

Traffic Scanner

Petr Kobierský, Jan Kořenek, Andrej Hank

30.11.2006

1 Abstract

Intrusion detection system is an integrated software/hardware tool capable of detecting unauthorised access to computer systems and malicious network traffic such as viruses, trojan horses and worms. This technical report presents the system architecture of the Traffic Scanner which is a hardware accelerated IDS based on Field-Programmable Gate Arrays (FPGAs). The designed system supports rules described in Snort-compatible format and can be configured using a web interface. System uses an architecture based on non-deterministic finite automaton for fast pattern matching. Using this approach, throughput up to 3.2 Gbps is achieved on the COMBO6X card for all rules from Snort database.

Keywords: Snort, acceleration, FPGA

2 Introduction

Network intrusion detection system (NIDS) is an integrated tool capable of detecting intrusions or malicious traffic. NIDS can analyse data packets that travel over the actual network and examine packets to verify their purpose (malicious or benign). Software, or appliance hardware in some cases, resides in one or more systems connected to a network, and is used to analyse data such as network packets.

Snort is an open source network intrusion detection system utilizing a rule-driven language, which combines the benefits of signature, protocol and anomaly based inspection methods. The Snort database contains thousands of rules which describe most of the known viruses and attacks. In Snort, more than 80% of the CPU time is consumed by the string matching task [Perf]. As network traffic speeds increase faster than PC performance, PC-based solutions can not continue to process all traffic.

The system performance can be increased by moving Snort time-critical paths from software to hardware. The technical report describes NIDS architecture which consist of COMBO6X acceleration card and Snort running on host PC.

The acceleration card removes packets from incoming traffic and sends only malicious packets to Snort.

The COMBO6X card contains powerful FPGA Virtex-II Pro. Using FPGAs, large ruleset can be matched at multigigabit speed [Clark], [Baker]. The proposed architecture uses core generator to transform Snort rules to a circuit described in VHDL language and matching all patterns from ruleset.

In the third chapter, an architecture of the whole system is described and mapping of the Snort rules to FPGA configuration is shown. The fourth chapter contains experimental results and the fifth chapter is about Traffic scanner configuration interface. Finally the characteristics of created system are conclude.

3 System Architecture

System architecture consists of several layers shown on the next picture. Physical layer is represented by COMBO6X and SFPRO cards equipped with FPGA chips suitable for processing network traffic. Behaviour of hardware cards is determined by a firmware layer which is responsible for packet processing. Malicious packets are separated from regular traffic at firmware layer. For this purpose software described in firmware paragraph is used.

Next layers include drivers that provides access to physical layer for operating system. Thanks to them COMBO6X card act as ordinary NIC card in operating system. Card is used as standard network interface which ensures transparency for using Snort IDS software. All basic software operations on COMBO6X card, especially input/output operations within registers and memory, are realized through libcombo and libcompat libraries designed for COMBO6 family cards.

For managing and debugging purposes an 'ids_ctl' tool was created. It serves mainly higher layers for probe initialization and firmware loading. It can also be used for debugging purposes by advanced users. The user interface layer is represented by 'ids' and 'ids_lkm' scripts which are used for standard Traffic Scanner control. 'ids_lkm' is responsible for loading kernel modules and 'ids' for initialization and more convenient firmware loading. A simple GUI layer which simplify these operations has also been created.

Traffic Scanner supports two ways of plugging it into network topology. Four 1 Gbps span ports monitoring or an in-line mode can be used. In-line connection works as a T-splitter and allows duplex line. Both methods are outlined in Figure 2.

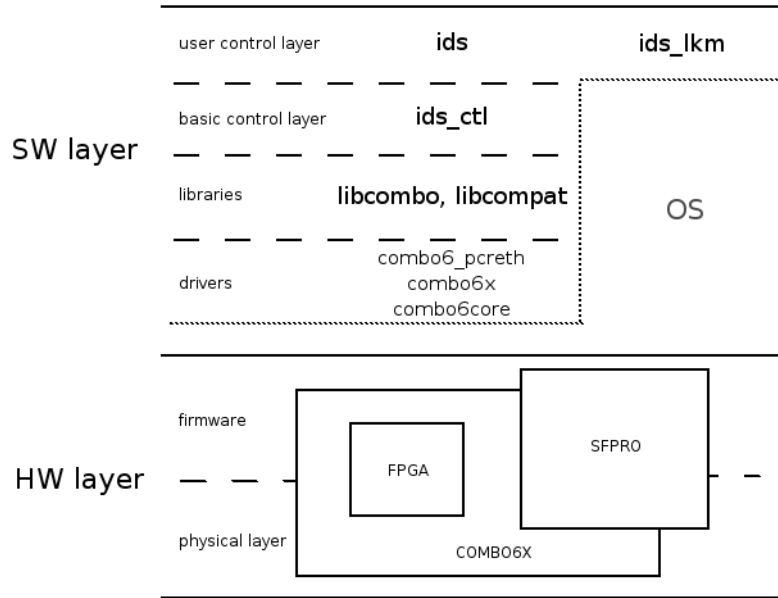


Figure 1: Traffic scanner system layers

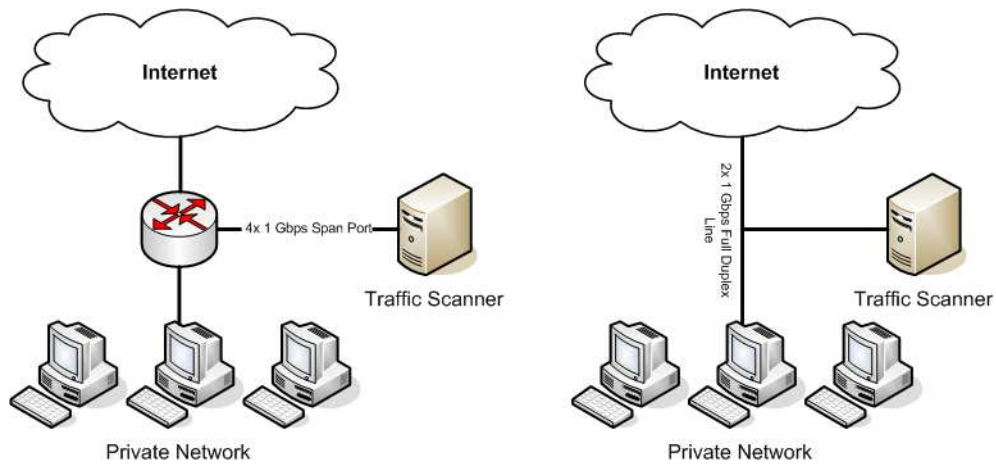


Figure 2: Traffic scanner network placement

3.1 Traffic Scanner platform

Target platform for Traffic Scanner is the COMBO6X card which was developed in the scope of the Liberouter project¹. It is an universal platform that can be used in several applications. Card is equipped with 3 SSRAMS, TCAM, DRAM, PCI communication core and FPGA which can be configured for specific application. Thanks to an extension connector, the COMBO6X card can be used with several add-on cards with different network interfaces. Currently only the SFPRO add-on card with SFP cages is supported. Both card cards contain a powerful FPGA Virtex II Pro chip with many resources which are needed to support thousands of IDS rules. New generation of COMBO cards will have more powerful FPGAs and will be able to support many more IDS rules and other IDS functionality like DOS detecting, TCP stream reassembly and filtering.

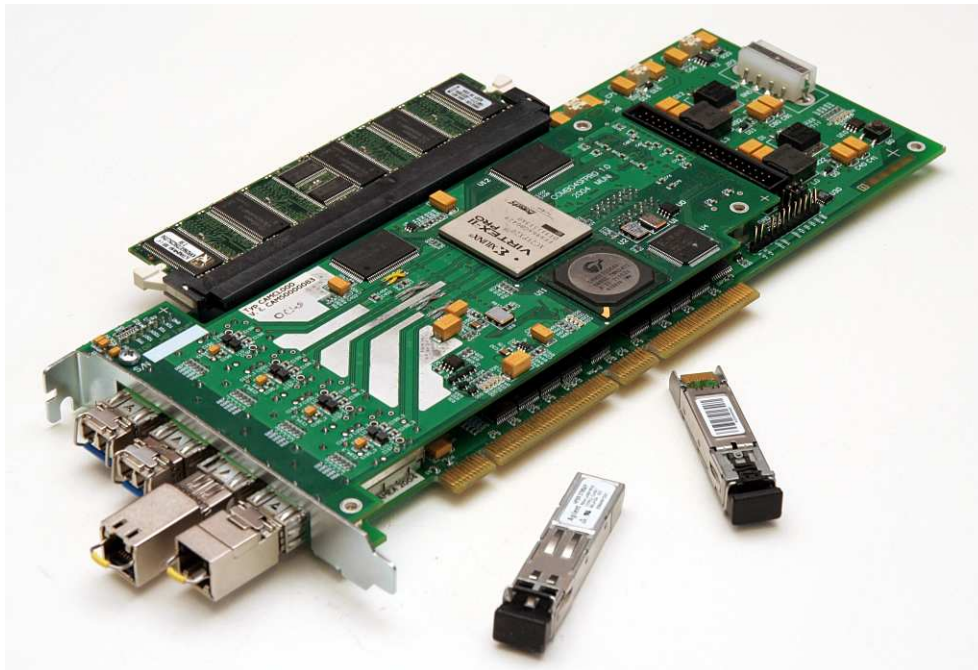


Figure 3: Traffic scanner on COMBO6x and SFPRO add-on card

3.2 Traffic Scanner firmware

Both FPGAs on COMBO6X platform must be configured to realize specified function. FPGAs are configured by loading firmware into FPGA configuration memory (SSRAM). This firmware can be changed anytime without replacing whole platform.

¹<http://www.liberouter.org>

Traffic Scanner firmware architecture is composed of classification unit, pattern match unit and several other units which are chained for pipelined packet processing. Incoming packets from four network interfaces are entering the *Input Buffer (IBUF)* block. For each incoming packet the CRC and packet length is checked and only valid packets pass through. Valid packets are processed by *Header Field Extractor (HFE)* which is assigned to each network interface. HFE extracts IPv4 header fields like source and destination address, ports, TCP Flags and much more. These fields can be used in Traffic scanner rules for detecting suspicious traffic. HFE also marks the start of packet payload where pattern matching is performed.

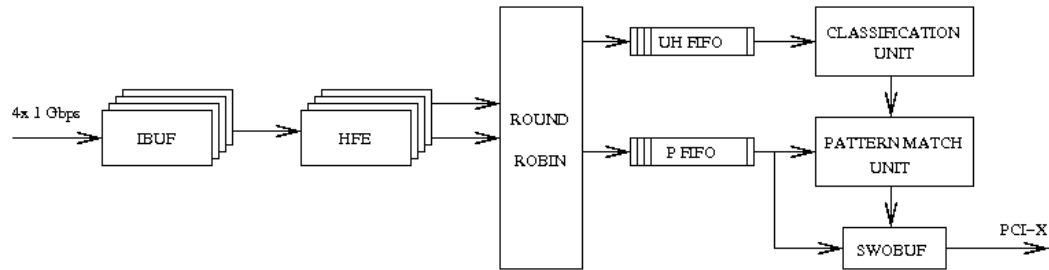


Figure 4: Traffic scanner firmware

The core of a Traffic Scanner consists of a *Classification Unit* and *Pattern Match Unit* which are configured for accelerating specified Snort ruleset. Classification unit compare extracted header fields to stored IDS rules. The main purpose of this unit is to restrict pattern matching – some patterns are searched only in specific flows not in every one. Depending on Classification unit results, specified patterns are searched in packet payload. Currently only string searching is supported but pattern match approach can also be used for regular expression matching; this will be supported in later Traffic Scanner versions. Whole packet is stored in *Software Output Buffer (SWOBUF)* and depending on a packet analysis result is either dropped or exported via PC bus for software processing.

3.3 Pattern match unit

From several pattern match approaches the NFA hardware generation was chosen as a most promising. This approach offers high throughput with possibility of searching both strings and regular expressions. High throughput compared to software solutions can be achieved thanks to simultaneous search of all strings in contents. Process of mapping patterns to hardware is composed of several phases which will be described below.

- Converting patterns to NFA
- NFA optimizations

- Converting NFA to Extended NFA
- Mapping Extended NFA to FPGA logic

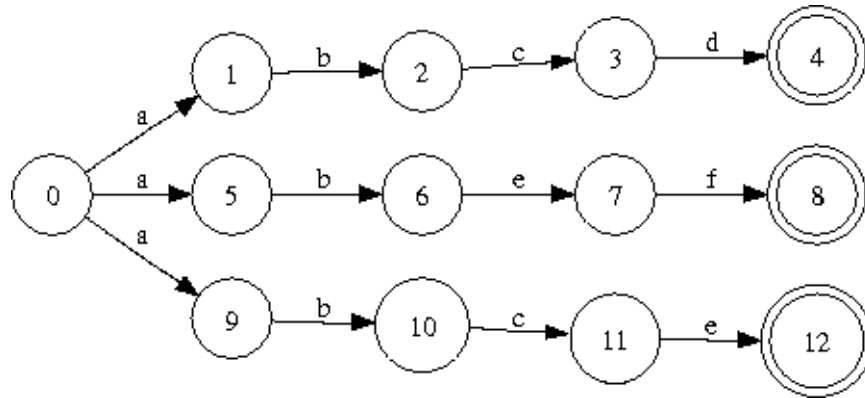


Figure 5: NFA before optimization

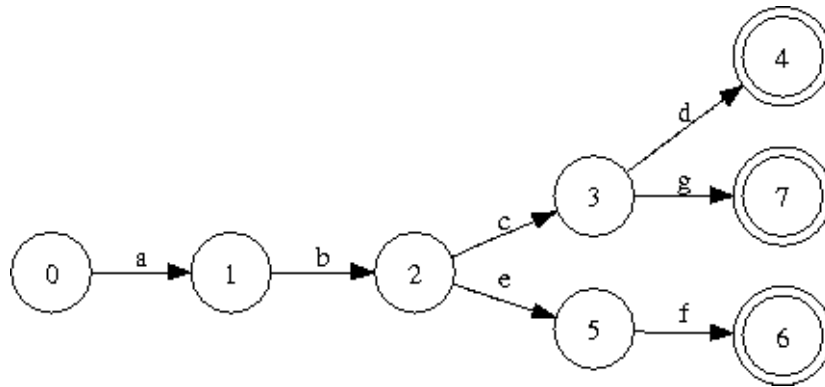


Figure 6: NFA after optimization

Patterns extracted from IDS rules are translated using generally known algorithms to NFA representation. This NFA can be further optimized. Optimization algorithms put focus on state minimization which is performed mainly through prefix sharing. Classical NFA accepts a char (8 bits) in each transitions. If we map this NFA to hardware we are able to achieve throughput up to 800 Mb/s with firmware running at 100 MHz clock frequency. The throughput can be increased by using Extended NFA (ENFA) that accept multiple chars in one clock cycle. In this case, throughput between 1 and 10 Gbps can be reached. Finally, ENFA is mapped into FPGA logic. Flip-flop is generated for every state and every transition is coded into AND-OR logic gates as show next figure. The number of FPGA gates used depends on number of patterns and their lengths. This logic utilization is also linearly increasing with number of chars which are accepted per clock cycle.

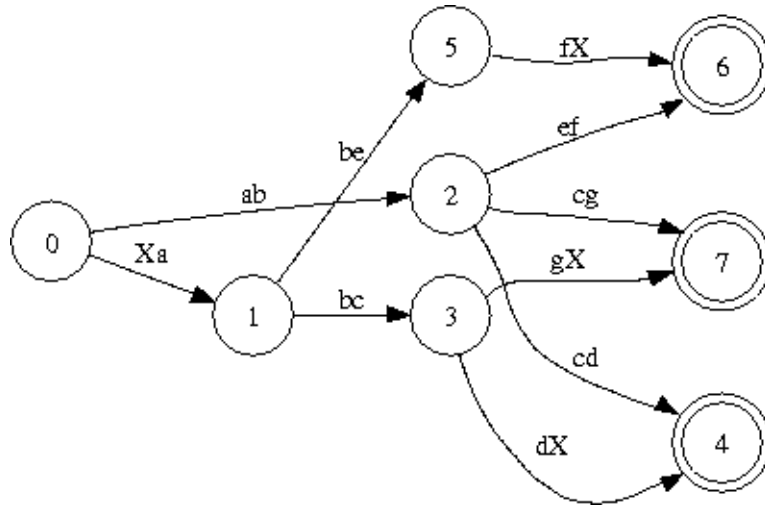


Figure 7: Extended NFA

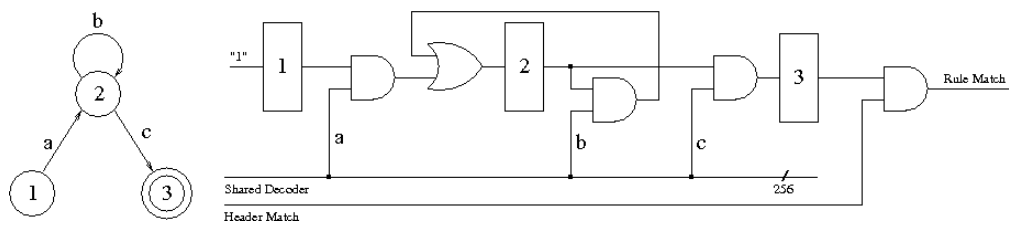


Figure 8: FSM hardware mapping

4 Experimental Results

Increasing system throughput affects requirements for chip capacity. This influence can be reduced by several optimizations focused on Snort patterns. Thanks to these optimizations all rules from default Snort installation can fit into FPGA logic with theoretical throughput 6.4 Gbps. Relation between used FPGA logic and maximal throughput for Snort rules and randomly generated strings can be seen on the next graph.

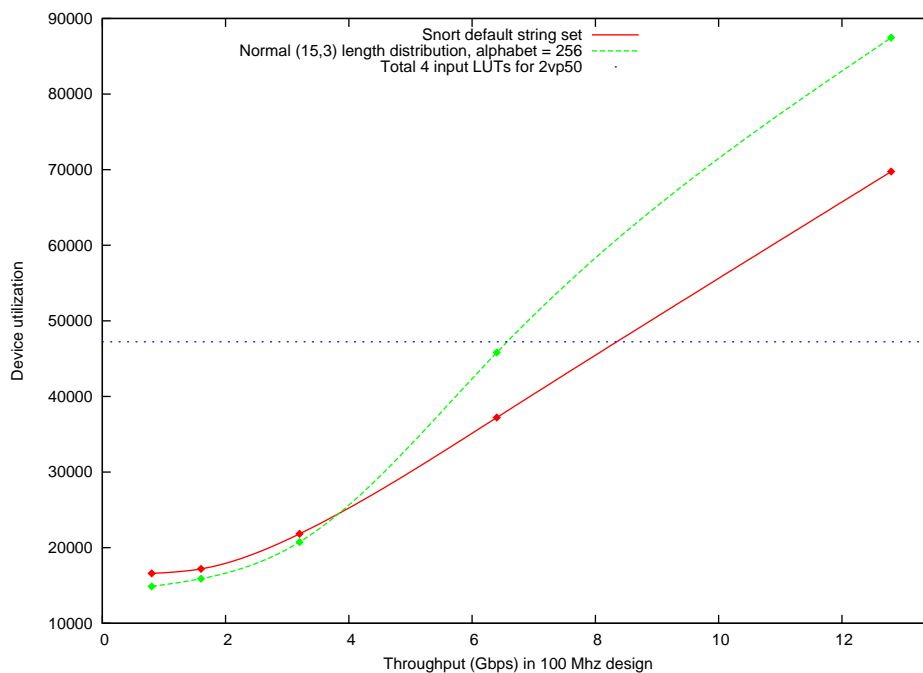


Figure 9: FPGA logic utilization for all patterns from Snort database and for randomly generated patterns

Further throughput tests over whole system were done. The aim of this test was to find out a real throughput of proposed architecture. A Spirent AX4000 network traffic generator which can generate 2 Gbps traffic with different datagram size distribution was used for testing. During the test Traffic Scanner was able to process whole 2 Gbps network traffic for short datagrams as well as long datagrams. Throughput 2 Gbps is not a Traffic Scanner limit. Probe is able to archive throughput about 3.2 Gbps and there is one important note: this throughput does not depend on the number of IDS rules. This statement is not valid for software based IDS system where throughput rapidly decreases with increasing number of rules. Results of throughput test can be seen on the following graph. Measured throughput is exactly same as a theoretical so Traffic Scanner is able to process packets on full network line speed.

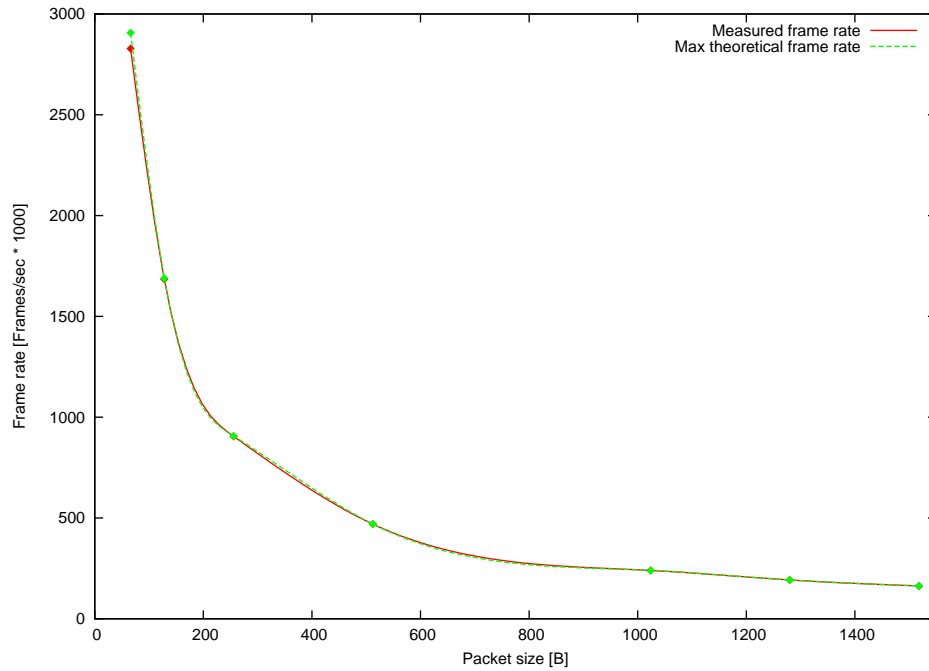


Figure 10: Theoretical throughput and throughput of proposed IDS

The last test focuses on comparing Traffic Scanner accelerated Snort and classical non-accelerated solution. Both probes were configured for detecting virus and worms threats. IDS sensors monitored the same 1 Gbps network span port from Masaryk University network for 24 hours. During this test, both solutions detected the same number and type of malicious packets. However Traffic Scanner filtered out 99.98 percent of non-malicious traffic so only 0.02 percent of all traffic had to be processed by Snort as shown on the next graph. This pre-filtration is very useful because Snort and other software based IDS solutions cannot process more than 100–300 Mbps.

5 Configuration Interface

Many rules can fit into the FPGA if core generator is used to generate and optimize the Pattern Match unit but the ruleset cannot be changed simply. If user wants to change the ruleset, new Pattern Match unit has to be generated and new FPGA configuration bitstream has to be created by synthesis tools. As users can't have all Traffic Scanner source codes and licenses to hardware synthesis tools, an issue of firmware distribution has to be solved.

One solution is to distribute prepared firmware versions which accelerate different types of rulesets (exploits, virus, p2p detecting, company network policy

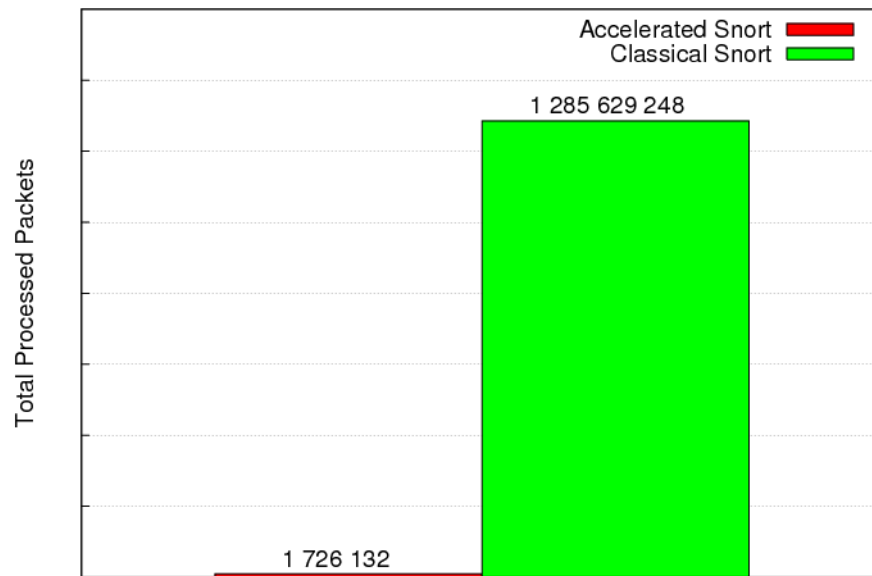


Figure 11: Comparison graph between classical Snort and Traffic Scanner Accelerated Snort

infringement etc.). After a new security threads is discovered and rule is written, firmware is generated and then distributed worldwide to Traffic Scanner sensors in similar way like anti-virus vendors distribute their updates.

Many system administrators need to create their custom IDS configuration for their specific network needs. For this reason a configuration web interface was created. Philosophy of this interface is clear. Traffic Scanner users can upload their custom ruleset to server and then start firmware generation process. After a while user receives a email notification with download link to custom firmware. This firmware can be easily loaded into COMBO6X platform using Traffic Scanner tools.

Several compilation parameters can be set by user.

- Path to Snort rules
- String length truncation
- Target throughput
- Email address

The parameters influence the number of rules which can fit into FPGA. If higher throughput is needed, number of supported rules decreases. One possible way

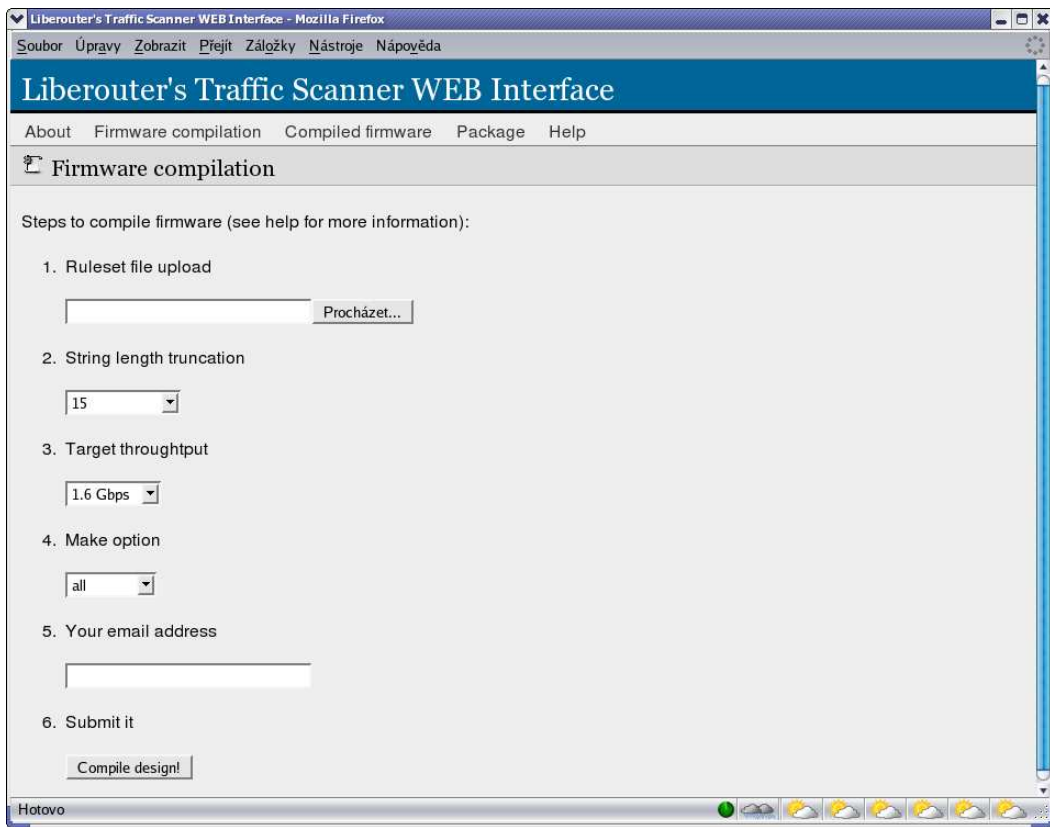


Figure 12: Traffic scanner Web interface

how to reach higher throughput with the same number of rules is truncation of search patterns. However this optimization can influence the number of packets exported to host computer, because very short patterns can be found in many packets. Currently design generation support three throughput rates – 800 Mbps, 1.6 Gbps and 3.2 Gbps.

6 Conclusion

This technical report describes an architecture and configuration of an IDS system based on the COMBO6X card. The proposed architecture uses core generator to transform Snort ruleset to optimized circuit described in VHDL and FPGA configuration. Using NFA approach for fast pattern matching, throughput up to 3.2 Gbps is achieved for all rules from the Snort database.

The core generator produces a VHDL description which is highly optimised for an input ruleset but it has to be run whenever the ruleset is changed. To simplify work with Traffic scanner, a web configuration interface was implemented. Using the configuration interface, any rules in Snort format can be transformed to FPGA configuration and downloaded to FPGA on COMBO6X card.

The performance of Traffic scanner was evaluated by Spirent AX4000 which can generate 2 Gbps traffic with different datagram size distribution. All generated traffic was processed by the Traffic scanner without losing any packet. Measurement on real network was also performed. Using the Traffic scanner, 99.98 percent of a non-malicious traffic were pre-filtered on COMBO6X card and only 0.02 percent had to be processed by Snort on a host computer.

References

- [Web] Traffic scanner web pages: <http://www.liberouter.org/ids.php>², 2006
- [Perf] Markatos, Antonatos, Polychronakis and Anagnostakis. Exclusion-based signature matching for intrusion detection. In: *IASTED International Conference on Communication and Computer Network (CCN02)*, 2002.
- [Clark] Clark and Schimmel. Efficient Reconfigurable Logic Circuits for Matching Complex Network Intrusion Detection Patterns. In: *Proc. Field Programmable Logic and Applications*, 13th International Conference, Lisbon, Portugal, 2003, p. 956–959

²<http://www.liberouter.org/ids.php>

- [Baker] Zachary K. Baker and Viktor K. Prasanna. Time and Area Efficient Pattern Matching on FPGAs. In: Proc. *12th ACM/SIGDA international symposium on Field programmable gate arrays*, New York: ACM Press, 2004, p. 223–232.