A Mining Based Inference Handling Approach for Message Blocking Filterset Policies of OSN User Wall

L.PRASANNA LAKSHMI
M.tech in Software Engineering
Aurora's Technological & Research Institute,
parvathapur, uppal, Hyderabad-500039

T.PADMAJA
ASSOCIATE PROFESSOR in CSE DEPARTMENT
Aurora's Technological & Research Institute,
parvathapur, uppal, Hyderabad-500039

Abstract: Firewall is in reality the core technology of network security and defense. The management of firewall rules however has proven to be cumbersome, frequently causes errors, is very costly, and not applicable for organizations having large networks. The rules that govern the firewall are generated as per user requirements and need to be updated according to the ever-changing requirements of network traffic characteristics, environment variables and the market demands.

In this paper, one of the main problem of usability of firewall rules in updating, organizing, and reflecting the efficiency of the existing characteristics of network traffics is approached. A set of methods, rules, and algorithms for firewall policy management to generate a new set of efficient firewall policy rules are presented. (1) Data mining of the network traffic log based on its frequency to set up effective policy rules for the firewall, (2) Filtering-Rule Generalization (FRG) to minimize the number of policy rules by generalization, and (3) A new approach to find and replace any decaying rule with a set of few dominant rules. Anomaly detection based on mining technique facilitates network security administrators to automatically review and update the rules. We developed a prototype system, discovered many hidden and undetectable anomalies, and demonstrated usefulness of our methodologies.

Keywords- firewall; data mining; network security; policy

I. Introduction

Firewall technology in the internet-networking world, is the first layer to secure the network and provide automated security against network vulnerabilities, threats, and attacks. The performance of the firewall is vulnerable to the inconsistencies in firewall policy management. Preset rules written extensively to enable the firewall to monitor, manage, filter and thus allow or deny the incoming or outgoing traffic and block any unwanted traffic traversing in out a secure network. The rules usefulness, update status, and efficiency depend on the current attributes and volume of traffic.

Let's consider an case where a network traffic trend shows few rules to be outdated and prompts one to consider deleting, summing or re-structuring the rules to optimize firewall policy and security though the server and network logs may authenticate the firewall policy rules to be up-to-date w.r.t. present network. Consider another issue where data mining over network traffic log, reveals a violation against the current firewall polices such as transferring plain text over a secure IPSec/VPN link or allowing traffic that may not be permitted by the other downstream devices behind firewalls. The firewall policy rules are designed as per user requirements and require frequent updating, tuning, and validating. As the number of filtering rules increases enormously to suit the increasing business and consumer requirements, manually managing the rules becomes very difficult, time consuming and expensive which is unsuitable to a business model even for large networking companies. Developing efficient firewall policy management techniques and tools will enable network administrator to optimize and validate firewall rules automatically and easily.

Current research combining traditional and classical research is employing innovative technologies like data mining of traffic logs for analysis and management of policy rules. This minimizes the number of rules, generates an efficient rule-set that monitors the network traffic in real time and provides policy update capability and security in scenarios such as a mail server detecting a huge incoming spam-mail from a host.

 All Rights Reserved © 2014 IJSETR

In this study, a set of techniques and algorithms for analyzing and managing firewall policy rules are presented: (1) Mining firewall Log using Frequency (MLF) a data mining technique to mine the network traffic log w.r.t. its frequency, (2) Filtering-Rule Generalization (FRG) to minimize by generalization the number of policy rules, and (3) A technique to find and replace decaying rule with dominant rule, for a new set of efficient firewall policy rules. An automated tool discerns frequent traffic behaviors and filtering rules, applying Log Mining Frequency and Filtering-Rule Generalization, to identify and deliver an effective and anomaly-free firewall policy rules.

The significant part of current research in policy anomaly detection is mostly about static analysis of the existing policy configuration. Our methodology of anomaly detection uses traffic and log mining techniques to establish other unknown major policy properties undetectable with previous approaches that analyze only the firewall rules.

II. Related Work

Firewall and policy based security management has generated immense research and various models proposed on the usability of firewall policy rules against security threats. The data mining approach of our study develops firewall policy rules that are generalized via a generalization model and applies an anomaly discovery algorithm to the rules. The two main fields of study in our work are (1) data mining and (2) firewall policy rule anomaly detection that are elaborated in the following section.

A. Data Mining Techniques

A predictive model using data mining technique is used to analyze huge quantity of information or datasets that provides insight into specific patterns or styles and facilitates decision capability and further analysis. Two approaches Decision Tree [11] and Association Rule Mining (ARM) [13] are followed.

In the two approaches, Decision Tree finds a function depicted as a decision tree where each node in the tree tests as an attribute, each branch corresponds to a value comparison and each leaf node assigns classification. The shortcoming of this approach is handling of continuous attributes, growing tree and computational inefficiency. The other approach Association Rule Mining algorithm examines the space for all probable patterns to define rules that meet the user-specified support and confidence thresholds. One example of an association rule algorithm is the Apriori algorithm and a comprehensive survey of for Association Rule Mining [18] designed by Srikant and Agrawal [15]. Lee [32] and Mahoney [33] pioneered a recent, significant progress in the technology of intrusion and anomaly detection using data mining techniques, particularly Association Rule Mining technique.

Association Rule Mining (ARM) is a nontrivial process of identifying valid, potentially useful, and ultimately desirable pattern (rule) in data. An Apriori algorithm generates rules in huge numbers, however most of the rules are found to be useless or impractical for analysis and some rules may have no further impact on firewall action such as rules repeated and rules that have zero impact on the right hand side which are filtered during further analysis. The major objective is to generalize specific and unique rules to more general rules. Generalizing using Association Rule Mining algorithms specifically identifies styles that occur in the original form all over the database. However, the major drawback of ARM and many other association rule-mining algorithms such as Apriori algorithm, is that only patterns in database similar to the query pattern support the query and not the databases with various minor variations between otherwise similar patterns.

B. Firewall Policy Rules and Anomaly Detection

The current study focuses on four areas i.e., (1) data mining of packet filtering rules from firewall log file, (2) aggregation of packet filtering rules, (3) anomaly detection of firewall policy rules in study of firewall policy rules and data mining of filtering packets, and (4) discovery of decaying or dominant firewall policy rules through data mining.

Several models have been proposed for filtering rules. A model by name ordered binary decision diagram is utilized to optimize packet classification in [11]. A different model using tuple space is designed [16] that combines a set of filters in one tuple and stored in a hash table. Another model in [17] uses bucket filters indexed by search trees and multi-dimensional binary trees to model filters [15]. In [16] a geometric model is used to represent 2-tuple filtering rules. These models designed specifically to optimize packet classification in high-speed networks are too complex to be used for firewall policy analysis. From the analysis in [2] tree-based model was uncomplicated and effective in packet filtering. In [10] diagrams and pre-processing of firewall rules were used to resolve rules overlapped and compact firewall policy rules and however it cannot be used for anomaly detection. A lot of papers highlight

 All Rights Reserved © 2014 IJSETR

filtering performance [9, 10] and some associated work [7, 11] deals with rule combination in filtering policies. Further, some approaches [3, 10] project using a high-level language to define rules to avoid rule anomalies but is not practical.

In our study, we implement a tree-based filtering representation to discover anomaly using an algorithm similar to the work performed in [1]. In [1, 2], we develop a tool called "Policy Advisor" for evaluating firewall polices and has been extensively used in academic and industrial communities. This tool does not examine network traffic or device logs, which results in revealing other serious and non-systematic anomalies such as traffic blocked to existing legitimate services, permitting or blocking traffic to non-existing services. In this paper, we will present traffic data-mining techniques that go beyond static analysis of these polices to find other unknown characteristics, or embedded anomaly that cannot be detected with current anomaly detection methods and firewall policy rules.

III. Mining Firewall Log File

This section provides the complete process of mining a log file for generating policy rules and its architecture as shown in Fig. 1.

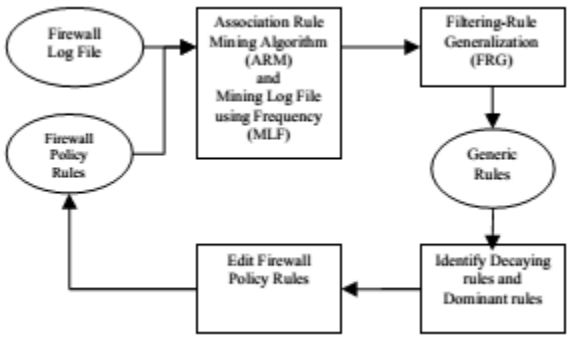

Figure 1. Flow Diagram of Our Approaches

The process consists of the following five iterative components in sequential order: (1) to assess and create a preliminary set of firewall policy rules, gather firewall log raw data, and acquire attribute data for data mining. (2) to use ARM and MFL techniques conduct data mining along generalization to uncover new rules to update policy existing rules of the firewall. (3) to recognize decaying and dominant rules, (4) to generate a generalized and updated firewall policy rules which is anomaly free.

A. Analysis of Firewall Policy Rules

An organization's network firewall has an initial set of policy rules determined by the administrator of network security with respect to the organization's security policy. A rule is written to accept or deny a packet based on its attributes. Ex: A rule can be written to allow a packet of Hyper Text Transport Protocol (HTTP) to transfer HTML documents via internet protocol on port 80 with a particular Source IP Address.

The attributes in a rule can be of maximum seven and the format is follows "<direction> <protocol> <source IP> <source port> <destination IP> <destination port> <action>". The first attribute defines the packet direction in both in-coming and out-going. The second attribute defines the protocol of packet such as TCP or UDP. The four subsequent attributes are two pairs of IP address and their port for source and destination packets. The last attribute defines the action upon a packet for being accepted or denied by the firewall.

An attribute may also not be actually present in a policy rule of a firewall to imply all or any. It can be similar to the TCP port range from 1024 to 65535 or the IP address range in a local area network under mask.

B. Association Rule Mining (ARM)

This part deals with the attributes extraction from the log file data collected for performing Apriori analysis. The attributes extracted have protocol (TCP or UDP), directions (incoming or outgoing), source IP, destination IP, source port, destination port, action (accept or deny) and further defined as nominal to avoid functional importance to its values.

In ARM, an association rule is of the type X => Y where X and Y represent disjoint conjunctions of attribute-value pairs. The confidence of the rule is the conditional probability of Y given X, Pr(Y|X), and the support, of the rule is the prior probability of X and Y, Pr(X and Y) where probability is assumed to be the observed frequency in the data set. The conventional rule-mining problem can be explained as follows. If given a data set of transactions and minimal confidence as well as support thresholds and requirement to find all the rules of association with confidence and support above the specified thresholds, ARM gives the maximum item-set and excellent rules from the log dataset. With the Confidence value set to 1 for 100 percent confidence, input data of small values for minimum support, (e.g., 0.001), the algorithm would output more rules

 All Rights Reserved © 2014 IJSETR

compared to large value (e.g., 0.9) input whi generates output of lesser rules. Thus working wi minimum support and also minimum confidenc ARM algorithm finds the largest item set and al discovers the best rules, specifically rules with acti field (i.e., =Y) since rules with no action have ze effect on firewall policy rule.

C. Mining firewall Log using Frequency (MLF)

Besides ARM, we have used MLF an uncomplicated and effective data-mining algorithm applying simple frequency-mining firewall log and algorithm. The MLF algorithm reads every line of firewall log file, extracts the attributes from each log record, counts its occurrence, and outputs the count for each unique combination of these attribute- values. As each rule identified is recorded, summed, and used in its probability and statistical processing. Similarly every log record of a packet in firewall log file is processed to form a primitive rule or a particular firewall policy rule with all its attributes instantiated, specific to its value observed in the firewall log file. Next we determine and create the associate primitive firewall rule using instantiated attributes set of (1) direction like incoming or outgoing packet, (2) protocol like TCP or UDP, (3) source IP, (4) source port, (5) destination IP, (6) destination port, and (7) the action to accept or deny for a packet satisfying these attributes.

The MLF algorithm is as below,

Mining firewall Log using Frequency (MLF) Algorithm

Input: Firewall Log file
Output: Unique Rules and their Frequency

1. Packet# ← 0
2. **FOR EACH** Line in Firewall Log file
3.FirewallRule[Packet#] ← Protocol, Direction, SRC-IP, SRC-Port, DST-IP, DST-Port, Action
4.Increment Packet#
5. END FOR
6. FOR EACH i WHERE 0 < i < Packet#
7. Frequcney ! 0
8.FOR EACH j WHERE i < j < Packet#
9.IF FirewallRule[i] = FirewallRule[j]
10.Increment Frequency
11.END IF
12.END FOR
13. IF FirewallRule[i] NBT discovered previously
14. Write FirewallRule[i] and Frequency
15. ELSE
16. Continue
17. END IF

Mining firewall Log using Frequency (MLF) Algorithm

D. Filtering-Rule Generalization (FRG)

FRG is an aggregation algorithm for generating least number of firewall policy rules while achieving high performance combined with anomaly detection. Filtering Rule Generalization creates a decision tree where a level or a branch is equivalent to one of the attributes belonging to a primitive rule extracted from log record of a packet, which forms the basis for a rule to be generated. The rules drawn from the log file of firewall are assessed and branched. They are further analyzed to be grouped and categorized by collecting the common fields to generate the superset rules that are equivalent to unique rules.

Let us consider an example: A few rules with similar fields with the exception of source port field are aggregated by combining all the matching fields, that is, aggregation by considering all the source ports as a range of these source ports to generate one generalized rule. The algorithm is extended incrementally adding new rules from the firewall log file from varied period with time stamp of a packet to generate further general rules. For further explanation, let us consider a case with final action as "DENY" where for the purpose of generalization the action of "ACCEPT" is fundamentally same as the "DENY" action.

Linked with each rule are seven attributes with the following format of "<action> <protocol> <direction> <destination-port> <source-port> <source-IP> <destination-IP>". A primitive rule starts with the Action parameter and protocol with binary decision of whether it is UDP or TCP. Next node in the tree describes direction of the packet (either INPUT or OUTPUT) with a destination port and finally a decision made at each destination port. However, these attributes aggregation process is binary in action, protocol, and direction, and requires a range for destination-port, source-port, source-IP, and destination-IP.

The current data mining strategy to extract and collect IP addresses or its port numbers from the

 All Rights Reserved © 2014 IJSETR

firewall log is limited to the observed and actual data in the log file and the chance of observing all the IP addresses and all the port numbers in the log file is very less requiring further aggregation for the corresponding range of IP addresses or port numbers or "ANY" OR "*" to cover all of the specified range for subsequent firewall rules. In the ideal scenario the rules resulting from such a data mining approach is most probably a subset of the firewall rules. The real scenario is of assessing, comparing and classifying the difference between the aggregated rules generated from the network firewall log files with the existing firewall policy rules, network data modeling for a schematic view of a conceptual network data space and identifying any anomalies in the firewall policy rules. Thus, enhance the schematic view to the logical view of network data space for projecting the firewall and firewall policy rules.

A comparable approach can be used for the aggregation of the port numbers. One approach is to set a tier of port numbers such as 1-1023, 1024-1999, 2000-2999, and so on, to aggregate any port number falling into one of the tiers to be its range of the tier (for example, 2000:2999) with its subrange incrementing in the unit of 1000, then incrementing in the unit of 10000, and so on. The final port ranges for our experiment has been (1) two tiers of 1-1023 and 1024-65535, and (2) one range of port 1-65535, to be further simplified. The final range is large but much simpler for the prototype implementation to aggregate to two distinct port numbers to be "ANY". Further a predefined table of the important or critical port could be used to simplify the process of aggregation, instead of checking all the ports available (from port 1 to port 65535 where the well-known or reserved ports are port 1 through port 1023).

Generalization Algorithm Pseudo Code
Input: *MM*
Output: The Tree

```
1. FOR EACH attribute € {User message request}
2. IF attribute doesn't exist in the tree AND DST Policy set
'3. Create new branch with attribute
4,   ELSE
5,       If DST policy set exists in the table
6,            if DST Policy set doesn't exist in Tree
7            Create new branch with DST Policy set
8.            ELSE
9,            Follow existing branch with  DST Policy set
10,           ELSE IF DST Policy set does not exist in Table
11,           IF DST Policy set doesn't exist in Tree
12.          Create new branch with DST Policy set
13,           ELSE
14,                Determine range of D.ST Policy set
15,                Update existing branch with DST Policy set range
16,           END IF            '
17,      END IF
18,      IF SRC Policy set exists in the table
19,           IF SRC policy set does not exists in Tree
20,                Create new  branch with  SRC Policy set
21,           ELSE
22,                Follow-existing branch with SRC policy set
23,      ELSE IF SRC Policy set does not exist in Table
24,           IF SRC doesn't exist in Tree
25,                Create new branch with SRC Policy set
26,              ELSE
,27,          Determine Range of SRC Policy set
28,                UpdateSRC Fort Range
29,           END IF
30,              END IF
31,      IF SRC USERWALL does not exist in Tree
32,           Create new branch with SRC USERWALL
33,              ELSE
34,           Determine Superset of SRC IF
35,           update existing branch With SRC USERWALL Superset
36,      END IF
37.  If DST USERWALL does not exist in Tree
38,           Create new branch with DST USERWALL
39,      ELSE
40,           Determine Superset of DST IF
41.  Update existing branch with DST USERWALL Superset
42. END IF
43.    END IF
44, END LOOP
45. Print the generated Policy (Each path of the tree will,
produce a new generated Policy).
```

Figure: Filtering Policy Generalization (FPG) Algorithm

It takes as input the policies and generalizes them. Let us consider the following two policies where the algorithm loops over each attribute field. (attributes are seen in line #1).

Policy Ordering

The general order of the policies can substantially affect the outcome of the filter policies and the performance also. This part of the process is however not to generate a set of the order of the policy based on its generalization. A policy if it is more specific should be used immediately in its policies to distinguish itself from other general policies. The underlying assumption here is that a typical policy (i.e. a policy that is more specific) should be used first and should not be used at all if it results in conflicts among the policies. The iterative approach of revising the order is to give a score to each policy

 All Rights Reserved © 2014 IJSETR

depending on its generalization and sort the policies in an ascending order of their scores.

We may score a generalized policy depending on its degree of the range of IP address or of port number, to add 1 point for each "*" of source or destination IP address, and 1 point for "ANY" for port. For example, Policy 3 in Fig. 7 gets 3 points whereas Policy 6 gets 2 points. Thus Policy 3 will be placed ahead of Policy 6 as Policy 3 and Policy 6 have the same attributes except for source IP address. As a result of this ordering, the Policy 3 and 4 will be reversed for the Unwanted activity detection.

Unwanted activity Detection

A policy for centralized filter set policies, is essential for packet filtering process to determine if its filtering policies are disjoint or not, or worse yet in conflict. It is very common that some of the filtering policies are not connected and any ordering may generate varied and unwanted results, which further generate an incorrect policy for filter set policies.

The policy for filter set policies will report Unwanted activity when (1) two or more policies match the same packet or (2) if a Policy never matches any packet that is filtered by the Filter set policies. As the number of policies increases, it tends to include more Unwanted results. Thus, the detection and removal of such Unwanted in a timely manner to at least keep Filter set policies free of Unwanted activity is very important. The Filter set policies Unwanted can be classified into 4 categories:

1) Shadowing Unwanted activity: A Policy is shadowed when a Policy matches packets that have been matched by a previous policy, resulting in the shadowed Policy never being activated.

2) Correlation Unwanted activity: Two policies are correlated if they have different filtering actions and the packets matched by the first Policy are same matches matched by the second Policy.

3) Generalization unwanted activity: A Policy is a generalization of a preceding Policy if both have different actions, and if the first Policy can match all the packets matched by the second Policy.

4) Redundancy unwanted activity: A redundant Policy executes the same tasks on the same packets as any another Policy, such that if the redundant Policy is removed, the security policy will not be disturbed.

Further, our log based mining approach can ascertain the following non-systematic misconfiguration Unwanted activity.

5) Blocking existing service Unwanted activity: A common misconfiguration case is blocking a legitimate traffic from a trusted network of an "existing" service. The reason may be mis-configuration of port number or deleting by mistake the exception Policy that allows the traffic from the trusted network. Simple detection of such Unwanted activity can be done while mining the log file, where an analyst would know that the traffic from a trusted network or existing service/port is being denied access.

IV. Conclusion

Firewall policy rules are a principal element of a network security system and play vital role in management of the network and security infrastructure of an organization. The supervision of policy rules is an important task and various tools and techniques have been developed to test anomaly detection and rule editing utilizing known set of existing policy rules. The limitation of firewall based on assumptions is the rules are mostly static and not dynamic enough to change according to the network.

In our paper, we proposed the innovative process of managing firewall policy rules, composed of anomaly detection, generalization, policy update using Association Rule Mining and frequency-based techniques. The summary of advantages are; (1) to be able to adapt to the current network traffic trends by mining firewall log data files and update firewall policy rules in real time, (2) to provide a tool to assess the traffic patterns that generate datasets for further analysis and anomaly detection, both visible and hidden for decision making, (3) to apply several data mining techniques to work with both discrete and continuous attributes without loss of operational efficiency and flexibility, and (4) to prove the benefit of data mining based algorithms in terms of feasibility, accuracy and effectiveness while firewall rules and log dataset gets larger in size and variation. The mining based anomaly detection approach uncovers not only these four types of anomalies detected by analyzing the firewall policy rules [1] but even anomalies not detected by analyzing the firewall policy rules. In conclusion, data mining is practical, effective, viable and significant in firewall policy rules analysis and optimization in real time.

In future research, contemporary uncomplicated model of single firewall can be extended to contain., Efficient algorithms to manage massive volumes of

 All Rights Reserved © 2014 IJSETR

log files with data mining techniques. Further investigation and analysis on time-dependent statistical behavior of network traffic and policy rules.

References

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

[2] Ehab Al-Shaer and Hazem Hamed, "Discovery of Policy Anomalies in Distributed Firewalls" in Proc. of IEEE INF0C0M'04, vol. 23, no. 1, March 2004 pp. 2605-2616.

[3] Y. Bartal., A. Mayer, K. Nissim and A. Wool. “Firmato: A Novel Firewall Management Toolkit.” Proceedings of 1999 IEEE Symposium on Security and Privacy, May 1999.

[4] D. Chapman and E. Zwicky. Building Internet Firewalls, Second Edition, Orielly & Associates Inc., 2000.

[5] W. Cheswick and S. Belovin. Firewalls and Internet Security, Addison- Wesley, 1995.

[6] S. Cobb. “ICSA Firewall Policy Guide v2.0.” NCSA Security White Paper Series, 1997.

[7] D. Eppstein and S. Muthukrishnan. “Internet Packet Filter Management and Rectangle Geometry.” Proceedings of 12th0Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), January 2001.

[8] P. Eronen and J. Zitting. “An Expert System for Analyzing Firewall Rules.” Proceedings of 6th0Nordic Workshop on Secure IT-Systems (NordSec 2001), November 2001.

[9] Z. Fu, F. Wu, H. Huang, K. Loh, F. Gong, I. Baldine and C. Xu. “IPSec/VPN Security Policy: Correctness, Conflict Detection and Resolution.” Proceedings of Policy’2001 Workshop, January 2001.

[10] J. Guttman. “Filtering Posture: Local Enforcement for Global Policies.” Proceedings of 1997 IEEE Symposium on security and Privacy, May 1997.

[11] Mitchell, T.M., Machine Learning. 1997, Sydney: McGraw-Hill.

[12] Salzberg, S.L., Book Review: C4.5: Programs for Machine Learning by J.Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. Machine

[13] Agrawal, R., T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. in Proceedings of the 1993 Webb, G.I., Association Rules, in Handbook of Data Mining, N. Ye, Editor, Lawrence Erlbaum: To appear.

[14] Agrawal, R., et al., Fast Discovery of Association Rules, in Advances in knowledge Discovery and Data Mining, U.M. Fayyad, et al., Editors. 1996, AAAI Press: Menlo Park, CA. p. 307-328.

[15] Srikant, R., Q. Vu, and R. Agrawal. Mining Association Rules with Item Constraints. in Proceedings of the 3rd International Conference on Knowledge Discovery in Databases and Data Mining. 1997. Newport Beach, California.

[16] Piatetsky-Shapiro, G., Discovery, analysis, and presentation of strong rules. Knowledge Discovery in Databases, 1991: p. 229-248.

[17] Webb, G.I. Discovering Associations with Numeric Variables. In Proceedings of the International Conference on Knowledge Discovery and Data Mining. 2001: ACM Press.

[18] Agrawal, R. and R. Srikant. Fast Algorithms for Mining Association Rules. in Proceedings for the 20th Int. Conf. Very Large Data Bases. 1994.

[19] Borgelt, C., Apriori (Computer Software). http://fuzzy.cs.unimagdeburg de/~borgelt/ Accessed via the Internet 17/06/2002.

[20] B. Hari, S. Suri and G. Parnlkar. ‘‘Detecting and Resolving Packet Filter Conflicts.” Proceedings of IEEE INFOCOM'00, March 2000.

[21] S. Hazelhusrt. ‘‘Algorithms for Analyzing Firewall and Router Access Lists.’’ Technical Report TR-WitsCS-1999, Department of Computer Science, University of the Witwatersrand, South Africa, July 1999.

[22] S. Ioannidis, A. Keromytis, S Bellovin and J. Smith. ‘‘Implementing a Distributed Firewall.’’ Proceedings of 7th0ACM Conference on Computer and Comminications Security (CCS'00), November 2000.

 All Rights Reserved © 2014 IJSETR

[23] E. Lupu and M. Sloman. ‘‘Model-Based Tool Assistance for Packet- Filter Design.’’ Proceedings of Workshop on Policies for Distributed Systems and Networks (POLICY’2001), January 2001.

[24] I. Lck. C. Schfer and H. Krumm. ‘‘Conflict Analysis for Management Policies.’’ Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM’1997), May 1997.

[25] A. Mayer, A. Wool and E. Ziskind. ‘‘Fang: A Firewall Analysis Engine.’’ Proceedings of 2000 IEEE Symposium on Security and Privacy, May 2000.

[26] L. Qiu, G. Varghese, and S. Suri. ‘‘Fast Firewall Implementations for Software and Hardware-based Routers.’’ Proceedings of 9th0 International Conference on Network Protocols (ICNP’2001), November 2001.

[27] V. Srinivasan, S. Suri and G. Varghese. “Packet Classification Using Tuple Space Search.” Computer ACM SIGCOMM Communication Review, October 1999.

[28] J. Wack, K. Cutler and J. Pole. “Guidelines on Firewalls and Firewall Policy.” NIST Recommendations, SP 800-41, January 2002.

[29] T. Woo. “A Modular Approach to Packet Classification: Algorithms and Results.” Proceedings of IEEE INFOCOM’00, March 2000.

[30] A. Wool. “Architecting the Lumeta Firewall Analyzer.” Proceedings of 10th0USENIXSecurity Symposium, August 2001.

[31] “Cisco Secure Policy Manager 2.3 Data Sheet.” http://www.cisco.com/ public/cc/pd/sqsw/sqppmn/prodlit/spmgr ds.pdf

[32] W. Lee, “A Data Mining Framework for Constructing Features and Models for Intrusion Detection”, Ph.D. Dissertation, Columbia Univeristy, 1999.

 All Rights Reserved © 2014 IJSETR