Structural Framework for High Speed Intrusion Detection/Prevention Signature Based Systems

Vukašin Pejović, Slobodan Bojanić and Carlos CarrerasVaquer

Departamento de Electrónica, Escuela Superior Técnica de Ingenieros de Telecomunicación
Universidad Politécnica de Madrid, Madrid, Spain

Summary
Evident demand for higher speed and better performance products also strives to be a direction of development in the area of intrusion prevention technologies. Shifting of crucial parts of such systems into hardware might provide demanded improvements. Systematic framework, for design of hardware based intrusion protection system, deployed as a misuse detection system is presented together with the most of the problems that are not solved and need to be tackled.

Key words:
Hardware, NIP(D)S, Signature , Misuse Detection, High Speed

Introduction

Traditional approach to designing, or to put it better, developing and deploying a network intrusion detection and/or prevention system (NID(P)S) has been completely based on software implementations. Typical example is extremely successful SNORT intrusion detection and/or prevention system, an open source based ID(P)S, and as it is written on official website “the de facto standard in intrusion detection/prevention” [1]. Software implemented approaches have proved to be fast enough for currently implemented local area networks. For example benchmarking of Hogwash IDS [11], another prominent open-source solution shows it capable of sustaining functionality on throughputs of up to 100Mbps [12]. It is expected that, as Ethernet improvement process strives to provide support for greater throughputs, 10Gbps, 40Gbps or even 100Gpbs, software based ID(P)S would not be able to follow these speedup tendencies. That specific motivation has inspired global academic effort on providing solutions for hardware based NID(P)S.

Hardware based should be associated with hardware implementation of the most time consuming parts of complete ID(P)S, as this type of approach might significantly improve overall system performance. Without any doubt and because of the nature of the problem it solves the most time consuming part, as one could notice even in the SNORT users manual [2] is the signature or sting matching part. That is relevant for signature based NID(P)Ss. On the other hand, because of the way it solves the problem, or exactly, the usage of some kind of repetitive comparison engine, signature comparison part is suitable for hardware implementations, leading to potential speed up and improvement of performance.

Existent string comparison algorithms, and signatures are strings, such as classic Knuth-Morris-Pratt [4] (KMP), Boyer-Moore [5] (BM) or Aho-Corasick [6], are being shifted to hardware and optimized further in order to be capable of performing with higher throughput rates. There are implementations promising 16 Gbps throughputs [19]. Besides those general purpose string matching algorithms, exist other algorithms such as Setwise Boyer-Moore-Horspool [7] or E2xB [8] designed for IDS specific comparison space, that might be considered for hardware implementations in future. Independently of that kind of software to hardware translation approach significant number of hardware specific matching algorithms occurred. These such as Ternary Content Addressable Memory (TCAM) based [18], Bloom filter based [17], or discrete comparator based [14] algorithms need to be taken into consideration as potentially strong foundations for HBNIPS.

Having this short preview in mind the purpose of this paper will be to give deeper insight on problems that have to be solved in order to design hardware based NID(P)S and present available proposals as a guidelines in design process. The paper continues as follows: Section 2 presents concepts important for hardware based network intrusion prevention system (HBNIPS), potential block scheme, requirements, etc; Section 3 gives deeper and more specific analysis of present hardware based approaches for signature comparison algorithms. In the end the final Section 4 concludes the paper by final discussion.

2. Hardware based network intrusion prevention system

A black box that would get rid of unwanted traffic one gets, this in ideal case would be a simplification of functionalities that must be provided by HBNIPS. It is obvious that unwanted greatly depends on referent system

Manuscript received September 5, 2006.
Manuscript revised September 25 , 2006.

used to decide what is unwanted and that the term traffic in network sense is something nobody can control, thus this simplification can be seen as a real-life existing model of unpredictability. Such situation directly implicates the ease of the design of mentioned black box, especially knowing that there is an almost total correlation between "unwanted" and OS running on machine that is being protected, referring here to numerous flaws of not just MS products. Additional influences on "unwanted" is made by immense amount of potential applications that are designed to implement networking functionalities and are almost normally present on average personal platforms, here even game servers must be taken into account.

2.1 Global View

Focusing on potential solution presented on Figure 1, the environment in which the HBNIPS should be seeing itself implemented, in the first instance is supposed to be 10Gbit half-duplex Ethernet environment, specified in IEEE 802.3a series of specifications, whether it has copper or optical cables as physical connection media, and hoping that in second instance it might be suitable for full-duplex use. In such an environment it would be placed "in-line" so it could interact with traffic that goes through it. Traffic analysis and activation of prevention actions are main functionalities and these must be implemented. Details related to those will be mentioned later on.

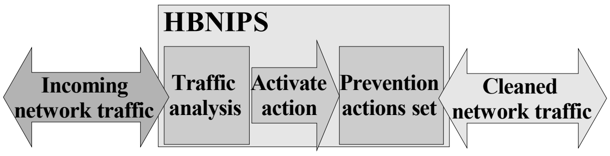

Fig. 1 Hardware Based Network Intrusion Prevention System Functional Overview

In this moment one important characteristic has to be highlighted. It is the necessity of this device to be immune to all the types of attacks known to be performed via TCP/IP network. Latter is achieved by designing the device as non TCP/IP device. That actually means that it does not participate in traffic as IP device, thus has no IP address assigned, but would need to have a MAC layer address as it needs to interact on physical layer. Device, thus only supervises traffic, seeing it as set of bits passing through it. Device performs analysis on bit level and by deriving actions from that leaves the traffic "cleaned". Although this representation is idealistic it should certainly be greatly welcome.

In more technical manner it is possible to give time representation of requirements for analysis. According to IEEE 802.3ae standard the worst case scenario is the shortest time period for analysis, for the minimal packet size transfer case. Again, according to the standard with 10Gbps throughput, worst case "packet throughput", the highest possible value, will be 10Gbps divided by minimal packet length of 672 bits. Stated packet length value includes 8 bytes long preamble, necessary for Ethernet physical implementation, 12 bytes of inter-packet gap and minimum packet length of 64byte. By simply calculating according to the specified values a "packet throughput" value is 10Gbps/672 = 14.88 mega packets per second [9]. This, translated to time requirements would mean that one 10 Gigabit Ethernet packet must be processed during 67.2 ns, in the worst case scenario.

Moving away from physical aspects and looking at packet analysis from prevention point of view, it is very important to have correct decision made, as the triggering of the actions on decisions is almost trivial. Since the passive vs. active role difference exists between intrusion prevention and detection systems, simple leaving out the prevention actions would transform prevention system into a detection system. Even such simpler, detection, system might be a good starting point for further development.

As prevention actions are concerned even the starter set of two actions might be satisfactory for the fist iteration. Those would be to drop the packet or to let it go through with alarm or warning. In any case for complete design of one HBNIPS issues like authority informing, configurability and user tuning, then user editable reaction actions, knowledge base refreshment and various others performance characteristics including possibilities of communication with user, here referring to GUIs or other type of user interface, such as console, must be developed and incorporated. All these aspects would than result in complete solution.

2.2 Zoomed View

The intention is for the device to be attached on 10Gbps Ethernet line, has been mentioned. With that in mind a more detailed block scheme would be as on Figure2. Functions which the blocks perform are clearly stated on the scheme.

Other important fact is that the scheme obviously corresponds to signature or misuse based solution, which is being proposed and discussed. There are different factors responsible for such approach of which the most important one is the destination platform, clearly stated to be a hardware one, FPGA or ASIC. Alternative would be anomaly based system. Since for such an implementation lots of statistical and other more general mathematical mechanisms are to be used, including training process, it is not seen easily implementable on hardware devices with requested speed goal in mind. On the other hand signature based system would, after fixing and structurally positioning the implementation mechanisms in hardware, be able to deliver maximum performance in terms of throughput. In such a case there is necessity to develop signature refreshment technique and allow a system to perform optimally while being constantly refreshed, we will get back to this specific topic later. Nevertheless the improvement of anomaly based techniques might lead to efficient hardware implementation, in which case the scheme would need to be upgraded.

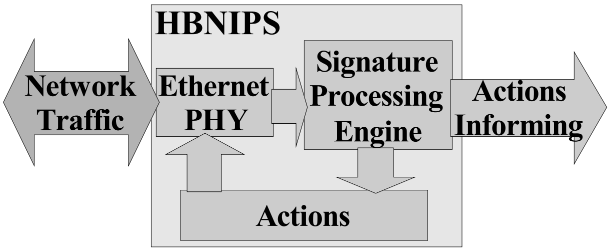

Fig. 2 HBNIPS Initial Block Scheme

2.3 Physical Layer

Although physical layer is not the crucial in achieving desired throughput, at least from the point of view of this work, the good solution on this design level certainly implies quality of overall solution. Dependently on selected hardware platform, FPGA or ASIC, there are different approaches available. In ASIC domain, there are various commercially available and standard compliant 10Gbps Ethernet ready-made IP cores. Similar situation is encountered in FPGA domain. These enable integration of design within one chip, either ASIC or FPGA.

Other commercially available approach, not so platform dependable, is to use a separate IC as MAC layer interface. That would raise the question of successful PCB design. Yet this question would remain in any selected scenario, because it might be expected to have a need for different memory access levels and interfacing between them. This appoints to that serious considerations of PCB design issues have to be also taken into account, with “all in one box” approach proposed here.

2.4 Functional Block Scheme

Block scheme to be described on Figure 3 has a goal to cover and present the mechanism that will probably need to be implemented on signature based intrusion detection system. It can be noted from mentioned figure that there are two separate functions that must be provided in signature processing mechanism, those are signature comparison and signature refreshment or updating, all making a part of signature processing engine. Upper must be present in order to establish the correctness of traffic going through the system, while latter must be present to provide correctness of comparisons performed, in the sense of global actualization of signatures.

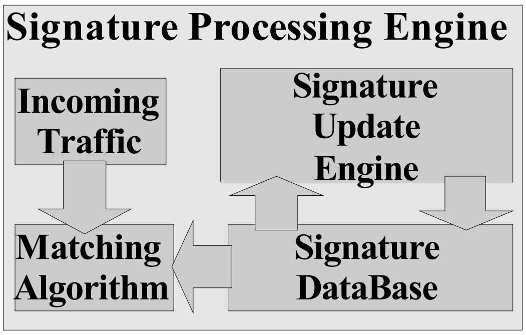

Fig. 3 Functional Block Scheme

Different questions arise form the mentioned facts, mostly targeting the way refreshment should be provided. It is surely not acceptable to have inline device out of function while refreshing, which restricts the way refreshment should be executed to, as might be called, “hot” procedures. If as a target device for design a FPGA chip is selected, there might be a possibility of applying dynamic reconfiguration techniques characteristic for this devices. Dynamic reconfiguration provides a way to reconfigure part of single FPGA chip, while the rest of it is still fully functional.

In intrusion detection context, as intensive real time computation is expected dynamic reconfiguration has to be used in processing free intervals, or the device has to be designed to provide time for dynamic reconfiguration.

If the targeted architecture is a yet to be develop ASIC, it obviously gives a liberty of configuring resources freely and making a specific layout design according to approach selected. The thing that must be noted, and is related to ASIC domain, is that the final solution must provide a way for updating signatures. This means that existence of certain “degrees of freedom” must exist. One way to do it that may occur at this moment is that if a memory structure is used for storage of signature data base, it must not be completely filled and it must be organized as a kind of “ping-pong” structure. Then one part of memory could be used to obtain current signatures to be sought in incoming traffic, while other would be used for refreshing the data base. At certain moments of time functions of the parts would need to be switched so that always the freshly updated one is used.

2.5 Additional Considerations

The essence of the speedup must be contained in signature processing engine. This part of device can be seen as working on network layer, as it receives IP packets and than compares them with signatures. As packet is composed of header and payload parts there must be made a consideration on how to process the signatures, whether whole signature, including header and payload parameters should be processed together, or the processing engines for both parts must be split. In order to obtain correct answer all the factors must be included: format of signatures, placement of signatures in design, and the most important one which algorithm is being used for string comparison.

Signature is actually a definition of conditions that a packet must fulfill in order to trigger an action. Almost standard set of signatures, and without the doubt excellent starting point for further signature generation, is the signature set available from SNORT. It is updated regularly and as one could note in Table 1 more than 97 percent of the rules contains a content field which actually meant that payload of packet needs to be processed. That re-brings to the light the header payload partition of packet, already mentioned, and how it should be done in order to achieve and potentially surpass the requirements.

Table 1: Achieved throughputs

Number of different Snort rules	6594
Number of rules containing content part	6440

In the case that rules are processed in cascade, the header first and than later upon materializations of header conditions described by header rules, payload rules, it might be possibly to keep the signature data base cleverly organized so that only specific payload rule set must be checked upon header rule positive match. This as cascading technique might increase system latency. As other option header and payload parts might be processed simultaneously. This would probably increase the latency, but both the options have to be studied carefully. Finally, it might be possible to design a rule set comprised of somehow merged header and payload rule conditions. What would be the consequence of such choice will remain unknown until the suggestion is made.

Making a right, optimal, choice on this issue is additionally complicated by the influence of search algorithm that is to be used, thus the potentials for research and classification, valorization, of all potential choices is quite opened.

Still, it is important to know that modern FPGA chips theoretically promise operation frequency of up to 550MHz [10]. Then the clever strategy of parallelization, which hardware permits by its nature, might lead to processing time less than 67.2ns as mentioned in previous analysis in spite of all the issues raised up to this point. For example, if we could process 32bits per clock cycle of 400MHz the delivered throughput would be 12.8 Gbps, which is equal to 52.5ns per worst case packet. Here the key words are “if we could”, leaving the answer yet to come.

3. Existing Algorithms Overview

The problem of processing data, in comparison sense, has different solutions. We will provide an insight on spectrum of those algorithms, some with software background implemented in hardware, and some only hardware implementable. Historically software based brute force algorithm as a first tentative that has appeared. Based on comparison of each and every character of both compared strings it needed improvements. It led to creation of mentioned KMP, Aho-Corasick and BM algorithms, some of which have important hardware implementations we are interested in viewing at, together with other hardware only based approaches, such as Bloom filter chips or Ternary Content Addressable Memory (TCAM).

3.1 KMP Approach

KMP uses two facts to speed up the brute force approach. The first is the fact that it might be possible to skip some character to character comparisons in the case of mismatch, according to partial match obtained before mismatch occurred. The second one is that some comparisons may also be skipped depending on a different type of previous partial matches. The result is somewhat similar to a kind of finite automata preprocessing of a shorter of strings being compared so that every match or mismatch corresponds to certain amount of jumps, actually to comparisons being skipped. This final characteristic provides a basis for hardware implementations of KMP based comparison machines for IDS implementations, which have been covered in the work of Baker and Prasanna [13], promising throughput of up to 2.4Gbps. This solution is also highly flexible and exploits potentials of hardware parallelism, while being good option for both FPGA and ASIC destination platforms.

3.2 Discrete Comparators Approach

It has already been mentioned how by parallelism in architecture throughput increase can be achieved. Excellent example of that approach is work presented by Sourdis and Pnevamatikatos [14]. They have used “deep-pipelining” optimized for FPGA devices, in their case Xilinx, to achieve maximum clock frequency, which consist in careful distribution of resources on the level of each CLB of FPGA device. Together with that approach they have applied parallel comparators matching the same character set on different, partially overlapping, positions in the incoming packet. These two approaches have led to throughput values for presented design of 12.672Gpbs in one particular case.

The work of Sourdis and Pnevmatikos is based on principle introduced earlier by Cho et al. [12], also deriving a system comprised of comparison units for each and every signature. For the starter work throughput achieved, having a value of 2.88 Gpbs, was extremely good.

3.3 Finite State Machines Approaches

This type of approach is very suitable for hardware because of regularity it imposes with its use. In string matching domain shifting through the states of single FSM is correspondent to the process of consecutive matching of characters. In signature context, every signature would have a corresponding FSM, which can be than implemented on hardware platform to work all together in parallel. This type of approach can be found in the work of Sidhu and Prasanna where they also use regular expressions [15]. The throughput achieved is slightly lower that 750Mbps.

A FSM constructed with capability of comparing multiple strings is a consequence of Aho-Corasick algorithm. The algorithm has two phases, preprocessing one, in which construction of multi-string FSM is taking place, and later on use of constructed automaton for comparison purposes. Most efficient implementation of this approach can be found in work of Tan and Sherwood [16], where they use machine splitting technique to improve memory demands of this algorithm. They also provide excellent solution for signature updating and the throughput is slightly higher than 10Gbps rate.

Another work based on the usage of finite automata is present in the work of Katashita et al. [19]. By implementing elimination of redundant states in machines generated by Aho-Corasick approach together with proper pipelining they managed to implement whole snort rule dataset on a single FPGA chip with throughput of up to 16.488 Gbps.

State machine approach might be a good option for either ASIC or FPGA implementations though some works like the work of Clark and Schimmel [21] are FPGA specific. I the mentioned work they presented a way to implement multi-character decoder nondeterministic finite automata with throughputs around 2 Gbps.

3.4 Content Addressable Memory

From the name of the memory can be seen that this special storage architecture implements address lookup by content, if one could even talk about address lookup. Benefits from such approach are obvious especially in content searching domain. Intrusion detection applied approaches based on this type of memories use intelligent schemes for storing and retrieving data providing throughputs bigger that values limited by constraints such as access time of CAM memories of 4ns corresponding to throughput of 2Gpbs, and are capable of achieving throughputs of up to 12.35Gbps.[18]

3.5 Bloom Filter Structure

Bloom filter is a structure that compresses amount of strings by transforming them to set of values obtained after hashing of original strings. It is actually composed of those “after hashing” values while the same hashing functions used in a process of forming of the filter are later used to establish dependency to the filter. By using these structures, problem of exact string matching is translated to the problem of establishing dependency to the structure. The usage of Bloom filter for signature comparison for intrusion detection domain has been suggested by Dharamapurikar et al. [17]. The structure implemented on FPGA has been able to achieve the throughput of 2.12Gbps.

3.6 Other Present Approaches

Specific approach present in the work of Singaraju et al. [20] is based on FPGA implementation of specialized “signature match processor”. Modular architecture based on content addressable memories, allows and implementation of signature based comparison engine as a generic approach for development of misuse intrusion detection/protection systems. Throughputs achieved by this specialized processor are up to 3.96 Gpbs. Usage of parallelization increases the performance, yet the architecture is dependable on the number of signatures entries implemented.

The work of Sourids et al. [22] shows potentials of usage of perfect hashing schemes in IDS domain with throughputs of up to 5.7 Gbps. The essence of this kind of approach is in the fact that by using hashing functions it is possible to quickly select potential set of matches and then by direct comparison the exact match can be found.

3.7 Summary

An obvious criterion for selection of the string search algorithm is the promised throughput value. Algorithms mentioned have been sorted according to upper. This is shown in Table 2.

Table 2: Achieved throughputs

Throughput [Gbps]	***Algorithm***
16.488	Katashita et al. [19]
12.35	Weinsberg et al [18]
12.672	Sourdis and Pnevamatikatos [14]
10.074	Tan and Sherwood [16]
5.7	Sourids et al. [22]
3.96	Singaraju et al. [20]
2.88	Cho et al. [12]
2.4	Baker and Prasanna [13]
2.12	Dharamapurikar et al. [17]
2	Clark and Schimmel [21]
0.75	Sidhu and Prasanna [15]

Independently of that the general impression is that acceptable solution is yet to come, as most of the approaches present have different problems that might be summed in the fact that none seems to have been tested in a real 10Gbps networking environment. The problem might be that there are quite a few high speed networking environments available, which points to another direction a necessity for development of objective 10Gbps security device testing platform.

Earlier discussion on importance of signature updating process has also to be considered when choosing an algorithmic approach. Actually none of presented solutions does provide swift updating mechanisms, the only exception being the work of Tan and Sherwood [16], up to some level. Absence of this shows another research direction, especially when high speed solutions are sought. In that light, maybe future improvements of FPGA technologies and mentioned dynamic configuration characteristics might be a good option.

The biggest concern arises from the fact that the very algorithm used for signature comparison is only a part of complete system, and that combination of factors such as high speed PCB design, high speed functioning, including high clock speeds, high speed updating capabilities, high speed memory accesses needed for some options can lead to extremely complex and demanding overall system design.

4. Conclusions, Potentials and Perspectives

Complexity of design of one hardware based network intrusion detection/protection systems is undisputable. Yet, besides highlighting the problems and existing potential solutions the goal of pointing to some other issues, not so directly related to the topic, has been tackled. Inexistence of test equipment, signature update mechanisms, maybe even global signature data base, are just some of the issues raised, waiting to be properly solved. Only upon overcoming most of the obstacles it is expected to have significant advance in this field of security applicable hardware design.

Another thing that has to be mentioned is certainly the fact that a signature based system is able to recognize only the things present as signatures in its constantly updated knowledge base. This means that in the case of malicious traffic not described by signature a system being protected would still be vulnerable. The proposed HBNIPS system might need an additional part, to tackle this issue. Something like test environment for suspicious packets. What should that look like, how can a signature system recognize a suspicious traffic to be passed to that additional unit, can that be done at all? These are some of the things that might be considered in future and be a good research topic.

In the end, it is worth mentioning that up-to-date commercial intrusion prevention hardware based solutions are working with throughputs of up to 4.4Gbps [4]. This can probably show a potential of commercialization of high speed hardware based intrusion detection technology and justify the need for improvements in the field, while having in mind all the problems foreseen.

Acknowledgments

This work has been partialy supported by European Comission through EU FP6 project IST 027095 and Spanish Government through TIC2003-09061-C03-02 project.

References

[1] www.snort.org

[2] The Snort Project, *"Snort Users Manual 2.6.0,"* May 2006.

[3] Top Layer Networks, *"IPS 5500, Intrusion Prevention System Data Sheet."*

[4] D.E. Knuth, J.H. Morris, and V.R Pratt, *"Fast Pattern Matching in Strings,"* SIAM Journal on Computing, Vol 6, No 2, pp. 323-350, June 1977.

[5] R.S. Boyer and J.S. Moore, *"A Fast String Searching Algorithm,"* Communications of the ACM, Vol 20, No 10, pp. 66-72, Oct. 1977.

[6] V. Aho and M. J. Corasick, *"Efficient string matching: An aid to bibliographic search,"* Communications of the ACM, 18(6):333–340, 1975.

[7] M. Fisk and G. Varghese, *"An Analysis of Fast String Matching Applied to Content-based Forwarding and Intrusion Detection,"* Techical Report CS2001-0670, University of California - San Diego, 2002.

[8] K. G. Anagnostakis, E. P. Markatos, S. Antonatos and M. Polychronakis, "*E2xB: A domain specific string matching algorithm for intrusion detection,"* Proceedings of the 18th IFIP International Information Security Conference (SEC2003), May 2003.

[9] Spirent Communications, *"How to Test 10 Gigabit Ethernet Performance,"* White Paper, May 2005.

[10] Xilinx Inc, *"Virtex-5 Datasheet: DC and switching characteristics,"* Aug. 2006.

[11] http://hogwash.sourceforge.net/

[12] Young H. Cho, Shiva Navab, W.H. Mangione-Smith, *"Specialized Hardware for Deep Network Packet Filtering Source,"* Proceedings of the 12th International Conference on Field-Programmable Logic and Applications, pp. 452 – 461, 2002.

[13] Z. K. Baker and V. K. Prasanna, *"Time and area efficient pattern matching on FPGAs,"* Proceeding of the 2004 ACM/SIGDA 12th International Symposium on Field Programmable Gate Arrays, pp. 223–232, Feb, 2004.

[14] Ioannis Sourdis and Dionisios Pnevmatikatos, *"Fast, Large-Scale String Match for a 10 Gbps FPGA-based Network Intrusion Detection System,"* Proceedings of the 13th International Conference on Field Programmable Logic and Applications (FPL2003), Lisbon, September 2003.

[15] R. Sidhu and V.K. Prasanna, *"Fast Regular Expression Matching using FPGAs,"* Proceedings of IEEE Symposium on Field-Programmable Custom Computing Machines, pp. Apr. 2001.

[16] Lin Tan, T. Sherwood, *"A high throughput string matching architecture for intrusion detection and prevention,"* Proceedings of the 32nd International Symposium on Computer Architecture, ISCA 2005, pp. 112-122, June 2005.

[17] Sarang Dharmapurikar, Michael Attig, John Lockwood, *"Design and Implementation of a String Matching System for Network Intrusion Detection using FPGA-based Bloom Filters"*, technical report, WUCSE-2004-12, Mar, 2004.

[18] Y. Weinsberg, S. Tzur-David, D. Dolev and T. Anker, *"High Performance String Matching Algorithm for a Network Intrusion Prevention System (NIPS),"P*roceedings of IEEE Workshop on High Performance Switching and Routing (HPSR'06), June 2006.

[19] Toshihiro Katashita, Atsusi Maeda, Kenji Toda, Yoshinori Yamaguchi, *"A Method of Generating Highly Efficient Matching Circuit for Intrusion Detection,"* Proceedings of FPL2006, Aug. 2006.

[20] J. Singaraju, , L. Bu, J.A. Chandy, *"A Signature Match Processor Architecture for Network Intrusion Detection,"* Proceedings of IEEE Symposium on Field-Programmable Custom Computing Machines, pp. 235-242, Apr. 2005.

[21] Christopher R. Clark , David E. Schimmel, *"Scalable Pattern Matching for High Speed Networks,"* Proceedings of the 12th FCCM, pp. 249-257, Apr. 2004.

[22] Ioannis Sourdis, Dionisios Pnevmatikatos, Stephan Wong, and Stamatis Vassiliadis, *"A Reconfigurable Perfect-Hashing Scheme for Packet Inspection,"* Proceedings of the 15th FPL, Aug. 2005.

Vukašin Pejović received the B.S. degree in the School of Electrical Engineering from Belgrade University, Serbia in 2004. Worked as R&D engineer on FPGA technologies for a year, and joined the Department of Electronics at ETSIT of Technical University of Madrid, where he is currently pursuing a PhD title. His topics of interest are intrusion prevention, detection, generation and classification, networking and hardware architectures for mentioned fields.

Slobodan Bojanić picture and CV not available.

Carlos Carreras Vaquer picture and CV not available.