IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 4 (April. 2013), ||V2 || PP 33-36

Diverse Firewall Policies with Change-Impact Analysis

Swapnil Hasabe[1,] Gaurav Naik[2,] Shrikant Nile[3,] Mukesh Rochlani[4]

Abstract: Nowadays, we are suffering from unintended security holes by unauthorized actions in enterprises as well as malicious attacks are very dangerous to the business services. For ensuring the security of private networks in most businesses and enterprises, firewalls are mostly deployed in security mechanism. The quality of policy configured in firewall decides how much it would be effective for securing the private networks. As we know, designing firewall policies are often error-prone due to the complexity of firewall configurations as well as the lack of knowledge of administrator. So, we have major task to discover the functional discrepancies between firewall policies and to resolve them to design diverse firewall policy which would be in most corrected form. We followed three phases in our project those are construction phase, comparison phase and resolution phase. Firewall policies often changes as networks evolve and new malicious attacks arrive. The methods for discovering functional discrepancies between two firewall policies are applied to perform firewall policy Change-Impact analysis as well.

Keywords: *Change-impact analysis, Design diversity, Firewall policy, Firewall Decision Diagram (FDD), Functional discrepancy, malicious attack, Network security, Policy configuration.*

I. INTRODUCTION

In recent years, there is a significant increase in the usage of computers and their capabilities to communicate with each other. It increased the need for more security and firewalls have proved themselves an important factor of the security architecture. As the policy is made up of rules, the quality of these firewall policies depends on knowledge of administrator. Unfortunately, there is little help for their administrators to understand the actual meaning of the firewall rules. A firewall policy consists of a sequence of rules where each rule is in form of <predicate> → <decision>. The predicate part contains packet fields such as source IP address, destination IP address, source port number, destination port number, and protocol type. The decision of rule can be accept, or discard, or logging option. The rules in a firewall policy can conflict to each other. To resolve such conflicts, priority of rule is taken into consideration.

II. OVERVIEW

To develop a Diverse Firewall policy, we need three phases: a design phase, a comparison phase, and a resolution phase [1]. Here, two firewall policies in terms of sequence of rules are taken as the input, in design phase, two firewall policies are converted into firewall decision diagram(FDD), in the comparison phase, the resulting firewall policies are compared with each other to detect all functional discrepancies between two given firewall policies. All functional discrepancies are resolved in resolution phase and a firewall that is agreed upon by both firewall policies is generated.

III. RELATED WORK

Our idea of design diversity is inspired by N-version programming [3]. The basic concept of N-version programming is to give the same requirement specification to N teams to independently design and implement N programs using different algorithms, languages, or methods. Then the resulting N programs are executed in parallel. A decision selection mechanism is applied to examine the N results for each input from the N programs and selects a correct or "best" result. It means we achieved diversity here. The key element of N-version programming is design diversity [1]. There are number of algorithms, methods which help the system administrator to manage or to configure firewall policies. We are using concept of Firewall Decision Diagram (FDD) to construct firewall policy. We are using algorithms: a construction algorithm [7], a comparison algorithm to detect functional discrepancies. This concept helps to understand how firewall policies are compared to each other to detect all functional discrepancies. Designing firewall policies suffers from three problems: the consistency problem, the completeness problem, and the compactness problem [6]. There are two approaches to reduce firewall policy configuration errors. The first approach is to cease errors from happening when configuring firewalls. The second approach is to detect errors after firewalls have been designed. We concentrate on second approach. In this approach, administrator manually examines every pair of conflicting rules to see whether the two rules need to be edited or a new rule needs to be added. These firewall policies are compared to each other in form of FDDs. In our paper, we take efforts to analyze change-impact of firewall

policies because as requirements of private networks changes, configuration of firewall policy needs to be changed [2].

IV. PROPOSED SYSTEM

Following are the major components of our system:

- Knowledgeable administrator
- Firewall Decision Diagram concept
- Firewall server Authentication for administrator
- Firewall policies

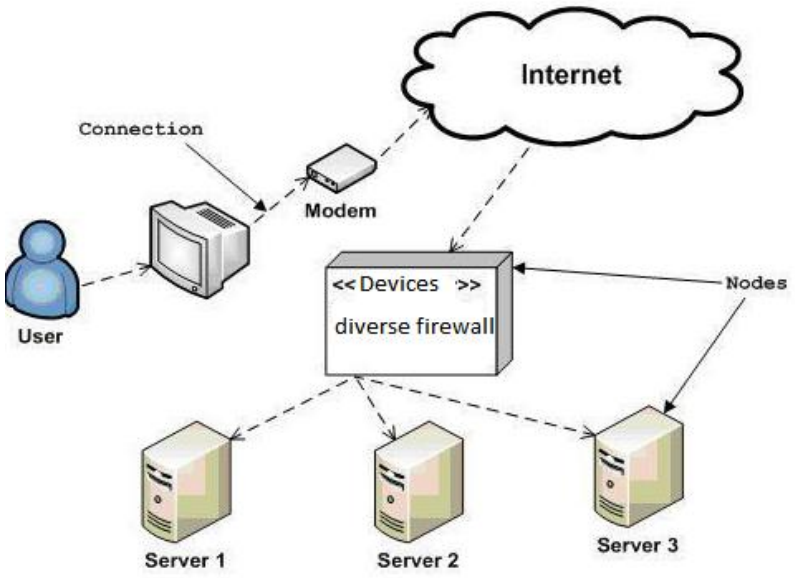


Fig. 1: Deployment of system

Above diagram shows the overall system works, how diverse firewall policy work in between internet and server.

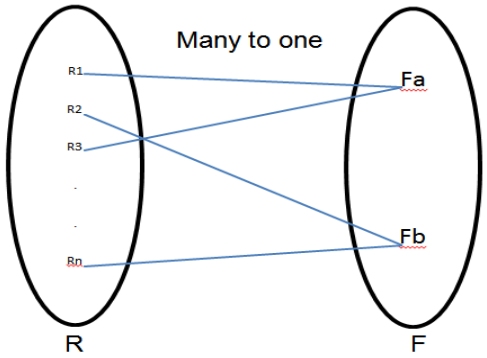
Fig. 2: Firewall Server Authentication

Administrator configures the firewall policy, so it is very important to keep firewall server authentication. Administrator has to log in before configuring firewall policy in form of rules.

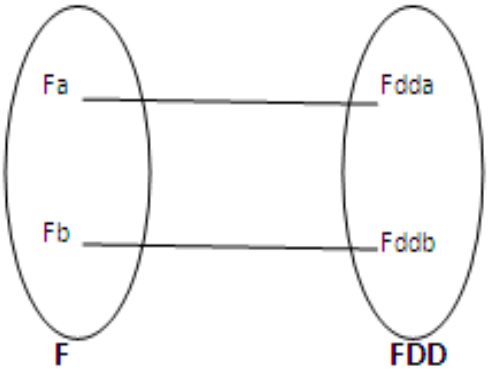
V. MATHEMATICAL MODEL

1. Let 'S' be the set of diverse firewall design system and representing byS= {R, F, FDD, Fs}.
2. R={R| R is set of all rule in requirement specification }R={ R1, R2, R3, …, Rn}Where R1={r1,r2,r3… rn},R2={r1`,r2`,r3`,…,rn`},R3={r1``,r2``,…,rn``}.. set represents Firewall policy set F = {Fa, Fb,…. , Fn} Where Fa={R1},Fb={R2},Fc={R3}
3. FDD set represents Firewall Decision Diagram for different firewall policy setFDD={Fa`,Fb`,…,Fn`}
4. Fs set represent semi isomorphic equivalent of FDD. Fs={Fa``,Fb``,….,Fn``}

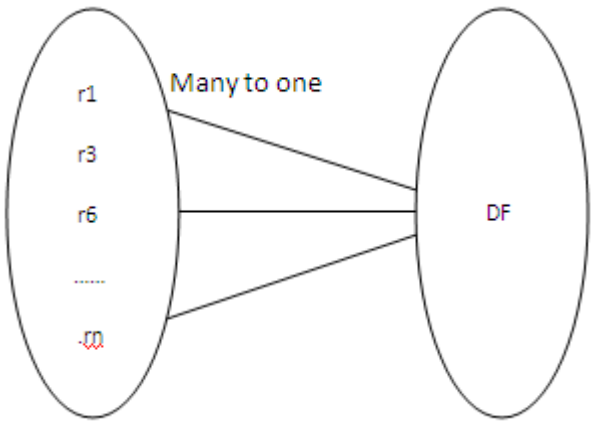
C={R,O,F} The process take rules(R) as input and combine to make Firewall Policy(F).



C={F,Fdd} This process take firewall policy as input and convert it into FDD.



D={RD,O,DF} Every discrepancy is discussed and resolved by all teams and finally generates Diverse Firewall. DF



VI. .CONCLUSION

The method of diverse firewall policy design is effective in practice and can be used flexibly in a variety of scenarios. This paper deals with the method that can compare two firewall policies and detect all functional discrepancies between them in human readable format. This method also can be used in change-impact analysis.

VII. ACKNOWLEDGEMENT

We express true sense of gratitude towards our project guide prof. D. P. Salapurkar. She contributed her valuable guidance and gave us plenty of their precious time to solve every problem that arose.

REFERENCES

[1]. Alex X. Liu, Member, IEEE, Mohamed G. Gouda, Member, IEEE ”Diverse Firewall Design”,2008
[2]. Alex X. Liu, Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824-1266, U. S. A., “Change-Impact Analysis of Firewall Policies”, 2012
[3]. A. Avizienis, “The Methodology of N-Version Programming,” Software Fault Tolerance, Chapter 2, M.R. Lyu, ed. Wiley, pp. 23-46,1995.
[4]. Y. Naveh, Y. Richter, Y. Altshuler, D. L. Gresh, D. P. Connors, Workforce optimization: Identification and assignment of professional workers using constraint programming, IBM J. RES. & DEV. VOL. 51 NO. 3/4 MAY/JULY 2007
[5]. Ada Yetunde Barlatt, Models and Algorithms for Workforce Allocation and Utilization, (Industrial and Operations Engineering) in The University of Michigan, 2009
[6]. M.G.Gouda and A. X. Liu. Firewall design: consistency, completeness and compactness. In Proceedings of the 24th IEEE International Conference on Distributed Computing Systems (ICDCS-04), pages 320–327, March 2004.
[7]. M.G.Gouda and A. X. Liu. Structured firewall design. Computer Networks Journal (Elsevier), 51(4):1106–1120, March 2007.
[8]. A.T. Ernst, H. Jiang, M. Krishnamoorthy *, D. Sier, Staff scheduling and rostering: A review of applications, methods and models, [Online] European Journal of Operational Research 153 (2004) 3–7
[9]. Stephane Bourdais, Philippe Galinier, and Gilles Pesant, A Constraint Programming Application to Staff Scheduling in Health Care, Departement de genie informatique, Ecole Polytechnique de Montreal
[10]. Hongxin Hu, Student Member, IEEE, Gail-Joon Ahn, Senior Member, IEEE, and Ketan Kulkarni, “Detecting and Resolving Firewall Policy Anomalies”