An Access Control System Based on Multistage Firewalls

Xingcheng Zhang
Cognitive Science Department, Xiamen University
Fujian Key Laboratory of Brain-like Intelligent Systems
Xiamen, China
zhxingcheng@gmail.com

Zhongpan Qiu
Cognitive Science Department, Xiamen University
Fujian Key Laboratory of Brain-like Intelligent Systems
Xiamen, China
zpqiu@xmu.edu.cn

Zhijun Song
Cognitive Science Department, Xiamen University
Fujian Key Laboratory of Brain-like Intelligent Systems
Xiamen, China
jason@xmu.edu.cn

***Abstract*—In accordance with the company's request, we have designed an Access Control System based on multistage firewalls, whose work flow consists of the centralized management, application, check and automatically alteration of firewalls, to effectively control the access to the server from the clients. In a most economical way, it saves the human resources and promotes the operation efficiency of the whole company.**

Keywords-access control; software engineering; firewalls; database

I. INTRODUCTION

The Access Control System is a significant branch of software engineering as well as being an effective tool to control the clients' access to the server within a company [1, 2]. The normal operation of a company and the working efficiency of the personnel will be influenced directly by the quality of the Access Control System. However, different companies use it in a quite different way. Generally speaking, there are four limitations of the Access Control tools in traditional IT industry.

First of all, with the increase in the quantity of clients and servers, access control chaos will easily occur if the devices have their clients and servers under no centralized management [3]. Second, along with the rise in the number of the requests for the access to the server and of the clients, the workload of the network administrator is also increasing even to the extent of being unable to control the Internet within a company [4]. Third, the traditional method of obtaining operating records needs a specific accessed server [5], which takes lots of manpower, material resources and time as well as causing data chaos. Last but not least, the traditional method to process the access control among different cities is too complicated to cause inefficiency.

II. OUR SOLUTIONS OF THE ACCESS CONTROL

Aiming at solving currently-existing problems, we have designed an Access Control System based on multistage firewalls, which is a kind of intermediate control device between clients and the sever [6]. When applying for the access to the server, clients need to fill in an application form, which will be checked after handing it in. If the application form passes, the system will generate the batch processing command of firewalls by extracting the application form's information. When it is the set time to start the access, the system will automatically execute the command-line configuration in order to open the specific server port; however, when it is the set time to terminate the access, command-line configuration will be executed again to close the server port. Nevertheless, if it fails to pass, the system will suspend the application immediately. Therefore, the safe access between the clients and server within a company including all the branch offices in different cities is implemented.

III. SYSTEM DESIGN

A. Business Logic

After analysis and simulation, we classify all the system users into three roles in the end. They are applicants, checkers and network administrators. The network administrators carry out the unified maintenance to each role of the system users, who have different using permissions.

For the applicants, they log in to fill in the application form which needs detailed information including source IP address, destination IP address, opening protocol, port and the access time etc.

For the checkers, they log in to check the application form. If the application form passes, the system will automatically generate relevant configuration command of firewalls and send processing batch command at the set time to the HyperTerminal of Windows or to the minicom of Linux to configure the settings of the firewalls. Only after the firewall configuration can the applicants have the direct access to the applied destination IP. The network administrators, meanwhile, can monitor the potential dangerous operation of the visitors as well as having the permission to maintain the information of the server, firewalls and the personnel. We also take advantage of the hardware firewalls. The hardware firewalls will have relevant version and log records about every configuration of

Published by Atlantis Press, Paris, France.

firewall settings, which will be automatically saved by our Access Control System. The figure 1 below shows the business logic of this system.

B. *Function Design*

As for our system, we will illustrate its function design from three perspectives.

First of all, permission application and query function. This function is the core function of the whole system, integrating the information of the filt-in application form, query for detailed information and state check etc. It is designed to satisfy the application, query and other operations of the applicants as well as meeting the network administrators' needs of unified management of the applicants' application.

Second, diversity management functions of the network administrators. Information of personnel, servers, firewalls and ports can be diverisifiedly managed by the network administrators, aiming at promptly update all kinds of important information of the system and ensuring that there is zero mistake in all applicants' information.

Third, function of multistage firewalls' automatic configuration. We have implemented the automatic configuration of the firewalls when there is cascading firewalls within a city or among several cities. Realized by the technology of processing batch, this function has an excellent safety performance and strong stability.

C. *Database Design*

The database design employs the PostgreSQL, an object - relation database management system [7]. The database is designed in both the applicants' and network administrators' respects.

With the application form as its core, it altogether consists of eleven forms. Associating the application form with other ten forms, they altogether generate a one-to-many date association [8] (as shown in the figure 2 about the database). Each form adopts our newly invented UUID [9]. Among these elven database forms, our UUID does not appear on the system interface or have any data redundancy besides functioning as a unique identifier. Furthermore, any field information can be configured at will without any influence on the database.

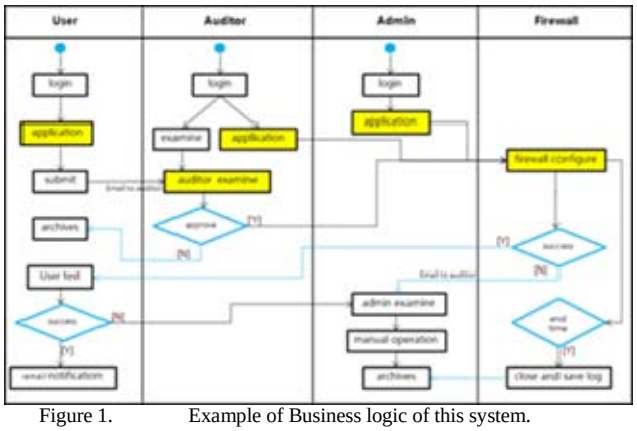

Figure 1. Example of Business logic of this system.

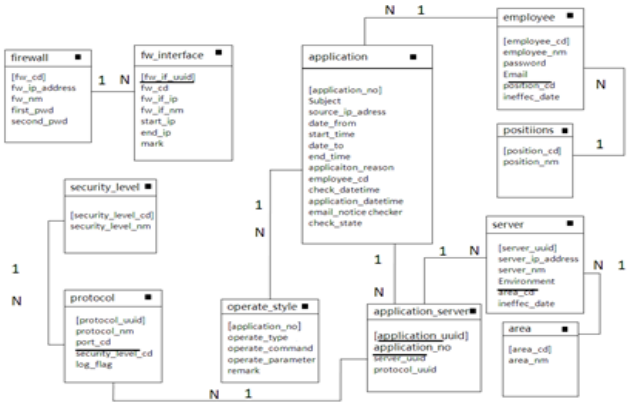

Figure 2. Example of Database Design.

IV. AUTOMATIC CONFIGURATION TECHNOLOGY OF MULTISTAGE FIREWALLS

A. *Fundamentals of Multistage Firewalls' Automatic Configuration*

The increasing number of clients and servers accompanies the increasing needs for more firewalls, and there may occur cascaded multistage firewalls. It is for this reason that we focus on the fundamentals of multistage firewalls' automatic configuration. According to the operation principles of firewalls [10, 11, 12], we have already known that every firewall will set an exit of its own to be the entrance from the outside. Hence, if there is a cascaded multistage firewalls [11], we will label the exit as 1 but no exit as 0. By labeling the positions, we can find the entrance of the next level firewall one by one and send the command one by one to implement the automatic configuration of the multistage firewalls.

From the multistage firewalls' situation, as shown in the figure 3, we can see the execution flow of the command. *Outside*, *outer* and *external* are the exits labeled as 1.

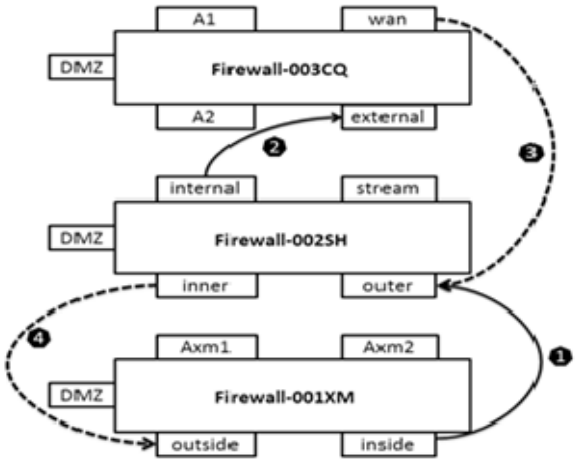

Figure 3. Example of the Opening Process of the Cascaed firewalls

B. *Work Flow and Algorithm*

First of all, the source IP address s_ip and the destination IP address d_ip of the application form should be extracted from the database to match the interface table of protocol. The compare () function implements the match of the IP scope, so that the source IP address and the destination IP address of the dependent firewalls, firewall interface and its area can be found. The required commands of the configured firewalls are screened according to the previous results. If two IPs belong to one firewall, two respective commands need generating; if not, four commands need generating.

Then the important information of the application form, such as the field of the opening protocol and port, should be extracted. By using the combination () function together with the formerly generated IP addresses could the final required command of the firewalls be generated [13].

Specified firewall formalization is expressed as: <order> <protocol> <s_ip> <s_port> <d_ip> <d_port> <action>

In the end, by using the timerTest () function to extract the field of the access time can the waiting state be activated. At the beginning of the access time, the system will automatically send the firewall command to the hyper terminal of the Windows in order to open corresponding protocol and port of the server to the clients. At the end of the access time, system will send the firewall command to the HyperTerminal of Windows again to end the access. Then the system will record and save all the commands sent during this access.

The algorithm for the automatic configuration of multistage firewalls [14], the generateCommand () function, is expressed as: generateCommand（s_ip, d_ip, ipScope）

Input:
s_ip: the source IP address of the applicant
d_ip: the destination IP address of the applicant
ipScope: the relations among the IP scope, port and firewalls maintained in the firewall interface table.

for i ∈{1...max} do
  compare(s_ip , ipScope)
  print firewall(s_ip) ,flag(s_ip)
compare(d_ip , ipScope)
  print firewall(d_ip),flag(d_ip)
combination(protocol, s_ip , d_ip, d_port)
bati ← combinationi
timerTest(myTime ,bati)
end of for
Output: h(x) ← timerTest

C. *Algorithm for the Rule Conflicts Detection*

In order to improve the accuracy and safety performance of the multistage firewalls, we try to detect the rule conflicts while the firewalls' commands are configurated at the same time [15, 16]. We use the improved algorithm for the rule conflicts detection based on the decision tree [17]. This algorithm detects the errors which frequently occur in the configuration of firewalls from five perspectives, such as the absence of default rule, existence of independence rule, mutual shielding among rules, cross-conflict of rules and the redundancy of rules [18].

In order to immediately determine [18, 19] whether there is a rule configuration error, we should first compare its filtration field and put the results into tempArray. TempArray has array items which correspondingly stores five results, namely, protocol field, source IP address field, source port field, destination IP address field and destination port field. Let's first compute the product of the five items by using the formula $\prod_{i=1}^{5} temp[i]$ If the product equals zero, there exists conflict; if not, the compute will continue next formula $\prod_{i=1}^{5} temp[i] - 1$. If the result does not equal zero, we continue the compute. In this way can we determine whether there is any rule conflict between T1 and T2.

Algorithm of the rule conflicts detection, the conflictDetection () function, is express as:

conflictDetection(rule_table t)
for k = 1 to t.count-1
  then for j = K+1 to t.count
   compare(t[k],t[j],tempArray);
   if($\prod_{i=1}^{5} temp[i] = 0$)
    if($\prod_{i=1}^{5} temp[i] - 1 = 0$)
     then if (t[k].action = t[j].action)
     then print t[j] is t[k]'s redundancy;
     else
      then print t[j] is shielded by t[k];
    else if(=0 and t[k].action! = t[j].action)
     print there is cross conflicts between t[j] and t[k];
  end of for
end of for

V. CONCLUSION

A. *Analysis of Experiment Results*

Experimental data are tested several times according to different network segment, areas, protocols and ports, as shown in figure 4.

Case NO.	Area	Experiment No.	Success rate
case1	XM——XM	30	96.7%
case2	XM——SH	20	95.0%
case3	XM——CQ	20	90.0%
case4	SH——SH	30	96.7%
case5	SH——CQ	20	90.0%
case6	SH——XM	20	95.0%
case7	CQ——CQ	30	96.7%
case8	CQ——SH	20	85.0%
case9	CQ——XM	20	90.0%

Figure 4. Experiment Results of multistage firewalls' automatic configuration. We conducted the experiments in three companies from three different cities, named as XM, SH and CQ.

From the analysis of the experimental results, we can discover that the highest success (96.7% on average) rata happens within the same area. Configuration failure happens

only once in the 30 experiments, which proves its relative stability. Under other circumstances, success rate are all above 90%. Here are several reasons causing the failure and instability. First, the matching arithmetic of firewalls among different cities may lack accuracy, which causes the opening of the wrong port; second, the scope of the algorithm for the rule conflicts detection is not comprehensive enough and can cause the omission; third, in rare cases, the commands of firewalls fail to be sent out, which causes partial implementation of the configuration command.

B. Contrast and prospects

This system implements the access control within a company located in two or even more cities. However, it still has many disadvantages and needs improving. First, screen recording function of the server should be added to enhance the access safety of the server. Second, this system has no alarm operation. We assume that it can be implemented by the means of matching the operation commands. This means needs to fill the relevant operations in the application form. The real operations will be recorded at the end of the access to match those filt-in application forms. If there is any unauthorized operation, the alarm will take actions. An Access Control System based on multistage firewalls like this will constantly improve itself and thus will play an important role in the field of software engineering.

ACKNOWLEDGMENT

At the end of my paper, I would like to express my sincere thanks to my supervisor, Professor Qiu Zhongpan, a respectable and resourceful scholar, Dr. Song Zhijun, a senior student, who have both provided me with valuable guidance in every stage of the writing of this paper and all the relevant companies, which offer me precious platform to do experiments. Professor Qiu and Dr. Song's keen and helpful guidance enlightens me not only in this paper, but also in my future study. Moreover, without the cooperation and support of relevant companies I will not have the chance to design and test the system.

REFERENCES

[1] Barkley J F. "Comparing simple role based access control models and access control lists"[EB/OL]. 1997.

[2] Ravi S.Sandhu. "Lattice-based access control models". IEEE Computer, 26(11):9-19,November 1993.

[3] David Ferraiolo and Richard Kuhn. "Role-based access controls". In 15th NIST-NCSC National Computer Security Conference, pages 554-563, Baltimore, MD, October 13-16 1992.\

[4] S. H. von Solms and Isak van der Merwe. "The management of computer security profiles using a role-oriented approach". Computer & Security, 13(8):673-680, 1994

[5] Ravi S. Sandhu. "The Typed Access Matrix Model". In Proceedings IEEE Computer Society Symposium on Research in Security and Privacy, pages 122-136, Oakland, CA, May 1992.

[6] Barkely J F, Cincotta A V. "Role based access control for the world wide web"[EB/OL]. 1998.

[7] Miller, G. A., Ed. "WordNet: an on-line lexical database". International Journal of Lexicography 3, 4 (Winter 1990), 235--312.

[8] Wheeler DL, Chappey C, Lash AE, Leipe DD, Madden TL, Schuler GD, Tatusova TA, Rapp BA: "Database resources of the national center for biotechnology information". Nucleic Acids Res 2000, 28:10-14.

[9] PostgreSQL: A Comprehensive Guide to Building, Programming, and Administering PostgreSQL Databases.

[10] Al-Shaer E, Hamed H. "Firewall Policy Advisor for Anomaly Detection and Rule Editing"[C]//Proceedings of IEEE/IFIP IM'03. Colorado Springs, USA: [s. n.], 2003.

[11] Al-Shaer E, Hamed H. "Design and implementation of firewall policy advisor tools"[R]. Technical Report CTI-techrep0801, School of Computer Science Telecommunications and Information System, DePaul University, August 2002.

[12] Wool A. A Quantitative "Study of Firewall Configuration Errors"[J]. IEEE Computer, 2004, 37(6): 62-67.

[13] GOUDA M, LIU X. "Firewall Design: Consistency, Completeness, and Compactness"[A]. Proceeding of the 24th IEEE International Conference on Distributed Computing Systems(ICDCS 04) [C], March 2004.

[14] Cisco System, Inc. "CiscoWorks2000 access control list manager 1.2 overview". November 2000.

[15] Hari B, Suri S, Parulkar G. "Detecting and Resolving Packet Filter Conflicts"[C]//Proceedings of INFOCOM'00. Tel-Aviv, Israel: IEEE Press, 2000.

[16] Taylor D E, Turner J S. ClassBench: "A packet classification benchmark"[R]. Saint Louis, USA: Washington University in Saint Louis, Tech. Rep.: WUCSE-2004-28, 2004.

[17] Gupta P, McKeown N. "Algorithms for Packet Classification":[J]. IEEE Network, 2001, 15(2): 24-32.

[18] BABOESCU F, VARGHESE G. "Fast and Scalable Conflict Detection for Packet Classifiers"[A]. Proceedings of the 10th IEEE International Conference on Network Protocols[C], 2002.

[19] HAN J, KAMBER M. "Data Mining: concepts and techniques"[M]. Morgan Kaufmann, 2000.