FPGA Design of IP Packet Filter based on SNORT rules

Danijela Efnusheva, Ana Cholakoska, Marija Kalendar
Computer Science and Engineering Department,
Faculty of Electrical Engineering and Information Technologies,
Skopje, Republic of North Macedonia
{danijela, acholak, marijaka}@feit.ukim.edu.mk

***Abstract*—Contemporary technology advances increase the pace of rapid expansion of the number of computers and devices connected to the Internet on a daily basis. Therefore, one of the highest priority issues that need to be considered in this enormous network is the network security. The greater the number of connected users and devices, the attempts to invade privacy and data of connected users becomes more and more tempting to hostile users. Thus, Network Intrusion Detection Systems (NIDS) become more and more necessary in any network device connected to the Internet, and are taking the lead in the battle against intruders. This paper addresses the network security issues by implementing NIDS style hardware implementation network packet filter intended to provide fast packet processing and filtering. The hardware is based on several NIDS rules that can be programmed in the system's memory, thus enabling modularity and flexibility. The designed hardware unit is described in VHDL and implemented in a Virtex7 VC709 FPGA board. The simulation timing diagrams and FPGA synthesis (implementation) reports are discussed and analyzed in this paper.**

I. INTRODUCTION

In the present rapid expansion of the number of computers and devices connected to the Internet, one of the top three topics that need to be addressed is the network security. The greater the number of connected users and devices, the attempts to assault privacy and data of connected users becomes more and more tempting to hostile users. Thus, network intrusion detection systems become more and more necessary and present in any network device enabling Internet connections.

Network Intrusion Detection Systems (NIDS) allow greater control over the traffic generated in the network while applying several mechanisms and rules for filtering known and sometimes predicting unknown types of network attacks according to anomalies detected in the monitored network traffic. For example, SNORT – a network IDS system, [1], is well known for its ever evolving architecture and the vast collection of rules for detecting unwanted network traffic. It is a free open source network IDS software intended to detect emerging threats.

In order to enhance the security NIDS have to inspect incoming network packets looking for unwanted and hostile traffic. This is foremost done via various software platforms (e.g. Snort), that naturally are still vulnerable to unknown or novel types of attacks. Despite the flexibility, modularity and upgradability, the main potential liability of these software based NIDS systems is their inability to handle and process the continuous and daily increasing quantities of network traffic. Thus, many researchers have already turned to hardware designs for many network issues, including security. One of the first topics for hardware processing is aimed to network packets header parsing, which is a prerequisite for the packet filtering operations. Namely, the process of identifying and extracting fields from a packet header and doing it in hardware for faster processing is being addressed in many research works, [2], [3]. In most cases network processors (NPs), [4], [5], are used to perform fast packet processing where the IP header is being processed, by analyzing, parsing and modifying its content. NPs might include some specialized hardware units to perform classification of packets, lookup and pattern matching, queue management and traffic control which on the other side can be used for the purpose of packet filtering for security reasons, [6].

Resuming this paper investigates the possibilities for implementing software IDS packet filters in hardware (after the network packet headers have been parsed and accessed in memory). Consequently, the primarily goal of this paper is to augment software defined IDS systems by implementing and simulating network packet filtering modules in hardware, using the faster hardware resources of an FPGA board while retaining the flexibility of the software based rules. Such hardware/software co-design would bring speed, as well as flexibility while implementing and applying the rules for network packets filtering.

The rest of this paper is organized as follows: Section II gives an overview of the state of the art in the area presenting different network processing and filtering hardware solutions. Section III describes the design of the rule filtering hardware intended for increased security and layouts, and also presents simulations and synthesis results from the FPGA implementation of the IP packet filtering hardware module in VHDL. Section IV concludes the paper, outlining the benefits of the proposed IP packet filtering module.

II. STATE OF THE ART

Network IDS systems are generally software defined, and are still vulnerable to unknown or novel types of attacks. On the other hand, the software defined NIDS systems are quite flexible, modular and easily upgradeable. Regardless of the flexibility, the main potential liability of these software based NIDS systems is

Copyright 2020 by Information Society of Serbia - ISOS, Serbia | Creative Commons License: CC BY-NC-ND

their inability to handle and process the continuous and daily increasing quantities of network traffic, [7].

Despite the software solutions, several specialized hardware solutions intended for packet filtering have also been proposed, in order to bring additional speed to the process of filtering. Regarding hardware network packet processing, one of the most popular and vastly used solutions are the Network processors (NPs) [4]. In general, they represent devices specially tailored to perform various network processing operations: header parsing, bit-field manipulation, pattern matching, table look-ups, and data movement, [5]. Similarly, one of the more renowned and studied architectures of network packet processing is the NetFPGA architecture [8]. NPs are usually used in different types of network equipment, including routers, switches, IDS or firewalls, [6].

In recent research, NP software is getting closer to the NP hardware, such as in [9] where part of the packet processing tasks such as classification or security are offloaded to application-specific coprocessors. Other proposals make big use of FPGA technology for packet parsing, enabling implementation of pipeline architectures and thus achieving high-speed network stream processing, [10], [11]. Actually, the reconfigurable FPGA boards can be used to design flexible multiprocessing systems that adjust themselves to the current packet traffic protocols, which in turn makes them very suitable for packet filtering regarding security, [12].

FPGA technology is widely used in combination with NIDS and packet filtering for security purposes. For example, the authors in [13] propose a modular approach for grouping homogeneous traffic, and then splitting it for filtering in different specialized hardware blocks, each supporting a (smaller) Snort-based rule set tailored for the specific traffic category. The rule sets are based on the well-known Snort NIDS. The paper concludes that the exploitation of traffic classification and load statistics may bring significant savings in the design of hardware NIDS. Similarly, [14] introduces a packet pre-filtering approach, based on the observation that very rarely a single incoming packet fully or partially matches more than a few tens of IDS rules. Finally, packet pre-filtering prevents matching at least 99% of the SNORT rules per packet, thus minimizing processing time and improving the scalability of the system.

Other reconfigurable FPGA approaches include [15] and [16], which propose a modular and programmable hardware accepting configuration changes (for the rules) in real time, while processing the main packet header fields and/or payload. Finally, the research given in [17] investigates an approach suitable for current and future high speed networks of 100 Gb/s and more, by trimming the traffic that needs to be filtered using FPGA-based packet filters. The goal is achieved by executing a novel "network grammar" for specifying protocols and filtering rules for continuous stream of data. The grammar compiles directly to Verilog code for packet filters. This new concept has been tested on two different proof-of concept designs: a DNS filter and a simple firewall.

III. Design and FPGA Implementation of IP Packet Filter Based on SNORT Rules

Regarding to network IP packets processing, it is well known that one of the main operations is packet header parsing, which is mostly used for packet routing purposes. Actually, NPs spend significant part of processor cycles on packet header parsing, especially when the packet header fields are non byte- or word-aligned. Improving the number of processor cycles needed for packet header parsing has been addressed in our previous work, explained in [18], where it was shown that the proposed IP header parser has enabled fast (single-cycle) memory access to these IP header fields (for both IPv4 and IPv6). Nevertheless, since the header fields of the packet have been already extracted, further processing is possible and very useful for the aim of traffic engineering, traffic shaping and network security.

This paper is considering further packet processing regarding security, i.e. applying specific IDS (SNORT-based) software rules on received network packets for the purpose of filtering unwanted packets from the network. The IP header that is received at the input of a previous IP header parser module, [18], is usually immediately written in memory without previously going through any inspection. Consequently, to be able to enforce greater security, in this architecture, a block for filtering IP headers (v4 or v6) is being implemented before writing the packet into memory. The filtering block uses rules previously defined by the system administrator (in the considering case, predefined rules from the Snort IDS for checking IP headers). The corresponding rules have been programmed into memory on a particular location.

A schematic of the HDL model of the proposed IP packet filter that enables applying SNORT-based rules for filtering network packets is presented in Fig. 1. Following the packet parsing module, a selected IP header field is being input in this proposed module (whether TTL, protocol etc.) and is then compared to the rule in memory. If these two fields match, then the Alert signal outputs a high level signal, which in turn allows the appropriate field to be written in memory. On the contrary, if these fields do not match, the packet is rejected and not written in memory. This is signaled again through the Alert signal that outputs a low level signal.

In the beginning the Alert signal may not be present, but however this kind of signaling has proven to be very useful to the previous module keeping the IP headers (packets). In such scenario, the value of the Alert signal would indicate to the IP packet parser module that in fact there is at least one field of the header that has not passed the filter, enabling it to make a rejection of the entire header and thus delete all the fields of the entire IP packet.

Fig. 2 and Fig. 3 present a simulation conducted with VIVADO simulator, showing the signal layout for an unsuccessful and successful rule matching procedure, accordingly. Figure 2 presents that the incoming IP header field (version) does not match the expected value in the filter field, so the Alert signal outputs a low level signal value (0 value), indicating that the incoming packet field is invalid. Figure 3 presents the contrary situation where the incoming IP packet header field (version) matches the rule, and the Alert signal outputs a high level signal value (1 value), thus indicating that the IP packet header field passes the security check and can be saved in memory.

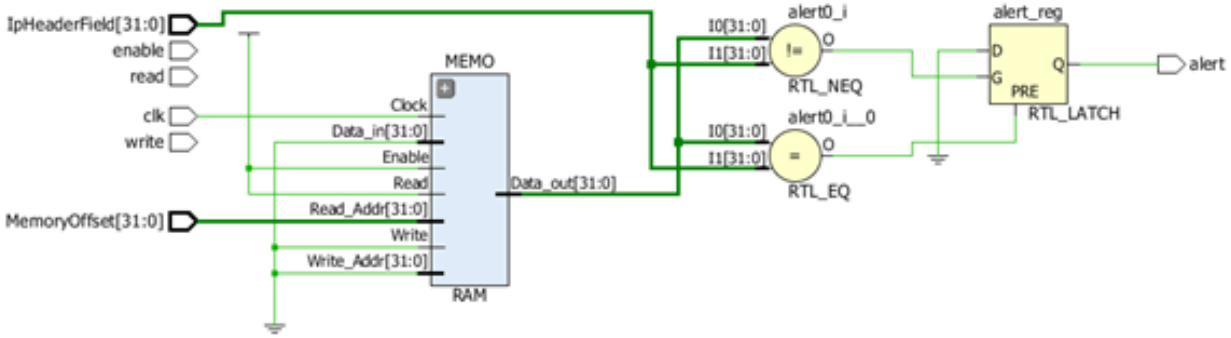

Figure 1. Schematic of IP header filter hardware.

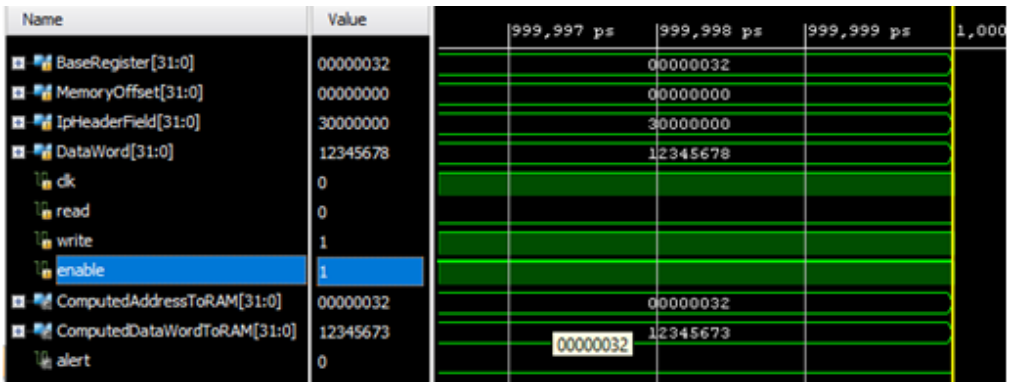

Figure 2. Simulation of non-matching IP header Field filtering.

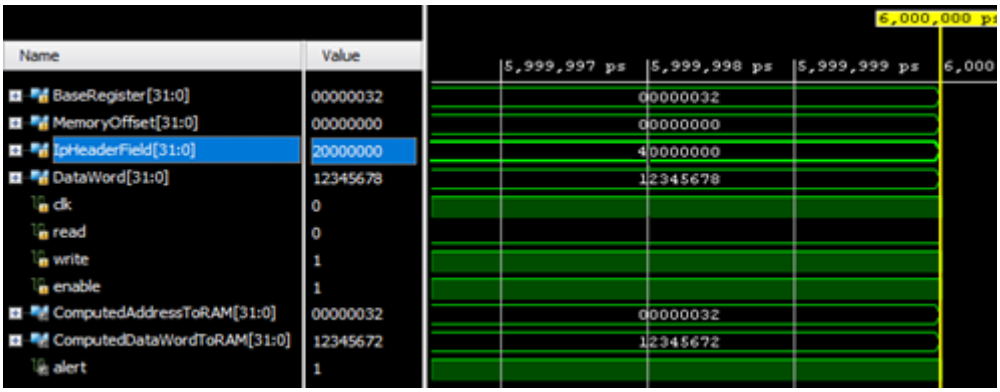

Figure 3. Simulation of matching IP header Field filtering.

Site Type	Used	Fixed	Available	Utilization(%)
Slice LUTs	689	0	433200	0.16
LUT as Logic	689	0	433200	0.16
LUT as Memory	0	0	174200	0.00
Slice Registers	36	0	866400	<0.01
Register as Flip Flop	0	0	866400	0.00
Register as Latch	36	0	866400	<0.01
F7 Muxes	0	0	216600	0.00
F8 Muxes	0	0	108300	0.00

Figure 4. Slice Logic utilization of Virtex7 VC709 FPGA resources for the proposed IP packet filter.

The simulation results and the flexibility of the proposed solution were investigated utilizing a reconfigurable Virtex7 VC709 FPGA hardware platform. After completing the functional simulation of the IP packet filter, next comes the synthesis and the implementation of the proposed hardware in Virtex7 FPGA evaluation board.

The results of the FPGA synthesis of the IP header filter are presented in Fig. 4. According to them, it can be noticed that the proposed filter of IP network packets can be implemented on Virtex7 VC709 evaluation platform by utilizing less than 0.01% of the slice registers and 0.16% of the slice LUT resources, which represents less than 1% of the available FPGA resources. As it is obvious from the low FPGA resource's utilization, the initial IP header filtering logic design can be further extended with additional selection filters to provide support for other types of network packet, protocols and packet behavior, and then re-implemented on the same FPGA board.

This means that as further research, the proposed IP packet filter can be expanded to incorporate more packet header fields and/or packet payloads, as well as additional and more complex packet header filters, all implemented on the same FPGA platform. From the ability to reconfigure the FPGA device, it can be concluded that this kind of module can be very easily adapted to work with other protocols, which in turn provides great flexibility at low cost.

IV. CONCLUSION

The presented hardware solution enables faster hardware comparison of the IP packet header fields against the programmed rules, compared to the software approach. Moreover, programming the filtering rules in memory imposes great flexibility and possibility for implementing and testing further rules for new and emerging threats. The presented module is a combined software/hardware solution intended to provide efficient network protection from unauthorized intrusions.

The concept of filtering network packets has been based on an open source network IDS system - SNORT that is well known for its ever evolving architecture and the vast collection of rules for detecting unwanted network traffic. Exactly those rules are input as the basis for hardware implementation. Therefore, the VHDL hardware design encompasses a packet filter based on hardware implementation of Snort rules, as well as a hardware accelerator for IP packet header fields extraction and rewriting. The hardware design of the module for network packet filtering, presented solid results showing that the proof of concept filter can be implemented by using only <0.01% of the slice registers and as little as 0.16% from slice LUT resources, which represents less than 1% of the available Virtex 7 VC709 FPGA resources in total.

The results obtained in this manner indicate the great flexibility and low cost of the module, as well as the possibility for its expansion towards more complex packet header filtering. As for the future development of this module, one of the possibilities is to enable more deep inspection of the packet headers, in parallel with other network processing tasks, in order to provide increased protection against unauthorized intrusions and fast detection of various security threats.

REFERENCES

[1] Brian Caswell, Jay Beale, Andrew Baker, *Snort Intrusion Detection and Prevention Toolkit*. Syngress, MA: Burlingthon, 2007.

[2] G. Gibb, G. Varghese, M. Horowitz, N. McKeown, "Design principles for packet parsers," in *Proc. of ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, 2013, pp. 13–24.

[3] J. Kořenek, "Hardware acceleration in computer networks," in *Proc. of 16th International Symposium on Design and Diagnostics of Electronic Circuits Systems*, 2013.

[4] P.C. Lekkas, *Network Processors: Architectures, Protocols and Platforms*. McGraw-Hill Professional, 2013.

[5] R. Giladi, *Network Processors - Architecture, Programming and Implementation*, Ben-Gurion University of the Negev and EZchip Technologies Ltd., 2008.

[6] B. Wheeler, *A New Era of Network Processing*. LinleyGroup Bob Wheeler's White paper, 2013.

[7] L. H. Yeo, X. Che, S. Lakkaraju, *Understanding Modern Intrusion Detection Systems: A Survey*, Eastern Michigan University, United States, 2017.

[8] J. Naous, G. Gibb, S. Bolouki, N. McKeown, "NetFPGA: reusable router architecture for experimental research," in *Proc. of Sigcomm Presto Workshop*, 2008.

[9] L. Kekely, V. Puš, J. Kořenek, "Software defined monitoring of application protocols," in *Proc. of IEEE Conference on Computer Communications*, 2014, pp. 1725–1733.

[10] V. Puš, L. Kekely, J. Kořenek, "Design methodology of configurable high performance packet parser for FPGA," in *Proc. of 17th International Symposium on Design and Diagnostics of Electronic Circuits Systems*, 2014, pp. 189-194.

[11] M. Attig, G. Brebner, "400 Gb/s programmable packet parsing on a single FPGA," in *Proc. of 7th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, 2011, pp. 12-23.

[12] G. Brebner, W. Jiang, "High-speed packet processing using reconfigurable computing," *IEEE Micro*, vol. 34, no. 1, 2014, pp. 8-18.

[13] S. Pontarelli, G. Bianchi, S. Teofil, "Traffic-aware design of a high speed FPGA network intrusion detection system," *IEEE Transactions on Computers*, Vol. 62, Issue 11, 2013, pp. 2322 - 2334.

[14] I. Sourdis, V. Dimopoulos, D. Pnevmatikatos, S. Vassiliadis, "Packet pre-filtering for network intrusion detection," in *Proc. of ACM/IEEE symposium on Architecture for networking and communications systems*, 2006.

[15] R. Ajami, A. Dinh, "Embedded network firewall on FPGA," in *Proc. of 8th International Conference on Information Technology: New Generations*, 2011.

[16] A. Wicaksana, A. Sasongko, "Fast and reconfigurable packet classification engine in FPGA-based firewall," in *Proc. of International Conference on Electrical Engineering and Informatics*, 2011.

[17] J.F. Zazo, S. Lopez-Buedo, G. Sutter, J. Aracil, "Automated synthesis of FPGA-based packet filters for 100 Gbps network monitoring applications," in *Proc. of 2016 International Conference on Reconfigurable Computing and FPGAs*, 2016.

D. Efnusheva, A. Cholakoska, A. Tentov, "Design of reconfigurable memory for fast network packet header parsing," in *Proc. of 7th ICIST Conference*, Kopaonik, Serbia, March, 2017.