Ph.D. Synopsis

Subject: Electronics

1. Name of the Research Student : Mr. Vinod G. Shelake

   Qualification : M.Sc. Electronics

   Address : Plot Number 361, Dadu Chougale Nagar, Kalamba Road, Kolhapur – 416 007

2. Name of the research Guide : Dr. R.K. Kamat

   Qualifications : M.Sc., M.Phil, Ph.D. (Electronics)

   Address : Reader
   Department of Electronics
   Shivaji University,
   Kolhapur – 416 004

3. Title of the proposed Thesis : Design and Development of a FPGA based Firewall

4. Introduction:

A firewall is a protective system that lies between the computer network and the Internet. The purpose of using it is to prevent unauthorized use and access to the network. It carefully analyzes the data entering and exiting the network based on the configuration chosen by the user. The configuration may be chosen so as to ignore the information coming from an unsecured, unknown or suspicious locations. A firewall plays an important role on any network as it provides a protective barrier against most forms of attack coming from the outside world. In addition to limiting access to you computer and network, a firewall is also useful for allowing remote access to a private network through secure authentication certificates and logins.

Hardware and software firewalls:

Firewalls are of two-type hardware and software. In most of the hardware firewalls the principle of “address filtering” is used. This is a sort of ‘packet filtering’ which enables to examine the header of a packet to determine its source, destination and port number. This information inturn is compared to a set of predefined or user-created rules and a packet is forwarded or dropped on the basis of the decision. The other types of firewalls are based on the principle of ‘protocol filtering’ wherein the forwarding or rejection of the traffic is dependant upon the protocol used, for example HTTP, ftp or telnet. Even the packet attributes or states have also been used as a basis to forward or reject a packet.

Firewalls are also available in the form of a software which protects the computer on which it is installed from outside attempts to control or gain access and to provide protection against the most common Trojan programs or e-mail worms. Almost all the software firewalls extend a user defined control panel for setting up safe file and printer sharing and to block unsafe applications from running on the system. The additional features available are privacy control, web filtering etc. However, the software firewalls offer only a client based computer security, which entails their installation on each and every machine intended to be secured. In practice a combination of hardware and software gives the most optimum performance for network security.

Another approach of firewall design is the application proxy in which no traffic is passed on through the firewall. Here the application proxy behaves like a server to clients on the trusted network and like a client to servers outside the trusted network.

Firewalls and their layer of operation:

Conventional firewalls operate at the network layer and their operation is based on stateful or non-stateful type. The former functions on the basis of information on the state of connections (for example: established or not, initiation, handshaking, data or breaking down the connection) as part of their rules (e.g. only hosts inside the firewall can establish connections on a certain port). The later type has packet-filtering capabilities however, it is unable to make more complex decisions as regards to the stage or level of communications between the hosts. This leads to less security and functioning more like a router from the packet filtering point of view.

5. Research Problem:

Need for the present investigation:

A firewall is an essential but neglected component of many network architectures. The proposed FPGA based firewall will address many pitfalls of the state of the art devices. Although the state of the art firewalls are more stringent from security point of view, they suffer many drawbacks. Many times they create a traffic bottleneck as they force all the packets to go through them leading to network congestion. A complete failure of the network is possible in case the firewall (which is a single link between two networks) is not configured properly or fails. The added security provided by the firewall may not be perceived as worth the increase in the technical support load. Firewalls also increase the network management responsibilities and makes network troubleshooting more complex. It is a common experience of the network administrators that they have to respond to each and every alarm of the firewall and examine logs on a regular basis, to ensure its functioning.

The proposed work will try to circumvent most of the above mentioned shortcomings. The ‘two tier security implementation algorithm’ will automatically customize the security levels so as to avoid the network congestion. The fault tolerant architecture of the FPGA will make the firewall more immune to failures. The intelligent and concurrent implementation using VHDL will reduce the burden on the network administrators.

Details of the proposed implementation:

The proposed firewall design will be based on FPGA. With the sate of art FPGA technology it is proposed to implement the

required firewall functionality directly in hardware. This will serve as a flexible platform for the processing of packets in hardware. The main feature of the implementation will be reprogrammability of the hardware , which facilitates dynamic loading of the new functionality in the firewall. As an example this feature can be used to add new filters that can run at line speed.

The literature survey depicts that the firewalls available today are based on wither packet filtering or circuit gateways or application gateways. The proposed product will be architecturally hybrid in nature with the optimum combination of all the three mentioned above. it will be based on multi-tier implementation with a provision to select a particular level by the user.

The basic security levels proposed at IP packet level are as follows:

- Level 1: static filtering with implementation of filter rules
- Level 2: dynamic filtering with provision of changing the filtering rules based on router observed events.
- Level 3: keeping a dynamic state table to make changes to the filtering rules based on events.

The level based implementation methodology proposed above will lead to many advantages such as rule storage space optimization, fast search times and generalized approach for transport and application level filtering. The other provisions proposed to be implemented are authorization based on addresses and decisions based on application data, such as commands passed to FTP, or a URL passed to HTTP.

The following standard firewall functions will be implemented and tested in VHDL on the FPGA platform:

- Authentication Functions
- Integrity Function
- Access Control Function
- Audit Function
- Access Enforcement Function

The hardware platforms used for implementation will be XILINX or ALTERA FPGAs. The software will be developed in VHDL. The main goal of the research work is exploring the benefits such as size, speed and flexibility of advanced reconfigurable FPGAs to implement firewalls with tight security and without detoriation of the network speed with fail safe architecture. FPGA based intelligent implementation will allow optimization of the rules as well as their dynamic adaptation based on the network traffic matching ratio.

The upper layers/levels will have most active rule sets to find out matching and filtering of the packets while the lower layers/levels will be based on less active rule sets. The challenging part of the research work will be dynamic updation of the rule set on the basis the network traffic so as to get optimum network performance and less end to end delay.

6. Significance of research work:

With the global Internet connection, network security has gained significant attention in the research and industrial communities. Due to the increasing threat of network attacks, firewalls have become important integrated elements not only in enterprise networks but also in small-size and home networks. Firewalls have been the frontier defense for secure networks against attacks and unauthorized traffic by filtering out unwanted network traffic coming into or going from the secured network. The filtering decision is taken according to a set of ordered filtering rules written based on predefined security policy requirements.

Although deployment of firewall technology is an important step toward securing the networks, the complexity of managing firewall policy might limit the effectiveness of firewall security. A firewall policy may include anomalies, where a packet may match with two or more different filtering rules. When the filtering rules are defined, serious attention has to be given to rule relations and interactions in order to determine the proper rule ordering and guarantee correct security policy semantics. As the number of filtering rules increases, the difficulty of writing a new rule or modifying an existing one also increases. It is very likely, in this case, to introduce conflicting rules such as one general rule shadowing another specific rule, or correlated rules whose relative ordering determines different actions for the same packet In addition, a typical large-scale enterprise network might involve hundreds of rules The proposed FPGA based firewall implementation will be designed to address all the above mentioned shortcomings.

7. Literature Survey:

The first network firewalls appeared in the late 1980s and was architecturally similar to routers used to separate a network into smaller LANs. The first security firewalls were used in the early 1990s. They were IP routers with filtering rules. The first security policy was something like the following: allow anyone "in here" to access "out there." Also, keep anyone (or anything I don't like) "out there" from getting "in here." These firewalls were effective, but

limited. It was often very difficult to get the filtering rules right, for example. In some cases, it was difficult to identify all the parts of an application that needed to be restricted. In other cases, people would move around and the rules would have to be changed [1].

The next security firewalls were more elaborate and more tunable. There were firewalls built on so-called bastion hosts. Probably the first commercial firewall of this type, using filters and application gateways (proxies), was from Digital Equipment Corporation, and was based on the DEC corporate firewall. Brian Reid and the engineering team at DEC's Network Systems Lab in Palo Alto originally invented the DEC firewall. The first commercial firewall was configured for and delivered to the first customer, a large East Coast-based chemical company, on June 13, 1991. During the next few months, Marcus Ranum at Digital invented security proxies and rewrote much of the rest of the firewall code. The firewall product was produced and dubbed DEC SEAL (for Secure External Access Link). The DEC SEAL was made up of an external system, called Gatekeeper, the only system the Internet could talk to, a filtering gateway, called Gate, and an internal Mailhub.

Firewalls from different vendors may vary significantly in terms of configuration languages, rule organizations and interaction between lists or chains. However, a firewall generally consists of a few interfaces and can be configured with several access control lists (ACLs). Both the ingress and egress of an interface can be associated with an ACL. If an ACL is associated to the ingress, filtering is performed when packets arrive at the interface. Similarly, if an ACL is associated to the egress, filtering will be performed before packets leave the interface[2].

The literature survey reveals that researchers have rated firewall security as the most focal issue. The emphasis was found to be mostly on the filtering performance issues [3-5]. On the other hand, a few related works [6,7] attempt to address only one of the conflict problems which is the rule correlation in filtering policies. Other approaches [8,9] propose using a high-level policy language to define and analyze firewall policies and then map this language to filtering rules. Although using such high-level languages might avoid rule anomalies, they are not practical for the most widely used firewalls that contain low-level filtering rules. It is simply because redefining already existing policies using high-level languages require far more effort than just analyzing existing rules using stand-alone tools such as the Firewall Policy Advisor.

Further work on managing firewall rules, particularly in multi-firewall enterprise networks, has been reported by many authors

[10-15]. They have reported a methodology for firewall filtering rules in a systematically written, ordered and distributed carefully in order to avoid firewall policy anomalies that might cause network vulnerability. The authors have identified all the anomalies that could exist in a single- or multi-firewall environment and presented a set of techniques and algorithms to automatically discover policy anomalies in centralized and distributed firewalls. These techniques are implemented in a software tool named as the “Firewall Policy Advisor” that simplifies the management of filtering rules and maintains the security of next-generation firewalls.

Management of the firewall rules has been proven to be complex, error-prone, costly and inefficient for many large-networked organizations. These firewall rules are mostly custom-designed and hand-written thus in constant need for tuning and validation, due to the dynamic nature of the traffic characteristics, ever-changing network environment and its market demands. This problem has been addressed by a number of researchers [16-20].Ordered binary decision diagram is used as a model to optimize packet classification in [16]. Another model using tuple space is developed in [17], which combines a set of filters in one tuple and stored in a hash table. The model in [18] uses bucket filters indexed by search trees. Multi-dimensional binary trees are also used to model filters [15]. In [16] a geometric mode is used to represent 2-tuple filtering rules. The reason is that these models were designed particularly to optimize packet
classification in high-speed networks too complex to use for firewall policy analysis.

Although firewall security has been given strong attention in the research community, the emphasis was mostly on the filtering performance and hardware support issues [19-23]. On the other hand, few related work [20] present a resolution for the correlation conflict problem only. Other approaches [19-23] propose using a high-level policy language to define and analyze firewall policies and then map this language to filtering rules. Firewall query-based languages based on filtering rules are also proposed in [23].

There are also research papers reporting ASIC based packet classification co processors [24]. The advantage of FPGA based coprocessing approach is due to the reconfigurable nature of the FPGA that adds additional flexibility in filtering mechanisms compared to ASIC solutions[25].

References:

1. Firewalls and Internet Security, the Second Hundred (Internet) Years by Frederic Avolio, Avolio Consulting, The Internet Protocol Journal,http://www.cisco.com/web/about/ac123/ac147/ac174/ac200/about_cisco_ipj_archive_article09186a00800c85ae.html
2. "FIREMAN: A Toolkit for FIREwall Modeling and Analysis" Lihua Yuan, Jianning Mai, Zhendong Su www.cs.ucdavis.edu/~su/publications/fireman.pdf
3. L. Qiu, G. Varghese, and S. Suri. "Fast Firewall Implementations for Software and Hardware-based Routers." Proceedings of 9th International Conference on Network Protocols (ICNP'2001), November 2001.
4. V. Srinivasan, S. Suri and G. Varghese. "Packet Classification Using Tuple Space Search." Computer ACM SIGCOMM Communication Review, October 1999.
5. T. Woo. "A Modular Approach to Packet Classification: Algorithms and Results." Proceedings of IEEE INFOCOM'00, March 2000.
6. D. Chapman and E. Zwicky. Building Internet Firewalls, Second Edition, Orielly & Associates Inc., 2000.
7. W. Cheswick and S. Belovin. Firewalls and Internet Security, Addison-Wesley, 1995.
8. Cisco Secure Policy Manager 2.3 Data Sheet." http://www.cisco.com/warp/public/cc/pd/sqsw/sqppmn/prodlit/spmgr ds.pdf
9. "Check Point Visual Policy Editor Data Sheet." http://www.checkpoint.com/products/downloads/vpe datasheet.pdf
10. Ehab Al-Shaer and Hazem Hamed, "Taxonomy of Conflicts in Network Security Policies", IEEE Communications Magazine, Vol. 44, No. 3, March 2006
11. Lopamudra Roychoudhuri, Ehab Al-Shaer and Gregory B. Brewster, "On the Impact of Loss and Delay Variation on Internet Packet Audio Transmission." In Journal of Computer Communications, Volume 28, 2005.
12. Lopamudra Roychoudhuri and Ehab Al-Shaer, "Real-Time Packet Loss Prediction based on End-to-end Delay Variation." In IEEE Transactions on Network and System Management (TNSM), Volume 2, No. 1, November 2005.
13. Ehab Al-Shaer, Hazem Hamed, Raouf Boutaba and Masum Hasan, "Conflict Classification and Analysis of Distributed Firewall Policies." In IEEE Journal on Selected Areas in Communications (JSAC), Volume 23, Issue 10, October 2005. (Nominated for Best JSAC Award paper for year 2005)
14. Hazem Hamed and Ehab Al-Shaer, " Dynamic Rule-ordering Optimization for High-speed Firewall Filtering", ACM Symposium

on InformAtion, Computer and Communications Security (ASIACCS'06), March 2006.
15. Korosh Golnabi, Richard Min, Latifur Khan, Ehab Al-Shaer, " Analysis of Firewall Policy Rule Using Data Mining Techniques", In the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006), April 2006.
16. Mitchell, T.M., Machine Learning. 1997, Sydney: McGraw-Hill.
17. Piatetsky-Shapiro, G., Discovery, analysis, and presentation of strong rules. Knowledge Discovery in Databases, 1991: p. 229-248.
18. Webb, G.I. Discovering Associations with Numeric Variables. In Proceedings of the International Conference on Knowledge Discovery and Data Mining. 2001: ACM Press.
19. S. Cobb. "ICSA Firewall Policy Guide v2.0." NCSA Security White Paper Series, 1997.
20. 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.
21. B. Hari, S. Suri and G. Parulkar. "Detecting and Resolving Packet Filter Conflicts." *Proceedings ofIEEE INFOCOM'00*, March 2000.
22. S. Hazelhusrt. "Algorithms for Analyzing Firewall and Router Access Lists." *Technical Report TRWitsCS-1999*, Department of Computer Science, University of theWitwatersrand, South Africa, July 1999.
23. T. Woo. "A Modular Approach to Packet Classification: Algorithms and Results." *Proceedings of IEEE INFOCOM'00*, March 2000.
24. Specialized Hardware for Deep Network Packet Filtering (2002) Young H. Cho, Shiva Navab, William H. Mangione-Smith at http://citeseer.ifi.unizh.ch/cho02specialized.html
25. P.W. Dowd, J.T. McHenry, F.A. Pellegrino, T.M. Carrozzi and W.B. Cocks, "An FPGA-Based Coprocessor for ATM Firewalls,"Proceedings of the IEEE Symposium on FPGA's for Custom Computing Machines (FCCM97), April 1997
26. "Design and Implementation of a Full Bandwidth ATM Firewall", O. PAUL, M. LAURENT, S. GOMBAULT ENST, C. DURET, H. GUESDON, V. LASPRESES, J. LATTMAN, J. LE MOAL, P. ROLIN, J-L. SIMON at http://www-lor.int-evry.fr/~paul_o/tissec01.pdf

Paper 10-15 are available online at
http://www.mnlab.cs.depaul.edu/mnlab/publications.htm

8. Objectives:

The main objective of the proposed research work is development of general purpose FPGA based firewall processor for high degree of traffic selectability while avoiding the usual performance penalty associated with IP level firewalls. The proposed development has following subsidiary objectives:

- Implementation on Xilinx or Altera based development platform
- Program development in VHDL using webpack
- A multi tier architecture to optimize the storage space
- An intelligent multilevel rule base to reduce the end to end delay
- Updating of the rule base on the fly on the basis of statistical techniques.
- Fault tolerant architecture to reduce the down time.

9. Methodology:

The methodology of implementing the proposed FPGA based firewalls is divided into the following steps:

- Planning: During the panning step the drafting of the rule sets, packet-filtering methodology and in general architecture of the proposed firewall will be done. The finite state machine (FSM) approach will be developed in the planning step.
- Explore step: During the explore step, the architecture of the different FPGAs will be explored. Comparison of system gates, logic blocks, registers, look up tables will be done amongst the Xilinx and Altera family of FPGAs. The FPGA architecture, which suits the proposed firewall, will be chosen for the implementation.
- Building: During this step the actual building of the devices will be undertaken. The FSM developed in the planning step will be coded using VHDL on the selected FPGA platform. A proper RJ45 connectivity provision will be made to the product. The proposed firewall will be developed as a tabletop model with built in PCI interface.
- Testing: In the testing step the firewall will be tested with the professional test bench programs. It is also planned to develop test benches.
- Installation: The finished product will be actually tested by connecting it to first to the LAN and then to internet.

- Documentation step: During this step the documentation will be written in the form of manual as well as thesis. The documentation will comprise of the listing of the entire VHDL code as well as the listing of the test-bench developed.

10. Time schedule:

Time Schedule:

Task	Year 1				Year 2				Year 3			
	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10	Q 11	Q 12
Literature survey	↔											
Planning	↔											
Drafting rule set		↔	↔									
Finalizing packet filtering methodology		↔	↔									
Developing FSM for the proposed firewall				↔								
Exploring architectures of different FPGA families of XILINX and ALTERA				↔	↔							
Finalizing the architecture				↔	↔							
Procuring the software tools					↔							
Developing the VHDL codes						↔	↔					
Testing the VHDL programs in simulated mode						↔	↔					
Synthesizing the code on the selected FPGA platform							↔	↔				
Packaging the device and providing RJ-45 connectivity									↔			
Procuring the test benches as well as developing the test benches									↔	↔		
Live testing on a LAN and internet.										↔	↔	
Documentation, thesis writing etc.										↔	↔	↔