A General Framework For Managing Firewall Policy Anomalies

First Author

Deepak Pedapenki

Department of Computer Science & Engineering, MLRIT
Hyderabad, Andhra Pradesh, India.

Second Author

Sheikh Gouse

Assisi tent Professor, Department of Computer Science & Engineering
MLRIT College, Hyderabad,

Andhra Pradesh, India

Third Author

R.Manasa Annapurna

Assisi tent Professor, Department of Computer Science& Engineering
MLRIT College, Hyderabad,

Andhra Pradesh, India

***Abstract*- Enterprise applications can be integrated to form chains of businesses. The technologies of distributed computing made it possible. This enables applications to interact with each other irrespective of the platform in which they are built. Though it is very useful in real world, such applications face security problems. To overcome this problem, firewalls are used in many networks that can monitor the incoming and outgoing flows. However, the efficiency of firewall depends on its security policies. The quality of security policies configured in firewall increases the level of security. In order to achieve this policies are to be created with plenty of rules and regulations as required. Such security policies are complex in nature but provide more quality rules to protect the systems. Nevertheless, it is the proven fact that creating and maintaining firewall policiesis error prone. The reason behind this is that firewall policies are very complex. Lack of sophisticated tool support is also a problem. In this paper we presented a framework for policy management for firewalls. The technique used by the framework is "rule-based segmentation". This technique could effectively identify anonomolies in firewall policies. We implement the firewall policy management framework customer Java Simulator. The prototype application is used to demonstrate the proof of concept. The simulation results revealed that the framework is able to detect and resovel anomoloies in filewall policies.**

***Index Terms*–Firewall, security, policy management**

I. INTRODUCTION

Firewalls have been around to protect networks. Firewall is the software which monitors incoming and outgoing flows in given network. Firewalls are used in various kinds of networks including Wire Area Networks (WANs). Generally the firewall lies between private network (trusted) and public network (untrusted) and monitors the flows in order to detect anomalies. The firewalls work based on the rules given to them. These rules are not simple. They are complex in nature. They are known as firewall policies. The firewall policies are used to protect networks from malicious attacks launched by adversaries. Security of a system depends on the quality of rules or policies that govern the behavior of a firewall. The firewall policies are defined by system administrators based on the network security requirements of the organization. As the firewall policies are so complex, they are not easy to define and maintain in tune with the motives of the requirements of an organization. Moreover, the emergence of new technologies, improvements in network infrastructure and protocols, it becomes so complex to manage firewalls. Wool [1] studies firewall policies and obtained from various departments and found many security flaws in them. It is not easy task to define and maintann firewall policies as they are error prone. Many researches were on the problem of firewall policy management [2], [3], [4]. There are policy anomaly tools such as FIREMAN and Firewall Plicy Advisor [4]. These tools are meant for finding anomalies in policies. Only pairwise anomalies are detected by Firewall Policy Advisor. Whereas the FIREMAN is capable of detecting based on multiple rules given to them. FIREMAN has drawbacks. It can only apply preceding rules but can't apply the subsequentrules while maiking anlaysis. Moreover FIREMAN can show only configiraiotn problemsbutcan't identity the rules in which actual error is.

As complexity is involved in policy management system, administrators face plethora of challenges. Particularly solving policy conflicts was not properly addressed by the tools available. In the process making changes to conflicted ruels is difficult. As thereare thousands of policy rules and conflicts are more, it is not easy to work with the policy management. The rules are so complex and also involve in conflict with multiple roles making itmore difficult to manage policies and to resolve conflicts. There might be legacy rules applied to firewalls as well and they are not compatible with new rule formats. In this case, changing rules may affect other rules and may lead to inefficiency of firewall. Sometimesadministrators may find inconsistencies and ignore believing that the first few rules only are important. In some cases, this approach might be useful to go ahead with the applications for the time being. However, in the long run it gives problems[4]. In this case conflicts are not errors in the view point of administrators. The tools in this case should focus on rules that conflict with multiple rules. Such conflicts are not easy to eliminate. A practical approach is to identify the rules that are making most conflicts and prioritizing them. First match resolution mechanism of firewall is used to resolve problems. Each packet when observed is mapped to the rules or policies. When rules are conflicting with the data available, the firewall detects it as an anomaly. First match policy can't effectively work with policy conflicts.

When conflict occurs and firewall finds it, the matching rule may not be the correct rule thus causing inconsistency. Such conflict has to be resolved. The existing mechanisms of the firewall are not effectively dealing with such problems. Thus the harmful packets are able to intrude into the security systems. Such incidents can harm to the genuinepackets travelling in the network. There isneed to fill the gap between the first match mechanism and conflict detection.

Hu et al. [5]presented a novel anomaly detection management framework based on the rule based segmentation technique. This tool an effectively detect and resolve firewall anomalies as it can make segmentation of disjoint packets. The introduced conflict resolution is very flexible and support fine-grained conflict management. This tool is effective in detecting and resolving firewall policy conflicts. Compared to other approaches such as [6], this approach has 70% improvement. The prior tools only showed results in the form of possible anomalies. They could not resolve it [4]. This tool also visualizes the anonolies effectively. Grid based visualization is used for this purpose.

In this paper we implement the framework proposed by Hu et al. [5]using a prototype application. The application is a customer simulator to demonstrate the proof of concept. The prototype is capable of helping system administrators to define and manage policies for firewalls. The rest of the paper is organized into the following sections. Section II review literature. Section III provides details of firewall policy anomaly detection management tool. Section IV discusses about the tool implementation details. Section V presents experimental results.

II. RELATED WORK

There exist a number of algorithms and tools designed to assist system administrators in managing and analyzing firewall policies. Lumeta [7] and Fang [8] allow user queries for the purpose of analysis and management of firewall policies. Essentially, they introduced lightweight firewall testing tools but could not provide a comprehensive examination of policy misconfigurations. Gouda et al. [9] devised a firewall decision diagram (FDD) to support

Consistent, complete, and compact firewall policy generation. Bellovin et al. [10] introduced a distributed firewall model that supports centralized policy specification. Several other approaches presenting policy analysis tools with the goal of detecting policy anomalies are closely related to our work. Al-Shaer and Hamed [11] designed a tool called Firewall Policy Advisor to detect pairwise anomalies in firewall rules. Yuan et al. [4] presented FIREMAN, a toolkit to check for misconfigurations in firewall policies through static analysis.

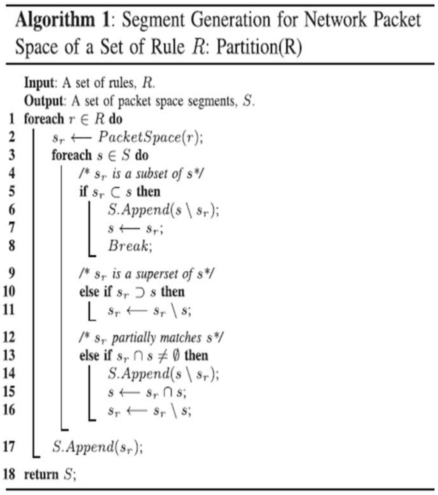
As we discussed previously, our tool, FAME, overcomes the limitations of those tools by conducting complete anomaly detection and providing more accurate anomaly diagnosis information. In particular, the key distinction of FAME is its capability to perform an effective conflict resolution, which has been ruled out in other firwall policy analysis tools. Hari et al. [12] provided an algorithm for detecting and resolving conflicts in a general packet filter. However, they can only detect a specific correlation conflict, and resolve the conflict by adding a resolving filter, which is not suitable for resolving conflicts identified recently in firewall policies. Fu et al. [13] examined conflict detection and resolution issues in IPSec policies, which is not directly

Applicable in firewall policy analysis. Also, there exist other related work to deal with a set of conflict resolution strategies for access control including Fundulaki and Marx [16], Jajodia et al. [15] and Li et al. [16]. These conflict resolution mechanisms can be accommodated in our fine grained conflict resolution framework.

There are several interfaces that have been developed to assist users in creating and manipulating security policies. Expandable Grid is a tool for viewing and authoring access control policies [17]. The representation in Expandable Grids is a matrix with subjects shown along the rows, resources shown along the columns, and effective accesses for the combinations of subjects and resources in the matrix cells. The SPARCLE Policy Workbench allows policy authors toconstruct policies in a natural language interface, which are in turn translated into machine-readable policies [18]. Even though these tools are useful for authoring access control policies, they cannot effectively represent the results of policy analysis for firewalls.

ALGORITHM

As part of the tool we implemented the algorithm proposed by Hu et al. [5]. The algorithm is as shown in fig. 1.



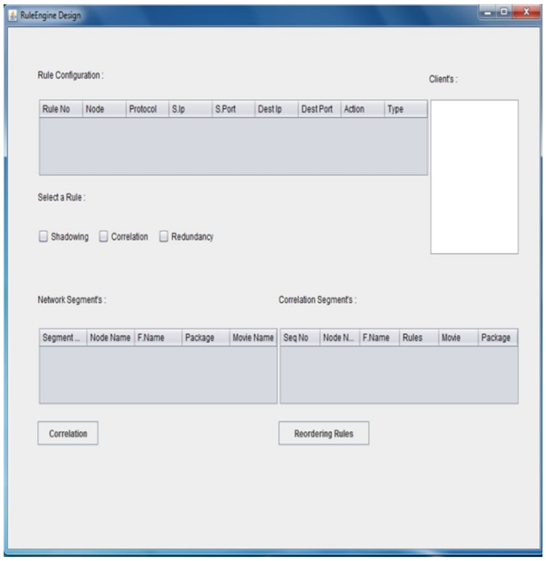
```
Algorithm 1: Segment Generation for Network Packet
Space of a Set of Rule $R$: Partition(R)

Input: A set of rules, $R$.
Output: A set of packet space segments, $S$.
foreach $r \in R$ do
    $s_r \longleftarrow PacketSpace(r)$;
    foreach $s \in S$ do
        /* $s_r$ is a subset of $s$*/
        if $s_r \subset s$ then
            $S.Append(s \setminus s_r)$;
            $s \longleftarrow s_r$;
            $Break$;
        /* $s_r$ is a superset of $s$*/
        else if $s_r \supset s$ then
            $s_r \longleftarrow s_r \setminus s$;
        /* $s_r$ partially matches $s$*/
        else if $s_r \cap s \neq \emptyset$ then
            $S.Append(s \setminus s_r)$;
            $s \longleftarrow s_r \cap s$;
            $s_r \longleftarrow s_r \setminus s$;
    $S.Append(s_r)$;
return $S$;
```

Fig. 1 – Algorithm for Segment Generation for Network Packet Space

As can be seen in fig. 1, the algorithm takes a set of rules and network packet space as input and applies algorithm on it. Any pair of packet space should satisfy the relations such as disjoint, partial match, superset, and subset. After completing the job the algorithm returns a set of packet space segments. More information can be found in [5].

III. TOOL IMPLEMENTATION

The environement used to built the prototype application is a PC with 4 GB or RAM with Core 2 processor. Java programming language is used to provide user interface and functionality. NetBeans is used as an IDE (Integrated Development Environment). Fig. 1 shows the main user interface of the application.



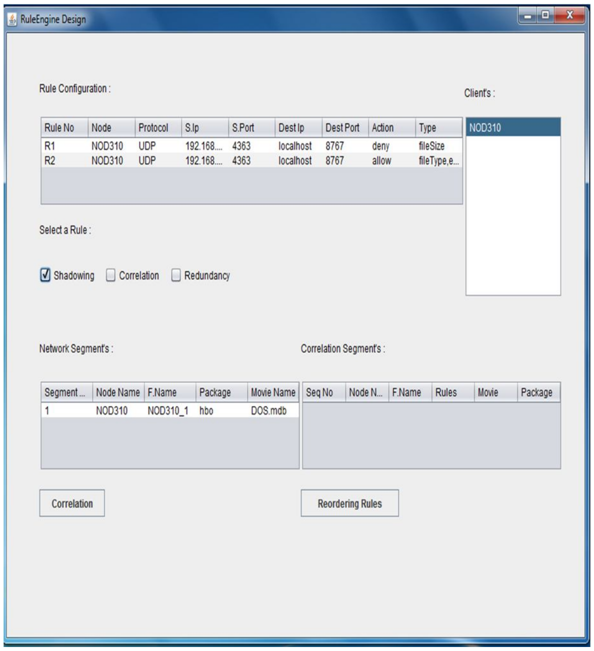
Rule No	Node	Protocol	S.Ip	S.Port	Dest Ip	Dest Port	Action	Type

Segment ...	Node Name	F.Name	Package	Movie Name

Seq No	Node N...	F.Name	Rules	Movie	Package

Fig. 1 –The main UI of the tool

As can be seen in fig. 1, the application has user interface for defining and resolving firewall policy anomalies. The tool has provision for rule configuration, viewing clients, viewing network segments, viewing correlation segments. Mainly three kinds of rules are supported. They are shadowing, correlation and redundancy [38].



Rule No	Node	Protocol	S.Ip	S.Port	Dest Ip	Dest Port	Action	Type
R1	NOD310	UDP	192.168....	4363	localhost	8767	deny	fileSize
R2	NOD310	UDP	192.168....	4363	localhost	8767	allow	fileType,e...

☑ Shadowing □ Correlation □ Redundancy

Segment ...	Node Name	F.Name	Package	Movie Name
1	NOD310	NOD310_1	hbo	DOS.mdb

Seq No	Node N...	F.Name	Rules	Movie	Package

Fig. 2 –GUI showing the results of anomaly detection

As can be seen in fig. 2, shadowing, correlation and redundancy are the three anomalies. As seen in the screen the shadowing results are shown as this option is selected.

IV. EXPERIMENTS AND RESULTS

The experimental resultsof the tool are recorded and presented in terms of policy vs. conflicts; policy vs. security risk value; policy vs. availability loss value; and policy vs. number of redundant rules.

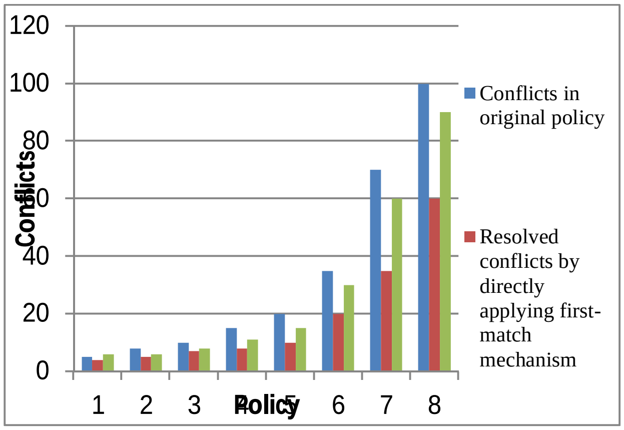


Figure 2. Resolution Rate

As can be seen in fig. 2, it is Conflicts in original policy, resolved conflicts by directly. The average number of nodes per hop is presented in horizontal axis while the vertical axis presents throughput.

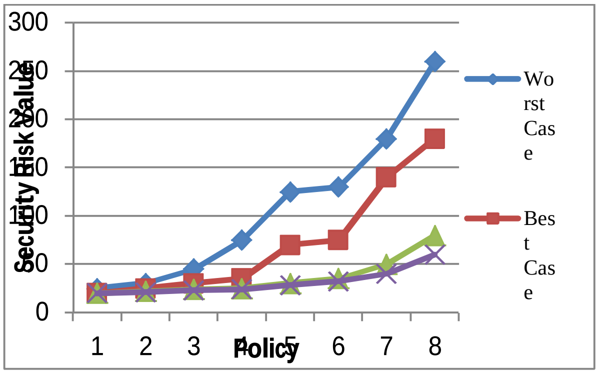


Fig. 3 - Risk Reduction

As can be seen in fig. 3. The security risk value is presented in horizontal axis while the vertical axis presents policy.

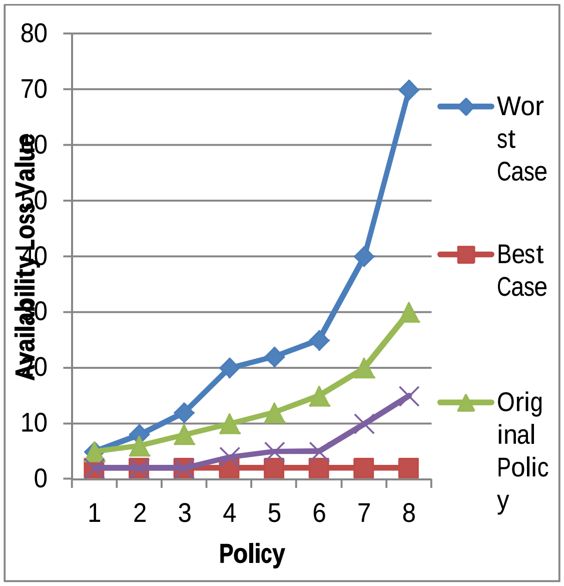


Fig 4 Availability Improvement

As can be seen in fig.4. The availability loss value is presented in horizontal axis while the vertical axis presents policy.

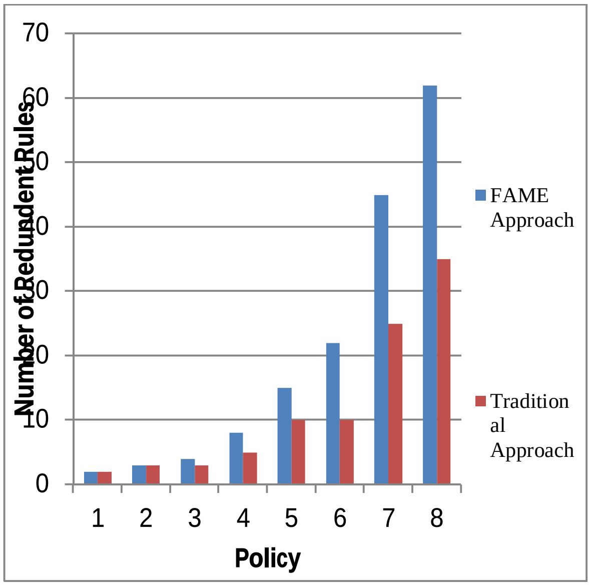


Fig 5 - Evaluation of Redundancy removal

As can be seen in fig. 5. The number of redundant rules is presented in horizontal axis while the vertical axis presents policy.

V. CONCLUSION

In this paper we have implemented a framework proposed by Hu et al. [5] is implemented using a custom Java simulator. The tool is meant for detecting and resolving firewall policy anomalies. The tool is based on rule-based segmentation technique. Grid based visual representation is also used to effectively present the abilities of the tool. The application allows system administrators to define policies and also detect and resolve anomalies. The experimental results revealed that the tool is efficient and can be used to explore possibilities of using it in the real world.

REFERENCES

[1] A. Wool, “Trends in Firewall Configuration Errors: Measuring theHoles in Swiss Cheese,” IEEE Internet Computing, vol. 14, no. 4,pp. 58-65, July/Aug. 2010.

[2] J. Alfaro, N. Boulahia-Cuppens, and F. Cuppens, “CompleteAnalysis of Configuration Rules to Guarantee Reliable NetworkSecurity Policies,” Int’l J. Information Security, vol. 7, no. 2, pp. 103-122, 2008.

[3] F. Baboescu and G. Varghese, “Fast and Scalable ConflictDetection for Packet Classifiers,” Computer Networks, vol. 42, no. 6, pp. 717-735, 2003.

[4] L. Yuan, H. Chen, J. Mai, C. Chuah, Z. Su, P. Mohapatra, and C.Davis, “Fireman: A Toolkit for Firewall Modeling and Analysis,”Proc. IEEE Symp. Security and Privacy, p. 15, 2006.

[5]Hongxin Hu,Gail-Joon Ahn, IEEE, and Ketan Kulkarni., 2012. “Detecting and ResolvingFirewall Policy Anomalies”, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 9, NO. 3, MAY/JUNE 2012.

[6] E. Lupu and M. Sloman, “Conflicts in Policy-Based DistributedSystems Management,” IEEE Trans. Software Eng., vol. 25, no. 6,pp. 852-869, Nov./Dec. 1999.

[7] A. Wool, “Architecting the Lumeta Firewall Analyzer,” Proc. 10thConf. USENIX Security Symp., vol. 10, p. 7, 2001.

[8] A. Mayer, A. Wool, and E. Ziskind, “Fang: A Firewall AnalysisEngine,” Proc. IEEE Symp. Security and Privacy, pp. 177-189, 2000.

[9] M. Gouda and X. Liu, “Firewall Design: Consistency, Completeness, and Compactness,” Proc. 24th Int’l Conf. Distributed ComputingSystems (ICDCS ’04), p. 327, 2004.

[10] S. Ioannidis, A. Keromytis, S. Bellovin, and J. Smith, “Implementinga Distributed Firewall,” Proc. Seventh ACM Conf. Computer andComm. Security, p. 199, 2000.

[11] E. Al-Shaer and H. Hamed, “Discovery of Policy Anomalies inDistributed Firewalls,” IEEE INFOCOM ’04, vol. 4, pp. 2605-2616, 2004.

[12] A. Hari, S. Suri, and G. Parulkar, “Detecting and Resolving PacketFilter Conflicts,” Proc. IEEE INFOCOM, pp. 1203-1212, 2000.

[13] Z. Fu, S. Wu, H. Huang, K. Loh, F. Gong, I. Baldine, and C. Xu,“IPSec/VPN Security Policy: Correctness, Conflict Detection andResolution,”

Proc. Int’l Workshop Policies for Distributed Systems andNetworks (POLICY ’01), pp. 39-56, 2001.

[14] I. Fundulaki and M. Marx, “Specifying Access Control Policies forXML Documents with Xpath,” Proc. Ninth ACM Symp. AccessControl Models and Technologies, pp. 61-69, 2004.

[15] S. Jajodia, P. Samarati, and V.S. Subrahmanian, “A LogicalLanguage for Expressing Authorizations,” Proc. IEEE Symp.Security and Privacy, pp. 31-42, May 1997.

[16] N. Li, Q. Wang, W. Qardaji, E. Bertino, P. Rao, J. Lobo, and D. Lin,“Access Control Policy Combining: Theory Meets Practice,” Proc.14th ACM Symp. Access Control Models and Technologies, pp. 135-144, 2009.

[17] R. Reeder, L. Bauer, L. Cranor, M. Reiter, K. Bacon, K. How, andH. Strong, “Expandable Grids for Visualizing and AuthoringComputer Security Policies,” Proc. 26th Ann. SIGCHI Conf. HumanFactors in Computing Systems, pp. 1473-1482, 2008.

[18] C. Brodie, C. Karat, and J. Karat, “An Empirical Study of NaturalLanguage Parsing of Privacy Policy Rules Using the SPARCLEPolicy Workbench,” Proc. Second Symp. Usable Privacy and Security,pp. 8-19, 2006.

Authors

Deepak Pedapenki , he is pursuing M.Tech (CSE) MLRIT, Hyderabad, AP, INDIA. He has received B.Tech Computer Science and Engineering in the year 2010. His main research Interest includes web Technologies & Computer Networking.

2. Sheikh Gouse B.Tech, M.Tech CSE He is currently with the Department of Computer Science and Engineering, MLRIT, Andhra Pradesh, India. He is having 7 years of teaching experience. He is Certified in Oracle 9i: SQL & Java SE 6.His research interest areas Programming (C&DS, C++, and JAVA), Computer Graphics, Data Mining, Software Engineering, Web Technologies & Computer Networking.

R.Manasa Annapurna B.Tech, M.Tech CS She is currently with the Department of Computer Science and Engineering, MLRIT, Andhrapradesh, India. She is having 4 years of teaching experience**.** Her research interest areas Programming (C&DS, C++, and JAVA), Computer networks, Web technologies