domains have separate policy priorities for each range. Each domain divides privileges for policy enforcement. Figure 5 shows the information structure for domain negotiation.

If the priority for the entire range is 5, the priority for TCP protocol is 4 and the priority for destination port from 2000 to 3000 is 3, then the priority for destination port from 2000 to 3000 is duplicated in three ranges: entire range, TCP protocol range, destination port range. In this case, the selected priority is 3 because the highest priority among the priorities for duplicated ranges is 3.

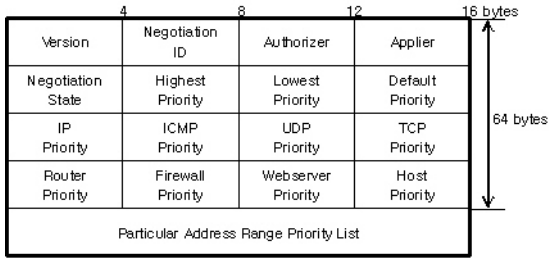

Fig 5. The Structure of Domain Negotiation

Basically the higher domain has a higher priority than lower domain. Therefore, the higher domain has privilege for policy enforcement about ranges not mentioned in domain negotiation. The core of the domain negotiation is to guarantee the absolute privilege for the minimum range that the lower domain man-ages. A higher domain administrator examines such requirement of a lower do-main administrator. Figure 6 shows the sequence of domain negotiation.

The final result of domain negotiation is a contract for dividing the privileges of policy enforcement between the higher domain and the lower domain. This result specifies the privilege range for each domain to enforce a policy.

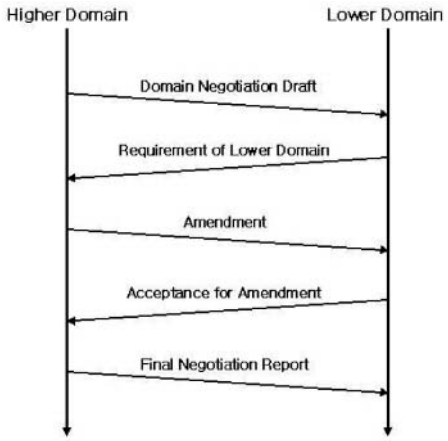

Fig 6. The Sequence of Domain Negotiation

For example, in a domain negotiation such as that performed above, the higher domain has the highest priority 3 for the entire lower domain and the lower domain has highest priority 5. However, for port range1025-2000, the lower domain can have the highest priority 2. That is, for a particular range, the lower domain has a relatively higher privilege than one of the higher domain.

4. Evaluation

Figure 7 represents the policy enforcement time according to the number of nodes and layers. As shown in Figure 7, the lower layer devices, the longer the response time for the policy is enforced. However, the rate of increase becomes blunter and blunter.

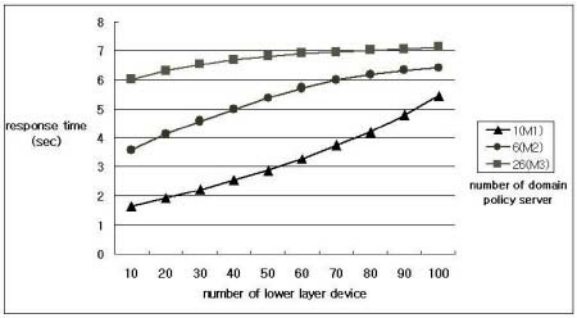

Fig 7. Simulation Result

There are three graphs which represent different environments. M1 shows the environment constructed by single domain policy server and a single layer. M2 shows the environment constructed by the 6 domain policy servers and 2 layers, and M3 shows the environment constructed by 26 domain policy servers and 3 layers. According to each environment, the enforcement time is a little different no matter how similar the time I s in a large number of nodes. Consequently, granting that a policy is enforced to a multiply layer network environment, the response time is very short. If that enforcement structure is complex, the entire response time is not affected.

5. Conclusion

We propose policy enforcement and a mediation mechanism for large-scale networks. The proposed mechanism allows an administrator to simultaneously enforce policies to the entire network. The large-scale network can, therefore, respond to harmful traffic that has rapidly propagated.

The high-level language is provided to effectively authorize a policy, enabling it to rapidly enforce a policy onto a large-scale network through an abstraction of information and a control form for lower layer devices. The policy collision in the multiple-layer structures of a large-scale network leads to confusion. Therefore, such collision should be detected and resolved. The proposed mediation mechanism selects the most important policies among the colliding polices, according to the administrator's intentions.

6. References

1. S. Hariri, Qu Guangzhi, T. Dharmagadda, M. Ramkishore: "Impact analysis of faults and attacks in Large-scale networks", Security Private Magazine, IEEE,Volume1, pp.49-54, Sept-Oct2003
2. Duan Haixin, Wu Jianping, "Security management for large computer networks", APCC-OECC99, Vol2, pp.1208-1213, 18-22Oct.1999
3. Sung Kang, "An efficient design of large-scale communication network with a decomposition technique", Circuits and Systems, IEEE Transactions on Vol27, pp.1169-1175, Dec1980
4. E.C. Lupu, M. Sloman, "Conflicts in policy-based distributed systems manage-ment", Software Engineering, IEEE Transactions on Vol25, pp.852-869, Nov-Dec.1999
5. D.C. Verma, S. Calo, K. Amiri, "Policy-based management of content distribution networks", Network, IEEE, pp.34-39, Mar-Apr2002
6. D.C. Verma, "Simplifying network administration using policy-based manage-ment", Network, IEEE, Vol16, pp.20-26, Mar-Apr.2002