A Matrix Model for Designing and Implementing Multi-firewall Environments

Loye L. Ray

Department of Cyber Security and Information Assurance, University of Maryland University College, 3501 University Blvd East, Adelphi, MD 20783

e-mail: loye.ray@faculty.umuc.edu

Abstract- Firewalls are core elements in network security, the effectiveness of firewall security is dependent on configuring the firewall policy correctly. A firewall policy describes the access that will be permitted or denied from the trusted network. In a corporate network several firewalls are setup and administrated by different individuals. The consistency between those firewall policies is crucial to corporate network security. However, the managing of these has become a complex and error-prone task. Bad configurations may cause serious security breaches and network vulnerabilities. In particular, conflicting filtering rules lead to block legitimate traffic or to accept unwanted packets. In this paper, we provide a firewall policy matrix for helping guide firewall administrators and designers overcome differences in interpreting firewall policies. The matrix presents how each firewall policy allows or denies traffic through the various firewalls in a distributive environment. The model was also tested in a university environment.

Keywords- Firewall policy; Multi-firewall environments; Firewall design; Firewall management; Inter-policy Errors.

1. Introduction

The firewall has become the primary security device for protecting networks connected to the Internet. However, to be effective, a firewall must be configured properly [1][13][18]. Firewall rule sets usually are not commented or described anywhere. This causes the administrator to rely on the firewall policy to be correct. If the written firewall policy doesn’t state what a firewall must do, implementing the policy relies on the interpretation of firewall administrator [18]. According to Hamed and Al-Shaer [11], even expert firewall administrators can make serious mistakes in configuring firewall policy of large networks. Additionally, the lack of a global view a network. Either may allow illegitimate traffic to pass or deny legitimate traffic through the firewall [15]. Lastly, design or deployment errors can lead to the same problems as with conflicts.

Large universities usually have several firewalls distributed across a campus network to protect various areas. These firewalls may be located at the point of entry between various campus network areas (trusted) and the Internet (untrusted) to serve as the first line of defense against of the multi-firewall environment configuration is ripe for misconfigurations that cause errors and major vulnerabilities [23]. Therefore, unawareness of policy conflicts and errors can significantly increase the risk of policy inconsistency thus increasing network vulnerability [11].

There are many challenges confronting the correctness and consistency of firewall policy configuration in enterprise networks [11][24]. One challenge is that large networks usually have several firewalls scattered across the network each with their own firewall policy. This makes designing and deploying an effective firewall policy difficult. Another is rule conflict across multi-firewalls (inter-policy) in unauthorized or malicious intruders [7]. The interaction between different firewall policies can introduce inconsistent rule matching between two firewalls [11][20]. This can result in illegitimate traffic to be allowed in the network which can lead to serious security threats such as denial of service attacks [11].

Policy conflicts can occur from a rule misconfiguration within a single policy or between security policies in different firewalls. Inter-policy conflicts are similar except the rules conflict across multiple firewalls. For a successful firewall deployment is dependent on a through understanding and identification of firewall rules.

An error in a firewall policy can be a wrong definition of being legitimate or illegitimate for some packets. This can lead to a firewall either accepting some malicious packets, which consequently creating security holes in the firewall, or discard some legitimate packets, which consequently disrupt normal business [3][15][20]. Either case could cause irreparable and tragic consequences. Given the importance of firewalls, such errors are not acceptable. Unfortunately, it has been observed that many firewalls are poorly designed and have many errors in their policies [13][20]. Therefore, how one can design firewall policies correctly is an important issue.

We categorize errors as specification-induced and design-induced errors. Specification-induced errors are caused by the inherent ambiguities of informal requirement specifications, especially if the requirement specification is written in a natural language [14]. Design-induced errors are caused by technical incapacity of individual firewall administrators and designers. Different administrators and designers may have different understandings of the same firewall requirement specification [1][2][14]. They may also exhibit different technical strengths and weaknesses in designing and managing firewalls.

Therefore, the effectiveness of firewall security is dependent on providing policy management techniques and tools that enable firewall administrators to simplify the writing of firewall policy across distributed environments. We firmly believe in the need for using a firewall matrix model and formal methods to reason about firewall policy in order to detect and identify policy errors. The goal of this article is to provide a detailed model for firewall administrators to use in deploying firewall policies.

In this paper, we define a formal firewall policy matrix model to transform firewall rule relations and their filter representation. The proposed matrix model is simple and visually comprehensible. We also propose using this matrix model (we have tested) to illustrate the inter-firewall relationships of traffic passing through the distributive network. We finally use the matrix model to reduce complexities in identifying various networks using virtual local area networks (VLANs). The matrix model was tested at Towson University using their multi-firewall network. Also we used the matrix model to redesign the firewall rules and construct an effective VLAN structure to simply management of the network.

2. Basic Firewall Background

A simple firewall is a network device that controls the flow of packets across the network based on a specific security policy [11]. A firewall security policy is a list of ordered rules that define the actions performed on network packets based on specific filtering actions [5][12]. Each rule is composed of filtering fields such as source IP address, source port, destination IP address, destination port and action field (Table 1). These fields correspond to the possible values of the corresponding fields in the actual traffic that matches the rule. The source IP address field represents one possible IP of the network address space where the source device is connected. The destination IP address field represents one possible IP that the device is trying to connect to. The protocol represents many different protocols (TCP, UDP, ICMP, etc.) a device may use as a service in order to communicate with. The service represents the destination port number where the service is located [18]. Basically, firewalls filter actions by either permit a packet to flow through (allow) or block a packet (deny). Packets are allowed or denied by a specific rule if the packet header information matches all the network fields of this rule. Otherwise, the following rule is examined and the process is repeated until a matching rule is found or the default policy action is performed [4]. The default policy action in all firewalls is to deny all traffic at the end of the rules. This same example was used in implementing the firewall matrix at Towson University. This model doesn't involve or consider more sophisticated firewall features such as packet tagging, stateful inspection, variables or control flow. These were not used in our implementation of the model at Towson University (TU). The reason was they were not used.

Table 1. Firewall rule example

Protocol	Source IP	Port	Destination IP	Port	Action
TCP	192.168.1.1	any	64.20.1.55	21	allow

The manual design of firewall rules from a security policy is complicated and prone to errors [15]. This is because of a person's interpretation of what the policy states and how it applies to the network. Thus, each person's knowledge and experience in designing firewall rules creates errors [2]. These errors can allow intruders to access or attack the network. This implies that intruders could access sensitive information stored on a database or disrupt networks services with a Denial of Service attack.

3. Firewall Rule Conflicts

Conflicts between firewall policies can come from several different firewalls (inter-firewall policy). In multiple firewall environments, firewalls may be individually configured without realizing the relationships between packets traveling from one firewall zone to another. With high level and systematic look at firewall rules, one can easily configure a set of firewalls. It is very common to find filtering rules that are inter-related in a single firewall or several firewalls. These may match exactly (duplicated), inclusively matched or correlated [11]. In this particular instance, different rules may imply different and incorrect policy semantics such as different actions. Thus, some rules may be over come by other rules resulting in firewall rule conflicts. Firewall inter-policy conflicts follow one of two conditions. These are when the last downstream firewall allows packets that were blocked by previous firewalls or when the first firewall permits packets that are blocked by other downstream firewalls. The rule of thumb here is that all downstream firewall policies should match the upstream firewall policies and vice versa. This is so that the traffic can reach its destination. Our firewall matrix model looked at end-to-end packet flows. This simplified the rules for all firewalls along the same path. If there is a specific restriction in a zone, it would be considered whether the packet gets blocked or allowed to pass.

4. Firewall Policy Matrix Model

Modeling of firewall rule relations is necessary for properly designing and managing a firewall policy to avoid firewall inter-policy errors. To solve this, one uses a high level look at firewall policy relationships that can be implemented across a distributed multi-firewall architecture. This model helps to devise the same rule that can be deployed across many firewalls thus avoiding inter-policy errors. Our matrix model was based on using a single firewall policy and deployed across various firewalls. Mayer, Wool and Ziskind [17] recommended this approach. In this section, we formally describe our model of firewall policy matrix.

4.1 Formalizing Firewall Rule Relationships

Firewalls are placed in different zones of a computer network such as the perimeter (Internet), server farm or campus. To be able to build a useful matrix model for filtering rules, we need to determine all the possible routes to and from various zones in an organization. These zones include the Internet, de-militarized zone (DMZ), server farm, data base servers, workstations, etc. This was used to determine the size of the matrix. It should be noted that the more the specific traffic paths, the bigger the matrix model. In this section, we define some of the possible relations that may exist between filtering rules. We determine these relations based on the network topology and the security policy. We then build the matrix framework illustrating the flow of traffic from source to destination (Table 2).

Table 2. Firewall Policy Matrix Template

	Source			
Destination	Internet	DMZ	Campus	Server farm
Internet	xx	xx	xx	xx
DMZ	xx	xx	xx	xx
Campus	xx	xx	xx	xx
Server farm	xx	xx	xx	xx

At the top is the source location where the traffic is coming from. The left side identifies the destination of where the packets are heading.

Traffic moving from one zone to another will pass through the zones and any others in the path. Thus, the same rule will be implemented across each firewall in the path. This matrix is used for breaking up a firewall policy into packets entering a firewall (ingress) and leaving a firewall (egress). An example for ingress is where the source is the Internet and the destination is the DMZ (Table 2). For egress a good example is when the source is the Campus zone and the destination is the Internet zone (Table 2). Each zone may have different security requirements and the firewall matrix helps to address traffic flow across the path between zones. This way we address a firewall ingress and egress rule that flows out of the network. The firewall matrix provides a simplified method to constructing rules that can help avoid inter-policy problems.

For small organizations, these designations may be adequate. However, in large organizations, such as a university campus, these areas may require further restrictions beyond just the name. In our large university, we decided to expand the matrix based on virtual local area networks (VLANs). VLANs are logical networks where devices can be grouped together based on their function such as Web servers. VLANs can further breakdown the relationship between traffic passing through different firewalls. The denoting of individual VLANs is up to the firewall administrator. For example an administrator can define the DMZ as VLAN 500 and could further break it up into smaller VLANs to further define specific areas within the DMZ. An example could be VLAN 501 (Web servers) and VLAN 502 (e-Mail). This concept allows one to define groups of servers using the same services to be grouped under similar rules and access control. A similar grouping can be done for the server farm where several types of servers are used. A good example may be VLAN 401 (file servers), VLAN 402 (databases) and VLAN 403 (application servers). The campus area can also be further broken down into VLAN 301 (class labs), VLAN 302 (research), VLAN 303 (admin workstations) and VLAN 304 (wireless) (Table 3). The result of using VLANs is to better define the security between different areas and to simplify the design of firewall rules from the security policies.

Table 3. Firewall policy matrix template using VLANS

	Source (VLANS)						
Destination	Internet	301	302	401	402	501	502
Internet	xx	xx	xx	xx	xx	xx	xx
VLAN301	xx	xx	xx	xx	xx	xx	xx
VLAN302	xx	xx	xx	xx	xx	xx	xx
VLAN303	xx	xx	xx	xx	xx	xx	xx
VLAN304	xx	xx	xx	xx	xx	xx	xx
VLAN401	xx	xx	xx	xx	xx	xx	xx
VLAN402	xx	xx	xx	xx	xx	xx	xx
VLAN403	xx	xx	xx	xx	xx	xx	xx
VLAN501	xx	xx	xx	xx	xx	xx	xx
VLAN502	xx	xx	xx	xx	xx	xx	xx

4.2 Populating the firewall matrix

Whether one uses VLANs or not, the next step is to populate the firewall matrix with a particular firewall policy. A good rule of thumb is to start with the Internet side (least restrictive) and work you way to the most restricted policy statements such as databases. Individual policy statements are written in a high level language so that administrators can read and understand them. A policy statement may be written as: Allow only ports 25, 80 and 443 from the Internet to the DMZ (10.10.1.2). This states that the firewalls should only allow e-mail (port 25), Web (port 80) or secure Web (port 443) traffic through the firewall(s). The same will need to be done concerning the egress direction of similar traffic. This security policy may be written as: Allow only ports 25 and 80 from the DMZ to the Internet (Table 4).

Table 4. Example egress and ingress firewall rules

Protocol	Source IP	Port	Destination IP	Port	Action
TCP	any	any	10.10.1.2	25, 80, 442	allow
TCP	DMZ	25, 80, 443	any	any	allow

This process continues until all the firewall policies are entered. When finished, the firewall policy matrix should look like Table 5.

Table 5. Firewall Policy Matrix including policies

	Source			
Destination	Internet	DMZ	Campus	Server farm
Internet	Allowed	Allow only ports 25, 80, 443	Allow all	Allow all
DMZ	Allow only port 80, 443	Allow all	Allow only ports 25, 80, 443	Allow ports 1521, 1433
Campus	Allow only port 80, 443	Allow all	Allow all	Restrict to specific rules
Server farm	Denied	Restrict to specific rules	Restrict to specific rules	Restrict to specific rules

Once all the firewall policy rules are added, we need to make sure all other traffic will be blocked. This is because the last firewall rule will need to make sure no unwanted traffic will be allowed into the network (Table 6). The firewall rule states that any IP address using any port/service is denied access. This is called the default rule and is placed at the end of the firewall rules. This is not placed in the firewall policy matrix since it is always put at the end of any firewall rules.

Table 6. Default firewall rule

Protocol	Source IP	Port	Destination IP	Port	Action
TCP	any	any	any	any	deny

5. Related Work

A significant amount of work has been reported in the area of firewall and policy-based security management. Several models have been proposed for designing and managing firewall policy. Gouda and Liu [10] and Liu and Gouda [14] used firewall design diagrams to specify firewall policies. The FIREMAN static analysis toolkit and SAT-solvers utilized binary decision diagrams to represent firewall rules [21]. Marmorstein [15] used multi-way decision diagrams to analyze firewall IPTables. Abbes, Bouhoula and Rusinowitch [1], Golnabi, Min, Khan and Al-Shaer [9], Tongaonkar, Inamdar and Sekar [19], and Zaliva [22] also used decision tree graphs to perform firewall policy management. These used design models to help create firewall rules. However, they were designed for single firewalls and not for a multi-firewall environment. Also they use decision trees to generate high-level policies from low level policies. However, they have handled only single firewalls. Our firewall matrix works with high-level firewall policies that can be used by firewall administrators with multi-firewall networks.

Also policy trees have been used to describe firewall policies [4][5][6]. Deng, Liang and Gao [8] and Liang and Deng [13] used Answer Set Programming (ASP) to define firewall policies. These worked well for single firewalls but not tested using multi-firewall environments. Also ASP works well for experienced firewall administrators but difficult for inexperienced firewall engineers. The firewall matrix model is simple to use for both experienced and inexperienced firewall administrators. Also our model was designed to work in single or multi-firewall environments.

Marmorstein and Kearns [16] utilized host-based classification for simplifying rules related to common host (Web servers, etc.). Their model utilized classification by IP addresses or networks. However, devices such as Web servers may have very different IP addresses. Our model provides better coverage and flexibility when using VLANs.

Pozo, Ceballos and Gasca [18] believed that firewall administrators should use models and formal methods to detect and identify errors. They utilized a constant satisfaction problem (CSP) technique for comparing firewall rules against the firewall policy. The CSP represented the policy and firewall rules independent of the network topology [18]. However, their model was constructed for a limited network and hasn't been used in a multi-firewall environment. Our firewall matrix model uses the network topology (VLANs) to construct the matrix since changes in the network topology can affect what firewall rules may need to be changed or updated.

6. Implementation

The firewall policy matrix was tested at TU using their multi-firewall network based on our association and knowledge of the university network. The TU network was composed of a database farm, server farm, perimeter, campus, de-militarized zone and over 1500 servers located throughout the campus. The TU network utilized a Cisco Firewall Service Module (FWSM) virtual firewall system. This system is a single stateless hardware firewall broken logically into multiple firewall interfaces. Each firewall interface acted as a single firewall and protected its individual network area through a central firewall policy. Interconnection between the firewall interfaces was done through the FWSM backplane and processors. There was a primary and backup FWSM supporting the whole campus network.

Utilizing VLANs and grouping server services as objects (Web servers, File servers, etc.) helps reduce firewall errors when server placement is changed. The same type of server uses the same protocols and ports. For example, Web servers use HTTP (port 80) and HTTPS (port 443). To prove this, we implemented the firewall policy matrix at TU. This tested the usability of the matrix in designing and managing firewall policies in a multi-firewall environment. The tool was able to simplify firewall policy design and management for firewall administrators.

In our implementation we started off with only 4 VLANs for breaking up the network based on the functionality described above. These four VLANs were the same areas depicted in Table 2 (Internet, DMZ, server farm and campus. These were the most common areas for communication across the network. These four VLANS were then established and incorporated within the firewall matrix. They were then incorporated within the firewall matrix of the FWSM. Then additional VLANs were established based on specific requirements of applications running on servers and user requirements. Being able to adjust the firewall rules matrix for different network connectivity helped reduce the time for setting up the new paths. Thus the firewall matrix can help setup a small and medium sized network using VLANs as seen in Table 3.

6.1 Setting up the Firewall Rules

The TU firewall team first identified all the necessary paths packets need to communicate. They started from the simple connectivity requirements such as Web and Internet communications. This was because most traffic on TU was from use of HTTP (port 80) and HTTPS (port 443). Then e-mail, enterprise resource planning and other common application communications requirements were identified. Other specific use communication from unique applications was done last. Next, the team determined what major functions different server performed such as e-mail, file storage, web, database and etc. Also the location of these servers and who communicated with them was investigated.

From these categories, VLANs were setup based on the server functionality. This resulted in constructing over 10 specific VLANs that broke down server and network areas into a more defined services (Table 3). This VLAN breakdown was also based on protecting sensitive data on databases such as medical and student data. Best practices from Defense Information Systems Agency Security Technical Implementation Guides, National Institute of Standards and Technology Special Publication 800-41 and vendors were utilized. These were used to ensure compliance with federal and state laws such as Health Insurance and Portability Act, Gramm-Leach Bliley Act and the State of Maryland Information Security Policy.

To simplify the matrix, VLANs was devised for each area such as 100 – DMZ, 200 - server farm, 300 – databases, 400 – campus and 500 – special. Then these were broken down according to different servers or services housed in the VLAN. Thus, VLAN 101 was for Web servers, VLAN102 for email servers, etc (Table 7).

Table 7. VLAN breakdown

VLAN	Service
100	DMZ
101	Web servers
102	eMail servers
200	Server farm
201	ERP
202	File servers
300	Databases
301	Oracle
302	SQL
303	MySQL
400	Campus
401	Human Resources
402	Provost
403	Finance
404	Police

From this table one can construct a matrix chart of the different VLAN zones that require firewall connectivity (Table 8).

Table 8. Example firewall policy matrix using VLANs

					Source (VLAN)				
Destination (VLAN)	Internet	101	102	201	202	301	302	401	402
Internet	Allow all	Allow only ports 80, 443	Allow only port 25	Allow all	Allow all	Allow all	Allow all	Allow all	Allow all
101	Allow only port 80, 443	Allow all	denied	Allow all	Allow all	Allow all	Allow all	Allow only ports 80, 443	Allow only ports 80, 443
102	Allow only port 25	denied	Allow all	Allow all	Allow all	Allow all	Allow all	Allow only port 25	Allow only port 25
201	denied	Allow only port 1300	denied	Allow all	denied	Allow all	Allow all	denied	denied
202	denied	Allow only port 21	Allow only port 21	denied	Allow all	Allow all	Allow all	Allow only port 21	Allow only port 21
301	denied	denied	denied	Allow only port 1521	Allow only port 1521	Allow all	Allow only ports 1433, 1521	denied	denied
302	denied	denied	denied	Allow only port 1433	Allow only port 1433	Allow only ports 1433, 1521	Allow all	denied	denied
401	Allow only port 80, 443	Allow only port 80, 443	Allow only port 25	denied	denied	Allow all	Allow all	Allow all	Allow all
402	Allow only port 80, 443	Allow only port 80, 443	Allow only port 25	denied	denied	Allow all	Allow all	Allow all	Allow all

Once the VLANs were defined, the team populated the matrix based on what firewall policies needed to be added according to source and destination VLANs. The idea was to show that the rule was used between two points and any firewalls in that path had that same rule applied. So that the firewall interface supporting the Internet and the one supporting the DMZ had the same rules for traffic across these two firewalls.

The firewall team implemented the firewall rule matrix, similar to Table 8, from the least restrictive (Internet and DMZ) and worked their way toward the most restrictive (databases). The specific protocols that needed to be passed were then added to the matrix. Examples included 80 and 443 for web servers, 1521 for Oracle database servers, 1433 for MS SQL database servers, etc. If a specific firewall needed more restricted access, a separate path was added to the matrix (Table 8).

6.*2 Implementing the Firewall Rules*

Once the basic firewall rules were added to the matrix, it was time to implement them. The team then used the Cisco Security Manager (CSM) to input the firewall rules based on the matrix. The CSM uses objects to identify similar servers such as all Web servers are in the Web server object. The same for databases was grouped by type (Oracle, SQL, etc.). The email servers were also grouped under its own object. The groups were associated with the VLAN numbering system so are to help keep things straight. Table 7 shows the objects used in configuring the firewall rules with CSM.

This console was used to configure and input defined firewall rules into various firewall interfaces of the virtual firewall system. Once all rules were installed, they were tested to ensure all rules worked. The firewall administrators found that using the model they were able to quickly redesign and implement the changes to the firewall interfaces. The administrators found that the matrix model simplified the firewall rules redesign and helped streamline the placement and movement of servers in and out of VLANs.

7. Conclusion and Future Work

Firewall security, like any other technology, requires proper management in order to provide proper security services. Thus, just having firewalls on the network boundaries or between different elements in the network may not necessary make the network secure. One reason for this is the complexity of designing and managing firewall policies in a multi-firewall environment. This can lead to firewall policy errors and conflicts that may degrade the effectiveness of firewalls. The firewall policy matrix presented in this paper provides a technique for designing and managing firewall policies to reduce the complexity of implementing these rules. It was found that a firewall administrator could use the firewall policy matrix to easily design and manage firewall policies.

The firewall policy matrix was used to redesign the Towson University multi-firewall environment to better handle changes to servers and reduce the chance of firewall errors and conflicts in the network. Using the firewall matrix, all servers in the Towson University network were reassigned to VLANs and server object groups according to the firewall policy matrix. This in turn, helped the university network to be more secure and effective in reducing anomalies. Using the firewall policy matrix helps the firewall administrator to easily align the servers with the appropriate firewall policies.

We believe that there is more to do in the firewall management area. Our future research plan includes devising algorithms to automate the matrix. In this case, the filtering rules and VLANs need to be well defined such that no desired traffic is blocked before reaching its destination and no undesired traffic is allowed to flow through the various firewalls in the distributive environment. Another is to incorporate a means to convert the security policy into specific firewall rules in a particular order to provide the best firewall performance. Also we intend to expand the matrix to work with stateful inspection, packet tagging and control flow. This way it can be tested with different models of multi-firewall environments.

References

[1] T. Abbes, A. Bouhoula and M. Rusinowitch, “An inference system for detecting firewall filtering rules anomalies”, SAC 08, Ceara, Brazil, pp. 2122-2128, 16-20 March 2008.

[2] J. G. Alfaro, N. Boulahia-Cuppens, F. Cuppens, “Complete Analysis of Configuration Rules to Guarantee Reliable Network Security Policies”, International Journal of Information Security, Vol. 7, Issue 2, pp. 103-122, 2008.

[3] J. G. Alfaro, F. Cuppens and N. Cuppens-Boulahia, “Aggregating and deploying network access control policies”, ARES 07, Vienna, Austria, pp. 532-542, 10-13 April 2007.

[4] E. Al-Shaer, H. H. Hamed, “Modeling and Management of Firewall Rules”, IEEE Transactions on Network and Service Management, Vol. 1, No. 1, pp. 2-10, April 2004a.

[5] E. Al-Shaer and H. H. Hamed, “Discovery of policy anomalies in distributed firewalls”, IEEE Communications Society, Hong Kong, China, pp. 2605-2616 7-11 March 2004b.

[6] E. Al-Shaer, H. Hamed, R. Boutaba, M. Hasan, “Conflict Classification and Analysis of Distributed Firewall Policies”, IEEE Journal on Communications, Vol. 23, No. 10, pp. 2069-2084, October 2005.

[7] F. Cuppens, N. Cuppens-Boulahia and J. Garcia-Alfaro, “Detection and removal of firewall misconfiguration”, CNIS 05, Phoenix, AZ, pp. 154-162, 14-16 November 2005.

[8] W. Deng, Y. Liang and K. Gao, “Discover inconsistencies between firewall policies”, KAM 08, Wuhan, China, pp. 809-813, 21-22 December 2008.

[9] K. Golnabi, R. K. Min, L. Khan and E. Al-Shaer, “Analysis of firewall policy rules using data mining techniques”, Vancouver, Canada, pp. 305-315, 3-7 April 2006.

[10] M. G. Gouda, A. X. Liu, “Structured Firewall Design”, Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 51, No. 4, pp. 1106-1120, August 2006.

[11] H. Hamed, E. Al-Shaer, “Taxonomy of Conflicts in Network Security Policies”, IEEE Communications Magazine, Vol. 44, Issue 3, pp. 134-141, March 2006a.

[12] H. Hamed, E. Al-Shaer, “On Autonomic Optimization of Firewall Policy Organization”, Journal of High Speed Networks, Vol. 15, Issue 3, pp. 209-227, August 2006b.

[13] Y. Liang and W. Deng, “Verify consistency between security policy and firewall policy with answer set programming”, CSSE 08, Wuhan, China, pp. 196-200. 12-14 December 2008.

[14] A. Liu, M. G. Gouda, “Diverse Firewall Design”, IEEE Transactions of Parallel and Distributed Systems, Vol. 19, No. 8, pp. 1237-1251, September 2008.

[15] R. M. Marmorstein, “Formal Analysis of Firewall Policies”, College of William and Mary, doctoral dissertation, 2008.

[16] R. Marmorstein and P. Kearns, “Firewall analysis with policy-based host classification”, LISA 06, Washington, DC, pp. 41-51, 3-8 December 2006.

[17] A. Mayer, A. Wool, E. Ziskind, “Offline Firewall Analysis”, International Journal of Information Security, Vol. 5, Issue 3, pp.125-144, July 2006.

[18] S. Pozo, R. Ceballos and R. M. Gasca, “CSP-based firewall rule set diagnosis using security policies”, ARES 07, Vienna, Austria, pp. 723-729, 10-13 April 2007.

[19] A. Tongaonkar, N. Inamdar and R. Sekar, “Inferring higher level policies from firewall rules”, LISA 07, Dallas, TX, pp. 17-26, 11-16 November 2007.

[20] A. Wool, “A Quantitative Study of Firewall Configuration Errors”, Computer, Vol. 37, No. 6, pp. 62-67, June 2004.

[21] L. Yuan, H. Chen, J. Mai, C. Chuah, Z. Su and P. Mohapatra, “FIREMAN: A toolkit for firewall modeling and analysis”, IEEE Security and Privacy, Berkeley, CA, pp. 199-213, 21-24 May 2006.

[22] V. Zaliva, “Firewall Policy Modeling, Analysis and Simulation: A Survey”, http://www.crodile.org/lord/fwpolicy.pdf, 2008.

[23] B. Zhang, E. Al-Shaer, R. Jagadeesan, J. Riely and C. Pitcher, (2007), “Specifications of a high-level conflict-free firewall policy language for multi-domain networks”, SACMAT 2007, Sophia Antipolis, France, pp. 185-194, 20-22 June 2007.

[24] C. C. Zhang, M. Winslett and C. A. Gunter, (2007), “On the safety and efficiency of firewall policy deployment”, IEEE Security and Privacy, Berkeley, CA, pp. 33-50, 20-23 May 2007.