Volume 4, Issue 3, March 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Survey: Firewall Anomaly Management

Hani Patil*, Mrs. A. A. Agarkar

IT, SCOE, Pune University,
India

Abstract— Internet has connected the networks globally. The vast span of interconnection between various networks makes security an important element for enterprise networks as well as other personal systems. This led to the development of certain security measures to immune the systems and networks. Firewall is one such element present to deal with network attacks which maybe in the form of illegitimate traffic. It is a defense mechanism which filters the unwanted packets coming in or going out from a secured network. Such decisions are taken on the basis of the filtering rules which are based on the policies predefined inside the firewall. In this paper, a brief description of the issues present in firewall policies as well as the identification and correction of the same is discussed by considering various approaches.

Keywords— Anomaly Management, Firewall, Firewall Policy, Policy conflicts, Test Packets.

I. INTRODUCTION

Internet has connected the networks globally. The vast span of interconnection between various networks makes security an important element for enterprise networks as well as other personal systems. This led to the development of certain security measures to immune the systems and networks. Firewall is one such element present to deal with network attacks which maybe in the form of illegitimate traffic. It is a defence mechanism which filters the unwanted packets coming in or going out from a secured network. Such decisions are taken on the basis of the filtering rules which are based on the policies predefined inside the firewall.

Firewalls are designed to ensure proper access control automatically. They can provide access control between various types of networks. The security issue can be within a public network or private network or within various local networks. There can be blocking of packets or rerouting them.

In any business organization or enterprise the compromise of even a small part of internal network provides major threat to the whole organization. Due to the reason that the firewalls are the basic elements of network security it becomes very important to deploy them very carefully so that they must fulfil their purpose to the fullest. Thus firewalls are deployed at the boundary of the networks either to filter the packets or reroute them. It is difficult to design firewalls in one go and thus they keep on getting modified internally. This can be in the form of adding new filters etc. The design and implementation are updated for providing the maximum security if required.

Firewalls are present to act as a screener for various kinds of traffic. Only the deployment of firewall does not fulfil the requirement of providing security to the network. It is very important that it should be configured according to the needs of the particular organization or network. The configuration is generally done by the administrator and it is not an easy job. The firewall policy configuration requires a very exhaustive study of the needs of the network. Any small part left can cause a series of unwanted data coming and going from the organization or network.

Firewalls can be software, hardware devices or a combination of the two. Firewalls can be used everywhere i.e. in centralized network or distributed networks as well where there can be multiple access points for data circulation. As defining the rules in a firewall policy is a very comprehensive process there are various kinds of errors or slips might take place. There have been various issues in the existing policies which need to be considered and managed carefully. Also various works has also been done to improvise the firewall policies.

II. FIREWALL POLICY MODEL AND ANOMALIES OVERVIEW

A firewall policy is sequence of rules containing two parts i.e. <condition and action>. Each rule in a firewall policy clearly states what action has to be taken if the data packet satisfies that rule. For matching the data packet to its respective rule, various fields are present in the <condition > part of the rule. These fields have the similar entities which are generally present in any data packet. For example the fields can be protocol, source IP, source port, destination IP, destination port etc. The incoming or outgoing data packet specifications are compared with the fields of the <condition> of the rules present. Also, as there is a very vast address space in global networks, it is not possible to individually mention all the addresses in the fields. Therefore certain field values can also show a particular range in the rules. For example a single "*" in the IP address depicts an entire address range of 0.0.0.0 to 255.255.255.255.

TABLE 1
An Example of Firewall Policy

Rule	Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
r1	UDP	10.1.2.*	*	172.32.1. *	53	deny
r2	UDP	10.1.*.*	*	172.32.1. *	53	deny
r3	TCP	10.1.*.*	*	192.68.*. *	25	allow
r4	TCP	10.1.1.*	*	192.68.1. *	25	deny
r5	*	10.1.1.*	*	*	*	Allow

Various research works carried out on firewall policies
Have shown that there are basically two types of issues in firewall policies which may be further refined into their sub issues. The basic issues are:

1. Conflicts
2. Redundancies

Based on the classification the firewall policy issues are further sorted and explained based on the table1 as follows:

A. Generalization

A rule is said to be generalized if a subset of packets matching this rule is also covered in a previous rule or previous set of rules but having different action specified e.g. r5 is a generalized version of r4.

B. Shadowing

A rule is said to be shadowed by one or a set of previous rules if all the packets matching with the shadowed rule also get covered due to matching with a previous rule or set of rules. In that case the previous rules will be applied to the packet and the action specified in the shadowed rule will not take place at all e.g. r4 is shadowed by r3.

C. Correlation

Correlation is said to occur when different rules match the same packets but specify different action. In this case positioning of the rules matter a lot. This is because if their actions are different then whichever rule is positioned earlier will be considered for the packet. The action may also be reversed if the rule positions are swapped e.g. r2 correlates with r5.

D. Redundancy

A rule is said to be redundant if there exists a similar rule or a general rule having the same action specified e.g. r1 is redundant with respect to r2.

The firewalls need to be analyzed for the above mentioned anomalies. Also it is possible to prevent the policies at the time of designing itself by following a proper step by step process which can overcome these anomalies. Firewall faults can be removed both automatically and manually.

Let us have a look on both of them one by one.

III. AUTOMATIC CORRECTION OF FIREWALL POLICY FAULTS

Not much work has been done in the field of automatic correction of firewall policy faults. But still a few steps have been taken to automate the correction process as it proves to be very hectic to the administrator to go through the entire firewall policy. The authors [1] found out three key challenges which are:

1. It is very difficult to find out the total number of firewall policy faults and also to determine the type of fault present.
2. It is very difficult to correct the firewall fault as there is enormously large number of rules involved.
3. It is even more difficult to correct the firewall policy faults without introducing new faults as any change made in the previous rule might affect the subsequent rules.

The automatic correction first required the generation of test packets which was based on the local constraint solving. It considers a rule field and generates the packets accordingly to be true or false w.r.t. to the constraints i.e. the rule fields.

The fault model considered in the automatic correction consists of the following five general types of faults:

A. Wrong order

This type of fault means that the order of the rules is wrong as generally the first match semantics is followed in a firewall policy. Wrong order of rules can inverse the action in case of a conflict which may lead to misconfigurations in a firewall policy.

B. Missing rules

This fault arises due to the fact that there is not enough number of rules to cover all the packets and the administrator needs to add a few rules to provide a proper coverage.

C. Wrong predicates

This fault indicates that there are some predicates present which can be wrong or must be overlooked in some cases.

D. Wrong decisions

This type of fault shows that few decisions are wrong in some rules.

E. Wrong extra rules

This fault indicates that there are few rules which need to be deleted as they are anyways being covered by the new rule additions being done by administrators while resolving some other conflicts related to the same firewall policy rules.

In the automatic correction of the above mentioned faults an assumption is made that states that all the faults are a set of misclassified packets. The correction techniques for the faults are called as order fixing, rule addition, predicate fixing, and decision fixing and rule deletion respectively.

Normally a fault is said to be detected when the administrators find out that some illegitimate packets have been allowed or some legitimate packets have been blocked. As the number of such packets is very small it is difficult to find out the proper information about the fault present in the policy. Therefore after finding a faulty firewall policy a set of packets are generated by the automated packet generation technique. Next step is that the administrator finds out the set of failed tests and passed tests. Any packet is said to have passed test if it is classified in tune with the action specified in the matching rule from the firewall policy, Fig. 1 shows an example of faulty firewall policy with passed and failed tests.

$r1 : F1 \in [1, 5] \wedge F2 \in [1, 10] \rightarrow a$

$r2 : F1 \in [1, 6] \wedge F2 \in [3, 10] \rightarrow a$

$r3 : F1 \in [6, 10] \wedge F2 \in [1, 3] \rightarrow d$

$r4 : F1 \in [7, 10] \wedge F2 \in [4, 8] \rightarrow a$

$r5 : F1 \in [1, 10] \wedge F2 \in [1, 10] \rightarrow d$

(a) An example faulty firewall policy

$p1 : (3, 2) \rightarrow a$

$p2 : (5, 7) \rightarrow a$

$p3 : (6, 7) \rightarrow a$

$p4 : (7, 2) \rightarrow d$

$p5 : (8, 10) \rightarrow d$

(b) A set of passed tests

$p6 : (6, 3) \rightarrow d$

$p7 : (7, 9) \rightarrow a$

$p8 : (8, 5) \rightarrow d$

(c) A set of failed tests

Fig. 1 Faulty firewall policy and its passed, failed tests

The aim of the automatic correction technique is to apply the five possible solutions i.e. order fixing, rule addition, predicate fixing, decision fixing and rule deletion in such a way so as to make all the failed tests come under passed tests in the minimum possible steps of modification. This is a global optimization problem for which the authors followed a greedy approach. In the automatic correction technique also, an administrator can be involved for supervision. He/she can decide which of the five correction techniques must be used. If it doesn't happen the greedy algorithm can itself choose this for the system. It is further mentioned that the administrator can induce restrictions if there are some critical requirements about few packets which must be accepted or discarded. At each step the modification generated must confirm with the critical requirement, and if it does not match the next correction technique is used. The five correction techniques described by the authors [1] are briefly discussed further.

1. Order fixing

As there is a first match rule semantics followed generally, any kind of swapping between a pair of rules can change the functionality of the firewall policy. The rules cannot be randomly reordered. So FDDs [2] are used, that is an all match decision diagram for firewalls. It is found out using FDD that which pair of rules after swapping maximises the set of passed test.

2. Rule addition

Rule addition is based on adding rules to maximize the number of passed tests.

3. Predicate fixing

This technique tries to fix the number of predicates in such a way that the new rule allows maximum tests to pass.

4. Decision fixing

This method tries to fix the decisions to allow maximum tests to be passed.

5. Rule deletion

It again uses FDD to identify which rules must be deleted to increase the number of passed tests. In the above mentioned techniques ,the experiments done by authors proved that this automatic correction technique is effective for the three types of faults i.e. wrong order, wrong decision and wrong extra rules.

IV. MANUAL FIREWALL POLICY CORRECTION TOOLS

There has been various works done for correcting or managing the firewall policies with the help of tools. These tools help the administrator to deal with the exhaustive firewall policy faults. One such work is the FPA[3], Firewall Policy Advisor. It is a tool which is capable of detecting only pair wise firewall anomalies which is not enough. Again there is another tool FIREMAN [4]. It can detect anomalies but only considers the preceding rules. Keeping the limitations of these two and various other tools in mind, the authors in another work [5] implemented a new technique which provided a complete coverage of all kind of firewall policy faults. Its idea was to represent anomalies in a grid based representation. This grid based representation again depended on the packet space segmentation and classification. After the grid based representation the authors discussed an anomaly management framework FAME[6]. FAME consisted of two major functionalities i.e. conflict detection and resolution and the second as redundancy discovery and removal. FAME was developed in java containing six modules two cover the above mentioned functionalities. The first three modules included conflicting segment identification and correlation, generating an action constraint for each conflicting segment which was based on the risk levels of conflicts and a reordering algorithm to discover an optimal solution by combining permutation and greedy algorithm. A threshold value was used to select a suitable rule reordering algorithm. Remaining modules included property assignment module, correlation module for redundancy removal. The FAME utilized the BDD [7], ordered binary decision diagrams to represent the firewall rules. FAME was a very innovative work in terms of providing proper visualization interface to the administrator. As two visualization interfaces i.e. policy conflict viewer and policy redundancy viewer were designed to manage the anomalies distinctly, it proves to be provide high clarity. FAME supported another lower layer for providing the underlying functionalities and resources.

V. Conclusions

In this paper both the automatic technique as well as manual technique for firewall policy anomaly detection and correction has been discussed. It is seen that the above mentioned techniques for firewall anomalies management have still got various areas to improve upon. Automatic correction technique discussed has not been able to correct all the five types of faults mentioned previously. Also it is providing no help in detecting the faults in firewall. In case of FAME, decisions like adjusting the threshold value can be a critical issue. Also more qualitative analysis needs to be done for the visualization interfaces. Firewall anomaly detection and correction still requires a lot of work to be done to handle the inadequacy in the automatic as well as manual techniques.

References

[1] F. Chen, A. X. Liu ,J. Hwang and T. Xie "*First Step Towards Automatic Correction of Firewall Policy Faults*", ACM Transactions on Autonomous and Adaptive Systems (TAAS) vol. 7 Issue 2, July 2012

[2] A. X. Liu, Y. Zhou and C. R. Meiners, "*All-match based complete redundancy removal for packet classifiers in TCAMs*" , In Proceedings of IEEE Conference on Computer Communications (INFOCOM) (2008), pp. 574–582.

[3] E. Al-Shaer and H. Hamed, "*Discovery of Policy Anomalies in Distributed Firewalls*," IEEE INFOCOM '04, vol. 4, pp. 2605-2616, 2004.

[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] H. Hu, G. Ahn, and K. Kulkarni, "*Detecting and Resolving Firewall Policy Anomalies*", IEEE Transactions on Dependable and Secure Computing, vol. 9, NO. 3, MAY/JUNE 2012.

[6] H. Hu, G. Ahn, and K. Kulkarni, "*Fame: a firewall anomaly management environment*", In Proceedings of the 3rd ACM workshop on Assurable and usable security configuration, pages 17–26, 2010.