UDC 004.72: 004.49

J.N. DAVIES[*], P. COMERFORD[**], M.V. VEROVKO[***], I.S. SKITER[***], I.S. POSADSKA[***]

INTRUSION PREVENTION WITHIN A SDN ENVIRONMENT

[*]Glyndŵr University, Wrexham, UK
[**] University of Derby, Derby, UK
[***]Chernihiv National University of Technology, Chernihiv, Ukraine

Анотація. *У статті звертається особлива увага на комплексність та взаємозв'язки між компонентами інфраструктури Інтернету. Для оптимізації управління даною інфраструктурою були проведені дослідження в області Software Defined мереж (SDN). У даній статті досліджуються значущі розробки для мережевої інфраструктури і способи їх розміщення в середовищі SDN. Зокрема, досліджується розгортання системи запобігання вторгнень, що є властивою більшості комп'ютерних мереж. Також запропоновано вирішення поставленого завдання за допомогою використання конструктивних особливостей апаратних засобів і методів їх інтеграції в SDN.*
Ключові слова: *Software Defined мережі (SDN), системи виявлення вторгнень, системи запобігання вторгнення, IP-маршрутизатор, мультипроцесорні системи, асоціативна пам'ять.*

Аннотация. *В статье обращается особое внимание на комплексность и взаимосвязи между компонентами инфраструктуры Интернета. Для оптимизации управления данной инфраструктурой были проведены исследования в области Software Defined сетей (SDN). В данной статье исследуются значимые разработки для сетевой инфраструктуры и способы их размещения в среде SDN. В частности, исследуется развертывание системы предотвращения вторжений, присущей большинству компьютерных сетей. Также предложено решение поставленной задачи с помощью использования конструктивных особенностей аппаратных средств и методов их интеграции в SDN.*
Ключевые слова: *Software Defined сети (SDN), системы обнаружения вторжений, системы предотвращения вторжений, IP-маршрутизатор, мультипроцессорные системы, ассоциативная память.*

Abstract. *Recent investigations have highlighted the complexity and interrelationship between components of the infrastructure of the internet. In an attempt to simplify the management of the infrastructure a great deal of research has taken place in the area of Software Defined Networks (SDN). This paper investigates the perceived developments in the network infrastructure and how they can be accommodated with a SDN environment. In particular the deployment of Intrusion prevention, a well-known function found in most computer networks, is investigated. A hardware design is offered as a solution and it is shown how this can be integrated into a SDN.*
Keywords: *Software Defined Networks (SDN), Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), IP Router, Multiprocessor Systems, Content Addressable Memory.*

1. Introduction to the topic of study

In 1989, Tim Berners-Lee proposed “a universal linked information system” to help itinerant academics from across the globe run a complicated particle accelerator” [1]. According to the Internet Society who are a global independent organization dedicated to ensuring that the Internet stays open, transparent and defined by everyone. “The genius of the Internet is that its decentralized architecture.” To ensure that individual users’ have the ability to use the hardware, software, and services that best meet their needs, its open and decentralized nature must be preserved. This must be kept in mind when the Internet policy, technology standards, and future development are considered [2].

© Davies J.N., Comerford P., Verovko M.V., Skiter I.S., Posadska I.S., 2017

ISSN 1028-9763. Математичні машини і системи, 2017, № 1

Unfortunately this does not preclude users from attempting to destroy this aim. Significant attempts, which are well documented, have been made to interfere with the smooth running of the services. Due to the complexity of the technology used in the provision of services this is not easy to identify the causes of such problems and prevent them from occurring. Typically the types of attacks are classified into a number of areas to help with the providing solutions.

Identified aims of intrusion include: Denial of Service Attacks including distributed, attacks in which a normal user exploit a vulnerability to gain enhanced level of permissions, users who do not have an account on that machine exploits some vulnerability to gain access, probing attacks.

Intrusion Detection (IDS) and Intrusion Prevention Systems (IPS) have been developed as a tool to address these issues. Ever improving technology in terms of increased computing power, storage and network bandwidth provide a double edge sword. Not only does it enable more protection to be put in place but it also provides attackers development tools to enable more sophisticated attacks to be attempted.

The main network infrastructure design criteria are to enable packets of data to be passed from the source to the destination with as little delay as possible. Since there are many devices involved in the transfer of data then it is necessary for these to cooperate, this involves IP Routers using Routing protocols or manual static routes. Clearly the requirement to move data packets very quickly requires these devices to have independent functions. It is necessary for routers to be able to communicate with other routers in the network domain to identify paths through a network. On receipt of a data packet routers have to make a decision about where to transmit the packet.

Communications between the routers means that they have distributed information available about the network. However this information is very limited. One of the major advances is the ability to utilize multiple processors in these devices. This enables greater functionality to be deployed in routers without interfering with the basic data packet passing.

Major rethinks around how the internet infrastructure should be developed have taken place recently. System engineering theory dictates that it is essential to identify traffic patterns in terms of type and volumes. Historic investigations of networks show that it is impossible to identify these with any certainty due to the range of people, devices and applications using networks. The only certainty is that the network infrastructure has to be agile i.e. must have the ability to change to meet user requirements. Additionally there is always a business need to reduce IT cost

Software defined networking (SDN) is seen as addressing this need for agility. Present networks are built around hardware boxes that have limited configurability. By adopting a philosophy of creating fully programmable networks creates an environment capable of delivering new services. To make this viable then it is necessary to have an underlying hardware structure to support it. A precursor to SDN was the research area of Active networks which made an attempt to improve the intelligence available within the infrastructure.

This paper considers how the network infrastructure is likely to develop by becoming more intelligent and how this would fit into an SDN environment. In particular the area of network security is investigated. The background section describes the principles behind the work while the Related Work covers published work in this area. The findings section contains the results of the investigations and conclusions are drawn from this with a consideration of future work.

2. Background

It is inconceivable to think that the internet will not develop both in terms of service provision and the technology required. The openness of the internet concept and design allows this to flourish. However there is a dichotomy, agility of development versus stability of service.

Work carried out by the Internet Engineering Task Force (IETF) and the standards bodies e.g. IEEE enhance the stability by considering future technologies and providing rules that ensure the health of the internet. But the implementation from a heuristic point of view is more difficult problem since there are many conflicting issues. Some services require real time responses (VoIP, Live streaming, Internet of Things) where as others secured data is of paramount importance (financial transactions, personal data).

This section considers the development of SDN, its ability to cope with the underlying hardware developments. In particular it investigates the very topical issue of security and its identification.

Software defined networks. According to IBM Global Services, Software defined networking (SDN) "creates a centrally managed network that can dynamically sense and respond to changing workload requirements." This is carried out by converting hardware-intensive networks into fully programmable software configurable networks which aids the agility of network service provision [3].

Clearly there are some lines that have to be drawn to enable optimum network services to be provided. The basic fundamentals of computing /electronics dictates that hardware is high speed but not easily adapted and software is very flexible but not as fast as hardware.

RFC7426 has been created by the IETF as an information document on Software Defined Networking (SDN): Layers and Architecture Terminology that addresses these issues. It defines as being a "programmable networks approach that supports the separation of control and forwarding planes via standardized interfaces." See figure 1.

Following the definitions of Network Device, Interface, Application (App), Service, the document breaks SDN into series of planes. Forwarding Plane (FP) – responsible for forwarding traffic, Operational Plane (OP) – management of the overall operation of individual network devices, Control Plane (CP) – to control one or more network devices. Management Plane (MP) – monitors, configuring, and maintaining one or more or parts of network devices. Figure 1 shows the relationship between these planes [4].

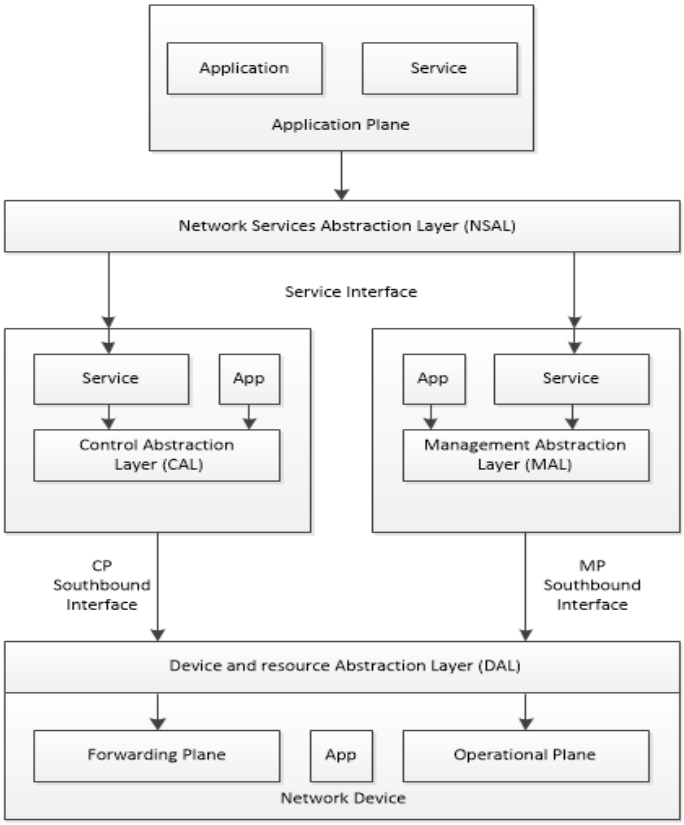

Fig. 1. SDN Layer Architecture [4]

Active Networks. Active Networks was an area of research that was centered around enabling Active Nodes (routers, switches) to perform defined functions on the data flowing through them. Much of this work was carried on around the year 2000 when the need was identified. It was intended to improve the intelligence of the infrastructure and provide the ability to accelerate infrastructure innovation [5].

The idea was to insert code into a programmable active node which then operates on the data packets passing through the network. "User" would send the program to the network node (switch or router), where it would be stored and later executed when the data arrives at the node. Detailed proposals were put forward including an Active Network Encapsulation Protocol (ANEP) along with Active IP an extension to the IP protocol that would retrofit active capabilities to the existing Internet. It was seen that this would build on existing applications that provided Network Monitoring and Measurement [6].

Despite the research work it was felt that this was a rather dangerous venture that could affect the stability of the internet. Additionally the technology was not in an advanced enough state to cope with this functionality. Technology and the understanding of security issues has advanced dramatically since this work was first carried out.

Network Security Problem. An obvious application / service that could be provided as part of the network infrastructure is the ability to deal with network security problems. Most networks contain Intrusion Detection whose aim is to identify intrusion of the type:

DDOS, Unauthorized access Denial of Service Attack (DoS): is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

User to Root Attack (U2R) is an attack in which the attacker starts out with access to a normal user account on the system and is able to exploit some vulnerability to gain higher access permissions to the system.

Remote to Local Attack (R2L): when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.

Probing Attack is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

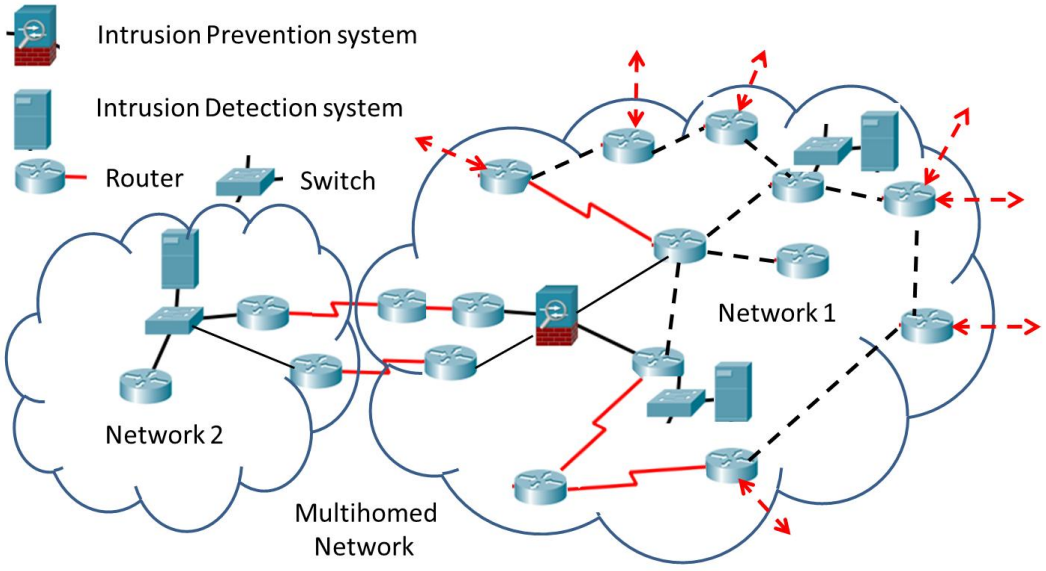

Fig. 2. Intrusion Detection / Intrusion Prevention Placement

Intrusion System (IDS, IPS). An Intrusion Detection System (IDS) is a passive device that analyses stored copies of network traffic. They operate offline and generate alerts when it detects malicious traffic. There are a number of advantages of utilizing IDS: it does not affect network performance, packet capture rate is well defined and so appropriate equipment can be employed, deep packet inspection and more advanced analysis can be carried out. Additionally since they are based on a relatively simple technology for the capture of packets then it is possible to utilize them many times in a network. See figure 2. This is the minimum requirement of most networks and does have a role in highlighting issues and identifying/classifying future possible attacks. Clearly this requires manual intervention.

An Intrusion Prevention Systems (IPS) is an active device. All traffic must pass through the IPS which means that it has access to traffic content all the way through to the application layer. Theoretically it therefore has the ability to stop attacks from reaching the target system.

A great deal of research has taken place based on these types of devices and the analysis techniques used. They provide invaluable information enabling research to be carried out in the development of optimizing of algorithms and early identification of future problems.

Distributed information. Requirements of modern networks dictate that they are designed with a high level of redundancy. Typically sites are multi-homed (see figure 2) which in itself can cause a major problem when attempting to identify network intrusions.

Hardware implications. Multiple processors are widely available in all types of hardware which opens up the possibility of providing much greater intelligence within the network infrastructure (Routers and switches). These devices frequently use special purpose hardware that can dramatically improve the performance of handling packets CAM (content-addressable memory).

3. Related Work

SDN Security Characteristics. Research has shown that protection of networks from abnormal behaviour or malicious applications is a serious security challenge [7]. The nature of software defined networks, using a centralised system of control, is the primary reason for these issues. This approach decouples the data and control planes of the network. An SDN controller is a single point of failure, additionally, its visible nature and limited resources make it especially attractive for an attacker to mount a denial of service attack [7]. The centralised control allows malicious activities and users to be traced through methods such as packet traceback [8]. This mechanism allows an administrator to obtain complete forwarding details for each hop the packet encounters including ingress and egress ports and any packet modifications made along the route. A framework for recording and replaying network events, OFRewind is proposed by Wundsam, et al [9] can also be used to identify the origination of malicious traffic. Despite its advantages, this system lacks a real-time, online replay facility and lacks timing accuracy [9].

Traffic flow analysis. Hu, et al. propose an access control system using an OpenFlow based firewall framework (FLOWGUARD) [10]. The system can track traffic flows and detect firewall policy violations. It is implemented as an application built on top of the SDN controller. Ahmad, et al. [7] highlight that there is a performance penalty compared to the integrated security provided by the controller, therefore a performance-security trade-off must be decided for the network.

An early study of denial of service attacks on software defined networks was performed by Shin et al [11]. The authors exploited the separation of control and data planes to launch an attack. This was facilitated using a scanning tool which uses flow response times to identify an SDN network. Subsequently, if such a network was detected, then flow request packets are sent to the target. This results in the switch sending excessive flow setup requests on the controller causing a denial of service. Similar work by Fonseca, et al [12] shows that using random headers, in continuous stream of IP packets can place an SDN controller in a non-responsive state. The

authors highlight the need for effective detection of such attacks as their work shows multiple controllers do not provide sufficient protection [12].

The analysis of traffic flow behavior and statistics in Openflow switches has been used to mitigate DoS or DDoS attacks. This process does not incur much overhead as the statistics can be easily retrieved from an Openflow controller [7]. Braga et al use an artificial neural network to transform data patterns into a one or 2 dimensional map [13]. Data which exhibits similar statistical features are identified as traffic which could signify a DDoS attack. Statistical anomalies for the traffic flows are collected at regular intervals. This includes average packets per flow and their duration and growth. The neural network is trained from samples collected during normal network operation or during an attack. Subsequently, it is able to identify any anomalous entries collected from the open flow switches.

Shin et al. developed a platform called FRESCO [14] which allows dynamic insertion of security rules by analysing switch statistics and network flows. The framework provides a scripting API which allows this information to be leveraged by intrusion detection systems and firewalls. The solutions are targeted to protect the control plane of the network as this is deemed the most vulnerable part of the architecture.

Flow sampling techniques. There are a number of schemes which utilise flow sampling techniques which can be used to perform statistical analysis. This typically involves samples of packet headers extracted from flows at a rate defined by the controller. For enterprise networks, the flow samples would be forwarded to dedicated security middle boxes for further analysis. In smaller networks, this could be achieved in software on the controller itself [7]. These schemes facilitate proactive security services and are able to operate at line rate due to low overhead of the switch and controllable network [7].

Shirali-Shahreza, et al. propose an implementation of this system, FleXam [15]. The scheme is flexible as it allows packet flows to be selected based on traffic patterns or with a predetermined probability. Dotcenko, et al propose an information security management systems using fuzzy logic techniques [16]. The system uses the techniques to make decisions based on the statistics gathered from the traffic flows. A similar machine learning approach is implemented by Skowyra, et al [17] for securing embedded mobile devices. Similarly, OpenSAFE (Open Security Auditing And Flow Examination) [18] is capable of packet filtering and traffic monitoring at line rate. The framework controls redirection of traffic using a customized language for route management.

Xing, et al. implemented SnortFlow [19], an intrusion detection and prevention system for cloud systems based on the popular open source Snort IDS [20]. The system uses a number of modules which monitor the security status of the network and generate appropriate actions using pattern matching and content analysis algorithms [19]. The system is able to dynamically generate security rules at runtime based on the traffic statistics. Yang et al., propose a traffic monitoring system for Openflow switches [21]. The system is capable of monitoring all traffic and can be used to detect anomalies and flooding attacks in Openflow networks.

Ahmad et al, notes that the view of the network provided by the control plane may be inconsistent with the actual network view [7]. If direct access to network resources and statistics is required, then this creates additional security challenges. The authors suggest that such applications should be categorized into security classes based on their functions and requirements. Security policies would be enforced for each class of application based on their access requirements.

Mechanisms for intrusion detection. Data mining techniques have been used successfully to address deficiencies in existing intrusion detection and prevention systems. These techniques use extremely large datasets to group objects into meaningful subclasses, a process known as clustering [22]. The systems assume that anomalous activity is much less than the normal actions of the network. Furthermore, it is assumed that intrusion activities are sufficiently different to baseline activity. Mingqiang, et al. propose a graph-based intrusion detection algorithm using

outlier detection [22]. The model requires a training dataset on which the algorithm is performed. Experimental results showed satisfactory performance, however the space complexity of the computation increases dramatically with the size of the dataset. Moreover, manual control of the proportion of normal and suspicious records is required which limits its value in practical applications.

A similar approach is provided by Kumar, et. al [23]. A key difference is that this solution uses unsupervised detection techniques. This does not require any previous knowledge or training dataset. Matsubara, et al. propose an algorithm for monitoring data streams that have the characteristics of a given hidden Markov model (HMM) [24]. The authors claim operation in constant O(1) time and high accuracy on a 67 GB dataset.

4. Findings

It has been identified that undertaking aspects of network security within the infrastructure devices would be a definite advantage. However there are a number of deeper investigations that need to be carried out. Firstly it is important to identify which aspects are capable of being carried out by the infrastructure. Having done this it is necessary to investigate the feasibility of this being carried out by highlighting the parameters required e.g. processing power and times involved. It is the important to clarify the hardware requirements for this to take place. Finally it is necessary to determine whether this would fit in with an SDM philosophy.

In this paper the Denial of Service Attack (DOS/DDOS), User to Root Attack (U2R), Remote to Local Attack and Probing Attack have been recognized as the most significant attacks to consider. As far as the infrastructure is concerned then DOS/DDOS and Probing attacks are the only ones that can be addressed.

DDOS Analysis. Due to simplicity these types of network attacks are the largest threat. For this analysis a statistical method based on time series analysis is used. A traffic structure analysis is needed to determine the most important metrics that can be used to identify the attack. These are: ratio of incoming packets to outgoing; number of HTTP flows; difference between quantity of outgoing and incoming ACK packets; UDP in IP-traffic; ratio of SYN to incoming traffic volume; ratio of PSH (push flag) to incoming traffic volume.

Ratio of incoming to outgoing packets in a unit of time is calculated using the formula

$$R_{ip} = \frac{T_i}{T_0},$$

where T_i and T_o are the volumes of incoming and outgoing IP-traffic. If the Incoming traffic speed is increasing without equivalent increase in the outgoing traffic speed this means there is a high possibility of an attack.

Value of threads that are critical for application attacks can be used for identification of application-layer attacks. Because HTTP-flood is an application-layer attack on Web-servers, it is necessary to calculate the difference between incoming and outgoing TCP-packets with set up ACK flag.

$$R_{ack} = N_{acko} - N_{acki},$$

where N_{acko} is number of outgoing ACK-packets, N_{acki} of incoming packets. During such attack the number of ACK packets typically decreases and the value of R_{ack} characteristics falls into the negative region.

The frequency of SYN and PSH flags in incoming packets allows the effectiveness of data transmission to be determined. Packets with SYN are transferred between client and server during TCP connection maintenance. In this case the number of SYN flags received on a TCP connection is equal to the number of requests for connection, and frequency of SYN identifies the level

of these packets. During the SYN-flood attack the goal is not to transfer data but to overflow the receiver queue. The frequency of SYN is measured by:

$$R_{syn} = \frac{N_{syn}}{N},$$

where R_{syn} is frequency of SYN, N_{syn} is the number of SYN in incoming packages, N is the total number of packets. Set up PSH flag signifies that data in packets must be transferred to program of application layer. The frequency of the PSH flag is a useful indicator of the channel load.

$$R_{psh} = \frac{N_{psh}}{N},$$

where R_{psh} is the frequency of PSH, N_{psh} – number of PSH in incoming packages, N – total number of packets.

UDP traffic impact coefficient is defined by ratio:

$$R_{udp} = \frac{T_{udp}}{T_{tcp}},$$

where T_{udp} is the volume of incoming UDP traffic, T_{tcp} is the volume of incoming TCP traffic. This coefficient characterizes the presence of a UDP-flood class attack. Usually there is only a small amount of UDP packets so excessive UDP traffic compared to TCP traffic allows for the detection.

SYN-flood attacks need to be the main component of all metrics since trends indicate that carries more than 90% of attacks. This component shows jump-like changes at the beginning of DDoS-attack.

The conclusion made that appearance of DDoS attack of any kind is reflected in dynamics of these components of time. The detection of attacks can be based on an analysis of the given metrics and expansion of the dynamic series on the main components. Components required for this analysis in a time sequence are all available in the network infrastructure.

Future Network Hardware trends. One of the main reasons for using a SDN is to enable the flexibility of software to be applied to network infrastructure devices. However for this to be achieved it is necessary to have an appropriate adaptable hardware configuration.

Utilizing a multiprocessor configuration enables the SDN to be supported and enhances the possibility to provide specialized functionality. An example of this is a CAM that is used in communication applications for a number of applications requiring comparisons to take place. It is a high-speed memory that searches its entire contents in a 1 or 2 clock cycle. Instead of the address matching used in standard memory a CAM performs content matching. This improves the speed of searches and is much faster than sequentially checking each address location in a standard memory for a particular value. The higher speed searches are achieved by using content values as an index into a database of address values.

Figure 3 shows a hardware block diagram of a typical Router hardware configuration and an enhanced interface card suitable for use with SDN. Routers would be made up of a motherboard containing a general processor and memory used as the management of a standard router. An Interface Card contain the components (in dotted box) i.e. independent processor and associated memory. Each Interface Card contains an Interface Processor and associated memory to handle the Interface Hardware. This is to allow the card to operate at the line speed. Specialized hardware is installed e.g. CAM or TCAM (ternary content-addressable memory). Additionally an

Intrusion Processor is added to capture packets and make decisions associated with the validity of the received packets.

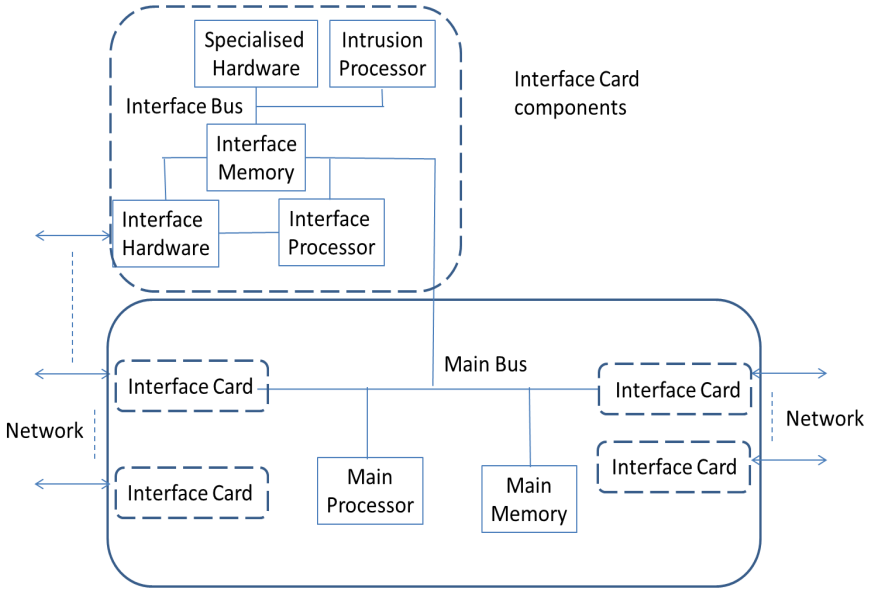

Fig. 3. Network Hardware Devices (Router) and Interface Cards

SDN in Intelligent Network. There will always be parts of the network infrastructure that need to be implemented in hardware due to timing constraints. However the trend in the design of hardware devices is to utilize general multiprocessors and specialized hardware to improve performance. Clearly there is a possibility of using SDN to reprogram these devices as network protocols change, new services become available etc. However this needs to be carried out in a very controlled manner. If this is not done then mistakes at the device level can be catastrophic.

5. Conclusions

There are many advantages in adopting a SDN since network services, functionality, protocols and devices are continually changing. It is clear to see the advantages and ways in which this can be implemented in a secure way. However for the device layer, as defined in SDNs, it is much more difficult to envisage how this will be handled.

Research in the area of Active Networks provided a structure and protocol to implement applications within a network device (router). This opens up the way of providing services within the network infrastructure. A candidate for this is network security and the use of multiprocessor and specialized hardware has opened up the possibility of implementing this. SDN has an important role in this since it has the ability to manage changes.

6. Future Work

Analysis of the timings associated with typical DDOS attacks need to be carried out to confirm requirements of hardware.

Simulation of the hardware environment and protocols associated with the loading and control of multiprocessors within routers.

REFERENCES

1. Shankland S. Tim Berners-Lee: 25 years on, the Web still needs work (Q&A) [Електронний ресурс] / S. Shankland. – Режим доступу: http://www.cnet.com/uk/news/tim-berners-lee-on-its-25th-anniversary-the-web-still-needs-work-q-a/.

2. Internet Society [Електронний ресурс]. – Режим доступу: http://www.internetsociety.org/who-we-are/mission/ values-and-principles.
3. White paper ICW03011-USEN-00 / IBM Global Services. – 2015. – July. – P. 1 – 12.
4. Haleplidis E. RFC 7426 SDN: Layers and Architecture Terminology / E. Haleplidis. – Режим доступу: https://tools.ietf.org/html/rfc7426.
5. Yasuda H. Lecture Notes in Computer Science / H. Yasuda // Active Networks: Second International Working Conference Proceedings, IWAN. – Tokyo, 2000. – P. 17 – 29.
6. Tennenhouse D.L. A Survey of Active Network Research / D.L. Tennenhouse // IEEE Communications Magazine. – January, 1997. – Vol. 35, N 1. – P. 80 – 86.
7. Security in Software Defined Networks: A Survey / I. Ahmad, S. Namal, M. Ylianttila [et al.] // IEEE Communications Surveys & Tutorials, Fourthquarter. – 2015. – Vol. 17, N 4. – P. 2317 – 2346.
8. Where is the debugger for my software-defined network / N. Handigol, B. Heller, V. Jeyakumar [et al.] // Proc. 1st Workshop Hot Topics Softw. Defined Netw. – 2012. – P. 55 – 60.
9. OFRewind: Enabling record and replay troubleshooting for networks / A. Wundsam, D. Levin, S. Seetharaman [et al.] // Proc. USENIX Annu. Tech. Conf. – 2011. – P. 29.
10. FLOWGUARD: building robust firewalls for software-defined networks / H. Hu, W. Han, G.-J. Ahn [et al.] // Proc. 3rd Workshop Topics Softw. Defined Netw. – 2014. – P. 97 – 102.
11. Shin S. Attacking software-defined networks: a first feasibility study / S. Shin, G. Gu // Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. – 2013. – P. 165 – 166.
12. A replication component for resilient OpenFlow-based networking / P. Fonseca, R. Bennesby, E. Mota [et al.] // Proc. IEEE NOMS. – 2012. – P. 933 – 939.
13. Braga R. Lightweight DDoS flooding attack detection using NOX/OpenFlow / R. Braga, E. Mota, A. Passito // Proc. IEEE 35th Conf. LCN. – 2010. – P. 408 – 415.
14. Shin S. FRESCO: Modular composable security services for software-defined networks / S. Shin // Proc. Netw. Distrib. Security Symp. – 2013. – P. 1 – 16.
15. Shirali-Shahreza S. Efficient implementation of security applications in openflow controller with flex-am / S. Shirali-Shahreza, Y. Ganjali // Proc. IEEE 21st Annu. Symp. HOTI. – 2013. – P. 49 – 54.
16. Dotcenko S. A fuzzy logic-based information security management for software-defined networks / S. Dotcenko, A. Vladyko, I. Letenko // Proc. IEEE ICACT. – 2014. – P. 167 – 171.
17. Skowyra R. Software-Defined IDS for securing embedded mobile devices / R. Skowyra, S. Bahargam, A. Bestavros // Proc. IEEE HPEC. – 2013. – P. 1 – 7.
18. Ballard J. R. Extensible and scalable network monitoring using OpenSafe / J.R. Ballard, I. Rae, A. Akella // Proc. INM/WREN. – 2010. – 8 p.
19. SnortFlow: A openflow-based intrusion prevention system in cloud environment / T. Xing, D. Huang, L. Xu [et al.] // Proc. 2nd GREE. – 2013. – P. 89 – 92.
20. Roesch M. Snort: Lightweight intrusion detection for networks / M. Roesch // Proc. LISA. – 1999. – Vol. 99. – P. 229 – 238.
21. Yang C.-T. Implementation of a virtual switch monitor system using openflow on cloud / C.-T. Yang // Proc. Int. Conf. IMIS Ubiquitous Comput. – 2013. – P. 283 – 290.
22. Mingqiang Z. A graph-based clustering algorithm for anomaly intrusion detection / Z. Mingqiang, H. Hui, W. Qian // Computer Science & Education (ICCSE), 7th International Conference, VIC. – Melbourne, 2012. – P. 1311 – 1314.
23. Kumar M. Unsupervised outlier detection technique for intrusion detection in cloud computing / M. Kumar, R. Mathur // Convergence of Technology (I2CT), International Conference. – Pune, 2014. – P. 1 – 4.
24. Fast and Exact Monitoring of Co-Evolving Data Streams / Y. Matsubara, Y. Sakurai, N. Ueda [et al.] // Data Mining (ICDM), 2014 IEEE International Conference. – Shenzhen, 2014. – P. 390 – 399.

Стаття надійшла до редакції 07.11.2016