A High Speed Network Intrusion Detection System Based On FPGA Circuits

Ahmed Riadh Baba-ali,

University of Sciences and Technology Houari Boumediene,
PO box 32, El-alia, Algiers, Algeria

Summary
This paper describes the implementation of a high speed hardware network intrusion detection system. The system is powerful as well as flexible due to the use of hardware configurable circuits such as FPGA (Field Programmable Gate Array). The developed circuit architecture uses a pipeline technique based on communicating finite state machines. The goal is to maximize the throughput while reducing the latency. One of the main characteristic of the circuit is that Ethernet packets are processed directly on the fly. As a consequence, a character is processed as soon as it is acquired from the medium without waiting for the complete arrival of the current packet. The pipeline allows all circuit modules to operate in parallel with minimal synchronization.

Key words:
NIDS,FPGA, Ethernet, hardware.

1. Introduction

NIDS (network intrusion detection system) are an essential element of network security. They are considered as the last defense after firewalls. They are generally used to filter packets coming from Internet. Their function is complementary to the one of firewalls. Firewalls perform filtering by analyzing packet headers while NIDS perform filtering by analyzing packet payload [1].

Payload analysis is much more complex than header analysis. It requires more resources because payload consists of unstructured text of varying length, in contrast with the header which is composed of data such as IP addresses that have a regular structure and a fixed length.

The goal of payload analysis is to verify that it does not contain any attack signature. The difficulty of this task is that networks speed is increasing almost everyday. Furthermore the number of attacks on Internet is also increasing. As a consequence for the NIDS, the rate of signatures analysis is increasing, whereas processors speed meanwhile is stagnant [2]. Given this situation, the conventional solution-based microprocessors are becoming increasingly dysfunctional. To such a point that some IDS software can not analyze all packets passing through a network when the speed exceeds a few hundred Megabits/s, while Gigabits/s networks already exists. In this context, SOC (System On Chip) hardware solutions based on FPGA (Field Programmable Gate Array) have emerged. In facts, NIDS based FPGAs offer the high performance of hardware while offering the flexibility of configurable circuits [3]. High performances are necessary in order to analyze high speed networks, while flexibility is necessary to cope with the constant evolution of hackers attacks.

2. Field Programmable Gate Array (FPGA) integrated circuits

A FPGA is an integrated circuit that contains a matrix of generic logic blocks and configurable switches as well as a set of configurable I/O ports. A logic cell is an universal boolean function which can be configured to realize basic logic functions. Configurable switches are programmed to realize interconnects between logic blocks. Finally configurable I/O ports can be programmed to realize either an input or an output port. All these configurable FPGA features permit to realize a complete custom design.

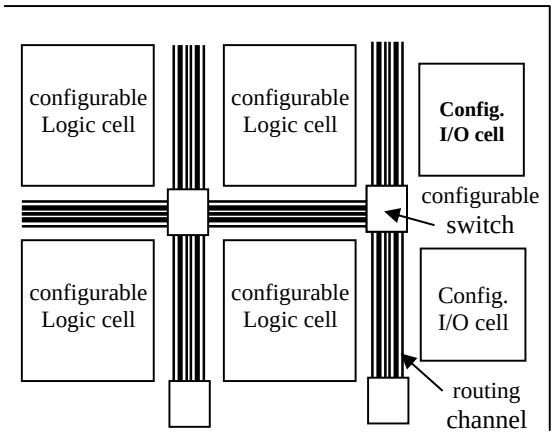

Fig. 1 FPGA structure

Manuscript received November 5, 2009
Manuscript revised November 20, 2009

FPGA circuits are nowadays a mean to realize custom integrated circuits, ranging from basic combinatorial circuits to complex multi-controller systems on chip. In one hand, their capacity is increasing every day, in another hand, their intrinsic parallelism permits to realize high performance systems.

3. NIDS Structure

NIDS are generally composed of several elements: One or more sensors and an IDS manager system [4]. IDS sensors are physically connected to the network wire. They read packets, compare their contents to an attack signatures database and send alarms to the manager when an attack is detected.

The IDS system manager receives alarms or suspicious packets, stores them in a file, and launches appropriate actions. Some IDS systems have an additional element called the console that enables remote view of alarms.

4. Structure of an IDS sensor

A sensor is generally composed of three components:

- Packet acquisition module
- Signature detection module
- Alarms generation module

4.1 Packet acquisition module

The acquisition circuit is connected directly to the network wire through a RJ45 connector. Ethernet has two levels called PHY (Physical Layer) interface and MAC (Media Access Control) interface. The PHY acquires electrical signals that are compatible to the IEEE 802.3 norm, to form 4 bits data if the network is 100 Mbits/s or 8 bits if the network is 1 Gbits/s. The transfer is synchronized by the clock network according to a communication standard called MII (Media Independent Interface). Subsequently the bytes are sent to the MAC which stores the basic information to form a complete packet. The obtained packet is then stored and managed in an accessible memory.

4.2 Signatures detection module

As soon as packet characters start entering inside the sensor circuit, the signature detection processing begins. Payload Substrings are compared to all signatures in the database. As the total number of signatures may be large, this process requires substantial resources. This process uses a database of signatures created and maintained by the Internet community. We can mention one of the most used, the Snort system [8].

There are currently two main types of hardware solutions [5]: finite state machine based solutions such as the AC (Aho-Corasick) algorithm [6], and hash tables-coding based solutions such as Bloom filters [7]. These two approaches have both advantages and disadvantages. Bloom filters have an efficient memory space usage reserved for signatures, whereas the AC algorithm is known for its ability to perform parallel signatures analysis.

The AC algorithm allows to search multiple keywords simultaneously. It uses a finite state machine constructed from all keywords. The machine is then used to analyze a text in one pass. The machine consists of states and transitions.

There are several types of states:

- The initial state
- The intermediate states
- The recognition states
- The error states

There are several types of transitions:

- Normal transitions
- Erroneous transitions
- Accepted transitions

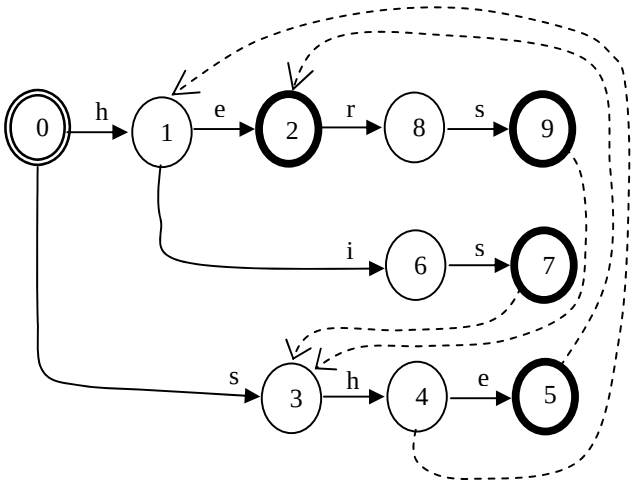

Fig. 2 Example of a finite state machine algorithm for the AC algorithm

In the example in the figure below, the state machine recognizes for example the following keywords: (*he*, *she*, *his* and *hers*). Normal transitions are drawn in solid line, while errors transitions are dashed. The initial state is the 0 state. Recognition states are the ones drawn in bold, are the 2, 5, 7 and 9 states. Error states are those that are pointed by wrong transitions and are the 1,2 and 3 states. Finally accepted transitions are those that end in recognition states.

Each machine state corresponds to an analysis stage. The machine processes a character per cycle, progressing from the starting state, through normal states, toward recognition states or error states. A recognition state is reached when a signature is detected. An error state is reached when there are several possible signatures.

Transitions represent possible evolutions between states, according to the current character read from the analyzed text. If a signature is found during a text analysis, a set of normal transitions are used to lead finally to an accepted transition. An accepted transition ends in a recognition state. An erroneous transition guides the analysis towards
finding an alternative signature.

4.3 Alarms module

Sending an alarm from an IDS sensor to the IDS system manager may be achieved by several means. One of the most economical means is to send an alarm as an Ethernet packet to the manager through the network being monitored.

5. The proposed architecture

The first choice that we have done is to acquire network packets directly from the wire through the PHY physical circuit. One of the advantages is that packets are read with a greater speed. Besides reading one character at a time without waiting the arrival of a complete packet, permits to search for signatures in parallel. Consequently, if the current packet contains a signature, an alarm is sent immediately without waiting the complete processing of the packet, reducing in turn the latency.

The hardware architecture that we proposed is based on a pipeline architecture containing three circuit stages that are:

- The packet acquisition circuit
- The signature detection circuit
- The alarm generation circuit

In our architecture, these three circuits operate in parallel despite the dependency that exists between them. As soon as the acquisition circuit has read a character, it sends it to the detection circuit. While the detection circuit starts processing the character, the acquisition circuit starts to read the next character. When the detection circuit has identified a signature, it sends an identifier as well as some packet parameters (source and destination IP addresses, source and destination ports etc.) to the generation circuit that emits an alarm. Consequently an alarm can be issued before a packet is completely acquired.

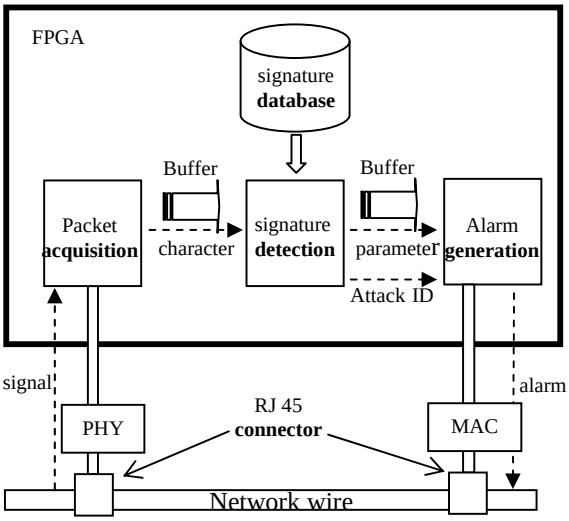

Fig. 3 System structure

The circuit works according to a producer/consumer mode. Synchronization between the three components is done by using conventional techniques such as synchronization variables. The main difference lies in the fact that these three circuits have different processing speeds clocks. The acquisition and alarm modules use the network clocks (RX_CLK and TX_CLK)) that have a standard rate. While the detection module must operate with the highest possible speed. The maximum speed is achieved with the use of an external clock whose frequency is set according to the placement and routing of the three circuits on the FPGA.

The processing speed of a character, within the detection circuit depends on the used algorithm. Indeed, the AC algorithm [6] requires in most cases one cycle (or one transition of the state machine) per character. But in some cases, the AC algorithm performs an additional transition to the appropriate error state through the failure table, which in turn slows the process.

The required processing time for one state machine cycle is much smaller than the one required for reading it from the PHY circuit. However, in order to avoid any synchronization problem, we chose to use two buffers: the first one is located between the reading circuit and the detection circuit. The first buffer is fed at one end by the acquisition circuit, while the processing circuit extracts the characters at the other end. The second buffer is located between the detection circuit and the generation circuit. It works on the same principle as the first and receives a character from the detection circuit when this character is completely processed. He also receives an attack identifier if a signature has been identified. In this case the generation circuit constructs an alarm message which contains the description of the attack and some parameters like IP addresses and port numbers to form a complete package. Then this packet is transmitted on the network to the IDS System Manager.

The structure of the used buffer does not require sophisticated synchronization techniques such as semaphores. Each process uses instead its private variable for buffer management. Besides, the buffer size is limited to the maximum size that may occupy an Ethernet packet. This buffer is managed as a circular list which in certain cases allow to manage up to two packets at the same time.

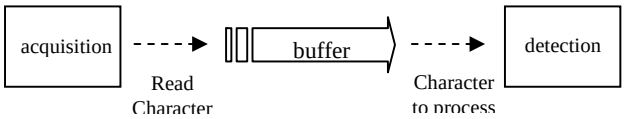

Fig. 4 Synchronisation between acquisition and detection

The generation module also uses the transmitting network clock that imposes a maximum writing speed. The choice that was made to the output circuit is to use the MAC interface which owns and manages a queue of packets to be transmitted, which allows to take into account the availability of the network.

6 Implementation and Testing

The circuit has been implemented and tested on a Spartan 3E low cost FPGA circuit, from Xilinx [9]. The clock speed of the detection circuit after placement and routing, is 200 Mhz, which gives a maximal throughput of up to 1.6 Gbits/sec. The circuit was tested on a real network, the results showed that the circuit works correctly with
different attack scenarios.

In this work we considered as a first priority the processing speed. We considered as second priority, the optimization of the FPGA resources in terms of CLBs (Configurable Logic Blocs). The circuit itself occupies an area of less than 5% of the FPGA, while the majority of the circuit is occupied by the matrix of the finite state machine. In fact, this matrix has 255 columns and thousands lines. The columns represent the ASCII codes, while the lines represent the states of the machine. As the number of states is high, the matrix size could be large. Techniques of representation of sparse matrices could be used but it would be at the expense of processing speed. The author believe that this problem can partially be solved by adopting FPGAs with more resources, but with higher cost. Another possibility would consist of using several FPGA circuits, containing each, only a part of the total detection rules. These FPGA could be plugged together in the same network and could in turn operate in parallel.

References

[1] Hong-Jip Jung, Zachary K. Baker and Viktor K. Prasanna
Performance of FPGA Implementation of Bit-split Architecture for Intrusion Detection Systems, PDPS 2006.
[2] V. Paxson, K. Asanovi, S. Dharmapurikar, J. Lockwood, R. Pang, R. Sommer, N. Weaver ,
Rethinking Hardware Support for Network Analysis and Intrusion Prevention in, HOTSEC'06, 2006.
[3] J. Sourdis.
Efficient and High speed FPGA based string matching for Packet Inspection, MSc. Thesis , Technical University of Crete, 2004.
[4] E. Maiwald.
Fundamentals of network security, Mc.Graw Hill, 2004.
[5] Jonnhy Tsung Lin Ho, Guy Lemieux.
PERG-Rx: A hardware pattern-matching engine supporting limited regular expressions, Proceedings of the ACM/Sigda Int. symposium on FPGA, Feb 2009.
[6] A.V. Aho and M. J. Corasick.
Efficient String Matching: an Aid to Bibliographic Search. *Communications of the ACM*, 18(6):333.340, June 1975.
[7] B. Bloom.
Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13 :422.426, 1970.
[8] Sourcefire. Snort: The Open Source Network Intrusion Detection System. http://www.snort.org, 2003.
[9] Xilinx, Inc. http://www.xilinx.com.

Dr. A.R Baba-ali received the B.S. and M.S. degrees in Computer Science from the University of Science and Technology Houari Boumediene in 1985 and 1991, respectively. He received his PHD degree in electrical Engineering in 1998 from the National polytechnic school of Algiers . He was with the Advanced Technology Development Center (CDTA) during 1986-1997. He is now with the University of science and technology Houari Boumediene where he is teaching a VLSI course and a course on embedded systems.