1000-9825/2001/12(12)1739-09 ©2001 Journal of Software 软件学报 Vol.12, No.12

Policy-Based Access Control Framework for Large Networks*

DUAN Hai-xin, WU Jian-ping, LI Xing

(*Information Network Engineering Research Center*, *Tsinghua University*, *Beijing* 100084, *China*)

E-mail: dhx@bjnet.edu.cn; {jianping,xing}@cernet.edu.cn

http://www.tsinghua.edu.cn

Received April 28, 2000; accepted June 13, 2001

Abstract: Efforts of this paper focus on the issues about the management and throughput of firewalls (or screening routers) applied in transit networks. On the one hand, manual configuration of large amount of firewalls distributed in many access points cannot meet the requirements of security management in the open and dynamic environment. On the other hand, the ordinal lookup of filtering rules in firewall results in decrease of throughput. Aimed at a typical transit network and its security policy requirements, a policy-based access control framework (PACF) is proposed in this paper. This framework is based on three levels of abstract access control policy: organizational access control policy (OACP), global access control policy (GAÇP) and local access control policy (LACP). The GACP, which comes from the results of IDS and search engines according to OACP, is automatically and dynamically distributed to firewalls as LACPs. Each LACP is then enforced by an individual firewall. Some algorithms fcr distribution of GACP and enforcement of LACP are described. A hash-based algorithm is proposed for lookup of filtering rules in LACP. PACF largely reduces the management labor of the security administrator for large transit networks. Under the environment with policy requirements described in this paper, the new algorithm reduces the time complexity of lookup from $O(N)$ of traditional sequential algorithm to $O(1)$, which increases largely the throughput of firewalls.

Key words: computer networks; network security; access control; firewall; security policy; hash table

Firewalls are widely used and become a fact of life in the Internet/Intranet, although they were a very emotional topic in the Internet community a few years ago[1]. The term "firewall" has been seen limited use since the late 1980s to describe an access control device to block unwanted network traffic between two networks. There are three types of firewall devices: packet filters (working at IP level), circuit gateways (TCP level) and application gateways (Application level)[2]. A reference model for firewall technology can be found in Ref. [3].

Firewalls are placed not only between organizational networks and outside networks, but also inside internal networks to isolate administrative domains (ADs). An AD is defined as a collection of network resources under

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No. G1999035810 (国家重点基础研究发展规划973资助项目)

DUAN Hai-xin was born in 1972. He received his Ph.D. degree in the computer science from Tsinghua University in 2000. His research interests are network security and network management. **WU Jian-ping** was born in 1953. He is a professor and doctoral supervisor of the Department of Computer Science at Tsinghua University. His research interests are computer network architecture, protocol engineering, interconnecting and routing technology, communication network management, and etc. **LI Xing** was born in 1955. He is a professor and doctoral supervisor of the Department of Electronic Engineering at Tsinghua University. His research interests are interconnecting and routing technology, communication network management, network security, and etc.

control of a single administrative entity, with common security policy.

Many security technologies other than access control, such as intrusion detection, audit and encryption, are integrated into firewalls. In the concept of the active firewall proposed by [4], a firewall is a set of systems that comprise traditional firewall, vulnerability scanner, anti-virus tool, encryption facility, and even PKI server. However, we don't advocate such a concept created by vendors who want to convince users of that "network security=firewall!". As pointed out in Ref. [3], firewalls cannot do everything to protect networks. Additionally, firewalls can result in some problems such as performance decreasing, asymmetric routing and multicasting problems, and etc. [1].

Some other issues, such as authentication[5], policy specification for packet filtering[6,7], etc., arrest great research interests. However, few efforts are addressed to the issues about firewalls applied in large-scale networks, especially in transit ADs[8,9].

The usage of firewall (or screening router/gateway) in transit ADs defers from that in stub ADs in the following aspects. In stub ADs, firewalls are used to protect hosts or information resources. Because the scale and the number of access points are limited, one or two firewalls with several filter rules is enough to enforce the perimeter access control policy. However, in transit ADs, firewalls are mainly used to protect communication resource (such as bandwidth and buffers); and in our case, firewalls are used to stop aggressive activities, or to block sites with harmful information. For one thing, a transit AD often connects many stub ADs as well as other transit ADs around its perimeter; therefore, multiple firewalls must be placed at those access points. On the other hand, the scale of a transit AD is often much larger than that of a stub AD, and the address space is open. The consequence is that the number of filtering rules in each firewall of transit AD is much larger than that of stub ADs'. In our case, there are often hundreds or even thousands of rules in some backbone gateways.

Consequently, it is a cumbersome work for the security administrator to configure the large number of rules in multiple firewalls distributed around the perimeter. Unfortunately, in the open, dynamic network environment, as one attacking or harmful site falls, another new one arises elsewhere. This means that the job of configuration is not something "done once and for ever". Our experience from management of CERNET backbone shows that the job of manual configuration and modification of filtering rules in multiple firewalls will overwhelm even an experienced administrator.

Another consequence is that the large number of filtering rules in each individual firewall largely reduces the throughput of the network. As shown in Ref. [10], TCP forwarding rates from many commercial high-end firewalls (such as Check Point, Lucent) are less than 10M bytes/s in a test bed in which TCP forwarding rate reaches 15.6M bytes/s without firewalls. With the revolutionary increase and requirement for high bandwidth, the decrease of throughput brought by firewalls cannot be ignored.

In this paper, we focus our efforts on the automatic management of a large number of filtering rules in multiple firewalls in a large, open and dynamic internetworking environment. In Section 1, such a typical transit network with specific security requirements is described. To meet these requirements, three levels of abstract access control policy are presented in Section 2: organizational access control policy (OACP), global access control policy (GACP) and local access control policy (LACP). A policy-based access control framework (PACF) is proposed, in which GACP comes from OACP combined with the results of Intrusion Detection Systems (IDSes) and search engines. The GACP is distributed automatically to multiple firewalls as LACPs, which are then enforced in each individual firewall. Some key algorithms, including distribution of GACP and enforcement of LACP, are described in Section 3 and Section 4. A new algorithm for lookup of filtering rules is proposed, which decreases the time complexity from $O(N)$ to $O(1)$ in our specific environment, therefore increases largely the throughput of filtering devices. The issues to implement PACF are discussed in Section 5. Finally, Section 7 summarizes this paper and presents some future directions.

1 Transit Network Environment and Security Policy Requirements

A typical transit network environment is illustrated in Fig. 1. The backbone of CERNET (Chinese Education and Research Network) can serve as the best example. In Fig. 1, the transit network (202.112.0.0/16) provides transit service for its own stub networks (e. g. campus networks of some Universities); at the same time, it connects other transit ADs (such as DFN, JANET, CHINANET, etc.). All the routers or gateways in the transit network (or the backbone) are administrated by the same administrative authority.

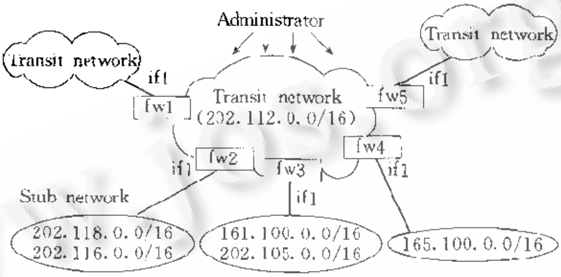

Fig. 1 The transit network environment

There are three aspects in high security policy requirements:

1) Some information (such as obscene picture) is harmful for the society; therefore some hosts with such information should be blocked;

2) Some activities (such as vulnerability scanning, proxy hunting, etc.) are illegal. Such aggressive activities across the transit AD should be terminated in time.

3) Because the harmful or aggressive sites fall and arise dynamically, the policy rules should be adjusted correspondingly in time.

To enforce such a high-level security policy, firstly, multiple firewalls or screening routers are necessary to be placed between the backbone and other ADs. Secondly, some automatic mechanisms to detect the blameworthy sites are necessary; in our considerations, IDSes and search engines serve the detection task. The detection mechanisms should work cooperatively with the access control devices; that is, global filtering rules should be generated dynamically from the results of IDSes and search engines.

Now, the problem comes: where and how to enforce the filtering rules to block these blameful sites? The following sections address to these problems with a policy-based access control framework.

2 The Policy-Based Access Control Framework

To implement the above security policy, a Policy-Based Access Control Framework (PACF) is proposed, as shown in Fig. 2. The administrator establishes the initial measures of access control, including reasonable placement of firewalls, IDSes and search engines according to organizational access control policy. Then global access control policies (global filtering rules) are dynamically generated from the results of IDSes and search engines by security manager. And then, security manager partitions the global rules into local access control policy and distributes them to each individual firewall. In PACF, administrators needn't know where and how to enforce the rules and, of course, needn't configure each filtering rule one by one, device by device.

2.1 Three levels of access control policy

Access control policy is the most important part of the overall security policy. In this paper, we distinguish

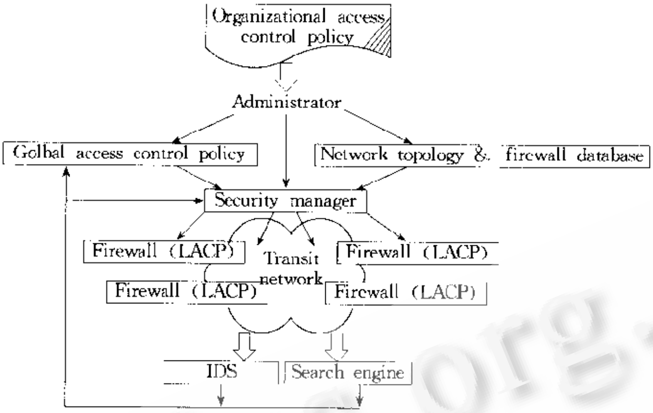

Fig. 2 Policy-Based access control framework

the term "access control policy" in three levels:

1) Organizational access control policy (OACP) is composed of a set of the high level policy statements, which regulate what resource to be protected, and which activities is to be forbidden in the organizational scope. OACP is often narrated in natural language, such as that in Section 1.

2) Global Access Control Policy (GACP), which can be performed automatically by computer systems (including routers/switches, firewalls, etc.), is the reification of the part of OACP in the scope of all AD. In the transit network discussed in Section 1, GACP is composed of a set of global filtering rules (or access control lists) which are enforced by a set of firewalls.

Definition (policy rule). A policy rule is a 3-tuple: ⟨source, destination, action⟩, where

source is the set of addresses matching where packets originate from, which is expressed in: *srcAddress/srcMask*;

destination is the set of addresses matching where packets are destined to, which is expressed in: *dstAddress/dstMask*;

action determinates what to do when an IP packet matches. Generally, $action = \{permit, deny\}$

For example, the filtering rule, ⟨20.1.1.0/24,10.1.1.1/32,permit⟩ means that packets from network 20.1.1.0/24 to host 10.1.1.1/32 are permitted to pass through. 'ANY' is the set of addresses which matches any site.

Unlike ordered access control lists in each individual firewall, the global set of filtering rules is unnecessary to be ordinal. Where and how to enforce the rules is the responsibility of the security manager.

In our specific policy, GACP can be partitioned into two classes: *dstRS* and *srcRS*. The *dstRS* contains sites (or networks) as the destinations that are prohibited from other sites. Rules in *dstRS* are used to satisfy the requirement 1) stated in Section 1. In the other, *srcRS*, some sites (or networks) are forbidden to access other sites to prevent the aggressive activities initiated from these sites. Rule in *srcRS* are used to satisfy the requirement 2) of Section 1. In a word,

$$GACP = dstRS \cup srcRS,\quad dstRS \cap srcRS = \varnothing$$

For $\forall\, r \in dstRS, r.source = ANY$;

For $\forall\, r \in srcRS, r.destination = ANY$;

3) Local Access Control Policy (LACP) is a subset of GACP, which is enforced by each individual interface of a device (such as routers, hosts, firewalls etc.). LACP of an interface consists of two separate sets of rules: inbound rule set (*inboundRS*) and outbound rule (*outboundRS*) set. The *inboundRS* is responsible for filtering incoming packets to the interface; the *outboundRS* is for filtering the outgoing packets leaving the interface.

2.2 Dynamic policy adjustment and enforcement

One of the features of PACF is that the policy is dynamically created, enforced and adjusted. After initialized, the global set of filtering rules is dynamically adjusted according to the results of IDSes and search engines. On the other hand, IDSes can serve as the verification of the access policy, and then feed back to the security manager, to reconfigure the firewall or network topology.

The discussion about implementation for IDS and search engine is outside the scope of this paper. We assume that the result about sites holding harmful information or initiating intrusion activities can be used by the security manager.

3 Distribution of the GACP

Given the set of global filtering rules and firewalls, the Algorithm 1, GACP-Distribution, shows how to distribute the rules to each individual interface of firewalls.

Algorithm 1.

```
GACP-Distribution (GACP,FWDB){
  $(srcRS, dstRS) = partition(GACP)$;
  FOR $fw \in FWDB \backslash OpenFW$ DO{
      FOR if$\in fw.interfaces$ DO{
        FOR $net \in if.networks$ DO{
          FOR $r \in srcSR$ DO
            IF $r.source \subseteq net$ THEN{
              $if.inboundRS = if.inboundRS \cup \{r\}$;
              $srcRS = srcRS \backslash \{r\}$
            }
          FOR $r \in dstRS$ DO
            IF $r.destination \subseteq net$ THEN{
              $if.outboundRS = if.outboundRS \cup \{r\}$;
              $dstRS = dstRS \backslash \{r\}$
            }
          } /* end of FOR net...
        } /* end of FOR if
      } /* end of FOR $fw$...
  FOR $fw \in OPENFW$ DO{
      $fw.default\_if.inboundRS = srcRS$;
      $fw.default\_if.outboundRS = dstRS$;
  }
}
```

The information model of the firewall database is shown in Fig. 3. Each firewall has $n(n \geqslant 2)$ interfaces, and every interface is associated with m $(m \geqslant 1)$ networks., the interface Open Interfaces is the set of interfaces which connect other transit networks, such as if1 of fw1 and fw5 In Fig. 1. The address spaces of the associated networks are open for Open Interfaces.

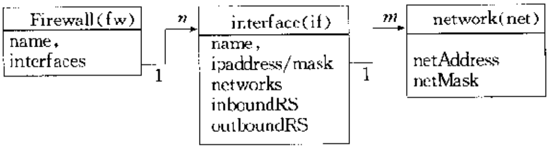

Fig. 3 Information model in firewall database

Firstly, the set of global rules is partitioned into two classes: *srcRS* and *dstRS*. Secondly, for each interface of firewalls other than Open Interfaces, the inbound and outbound rules set are generated. The rules in *srcRS* are distributed into *inboundRS* of interfaces; the rules in *dstRS* are distributed into *outboundRS* of interfaces.

For each rule in either *srcRS* or *dstRS*, the only and enough location is the interface of some firewall which is the nearest to the site (or network) in the rule. In Fig. 1, for example, the best place to enforce the rule ⟨*ANY*, 202.118.2.1/32, *deny*⟩ is in the outbound rule set of interface if1 of firewall fw2. To block access from 161.100.1.1, the most laborsaving way is to distribute the rule ⟨161.100.1.1/32, *ANY*, *deny*⟩ to the inbound set of interface if1 of fw3. Other rules whose source or destination addresses are not in any associated networks of any firewalls should be distributed to Open Interfaces connecting other transit ADs.

4 Enforcement of the LACP

As shown in Algorithm 1, every interface has been allocated by the security manager to its own LACP, i.e., *inboundRS* and *outboundRS*. Both *inboundRS* and *outboundRS* are partial ordered set. To enforce its LACP, each individual firewall must transform the sets into ordered access lists. Therefore, the relationship among the filtering rules in *inboundRS* or *outboundRS* must be analyzed. To sort these rules automatically, the correlative relationship between two rules is defined as following:

Definition (**correlative**). For R_1, $R_2 \in inboundRS$, if

$$R_1.source \cap R_2.source \neq \varnothing,$$

then R_1 and R_2 are correlative, with the notation: $R_1 \sim R_2$. Similarly, for $R_1, R_2 \in outboundRS$,

$$R_1.destination \cap R_2.destination \neq \varnothing \Rightarrow R_1 \sim R_2.$$

Otherwise, we say that they are independent. The order of correlated rules in access list is essential, while the order of independent rules is inessential.

4.1 Ordering routine for access list

Take the *inboundRS* as an example. According to the control granularity of each rule, the set can be divided into two parts: *siteArray* and *netList*. In *siteArray*, the srcMask field of each rule equals 32; that is, this filtering rule applies to a single site instead of a network. Because every two rules in *siteArray* are independent, the order of rules in *siteArray* is inessential. So, a firewall can perform non-sequential lookup in *siteArray*. In *netList*, the srcMask field of each rule is less than 32; that is, this rule applies to a network instead of a single site. Unlike *siteArray*, because rules in *netList* may be correlative, rules in the *netList* must be sorted.

The algorithm for generation of *siteArray* and *netList* is shown in Algorithm 2. The sort algorithm for netList is according to the following rules:

For $R_1, R_2 \in inboundRS$:

1) $R_1 \sim R_2 \wedge R_1.source \subseteq R_2.source \wedge R_1.action \neq R_2.action \Rightarrow$ ORDER(R_1) < ORDER(R_2), otherwise R_1 will never work; therefore R_1 should be inserted before R_2 in *netList*;

2) $R_1 \sim R_2 \wedge R_1.source \subseteq R_2.source \wedge R_1.action = R_2.action \Rightarrow R_1$ is excrescent and should be ignored;

3) if $R_1 \in siteArray, R_2 \in netList, R_1 \sim R_2 \Rightarrow$ ORDER(R_1) < ORDER(R_2). That is, lookup in *siteArray* should be previous to *netList*.

In a large transit network environment with security policy discussed above, the number of rules in *siteArray* is much larger than that of *netListc*, that is,

$$|siteArray| >> |netList|$$

Therefore, the improvement of lookup algorithm in *siteArray* approximates the increase of that in the whole access list. The rest of this section aims at increasing the lookup efficiency in *siteArray*.

Algorithm 2.

```
$InboundInitialize(inboundRS: RULE\_SET)\{$
  $netList = (); siteArray = (); i = j = 0;$
    FOR $r \in inboundRS$ DO{
        IF r.srcMask==32 THEN SiteArray[i++]=r;
        ELSE{
         WHILE (j<\netList\) DO{
          IF r.source⊆netList[j].source)
          THEN {
                 IF r.action≠netList[j].action THEN
                   INSERT (r.netList,j); BREAK;
                 ELSE BREAK; /* ignore r */
          } /* end of IF */
         j++;
        } /* end of WHILE */
        INSERT(r,netList,j);
    } /* end of ELSE */
  RETURN siteArray,netList
}
```

4.2 Hash based lookup in *siteArray*

Based on the previous analysis, a hash-based algorithm is proposed for lookup of rules in *siteArray*. Also taking the inboundRS as the example, the algorithms for the construction of the hash table and the routine for lookup of rules are shown in Algorithm 3.

On receiving an IP packets from one of its interface, the firewall looks up the *inboundRS* (first inboundHash and then *netList*) associated with this interface. If the action of the matched rule is "deny", then the packet is discarded. Otherwise, the firewall looks up its routing table and forwards the packet to the corresponding interface. Then, similar lookup routine in *outboundRS* associated with this interface is then performed.

Because of |*siteArray*|>>|*netList*|, the time consumed by lookup routine approximates the time performed in *siteArray*. Therefore, the time complexity of the routine is nearly $O(1)$.

Algorithm 3.

```
Lookup (ipPacket:Packet,inboundHash:Rule_SET,netList:Rule_SET){
    idx=Hkey(ipPacket.source);
    IF (inboundHash[idx].srcAddress=ipPacket.srcAddress)
      THEN return inboundHash[idx].action;
    WHILE (j<|netList|) DO
      IF (ipPacket.srcAddress∈netList[j].source) THEN
        return netList[idx].action;
}
siteArray_Initialization (siteArray:RULE_SET){
  FOR (r∈siteArray) DO{
    idx=Hkey(r.srcAddress);
    inboundHash[idx]=r;
  }
}
/***** *Hash Key calculation ****/
INTEGER Hkey(ipAddr:IP_ADDRESS){
      high=ipAddr[0..7]XOR ipAddr[16..23];
      low=ipAddr[8..15]XOR ipAddr[24..31];
      index=(high<<8 XOR low) MOD MAXTABLE;
      return index;
}
```

5 Implementation Issues

A prototype for PACF is being implemented, in which the security manager is part of the network management system developed in our project. The set of global filtering rules and configuration of firewalls are stored in a LDAP directory. CLI (Command Line Interface) is used as the communication protocol between the security manager and Cisco screening routers. Currently, security manager generates access list commands for each individual router. Because the firewalls/routers available now can not support the enforcement and lookup algorithms discussed above, a dedicated gateway based on PC/Linux is being developed to implement the function discussed in this paper.

6 Summary and Future Work

Aimed at the typical transit network and security requirements discussed in Section 1, three levels of access control policy are presented. Then, a policy-based access control framework (PACF) is proposed, addressing to automatic and dynamic enforcement of access control policy. To implement the framework, key algorithms for distributing the global policy and enforcing the local policy are described.

PACF is essentially automation of access control management. With the capability of generation and distribution of GACP and the enforcement of LACP, the PACF will largely liberate the administrator from the heavy, trivial configuration task of large amount of filtering rules.

Policy enforcement is described only for firewalls in this paper. In fact, policy-based IDSes and search engines can serve as the verification and feedback of the enforcement of the access control policy by firewalls. With the progress of another project about IDS and search engine, policy verification by IDSes and search engine will be integrated into the policy-based access control framework.

References:

[1] Braden, R., Clark, D. Report of IAB Workshop on Security in the Internet Architecture. RFC1636, 1994. URL:http://www.ietf.org/rfc/rfc1636.txt

[2] Bellovin, S.M., Cheswick, W.R. Network firewalls. IEEE Communications Magazine, 1994,32(9):50~57.

[3] Schuba, C.L. A reference model for firewall technology. In: Lyles, J.B. ed. Proceedings of the 13th Annual Computer Security Applications Conference. New York: IEEE Computer Society, 1997. 133~145.

[4] Network associations Corp. The Active Firewall: The End of the Passive Firewall Era, 1999. URL:http://www.nai.com/nai_labs/asp set/network_security.asp.

[5] Leech, M., Ganis, M. SOCKS Protocol Version 5. 1996. URL:ftp://ftp.isi.edu/in-notes/rfc1928.txt.

[6] Guttman, J.D. Filtering postures: local enforcement for global policies. In: Steve Kent, ed. Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy. New York: IEEE Computer Society, 1997. 120~129.

[7] Oppliger, R. Internet security: firewalls and beyond. Communications of the ACM, 1997,40(5):92~102.

[8] Estrin, D. Tsudik, G. Secure control of transit internetwork traffic. Computer Networks and ISDN Systems, 1991,22(5):363~382.

[9] Hares, S., Katz, D. Administrative domains and routing domains: a model for routing in the internet. RFC1136, 1989. URL:ftp://ftp.isi.edu/in notes/rfc1136.txt.

[10] Newman, D. Super firewalls. Data Communications, 1999,28(5):51~61.

面向大规模网络的基于政策的访问控制框架

段海新，吴建平，李 星

(清华大学 信息网络工程研究中心,北京 100084)

摘要: 研究防火墙(或过滤路由器)应用于传输网络中的管理问题与吞吐量问题.一方面,手工配置分布在各个接入点的大量防火墙,无法满足开放的、动态的网络环境的安全管理需求;另一方面,大量过滤规则的顺序查找导致了防火墙吞吐量下降.针对一个典型的传输网络和它的安全政策需求,提出了一种基于政策的访问控制框架(PACF),该框架基于3个层次的访问控制政策的抽象:组织访问控制政策(OACP)、全局访问控制政策(GACP)和本地访问控制政策(LACP).根据OACP,GACP从入侵监测系统和搜索引擎产生,作为LACP自动地、动态地分配到各防火墙中,由防火墙实施LACP.描述了GACP的分配算法和LACP的实施算法,提出了一种基于散列表的过滤规则查找算法.PACF能够大量减轻管理员的安全管理工作,在描述的安全政策需求下,基于散列表的规则查找算法能够将传统顺序查找算法的时间复杂度从$O(N)$降低到$O(1)$,从而提高了防火墙的吞吐量.

关键词: 计算机网络;网络安全;访问控制;防火墙;安全政策;散列表

中图法分类号: TP393 **文献标识码**: A