An Implementation of Anomaly Detection Mechanism for Centralized and Distributed Firewalls

Rupali Chaure
M.Tech Scholar, PG Dept. of Computer Science & Engineering
NRI Institute of Information Science and Technology
Bhopal, India

ABSTRACT

Due to the rapid growth in the field of Internet, the related security mechanisms are the key area of research. Firewalls serve the solution for secured Internet experience. Latest firewalls are fully-equipped for providing hi-end security to the network. However, due to the continuous growth of security threats, the firewall mechanisms and policies are compulsorily needed to get updated. The manual processing for detecting anomalies in firewall is complex and often error-prone. Any minor change in the rule set of firewall leads to the requirement of rigorous analysis for maintaining the consistency and efficiency of firewall mechanism. Many Data structures have been proposed for detection and removal of anomalies so as to reduce the burden of Network Administrator. In this paper I have shown the results of implementation of a mechanism for the anomalies detection in the centralized and distributed firewall systems. This paper also discusses the design implementation of the irrelevance anomaly for the intra firewalls. It is developed in VB.Net and SQL Server. The algorithm used in this paper purifies the rule sets of firewall in such a way that makes the rule set optimal and free from all known anomalies.

I. INTRODUCTION

A firewall is a system acting as an interface between a network and one or more external networks. It helps implementing the security policy of any network by deciding which packets to let pass through and which to block, based on the set of rules defined by the network administrator. Any error in defining the rules may compromise the system security by letting undesired traffic pass through or blocking the desired traffic. The rules when defined manually often results in a set that contains conflicting, redundant or overshadowed rules, which creates anomalies in the firewall policy. A network firewall protects a computer network from unauthorized access. Network firewalls may be hardware devices, software programs, or they may be a combination of the two. Network firewalls guard an internal computer network (home, school, business intranet) against malicious access from the outside. Network firewall may also be configured to limit access to the outside network of internal users. If passwords provide a 'door' to cover the 'doorway' into your 'house', then firewalls provide 'shutters' to cover the 'windows'. A firewall does absolutely nothing to protect the windows you leave open - that's the job of the programs, which provide the services at those windows.

The firewall is ideally a separate computer, which exists between a network and the Internet. It can be a purpose-built device - some of them are available as small black boxes which look like network hubs. This computer can be any old 486, with a highly secure operating system that provides an inbuilt firewall. None of the network computers should be able to access the Internet or can be accessed from the Internet without going through the firewall.

II. FIREWALL RULES

Whenever a packet is tested by the Firewall, it means that the header of the incoming or outgoing packet is tested against all the rules one by one, which are stored in the Firewall rule set. The rules in the Firewall rule set consists all the header information like source and destination address, source and destination port address and the corresponding action to performed i.e. whether to accept or deny any packet which matches all the other fields of any rule in the rule set. The rules are stored in the rule set in the following format,

<order> <prtcl> <S_ip> <S_port> <D_ip> <D_port> <action>

Here all the terms have respective meanings with properly defined domains. Order is the number at which the rule is stored in the rule set, prtcl is the type of the protocol specified in the packet's header, s_ip and s_port are the source machines' IP address and port number respectively. Similarly D_ip and D_port are the IP address and port number of the destination. In the last action field defines the resulting action to be performed on the packet which matches all the previous fields. The action field can be either ACCEPT or DENY.

These rule sets of any firewall defines the Security Policy of that organization. The security policy of any organization is very dynamic i.e. it can be altered anytime whenever the administrator wants to modify the rules. So such frequent changes are the reason for the inconsistencies in the rule set.

III. FIREWALL ANOMALIES

As the rule set is very large it becomes difficult to check all the rules for any redundancy. Hence the updating of rule set may generate erroneous set of rules which are unable to perform their intended job i.e. protection from unauthorized access to the network or from the network. These errors in the rule set are called anomalies that have to be detected and removed from rule set for the efficient working of any firewall. Firewall anomalies can be classified in to two major categories based on the architecture of the underlying firewall. Depending on the requirements of the organization there can be a single or multiple firewalls. If there exists only one centralized firewall that separates the internal private networks from the external network threats then it called the centralized firewall system and the anomalies for that firewall are classified as Intra-firewall anomalies. Similarly when the network of any organization has more than one firewalls then it is called distributed firewall system and the anomalies for distributed firewalls are classified under the inter-firewall anomalies. According to E Al-Shaer & H. Hamed [4] these anomalies are defined in the following points:

III (A) Intra-Firewall Anomalies

1. Shadowing Anomaly

Two rules are said to have shadowing anomaly, whenever the rule which comes first in rule set matches all the packets and the second

rule which is positioned after the first rule in rule set does not get chance to match any packet because the previous rule has matched all the packets. It is a very critical problem since the rule coming later to the previous rule will never get activated. Hence the traffic to be blocked will be allowed or the traffic to be permitted can be blocked.

2. Correlation Anomaly

Two rules are said to have correlation anomaly if both of them matches some common packets i. e. the rule one matches some packets, which are also matched by the rule second. The problem here is that the action performed by both the rules is different. Hence in order to get the proper action such correlated rules must be detected and should be specified with proper action to be performed.

3. Generalization Anomaly

Two rules which are in order one of them is said to be in generalization of another if the first rules matches all the packets which can be also matched by the second rule but the action performed is different in both the rules. In this case if the order is reversed then the corresponding action will also be changed. The rule, which comes later in the rule list, is shadowed by the previous rule and also it has no effect on incoming packets. The super set rule is called General rule and the subset rule is called Specific rule. If such generalization relation exists between two rules then the super set rule should be placed after the subset rule in the rule list.

4. Redundancy Anomaly

Two rules are said to be redundant if both of them matches some packets and the action performed is also the same. So there is no effect on the firewall policy if one of redundant rules will be removed from the rule set. It is very necessary to search and remove the redundant rules from the rule set because they increase the search time, space required to store the rule set and thus decrease the efficiency of the firewall. The firewall administrator should detect and remove such redundant rules to increase the performance of the firewall.

5. Irrelevance Anomaly

Any rule is said to be irrelevant if for a given time interval it does not matches any of the packets either incoming or outgoing. Thus if any type of the packets do not match a rule then it is irrelevant i.e. there is no need to put that rule in the rule set.

III (B) Inter-Firewall Anomalies

1. Shadowing Anomaly

A shadowing anomaly occurs if an upstream firewall blocks the network traffic accepted by a downstream firewall. Formally, rule Rd is shadowed by rule Ru if one of the following conditions holds:

$$Rd\_\text{EM}Ru,Ru[\text{action}]=\text{deny},Rd[\text{action}]=\text{accept} \quad (1)$$
$$Rd\_\text{IM}Ru,Ru[\text{action}]=\text{deny},Rd[\text{action}]=\text{accept} \quad (2)$$
$$Ru\_\text{IM}Rd,Ru[\text{action}]=\text{deny},Rd[\text{action}]=\text{accept} \quad (3)$$
$$Ru\_\text{IM}Rd,Ru[\text{action}]=\text{accept},Rd[\text{action}]=\text{accept} \quad (4)$$

In cases (1) and (2), the upstream firewall completely blocks the traffic permitted by the downstream firewall. In cases (3) and (4) the upstream firewall partially blocks the traffic permitted by the downstream firewall.

2. Spuriousness Anomaly

A spuriousness anomaly occurs if an upstream firewall permits the network traffic denied by a downstream firewall. Formally, rule Ru allows spurious traffic to rule Rd if one of the following conditions holds:

$$Ru\_\text{EM}Rd,Ru[\text{action}]=\text{accept},Rd[\text{action}]=\text{deny} \quad (5)$$
$$Ru\_\text{IM}Rd,Ru[\text{action}]=\text{accept},Rd[\text{action}]=\text{deny} \quad (6)$$
$$Rd\_\text{IM}Ru,Ru[\text{action}]=\text{accept},Rd[\text{action}]=\text{deny} \quad (7)$$
$$Rd\_\text{IM}Ru,Ru[\text{action}]=\text{accept},Rd[\text{action}]=\text{accept} \quad (8)$$
$$Ru\_\text{IM}Rd,Ru[\text{action}]=\text{deny},Rd[\text{action}]=\text{deny} \quad (9)$$

In cases (5) and (6), the rule Ru in the upstream firewall permits unwanted traffic because it is completely blocked by Rd in the downstream firewall. In cases (7) and (8) part of the traffic allowed by rule Ru in upstream firewall is undesired spurious traffic since it is blocked by rule Rd in the downstream firewall. Case (9) is not as obvious as the previous cases and it needs further analysis. Since we assume there is no intra-firewall redundancy in the upstream firewall, the fact that Ru has a "deny" action implies that there exists a superset rule in the upstream firewall that follows Ru and accepts some traffic blocked by Rd. This occurs when the implied "accept" rule in the upstream firewall is an exact, superset or subset match (but not correlated) of Rd.

3. Redundancy Anomaly

A redundancy anomaly occurs if a downstream firewall denies the network traffic already blocked by an upstream firewall. Formally, rule Rd is redundant to rule Ru if, on every path to which Ru and Rd are relevant, one of the following conditions holds:

$$Rd\_\text{EM}Ru,Ru[\text{action}]=\text{deny},Rd[\text{action}]=\text{deny} \quad (10)$$
$$Rd\_\text{IM}Ru,Ru[\text{action}]=\text{deny},Rd[\text{action}]=\text{deny} \quad (11)$$

In both of these cases, the deny action in the downstream firewall is unnecessary because all the traffic denied by Rd is already blocked by Ru in the upstream firewall.

4) Correlation Anomaly:

A correlation anomaly occurs as a result of having two correlated rules in the upstream and downstream firewalls. We defined correlated rules in Section III-A. Intra-firewall correlated rules have an anomaly only if these rules have different filtering actions. However, correlated rules having any action are always a source of anomaly in distributed firewalls because of the implied rule resulting from the conjunction of the correlated rules. This creates not only ambiguity in the inter-firewall policy, but also spurious, and shadowing anomalies. Formally, the correlation anomaly for rules Ru and Rd occurs if one of the following conditions holds:

$$Ru\_\text{C}Rd,Ru[\text{action}]=\text{accept},Rd[\text{action}]=\text{accept} \quad (12)$$
$$Ru\_\text{C}Rd,Ru[\text{action}]=\text{deny},Rd[\text{action}]=\text{deny} \quad (13)$$
$$Ru\_\text{C}Rd,Ru[\text{action}]=\text{accept},Rd[\text{action}]=\text{deny} \quad (14)$$
$$Ru\_\text{C}Rd,Ru[\text{action}]=\text{deny},Rd[\text{action}]=\text{accept} \quad (15)$$

IV. PROPOSED SYSTEM

The size of the rule set varies according to the type of the organization. Generally the rule set is very large because different administrators come and modify the policy rules according to their requirements and so is the reason of occurrence of anomalies. Because of the large size of the rule set it is difficult to detect anomalies by manually checking the rules one by one. So there is

different software implemented to perform the job of anomaly detection and removal automatically.

The endless growth of internet in today's commercial and technical scenario finds the need to secure the data which should be protected against unauthorized access. Firewalls perform this job of protecting any network. A lot of research work has been done in the field of Firewalls. The main problem that arises in firewalls is that anomalies are generated during updating the rules in the rule set. So the main interest of research is the detection and removal of firewall anomalies. There are a number of approaches for this, which varies to each other in some implementation context.

The proposed system is shown in the figure–1 which shows four stages. The Rule set Extractor stage generates policy rules for intra or inter firewall system. Then these randomly generated or user defined rules will be checked by the anomaly detection algorithm in the Rule set Analyzer stage. This stage generates the log file for the anomalies detected with the rule numbers and the corrective actions. User can now edit the anomalous rules as guided by the analyzer stage and a new anomaly free rule set can be achieved. After this stage comes the Rule set Updator stage which defies the manual updations done due to the policy changes in the organization. So this updation will again generate some anomalies since it is a manual process hence the rule set is again given as input to the Rule set Analyzer stage.

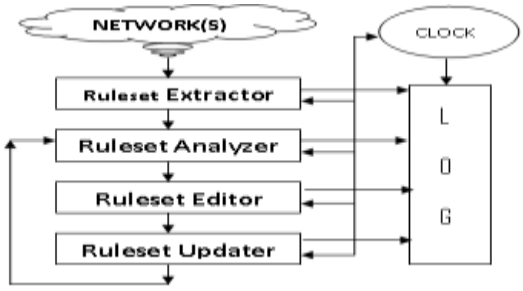

Fig. 1 :- Firewall Proposed Rule set Review Mechanism.

V. IMPLEMENTATION

We used VB.Net for developing the front end of this software and SQL Server for the back end. The reason for using VB.Net is its flexibility. We can add or remove any features without editing the whole code. We separated the standalone functions like port matching and IP address matching in separate functions which are reused again and again. For the back end we wanted a freely distributed and powerful database so SQL Server was a good choice. Whole of the rule list is stored in the database. All fields except the Rule No. are stored as the Strings. They are accessed and parsed according to the use, edited if necessary and stored again in the String form.

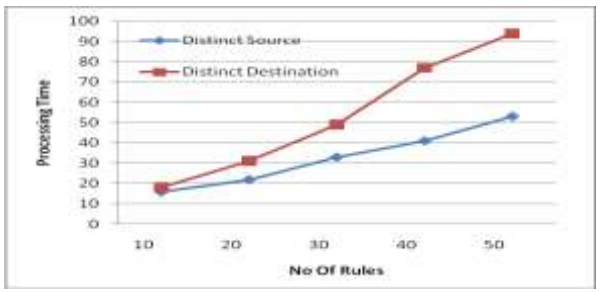

Fig. 2 :- Results of Processing Time for Intra Firewall.

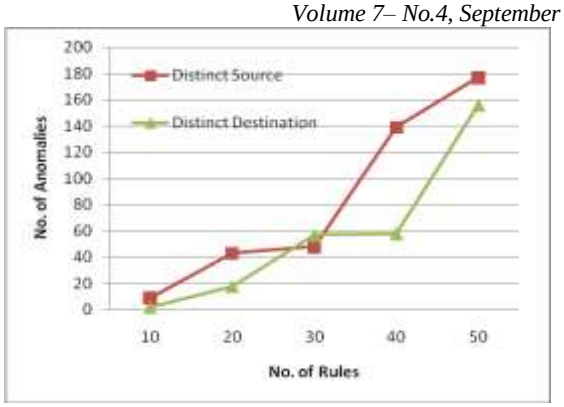

Fig. 3 :- Results of Number of Anomalies for Intra Firewall.

Figure 2 shows the results when running the anomaly discovery algorithm for distinct destination and distinct source. Furthermore the results of performance evaluation of anomaly discovery algorithm are shown in Figure 3 The increase in the processing time as the rule database size increases is due to the fact that the complexity of the algorithm is dependent on the number of propositions in the security policy of the firewall. The results indicate that in the worst case, the anomaly detection process takes 18-94 ms of processing time to analyze a security policy of 10-50 rules and anomaly detection process takes 16-53 ms of processing time to analyze a security policy of 10-50 rules. Compare to the other works, the lowest time was achieved to analyze the anomalies in firewall policies and the more the number of the rules, the more evident it take effects. Moreover, the framework proposed is not limited to the number of anomalies in the security policy of firewalls.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

Most of the papers discussed are intended to perform the anomaly detection and removal by using different techniques. All of them consider that the rules are written in predicate like language. The policy rules have very simple attribute like fields but in some cases some firewalls define the rules with time parameters defined within the rules, and the actions performed are restricted to be only accept and deny. One more observation was carried out about the anomalies that almost no paper includes irrelevant anomaly as important one, but we observe that due to the effects of it the rule size is increased enormously.

The Firewall Anomaly Detection System presented in this research work provides a number of techniques for purifying and protecting the firewall policy from rule anomalies. The administrator may use the firewall policy detection system to manage legacy firewall policies without prior analysis of filtering rules.

When an anomaly is detected, users are prompted with proper corrective actions. We intentionally made the tool not to automatically correct the discovered anomaly but rather alarm the user because we think that the administrator should have the final call on policy changes. Finally, we have presented a user-friendly VB.Net-based implementation of Firewall Anomaly Detection System. Using Firewall Anomaly Detection System was shown to be very effective for firewalls in real-life networks. In regards to usability, the tool was able to discover filtering anomalies in rules written by expert network administrators.

The proposed system can solve the problem of frequent changes in the policy rules for both intra and inter firewalls. Our future research

plan includes implementation optimization of intra- and inter-firewall anomaly discovery, online automatic discovery and recovery of anomalies created as a result of the rule editing, rule placement based on firewall performance, self-configurable firewalls, translating low-level filtering rules into high-level textual description and vice versa.

VII. REFERENCES

[1] Ehab S. Al-Shaer and H. Hamed. " Management and translation of filtering security policies". In IEEE International Conference On Communications (ICC '03), 2003.

[2] E. Al-Shaer and H. Hamed. "Firewall Policy Advisor for Anomaly Detection and Rule Editing." IEEE/IFIP Integrated Management Conference (IM'2003), March 2003

[3] E. S. Al-Shaer and H. H. Hamed. "Discovery of policy anomalies in distributed firewalls". In IEEE Infocom, 2004.

[4] Al-Shaer and H. Hamed, "Conflict classification and Analysis of Distributed Firewall policies", IEEE J SEL AREA COMM, 2005

[5] Chotipat Pornavalai and Thawatchai Chomsiri."Firewall Rules Analysis", International Technical Conference on Circuits/Systems, Computers & Comm. (ITC-CSCC 2004), JULY 2004.

[6] Thawatchai Chomsiri, Chotipat Pornavalai: Firewall Rules Analysis, International Conference on Security & Management, SAM 2006, Las Vegas, Nevada, USA, June 26-29, 2006.

[7] Deri Luca and Suin Stefano and Maselli Gaia (2003) Design and implementation of an anomaly detection system: An empirical approach. In Proceedings of Terena TNC .

[8] Y. Bartal, A.J. Mayer, K. Nissim, A. Wool, Firmato: A novel firewall management toolkit, in: Proceedings of the IEEE Symposium on Security and Privacy, 1999

[9] Errin W. Fulp. "Optimization of network firewall policies using ordered sets and directed acyclical graphs". Technical report, Computer Science Department, Wake Forest University, 2004

[10] Yu Gu, Andrew McCallum and Don Towsley. "Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation", Tech. rep., Department of Computer Science, UMASS, Amherst, 2005.

[11] F. Cuppens, N. Cuppens, and J. Garc´ıa. "Detection and removal of firewall misconfiguration". In International conference on Communication, Network and Information Security (CNIS2005), Phoenix, AZ, USA, November 2005. IASTED.

[12] Cuppens, F., Cuppens-Boulahia, N., and Garcia-Alfaro, J. (2005). "Misconfiguration Management of Network Security Components". In Proceedings of the 7th International Symposium on System and Information Security, Sao Paulo, Brazil.

[13] Muhammad Abedin, Syeda Nessa, Latifur Khan, Bhavani Thuraisingham. "Detection and Resolution of Anomalies in Firewall Policy Rules". In Proc. 20th IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec 2006), Springer-Verlag, July 2006, SAP Labs, Sophia Antipolis, France.

[14] T Katic, P Pale. "Optimization of Firewall Rules". Information Technology Interfaces, 2007.

[15] Grout V. , Davies J. and McGinn J. "An Argument for simple embedded ACL optimization", Computer Communications, Volume-30 No-2,

[16] L. Yuan, H. Chen, J. Mai, C. -N. Chuah, Z. Su, P. Mohapatra, Fireman: a toolkit for firewall modeling and analysis, In IEEE Symposium on Security and Privacy, May 2006

[17] S. Acharya, J. Wang, Z. Ge, T. F. Znati, and A. Greenberg.

Traffic-aware firewall optimization strategies. In Proceedings of the International Conference on Communications, 2006.

[18] Subrata Acharya , Jia Wang , Zihui Ge , Taieb Znati , Albert Greenberg, "Simulation Study of Firewalls to Aid Improved Performance", In Proceedings of the 39th annual Symposium on Simulation, 2006.

[19] K. Golnabi, R.K. Min, L. Khan, and E. Al.Shaer, "Analysis of firewall policy rules using data mining techniques", IEEE NOMS 2006, Vancouver, Canada, April 2006.

[20] K. Golnabi, R.K. Min, L. Khan, and E. Al.Shaer, "Analysis of firewall policy rules using data mining techniques", IEEE NOMS 2006, Vancouver, Canada, April 2006.

[21] Indrajeet S. Pabla, "A New Architecture For Conflict-Free Firewall Policy Provisioning**",** A minor thesis submitted at School of Computer Science and Information Technology Royal Melbourne Institute of\Technology, July 19, 2006.

[22] Optimization of firewall performance by Anssi Kolehmainen. In Home Networking, Seminar on Internetworking, Spring 2007 at Helsinki University of Technology.

[23] Salem, O., Vaton, S. and Gravey, A. (2007). A novel approach for anomaly detection over high-speed networks. In, Proceedings of EC2ND.

[24] E.-S. M. El-Alfy and S. Z. Selim, "On optimal firewall rule ordering," in Proceedings of IEEE International Conference on Computer Systems and Applications, 2007.

[25] V. Capretta, B. Stepien, A. Felty and S. Matwin, "Formal Correctness of Conflict Detection for Firewalls", FMSE'07, ACM, Virginia, USA, Nov 2007

[26] Haakon Ringberg , Matthew Roughan , Jennifer Rexford, "The need for simulation in evaluating anomaly detectors", ACM SIGCOMM Computer Communication Review, v.38 n.1, January 2008.

[27] J. Lane Thames, Randal Abler, David Keeling," A distributed firewall and active response architecture providing preemptive protection". In ACM Southeast Regional Conference Proceedings of the 46th Annual Southeast Regional Conference on Network and system security, Auburn, Alabama, 2008.

[28] Ricardo M. Oliveira, Sihyung Lee, and Hyong S. Kim, "Automatic Detection of Firewall Misconfigurations using Firewall and Network Routing Policies", [PFARM'09] IEEE DSN Workshop on Proactive Failure Avoidance, Recovery, and Maintenance (PFARM), Lisbon, Portugal, Jun. 2009.

[29] Ashish Tapdiya, Errin W. Fulp, "Towards Optimal Firewall Rule Ordering Utilizing Directed Acyclical Graphs," icccn, pp.1-6, 2009 Proceedings of 18th International Conference on Computer Communications and Networks, 2009.

[30] A Multi Agent framework for anomalies detection on distributed Firewalls using data mining techniques in 2009 by Kamel Karoui, Fakher Ben Ftima, Henda Ben Ghezala.