УДК 004.056:004.274

Hilhurt S. Ya., Kiev

APPLICATION OF FPGA-BASED RECONFIGURABLE ACCELERATORS FOR NETWORK SECURITY TASKS

Abstract. The paper presents a review of the FPGA-based implementations aimed to solve the network security tasks. The main directions of research in this area are defined. Typical examples of existing application are given.

Introduction

Securing the network infrastructure has become a high priority problem because of its great importance for data protection, e-commerce and national security [1]. The increase in both the number and sophistication of network attacks requires more strong and effective solutions. Recent years, the performance of security software doesn't keep up with the amount of data to be processed which grows noticeably every day.

The known solution to overcome this performance gap is to use hardware units to speed up the security tasks realization. The reconfigurable accelerators based on programmable logic integrated circuits are the most appropriate tool to achieve this goal. Reconfigurable devices of type Field Programmable Gate Array (FPGA) have become increasingly popular for this purpose, because they combine the flexibility of software and high performance of hardware due to powerful possibilities of paralleling [2].

Analysis of the recent investigations and publications shows that applying hardware solutions based on reconfigurable devices to solve network security tasks has become a relevant subject. Many researchers successfully work over this problem, many achievements have been reported. However, there are few publication aimed to analyze and systematize these investigations.

The objective of this paper is to present a short review of the known approaches using reconfigurable hardware in network security. The main goal is not to cover completely, but to note the fundamental and key researches conducted in this area.

1. Performance Acceleration Need and Requirements

The increasing number of attacks as well as steady growth of network throughput makes software solutions for network security less effective because of performance limitations of the off-the-shelf processors which functioning is naturally sequential. Under these conditions, it seems to be a good idea to use hardware devices to enable real-time implementation of sophisticated security functions.

Analyzing current network security applications, it can be put in the forefront following characteristics as requirements for security system [2]:

- real-time protection;
- flexible updating;
- well controlled scalability.

With the rapid development of technology, hardware devices become increasingly interesting in network security.

2. FPGA as a Solution

Because off-the-shelf processors have become overburdened by the bandwidth expansion of network connections the performance of purely software-based applications is often inadequate for practical deployment. To enable real-time implementation of security functions the shift towards hardware-based implementation is necessary.

On the other hand, it is also important to maintain the system flexibility equal to one that general purpose computers have. Since security threats are constantly evolving, defense systems require appropriate update mechanisms.

Thus, reconfiguration possibility of hardware implementation is also very important for security applications and must be as similar to software programmability feature as possible.

A Field-Programmable Gate Array (FPGA) is a type of general-purpose, multi-level programmable logic device that can be programmed by end users. FPGA devices have commonly been proposed because they feature both the flexibility of software and the high performance of hardware [3 – 8]. That is why the reconfigurable accelerator based on FPGA became a popular platform for network security applications [9 – 12].

Many FPGA-based infrastructure security solutions are developed for network traffic analysis, packet classification, pattern matching, worm and DDoS containment. Network Intrusion Detection Systems (NIDS) and Network Intrusion Prevention Systems (NIPS) were also created for the network security using reconfigurable platform.

Below, let us consider the main groups of FPGA-based applications in the network security area.

3. Packet Classification

A network packet contains information of two kinds: control data (packet header) and user data (payload). Packet header fields are essentially constant in length and appear at fixed locations in the packet, while packet payloads can be variable in length with no fixed format. Respectively there are two main techniques to handle network packets [13]:

- packet classification – processing of packet headers;
- deep packet inspection (DPI) – processing of packet payload.

Packet classification is an important technology for network security. Intensive research in this area has been carried out in recent years and many algorithms and architectures have been developed for this purpose. The major

packet classification techniques according to the work [14] are:

- exhaustive search;
- decision tree;
- decomposition;
- tuple space.

One of the first hardware implementations of packet classification called Flow Classifier was reported in the paper [15]. It was a module employing a hybrid hardware-software architecture performing straightforward operations.

A hardware-based approach for network intrusion detection combining the advantages of ternary content addressable memory and the bit vector algorithm was then introduced [16].

A Gigabit packet filter based on FPGA was introduced in the paper [17]. The high performance of hardware allowed the linear-search based algorithm to be used under multi-parallel operation mode.

4. Pattern Matching

As mentioned above, unlike packet classification deep packet inspection is more challenging since the payload contents can be any format that is determined by the applications. Pattern matching is the most popular approach for payload inspection [18, 19]. Such type of DPI is one of the fundamental functions for security purposes in the network infrastructure, including intellectual firewalls and Intrusion Detection Systems (IDS).

The principle of pattern matching is to compare incoming packets against a large number of patterns (or signatures) in order to make a decision if an attack takes place [20].

There are several FPGA-based pattern matching techniques such as, in particular, content addressable memory (CAM), finite automata (FA), regular expression matching (REM) and Bloom-filter. They will be discussed in the following sub-sections.

4.1. Content Addressable Memory Technique. Content addressable memory (CAM) returned interest to itself in recent years due to their ability to fulfill fast pattern comparison. Unlike traditional random access memory (RAM) CAM allows to check the presence of certain information in it without memory addressing [21].

A multiple-pattern matching approach based on so called ternary content addressable memory (TCAM) was proposed in paper [22] which added a "don’t care" state in addition to the traditional "0" and "1" states used in classical CAM. Therefore, this approach can handle complex patterns, in particular, arbitrarily long patterns, correlated patterns and patterns with negation.

High performance CAM based on discrete comparators was proposed in [23] and advanced as Decoded CAM (DCAM) in [24]. Demonstrating excellent velocity characteristics this approach unfortunately have high resource requirements.

To solve this problem a method based on binary decision diagram (BDD) CAM was introduced in [25]. The pattern-matching engine proposed uses the tree-based CAM structure. Exploiting logic optimizations for multiple strings in the form of BDD, the approach involves hardware sharing at the bit level.

4.2. Finite Automata (FA) Technique. A hardware approach frequently used for pattern matching based on finite-automata (FA) – a computation model that is characterized as having a finite number of states [26]. Applying to the pattern matching task FA processes input string to either accept or reject it. A successful matching occurs when the string of input characters match the certain pattern on any path of the FA that leads from the initial state to the final state.

Finite Automates provide low cost of design at relatively not high throughput. Due to the input data are processed in FA sequentially, the overall latency increases proportionally to the number of patterns. Moreover, FA implementations are restricted in operating frequency by the complexity of combinational logic of state transitions, where complex expressions result in multilevel implementations.

There are two types of finite automata reported for security implementation: deterministic FA (DFA) and non-deterministic FA (NFA). DFAs are usually used for matching against a fixed-pattern rule set whereas NFA are more frequently employed for regular expression matching (REM).

A large rule set using DFA approach was supported in the work [27]. Authors employed the Aho-Corasick algorithm for pattern matching in a parallel manner, where each sub-engine processes a part of traffic. To allow big number of large patterns, the rule set are stored in off-chip memory.

In the paper [28] the DFA architecture realizing the Aho-Corasick algorithm (AC-DFA) was extended to support multi-character input per clock cycle by mapping a compressed structure of finite automata onto multiple pipelines.

Simulation using real-life NIDS rule sets showed that a 8-stage pipeline can remove more than 99% of the transitions in the graph of original AC-DFA [29]. The implementation on a state-of-the-art reconfigurable devices shows allows such architecture to realize the full set of string patterns from the latest NIDS Snort rule sets [30, 31] using a single FPGA chip.

4.3. Regular Expression Matching. A regular expression (RE) is a non-fixed pattern that can match several strings simultaneously. Atomic element of RE is a character or a special symbol such as concatenation "(.)" or union "|". The question sign "?" means presence or absence of an expression in the string etc. For example, an expression "a(b|c)?" designate the following combinations of characters: "a", "ab" or "ac". Metacharacter Kleene-Star "*" denotes repetition of an expression any number times, including zero [32].

In network security, REs are widely used to present attack pattern because of its flexibility and compactness. Several variations of an attack can be defined by using one regular expression.

There are relatively few studies being reported for DFA-based implementation of regular expression matching (REM) [33]. NFAs are more

frequently used as platform for hardware implementation of REM [34, 35].

An NFA-based regular expression matching engine for NIDS on FPGA called ENREM proposed in paper [36]. Besides sharing common prefix, a new infix/suffix sharing model introduced which successfully handle patterns overlap problem.

4.4. Bloom Filter Technique. A Bloom filter is a specific data structure based on hash functions which allows to represent a set of elements with probabilistic membership queries [37]. An important property of Bloom filters is that the computation time for pattern searching is independent of the number of patterns contained in the database. The required memory space for hashed pattern storage is also independent of the number of patterns.

A basic structure using Bloom filter for packet payload inspection has been reported by a research team from Washington University [38 – 40].

An automatic compilation framework to automatically generate VHDL code for Bloom-filter based intrusion detection system on FPGA was proposed in [41].

To resolve the scalability issue a graph-based partition technique decomposing a large pattern database into certain basic pattern sets was adopted in [42]. Further optimization approach by combining graph-based partitioning and tree-based matching of large pattern databases was realized in [43].

5. Network Worms

Network worms and Distributed Denial-of-Service (DDoS) attacks are serious security threats towards the network infrastructure [44].

Security strategy of defense system may vary depending on the specific attack stage in progress. According to [2] attack stages can be categorized as:

- pre-attack;
- prevention;
- under-attack containment;
- post-attack update.

Most applications against present-day worm attacks are applied during the under-attack stage. Good masking and high outbreak speed make worms difficult to prevent at the pre-attack stage. Due to the property of self-propagation, worms can spread quickly over the whole network within a short time.

Probably the first hardware-based implementation of a worm containment system was reported in [45]. The system consists of three major modules: inspector, updater and manager, each of which performs specific functions.

An OS independent approach to zero-day worm detection and containment was proposed in [46]. Questions of its hardware implementation using FPGA-based accelerators are investigated in [47].

An approach for worm containment based on anomaly detection was proposed in [48]. In this paper the idea that a worm detection system should keep looking for frequently occurring contents was evolved [49].

6. DDoS Attacks

Security tools against DDoS attack are more complicated compared with worm containment. It can be defined two phase discipline of a typical DDoS [50]:

- the first phase is to compromise a large number of computers and recruit them into a zombie army;
- the second phase is to indirectly launch DDoS attacks towards a specific target through these zombies.

It is difficult to detect malicious activities of such type using signature-based technique because legitimate data packets, such as SYN-ACK, RST or ICMP packets in TCP flows, can be used for DDoS attacks [51]. On the other hand, the flooding property of DDoS attacks makes it possible to use anomaly detection. A work [52] is just based on the inspection of network traffic rates. A double token bucket mechanism was applied in it for bandwidth control.

One of approaches proposed for DDoS attack defense is based on its spectral features. A hardware application using the Power Spectral Density (PSD) analysis was developed [53].

The joint attack of worms and DDoS is more destructive. If attackers take the advantage of worm spreading to comprise thousands or millions of computers and to launch a DDoS attack, none network infrastructure will be able to resist such an attack without powerful protection.

Because of the importance of defending against Internet worms and DDoS attacks there are many researchers who address approaches based on hardware. Nevertheless results of this work are respectively modest.

7. TCP Stream Preprocessing

The transmission control protocol (TCP) is one of the widespread protocols used in the network infrastructure. More than 85% of traffic exploits the TCP protocol in global network environments [54, 55]. Many network security services including packet inspection, content-based routing, internet worm detection, DDoS attack resistance and spam removal require high-speed TCP traffic manipulation as an auxiliary operation allowing increasing the performance of applications mentioned.

FPGA-based solutions give possibility to separate TCP stream preprocessing from security functions due to modular design of the hardware. Since many security functions require TCP stream preprocessing, separating this function allows subsequent applications to be relieved of this task, so more resources become free for security functions.

A TCP stream reassembly and state tracking module for NIDS that implements TCP processing functions in a high performance reconfigurable network card was proposed in paper [56].

In [57] a reconfigurable hardware architecture to replace a software based STREAM4 preprocessor for a popular NIDS system Snort was implemented [58], which performs TCP stateful inspection and reassembly functions. By analyzing

the current and past packets of TCP stream, it is possible to predict what will happen next. This function is very helpful for detection of certain abnormal activities.

A TCP-splitter as a lightweight, high performance hardware module was designed in paper [59] to provide a consistent TCP data stream treatment for client application systems.

Conclusion

In this paper, the FPGA-based hardware implementations aimed to solve the network security tasks was reviewed. Such directions of research were defined:

- packet classification;
- pattern matching including regular expression matching;
- internet worms detection;
- DDoS attack prevention;
- TCP stream preprocessing.

Typical examples of state-of-the-art application realized by leading scientists in this area are observed.

Reconfigurable FPGA-based accelerators allow combining the flexibility of software and high performance of hardware. That is why application of reconfigurable devices for network security tasks is so widespread solution.

It is necessary to note that network attacks of different type may be applied together. For example, worms spreading is often followed by DDoS attack. On the other side, different approaches and techniques in security tools designs can also be combined in various ways.

1. *T. Grandison, M. Sloman* "A survey of trust in internet applications" IEEE Communications Surveys & Tutorials, vol. 3, no. 4, pp. 2–16, 2000.
2. *H. Chen, Y. Chen, D.H. Summerville* "A Survey on the Application of FPGAs for Network Infrastructure Security" IEEE Communications Surveys and Tutorials, 2011, pp.541-561.
3. *S. M. Trimberger* "Field-Programmable Gate Array Technology" Kluwer Academic Publishers, 1994.
4. *P. Garcia, K. Compton, M. Schulte, E. Blem, W. Fu* "An overview of reconfigurable hardware in embedded systems" EURASIP J. Embedded Syst., vol. 2006, no. 1, pp. 13–13, 2006, 1288236.
5. *Гильгурт С.Я.* Применение типовых устройств на базе программируемой логики для решения вычислительных задач // Труды II международной конф. «Параллельные вычисления и задачи управления» PACO'2004 памяти Е.Г. Сухова. Москва, 4–6 окт. 2004 г. – М.: Институт проблем управления им. В.А. Трапезникова РАН, 2004. – С. 514–530.
6. *Гильгурт С.Я.* О применении реконфигурируемых унифицированных вычислителей для решения научно-технических задач / Параллельные вычислительные технологии (ПаВТ'2008) // Труды международной научной конференции (Санкт-Петербург, 28 января – 1 февраля 2008 г.). – Челябинск: Изд. ЮУрГУ, 2008. – С. 358–363.
7. *Гильгурт С.Я.* Обзор современных реконфигурируемых унифицированных вычислителей // Моделювання та інформаційні технології. Зб. наук. пр. ІПМЕ НАН

України. – Київ: 2008. – Вип. 49. – С. 17–24.
8. *Гильгурт С.Я.* Реконфигурируемые вычислители. Аналитический обзор // Электронное моделирование. – 2013. – Т.35, № 4. – С. 49–72.
9. *Гильгурт С.Я.* Анализ применения реконфигурируемых устройств в системах обнаружения вторжений // Тез. доп. Міжнар. наук.-техн. конф. «Моделювання-2010». Т.1. – Київ: Інститут проблем моделювання в енергетиці ім. Г.Є.Пухова НАН України, 2010. – С. 260–267.
10. *Гильгурт С.Я.* Обзор возможностей реконфигурируемых устройств для применения в компьютерной безопасности // Зб. наук. пр. ІПМЕ НАН України. – Київ, 2010. – Вип. 55. – С. 117–124.
11. *Коростиль Ю.М., Гильгурт С.Я.* Принципы построения сетевых систем обнаружения вторжений на базе ПЛИС // Моделювання та інформаційні технології. Зб. наук. пр. ІПМЕ НАН України. – Київ, 2010. – Вип. 57. – С. 87–94.
12. *Гільгурт С.Я.* Множинне розпізнавання рядків у системах виявлення вторгнення на базі реконфігуровних обчислювачів / // Сучасні комп'ютерні системи та мережі: розробка та використання : матеріали 5-ої Міжнар. наук.-техн. конф. ACSN-2011, 29 вересня – 01 жовтня 2011, Львів, Україна. – Л. : Вид-во Нац. ун-ту «Львів. політехніка», 2011. – С. 54–56.
13. *J. Kurose, K. Ross* "Computer Networking: A Top-Down Approach Featuring the Internet" Addison-Wesley Longman Publishing Co., Inc., 2002.
14. *D.E. Taylor* "Survey and taxonomy of packet classification techniques" ACM Comput. Surv., vol. 37, no. 3, pp. 238–275, 2005.
15. *S. Yusuf, W. Luk, M. Sloman, N. Dulay, E. Lupu* "A combined hardware-software architecture for network flow analysis" IEEE International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA'05), Las Vegas, USA, 2005.
16. *H. Song, J.W. Lockwood* "Efficient packet classification for network intrusion detection using FPGA" Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays. Monterey, California, USA: ACM, 2005, pp. 238–245.
17. *J. Loinig, J. Wolkerstorfer, A. Szekely* "Packet filtering in gigabit networks using FPGAs" Austrochip 2007 - Proceedings of the 15th Austrian Workshop on Microelectronics (2007), 2007, pp. 53 – 60.
18. *Y.H. Cho, W.H. Mangione-Smith* "Deep packet filter with dedicated logic and read only memories" Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on, 2004, pp. 125–134.
19. *P.-C. Lin, Y.-D. Lin, T.-H. Lee, Y.-C. Lai* "Using string matching for deep packet inspection" Computer, vol. 41, no. 4, pp. 23–28, 2008.
20. *Z.K. Baker, V.K. Prasanna* "A methodology for synthesis of efficient intrusion detection systems on FPGAs" Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on, 2004, pp. 135–144.
21. *S. Guccione, D. Levi, D. Downs* "A reconfigurable content addressable memory" Proceedings of the 15 IPDPS 2000 Workshops on Parallel and Distributed Processing. Springer-Verlag, 2000, pp.882–889.
22. *F. Yu, R.H. Katz, T.V. Lakshman* "Gigabit rate packet patternmatching using tcam" Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on, 2004, pp. 174–183.
23. *I. Sourdis, D. Pneumatikatos* "Fast, large-scale string match for a 10gbps FPGA-based network intrusion detection system" Lecture notes in computer science. Berlin; New York: Springer Berlin / Heidelberg, 2003, vol. Volume 2778/2003, pp. 880–889.
24. *I. Sourdis, D. Pnevmatikatos* "Pre-decoded cams for efficient and high-speed nids pattern matching" Field-Programmable Custom Computing Machines, 2004. FCCM 2004.

12th Annual IEEE Symposium on, 2004, pp. 258–267.
25. *S. Yusuf, W. Luk* "Bitwise optimised cam for network intrusion detection systems" Field Programmable Logic and Applications, 2005. International Conference on, 2005, pp. 444–449.
26. *M.V. Lawson* "Finite Automata" Boca Raton: Chapman & Hall/CRC, FL, 2004, 144pp.
27. *H. Bos, K. Huang* "A network instruction detection system on ixp1200 network processors with support for large rule sets" Technical Report 2004-02. Leiden Univeristry, 2004.
28. *M. Alicherry, M. Muthuprasanna, V. Kumar* "High speed pattern matching for network IDS/IPS" ICNP ’06: Proceedings of the 2006 IEEE International Conference on Network Protocols. IEEE Computer Society, 2006, pp. 187–196.
29. *Jiang W., Prasanna V.* Scalable Multi-Pipeline Architecture for High Performance Multi-Pattern String Matching // IEEE International Parallel and Distributed Processing Symposium (IPDPS '10), April 2010.
30. *Давиденко А.Н., Гильгурт С.Я., Сабат В.И.* Аппаратное ускорение алгоритмов сигнатурного обнаружения вторжений в открытой системе информационной безопасности Snort // Моделювання та інформаційні технології. Зб. наук. пр. ІПМЕ НАН України. – Київ, 2012. – Вип. 65. – С. 94–103.
31. *Коростиль Ю.М., Гильгурт С.Я., Назаренко О.М.* Анализ базы данных системы информационной безопасности Snort и вопросы быстродействия // Моделювання та інформаційні технології. Зб. наук. пр. ІПМЕ НАН України. – Київ, 2012. – Вип. 66. – С. 77–84.
32. *R. Sidhu, V.K. Prasanna* "Fast regular expression matching using FPGAs" Field-Programmable Custom Computing Machines, 2001. FCCM ’01, 2001, pp. 227–238.
33. *J. Moscola, J. Lockwood, R.P. Loui, M. Pachos* "Implementation of a content-scanning module for an internet firewall" Field-Programmable Custom Computing Machines, 2003. FCCM 2003. 11th Annual IEEE Symposium on, 2003, pp. 31–38.
34. *C.R. Clark, D.E. Schimmel* "Efficient reconfigurable logic circuits for matching complex network intrusion detection patterns" Proc. 11th ACM/SIGDA Int. Conf. Field-Program. Logic Appl. (FPL), 2003, p. 956.
35. *C.R. Clark, D.E. Schimmel* "Scalable pattern matching for high speed networks" Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on, 2004, pp. 249–257.
36. *T.T. Hieu, T.N. Thinh, S. Tomiyama* "ENREM: An efficient NFA-based regular expression matching engine on reconfigurable hardware for NIDS" Journal of Systems Architecture - Embedded Systems Design 59(4-5): 202-212 (2013)
37. *S. Dharmapurikar, J.W. Lockwood* "Deep packet inspection using parallel bloom filters" Micro, IEEE, vol. 24, no. 1, pp. 52–61, 2004.
38. *M. Attig, S. Dharmapurikar, J. Lockwood* "Implementation results of bloom filters for string matching," in Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on, 2004, pp. 322–323.
39. *S. Dharmapurikar* "Design and implementation of a string matching system for network intrusion detection using FPGA-based bloom filters" in Proc. of 12 th Annual IEEE Symposium on FieldProgrammable Custom Computing Machines, 2004.
40. *S. Dharmapurikar, J.W. Lockwood* "Fast and scalable pattern matching for network intrusion detection systems" Selected Areas in Communications, IEEE Journal on, vol. 24, no. 10, pp. 1781–1792, 2006.
41. *D.C. Suresh, Z. Guo, B. Buyukkurt, W.A. Najjar* "Automatic compilation framework for bloom filter based intrusion detection" Lecture notes in computer science. Berlin; New York: Springer-Verlag, 2006, pp. 413–418.

42. *Z.K. Baker, V.K. Prasanna* "A methodology for synthesis of efficient intrusion detection systems on FPGAs" Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on, 2004, pp. 135–144.
43. *Z.K. Baker, V.K. Prasanna* "High-throughput linked-pattern matching for intrusion detection systems" Architecture for networking and communications systems, 2005. ANCS 2005. Symposium on, 2005, pp. 193–202.
44. *M. Cai, K. Hwang, Y.-K. Kwok, S. Song, Y. Chen* "Collaborative internet worm containment" Security & Privacy, IEEE, vol. 3, no. 3, pp. 25–33, 2005.
45. *J.W. Lockwood, J. Moscola, M. Kulig, D. Reddick, T. Brooks* "Internet worm and virus protection in dynamically reconfigurable hardware" Military and Aerospace Programmable Logic Device (MAPLD), 2003, p. 10.
46. *U. Savagaonkar, R. Sahita, G. Nagabhushan, P. Rajagopal, D. Durham* "An OS Independent Heuristics-Based Worm-Containment System" Intel Corp., 2005.
47. *Гильгурт С.Я.* Особенности применения реконфигурируемых вычислителей для аппаратной защиты информационных систем // Зб. наук. пр. ІПМЕ НАН України. – Вип. 38. – Київ: 2007. – С. 36–41.
48. *B. Madhusudan, J. Lockwood* "Design of a system for real-time worm detection" Proceedings of the High Performance Interconnects, 2004. on Proceedings. 12th Annual IEEE Symposium. IEEE Computer Society, 2004, pp. 77–83.
49. *S. Singh, C. Estan, G. Varghese, S. Savage* "Automated worm fingerprinting" Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6. San Francisco, CA: USENIX Association, 2004, pp. 4–4.
50. *T. Peng, C. Leckie, K. Ramamohanarao* "Survey of network-based defense mechanisms countering the dos and DDoS problems" ACM Comput. Surv., vol. 39, no. 1, p. 3, 2007.
51. *R.K.C. Chang* "Defending against flooding-based distributed denial-of-service attacks: a tutorial" Communications Magazine, IEEE, vol. 40, no. 10, pp. 42–51, 2002.
52. *J.-T. Oh, S.-K. Park, J.-S. Jang, Y.-H. Jeon* "Detection of DDoS and ids evasion attacks in a high-speed networks environment" International Journal of Computer Science and Network Security, vol. 7, no. 6, pp. 124–131, 2007.
53. *H. Chen, Y. Chen* "A novel embedded accelerator for Inline detection of shrew DDoS attacks" Networking, Architecture, and Storage, 2008. NAS '08. International Conference on, 2008, pp. 365–372.
54. *S. Shalunov, B. Teitelbaum* "TCP use and performance on internet2" ACM SIGCOMM Internet Measurement Workshop, San Francisco, USA, 2001.
55. *S. Kent, K. Seo* "Rfc 4301: security architecture for the internet protocol" 2005. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4301.txt
56. *M. Necker, D. Contis, D. Schimmel* "TCP-stream reassembly and state tracking in hardware" Field-Programmable Custom Computing Machines, 2002. Proceedings. 10th Annual IEEE Symposium on, 2002, pp. 286–287.
57. *S. Li, J. Torresen, O. Sorasen* "Exploiting stateful inspection of network security in reconfigurable hardware" Field-Programmable Logic and Applications. Springer Berlin / Heidelberg, 2003, vol. 2778/2003, pp. 1153–1157.
58. *S. Egorov, G. Savchuk* "Snortran: An optimizing compiler for Snort rules" Fidelis Security Systems, Inc., Tech. Rep., 2002.
59. *D.V. Schuehler, J. Lockwood* "TCP-splitter: A TCP/IP flow monitor in reconfigurable hardware" High Performance Interconnects, 2002. Proceedings. 10th Symposium on, 2002, pp. 127–131.

Поступила 18.08.2014р.