Firewall Anomaly Management: A survey

Ameya Hanamsagar, Ninad Jane, Bhagyashree Borate, Aditi Wasvand, S.A. Darade
Department of Computer Engineering, Sinhgad Institute of Technology and Science, Narhe, Pune-411041

ABSTRACT
Firewall secures a private network from intrusions from other networks. The firewall has ACLs (Access Control List) that contain rules used to allow or deny incoming traffic. These rules form the security policy of the firewall. The large size and complexity of modern networks result in large and complex firewall policies. Designing policies for a network of firewalls is a difficult task as a number of cases have to be taken into consideration for access control. Also, a network administrator may want to update the policies in order to replace them with new ones. The process of updating firewall policies is difficult and error prone. In this paper, we provide a structured and comprehensive overview of various techniques in regards to firewall anomaly detection. We briefly describe and compare various known algorithms and tools used to detect and/or resolve the firewall anomalies.

General Terms
Computer Networks, Network Security

Keywords
firewalls, ACL, rules, anomaly, Firewall Policy, Policy conflicts

1. INTRODUCTION
Due to the exponential growth of applications and services made available on the Internet, network security has become an imperative. To include the security requirements covering homogeneous and heterogeneous computing environments, the use of a policy-based approach in network administration has received considerable attention. Firewall is one such element commonly used to deal with network attacks not only in large enterprise networks but also in small home networks. Firewalls protect the private network from external attacks referring to predefined set of rules called ACL's (Access Control Lists). Another important application of firewall is to block specific inbound and/or outbound traffic from/to the network. A firewall ACL is a rule or set of rules which map the firewall policy to some predefined decision(s) which are in terms of *allow* or *deny*. Khummanee et al. [5] defined these firewall policies as a predicate which is defined over source or destination IP address, their port number and/or protocol type.

{Predicate}→ {Decision}

Typically, firewalls are deployed at the boundary of the network thus providing security to the private network. However, there has been a focus on distributed firewalls. The basic idea of distributed firewalls is to make every host in the network into a firewall that filters traffic to and from itself [10].

Modern firewalls have evolved from traditional packet filters to application gateways. Though the developments in firewall technology is important to secure private networks, the complexity and anomalies in firewall policies may limit the effectiveness of firewall security. Al-Shaer and Hamed [2] formally defined types of firewall anomalies as inter-firewall and intra-firewall anomalies. Intra-firewall anomalies are the anomalies present in the same firewall where the same packet may match more than one filtering rule. Inter-firewall anomalies arise when individual firewalls in the same network path perform different actions on the same packet. Typically, for large enterprise networks, the ACL's can go on to be several hundreds of lines large. Thus, the difficulty of management of ACL's (i.e. adding new rule or modifying existing one) significantly increases and an anomaly might be introduced especially in distributed networks. Al-Shaer and Hamed [2] classified the firewall anomalies as Shadowing, Correlation, Redundancy and Generalization, just to mention a few.

The challenges presented in the aforementioned discussion can be resolved by an efficient firewall anomaly management tool which will automatically detect and possibly resolve the anomalies either automatically or manually. In this paper, we provide a brief but effective overview of various firewall anomaly management algorithms and their implementations. We also categorize the methods according to the nature of algorithms, provide the pros and cons of each method and finally discuss the work which needs to be done in the field of firewall anomaly management.

In this paper, we provide a comprehensive survey of pre-existing firewall anomaly detection algorithms and tools. Section II discusses the problem of firewall anomalies in brief while section III presents the types of firewall anomalies. Section IV discusses in detail the existing approaches, their pros and cons. Section V is devoted to the issues and challenges faced in firewall anomaly detection. Opportunities for researchers and concluding remarks are presented in section VI.

2. PROBLEM DEFINITION
Firewalls are the network security elements which isolates an organization's internal network from other external networks. Firewalls can be hardware-based, software-based or a combination of the two. Typically, firewalls use certain rules as defined by network administrator(s) called ACL's (see table 1) to make a decision regarding incoming and outgoing traffic. A lot of research has been done on firewall design [16] which ranges from packet filtering firewalls to application gateways. Though modern firewall systems are efficient in terms of traffic filtering, firewalls contain hundreds or thousands of hundreds of ACL's which increase the complexity of rules giving rise to firewall policy anomalies.

Firewall anomaly has been formally defined in [2] and elaborated in [1] as the firewall rules which conflict each other in the same firewall as well as distributed firewalls. Lot of research has been done on detecting and resolving firewall anomalies which will be discussed in henceforth sections.

Table 1. A Firewall ACL

No	Source IP	Dest. IP	Src. Port	Dest. Port	Protocol	Inbound/Outbound	Action
R1	192.168.2.*	*	*	80	TCP	IN	Deny
R2	192.168.1.2	*	*	*	TCP/UDP	OUT	Allow
R3	*	*	*	22	TCP	IN	Deny

* indicates 'Anything

3. PREFACE TO FIREWALL ANOMALIES

Several types of firewall anomalies have been defined earlier in [2, 3 and 8]. Here, we briefly describe the anomalies:

3.1 Shadowing

Shadowing anomaly occurs when a rule matches a previously defined rule. A basic characteristic of any firewall is that when a packet arrives, firewalls scan the rules sequentially from top to bottom and when a matching rule is encountered, the specified action is taken. Now, if a new rule is inserted which is analogous to a previously defined rule, the firewall will consider the first occurrence which matches i.e. the previously defined rule.

3.2 Correlation

If a rule matches with a previously defined rule, keeping all the fields other than Action similar, correlation anomaly is said to have occurred. In this case the rule which appears first in the rule list is considered while filtering.

3.3 Generalization

An anomaly is said to be a generalization anomaly if a rule is a subset of previously defined rule. Thus, the rules with generalization anomaly increases time as well as space complexity of the firewall.

3.4 Redundancy

A rule is redundant if a previously defined rule exist which perform the same task as that of newly inserted rule. Redundancy increases the processing time of firewalls as firewalls will unnecessarily check the duplicate entries.

The firewall anomalies according to inter and intra firewall anomalies are presented in [2].

4. EXISTING APPROACHES TO ANOMALY MANAGEMENT

Hardware Firewalls from vendors like Cisco [17], Barracuda Networks, Check Point [19], Juniper [18], Lucent Technologies, etc. are available in the market ranging from firewalls for home networks to large enterprise networks. Also, various configuration tools are available to configure the aforementioned vendor-specific firewalls (e.g. Cisco ASDM [13]). However, none of the tools have the capability of addressing the firewall inconsistencies [10].

We have categorized existing work in the fields of constructing firewall queries, verification of firewall rules and detection of firewall rule anomalies.

4.1 Survey on "Constructing Firewall Queries"

Considering the work on constructing firewall queries, A. Mayer et al. [12] proposed a query-based firewall analysis system *Fang* which represents a firewall query by a triple containing source address, destination address and set of services. *Fang* is efficient in the sense that the time required to execute a query is independent of the number of machines in the network. However, the queries are scanned linearly which increases the time complexity. Moreover, it can't process queries over discard traffic which limits the tool to process only over accepts traffic.

Liu and Gouda [7] proposed Structured Firewall Query Language (SFQL) and Firewall Query Theorem as a foundation for developing firewall query processing algorithms. Also, a decision diagram based algorithm was proposed which reduces the processing time drastically as compared to Fang.

4.2 Survey on "Verification of Firewall Rules"

Alex Liu [11] implemented a verification and troubleshooting algorithm which improves the FDD-based (Firewall Decision Diagram) firewall verification algorithm proposed in [14] by additionally comparing the rule defined by the decision path (a path from root firewall to terminal node) and the given property value. The basic idea behind the algorithm is to check and verify whether the specified firewall policy satisfies the given property. Alex Liu [11] formally defined origin rule as the first rule which satisfies decision path by the condition that the subset of packets is a subset of decision path rule for all the rules defined for the firewall. The author also designed a FDD construction algorithm with origin rules marked (i.e. outputs the origin rule of each decision path). The algorithm is experimented on real-life firewall policies as well as synthetic firewall policies. But, the firewall rules have to be fed manually. Furthermore, the algorithm is not implemented on distributed firewalls.

Khummanee et al. [5] while continuing the research on firewall rule verification, proposed a novel firewall rule management policy called Single Domain Decision Firewall (SDD) to completely eliminate rule anomalies. Secondly, they proposed Binary Tree Firewall (BTF), a data structure and algorithm to increase the speed of checking firewall rules. Experiments conducted in [5] show that the time complexity of firewall rule checking was reduced from $O(N^2)$ while following sequential searching to $O(\log_2 N)$.

4.3 Survey on "Detecting Firewall Anomalies"

A lot of work has been done on analyzing firewall policies. Al-Shaer and Hamed [2] classified various inter-firewall and intra-firewall anomalies, prominent of which being Shadowing, Redundancy, Correlation and Generalization. Based on the specified anomalies, a *Firewall Policy Advisor* tool was developed to detect the existing firewall anomalies in the specified network. The disadvantages of the tool include detection of only pair wise firewall anomalies, using a State Machine based comparison which will be efficient only in small networks and inability to respond to anomalies. Furthermore, the tool cannot be used to obtain firewall rules from real-time networks. Instead the rules have to be manually entered into the proposed tool. Also, the searching of the existing rules is linear which increases the rule searching and rule insertion time complexity to O(N).

Bartal et al [4] demonstrated that firewall management can be done at a level of abstraction analogous to modern programming languages in their toolkit *Firmato*. For the purpose, they designed UML like language for specifying global policy rules for the usage of a compiler which converts the high-level language into individual ACL's. However, the disadvantage of the system includes the ability to detect inconsistencies limited to packet filtering firewalls. Also, the system is seen far from being implemented practically in real-time networks.

A. Mayer et al. [10] designed and implemented *Firewall Analyzer* (FA) which analyzes the policies enforced by the firewall in a passive environment i.e. no packets are sent into the network for policy analysis, instead the analysis is performed offline. The system takes configuration files and routing tables (vendor specific) manually as an input and thus simulates firewall behavior offline. The *Firewall Analyzer* is only limited to analyzing the global firewall rules offline and thus it is unable to detect inconsistencies in real-time firewall networks. Moreover, the system is incapable to work over distributed firewalls.

Similar to Firewall Policy Advisor proposed by Al-Shaer and Hamed [2], Lihua Yuan et al. [3] introduced *Fireman*, a toolkit for static analysis in distributed firewalls. *Fireman* also focusses on firewall anomalies like Shadowing, Redundancy, Correlation and Generalization. However, it evaluates firewall configurations as a whole piece rather than just limiting to relation between two firewall rules as in case of Firewall Policy Advisor [2]. The disadvantage of *Fireman* lies in its static analysis of firewalls i.e. it is unable to detect inconsistencies in dynamically changing firewall rule sets which is the case in real-time scenarios.

The algorithm proposed in [3] was improved by using bounded model checking instead of Binary Decision Diagrams to analyze firewall policy configurations in [9]. Alan Jeffrey and Taghrid Samak [9] implemented *Fireman* algorithm [3] using BDD's and it was experimentally found that the algorithm proposed in [9] is efficient than [3]. The experimental results can be found in [9]. However, the algorithm reads configurations in *IPTables* format, which clearly doesn't take inputs from real-time dynamic firewalls. Moreover, the algorithm is classified as NP-Complete but is still not implemented.

Chi-Shih Chao [6] proposed an effective anomaly diagnosis system in which a RAR (Rule Anomaly Relation) tree is created based on ACL's. The system was experimented with seven firewalls with 300 rules each and the system was tested with different situations relating to the number of overlapping rules. The system detect inter- as well as intra-firewall anomalies in a feasible time range. Also, the system suggests network administrators regarding correction in behavior mismatching errors. The author has implemented a GUI prototype of the system which diagnose single as well as multi firewall networks. The disadvantage of the system lies in its incapability of collecting firewall rules (ACL's) from dynamic real-time networks and automatically propagating the consistent ACL's into the network.

Hongxin Hu et al. [8] presented and implemented a policy analysis tool called Firewall Anomaly Management Environment (*FAME*) which too uses binary decision diagrams to represent firewall rule sets. *FAME* introduces a grid representation (matrix based) of the firewall anomalies which provides a better understanding of policy anomalies. The experimental results provided by authors show that *FAME* can resolve 92 percent of the firewall conflicts. However, *FAME* is unable to collect firewall rules in real-time dynamic systems. Moreover, *FAME* currently cannot handle distributed firewalls.

Pedditi et al. [1] presented a design of a new protocol namely *FIEP* (Firewall Information Exchange Protocol) which provides a communication mechanism for two or more firewalls to communicate with each other. The protocol works while considering parent-child relationships. The relationships can be defined by considering the distributed firewall network as a graph with a root node. The relationships are similar to that in a simple tree implementation (parent, child, siblings). The authors simulated the protocol in Java with a static parent-child relationship. The simulation results show that firewalls can successfully communicate the ACL's to their child firewalls on the event of network intrusion. Furthermore, like that of OSPF (Open Shortest Path First), the firewalls can automatically check for inconsistencies in their firewall configuration through message passing. The disadvantages of the protocol include the need to change the existing enterprise hardware which is time consuming and considering economic barriers, difficult to implement in existing networks.

The pros and cons of the aforementioned techniques are summarized in Table 2.

Table 2. Comparison of Firewall Anomaly Management Techniques

Method	**Pros**	**Cons**	**Remark**
A. Construction of firewall queries			
Fang - A Firewall Analysis Engine.[12]	Time required to execute a query is independent of number of machines.	1. More processing time 2. Only processes accept queries 3. Cannot identify troublesome rules	Implemented, but can't process queries over discard traffic
Structured Firewall Query Language and Firewall Query Theorem [7]	Processing time is reduced as compared to *Fang*.	--	Not implemented
B. Verification of firewall rules			
Verification and Troubleshooting algorithm [11]	Algorithm overcomes cons of *Fang*	Rules need be inserted manually.	Not implemented on distributed firewalls.
Single Domain Decision Firewalls [5]	1. SDD is used to completely eliminate firewall anomalies 2. Time complexity of firewall rules checking reduced to $O(\log_2 N)$	1. Space complexity is same as that of a general firewall. 2. Real-world factors not considered.	Not implemented for real-time dynamic networks.
C. Detecting firewall anomalies			
Firewall Policy Advisor [2]	1. Detects the existing firewall anomalies. 2. Prompts network administrator to resolve the detected anomaly.	1. Detection of only pairwise firewall anomalies. 2. Searching of rules is linear	Not implemented for real-time dynamic networks.
Firmato [4]	Firewall rules can be represented at a level of abstraction.	The ability to detect inconsistencies limited to packet filtering firewalls.	Not implemented for real-time dynamic networks.
Firewall Analyzer [10]	Performs the firewall analysis offline.	Only limited to analyzing the global firewall rules offline.	Not implemented for dynamic networks.
Fireman [3]	Eliminates cons of Firewall Policy Advisor [2]	Only Static analysis of firewalls.	Not implemented for dynamic networks.
Bounded Model Checking [9]	1. More efficient as compared to *Fireman* 2. Network Configuration model presented is proved to be NP-Complete Problem	Doesn't take inputs from real-time dynamic firewalls.	Algorithm not implemented.
Rule Anomaly Relation (RAR) [6]	1. Detect intra as well as inter firewall anomalies in feasible time range 2. Diagnose single as well as multi firewall systems.	Cannot collect ACL's from real-time dynamic networks.	Not implemented for real-time dynamic networks.
FAME [8]	1. Matrix representation of firewall anomalies. 2. Can resolve up to 92% of firewall anomalies	Cannot handle distributed firewalls.	Not implemented for real-time dynamic networks.
FIEP [1]	1. Communication between two or more firewalls. 2. Considers parent-child relationships	Need to change hardware/ firmware of existing firewalls.	Not implemented in real-time. Results of simulation are promising.

5. OPEN ISSUES AND CHALLENGES

Although, many methods and systems have been developed in the field of firewall anomaly management, there are still a number of issues and challenges which need to be addressed.

- None of the algorithms and/or tools are implemented in real-time dynamic networks i.e. the aforementioned techniques are incapable to obtain rules (ACL's) from real-time firewalls with dynamically changing rule sets. Pedditi et al. [1] have proposed a protocol which might shed light, but the protocol is still in simulation stage.
- The simulation developed in [15] is unable to detect and resolve inconsistencies that are already defined in the firewalls. The simulation results show that the rules are propagated to parent and/or child firewalls only when an attack is reported by IDS.
- The determination of parent-child relationships as defined in [1] are exceptionally challenging in dynamic networks i.e., it is challenging and equally difficult to determine the parent-child relationships dynamically.
- The collection of ACL's and propagation of consistent ACL's to respective firewalls in real-time still seems far from implementation.
- The techniques discussed earlier in the paper are incapable to detect anomalies in IPV6 based firewall rules.
- As discussed in [4], modern firewalls have advanced features like time-dependency and session-dependency. The detection of inconsistencies having these features, is another challenge.

6. FUTURE WORK

The future work includes practical implementation of inter- and intra-firewall management tool in real-time environment. Also, the ability to handle inconsistencies in distributed networks is also left as a part of future work.

7. CONCLUDING REMARKS

In this paper, we have examined the state-of-the-art in firewall rule anomaly management algorithms and tools. Firstly, we briefly discussed firewalls, their role in network security and introduced firewall anomalies. Further, we discussed various methods categorized into Constructing firewall queries, Verification of firewall rules and Detecting firewall anomalies with their pros and cons. Finally, we outline several issues and challenges while detecting and resolving firewall rule anomalies. The research frontier in firewall anomaly detection lies in implementing the algorithms on real-time dynamic systems which will help network administrators to effectively manage distributed firewalls in real-life scenarios.

8. REFERENCES

[1] Sandeep Reddy Pedditi, Du Zhang, and Chung-E Wang, "FIEP: An Initial Design of A Firewall Information Exchange Protocol," IEEE 14th International Conference on Information Reuse and Integration (IRI), 2013.

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

[3] 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, 2006

[4] Y. Bartal, A.J. Mayer, K. Nissim, A. Wool, "Firmato: A novel firewall management toolkit," ACM Transactions on Computer Systems 22, 2004, pp. 381-420

[5] Suchart Khummanee, Atipong Khumseela and Somnuk Puangpronpitag, "Towards a New Design of Firewall: Anomaly Elimination and Fast Verifying of Firewall Rules," 10th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2013, pp. 93-98

[6] Chi-Shih Chao, "A flexible and feasible anomaly diagnosis system for Internet firewall rules," 13th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2011

[7] A. X. Liu and M. G.Gouda, "Firewall policy queries," IEEE Transactions on Parallel and Distributed Systems (TPDS), 20(6), pp. 766–777, 2009

[8] Hongxin Hu, Gail-Joon Ahn and Ketan Kulkarni, "Detecting and Resolving Firewall Policy Anomalies," IEEE Transactions on Dependable and Secure Computing, vol. 9, issue 3, pp. 318-331

[9] Alan Jeffrey and Taghrid Samak, "Model Checking Firewall Policy Configurations," IEEE International Symposium on Policies for Distributed Systems and Networks, 2009, pp. 60-67

[10] A. Mayer, A. Wool and E. Ziskind, "Offline firewall analysis," International Journal of Information Security 5 (3), 2005, pp. 125–144

[11] Alex X. Liu, "Firewall policy verification and troubleshooting," The International Journal of Computer and Telecommunications Networking, Vol 53 Issue 16, 2009, pp. 2800-2809

[12] A. Mayer, A. Wool, and E. Ziskind, "Fang: A Firewall Analysis Engine," Proc. IEEE Symp. Security and Privacy, pp. 177-189, 2000.

[13] Cisco ASA Series Firewall ASDM Configuration Guide, Cisco Systems Inc., updated March 31, 2014

[14] A. X. Liu, "Formal Verification of Firewall Policies," IEEE International Conference in Communications, 2008, pp. 1494 - 1498.

[15] S. R. Pedditi, "An initial design of firewall information exchange protocol (FIEP)," MS Degree Project Report, Department of Computer Science, California State University, Sacramento, May 2012.

[16] Keromytis, A. D. and Prevelakis, V., "Designing Firewalls: A survey", in C.Douligeris and D.N. Serpanos, "In Network Security: Current Status and Future Directions", Wiley - IEEE Press, 2007

[17] Cisco Security Appliance Command Line Configuration Guide, Cisco Systems Inc., 2009

[18] Juniper Netscreen Series Security Systems, Juniper Networks Inc., Dec 2011

[19] Check Point Threat Prevention Appliances, Check Point Software Technologies, Ltd., 2012.