ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20201208122622
Preliminary communication

Security Features in a Hybrid Software-Defined Network

Igor FOSIĆ*, Drago ŽAGAR

Abstract: The paper presents a novel paradigm of software-defined network that is significantly different from previous traditional networks and enables new opportunities in the architecture and implementation of security solutions. The analysis of network environments will compare traditional networks and software-defined networks and emphasize significant differences. A survey of the existing research includes vector attacks and troubleshooting using the capabilities of SDN with an emphasis on access control, detection, and prevention of attacks. This paper uses previous research and results to obtain information that will be used in improving critical system network protection and compares it with the existing conventional approach as well as implements it through a hybrid software-defined network.

Keywords: Hybrid SDN; IDS; IPS; OpenFlow; Traditional network

1 INTRODUCTION

In traditional networks, the control and infrastructure layer of network devices are tightly integrated into physical devices. Security mechanisms, forwarding rules on routers, and transmission on switches are also tightly integrated into the physical network infrastructure, which makes implementing changes across a large number of devices difficult and complicated. Recent research suggests SDN-based mechanisms (software-defined network) enable greater flexibility, dynamic programmed performance, and reduction in operating costs.

SDN as a new paradigm of network architecture enables dynamic adaptation of the network environment to the current requirements or needs of users and applications, simplifies management to a great extent and increases network scalability. An additional advantage of SDN is the ability to use network components from different vendors that support SDN protocols, without knowing the devices themselves, because the entire network environment is managed via an SDN controller. The basic components of the SDN network architecture are the SDN controller, OpenFlow network devices and OpenFlow protocol of the communication channel that connects the components.

Traditional network security is often based on firewalls, intrusion detection systems (IDS), intrusion prevention systems (IPS), access control, audits, and ruleset management. Traditional firewall functions are constructed on a static set of rules without fine precision, traffic filtering is done based on the source or destination IP address and TCP/UDP port. Therefore, existing static safety mechanisms lack flexibility and scalability. Another aspect of security is the management of access to network resources used to set appropriate permissions for users or devices accessing the network [1-3].

On many devices, it is very difficult to achieve frequent and rapid changes of network settings and network device configurations due to security or user requirements. When specifics are added to the mix, the traditional network troubleshooting becomes complicated. Once bypassed, the security mechanism leaves a network inadequately protected from the inside. Allowed activities within the network are not always properly authorized, and new solutions that will enable monitoring and filtering of unwanted and unauthorized network traffic are required. This paper presents an overview of security solutions with an emphasis on access control, detection, and intrusion prevention in the SDN network environment. Software-defined network (SDN) is a new network architecture that allows greater flexibility in achieving network security. The fact that in traditional networks the identity, authentication, and authorization of users takes place when entering the network justifies the use of SDN network elements to reduce and address this shortcoming. By adding SDN elements, all the advantages of traditional networks are retained and, simultaneously, new possibilities are added via SDN. Such networks are called hybrid networks. Improving the current security mechanisms of a traditional VPN network is yet another motif for research, which aims to implement a hybrid SDN system in an organic environment that will prevent external and unauthorized VPN users from accessing critical network resources. The main goal of this paper is to research the existing hybrid SDN implementation and how to achieve improved security of the traditional network. A model with characteristics of a hybrid SDN will be able to block the unwanted traffic with the help of several SDN devices as one of the factors that can achieve a higher level of security. One of the issues of the existing security troubleshooting is the inability to monitor and verify the identity of users after entering the protected part of the network. This particularly refers to insufficient control of legitimate and authorized users and their potentially unauthorized activities in systems with critical data. Detecting and blocking unwanted activities is extremely important in enterprise systems and it is necessary to provide critical systems and infrastructure as well as implement new solution models in the existing environment. In fact, the implementation of the hybrid SDN can improve the existing IDS and IPS systems.

The rest of the paper is organized in sections, where Section 2 presents an overview of network environments. A brief description of security SDN solutions and attack vectors is described in Section 3. Previous research on these topics is presented in Section 4, while the conclusion and guidelines for future research are presented in Section 5.

2 NETWORK ENVIRONMENTS

The new technologies and devices are causing the change in modern networking concept. Traditional

network paradigms are seen as overly static and require greater efforts to physically be changed and reconfigured. The SDN paradigm has emerged as a new approach to better cope with the exponential growth of data traffic, network virtualization, and user mobility. SDN allows network administrators/operators program control of the network by helping them add new features without compromising performance, reliability, or user experience [4].

One of the goals of SDN is to enable the design of dynamic and programmable security controls that can fully operate with little or no human interaction in real time and provide access to legitimate users, protect systems from attacks and mitigate damage in the event of an attack. New controls and possibilities lead to new types of attacks, which did not exist in traditional networks [5].

2.1 Traditional Networks

In traditional networks, the characteristics of the device depend exclusively on the vendors. The control and infrastructure layer are connected, thus making it difficult to develop and implement new network features.

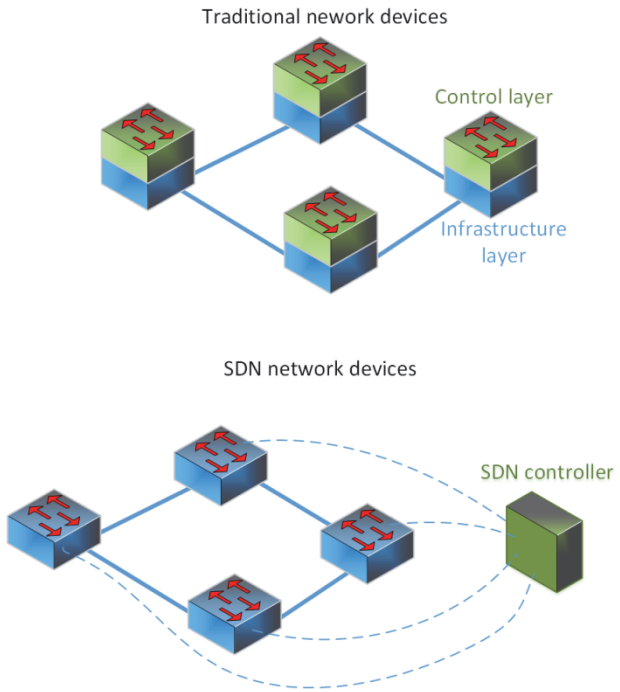

Figure 1 The comparison of traditional and SDN devices

Therefore, it is not easy to include new features because the process includes numerous different protocols built into different hardware such as firewalls, routers, switches, etc. The change in the network status needs to be communicated to all other devices so they could update the status of neighboring devices. The complexity of today's networks makes it difficult to apply a consistent approach, security, QoS and other features [6-7].

2.2 Software-defined Network

Virtual networks are not a novelty, they have existed in various forms over the years, such as MPLS, VPN, ATM, Frame Relay and VLAN. SDN has emerged as a new networking paradigm that separates the network control layer from the infrastructure layer. The goal is to separate the layers from specified hardware technologies and provide control, which significantly increases network agility. The layer separation allows the use of OpenFlow and other open protocols to access network switches and routers. Software networks are designed to automate and radically simplify the management of computer networks with a significant reduction in errors, unlike manual operation.

It allows networks to connect directly to applications through application programming interfaces (APIs), improving performance and creating a flexible and dynamic network architecture that can be modified when necessary. Controllers can dynamically reconfigure the network to avoid congestion, implement new services, add virtual infrastructure, etc. [8-9]. Some of the main features of the SDN architecture include [10]:

a. Programmable management - control and configuration of the network is directly programmed because the forwarding function is decoupled from the control function and allows very fast configuration, management, provision, and optimization of network resources using automated programs.

b. Agility - abstracting forwarding control allows dynamic adjustment of traffic flow across the entire network.

c. Central management - network intelligence is centralized in SDN controllers that maintain a global view of the network.

d. Open standards - SDN is open standards-based, which simplifies network design and operation because SDN controllers use universal protocols instead of vendor-specific ones.

Model SDN architecture presented in Fig. 2 is comprised of three layers connected via API:

a. The application layer consists of end-user applications that use SDN communication services.

b. The control layer provides consolidated management performance that monitors forwarding packets.

c. The infrastructure layer consists of network elements and devices that allow packet forwarding.

In a software-defined network architecture, layers are connected through API as shown in Fig. 2. "Southbound" interface allows a particular component of a network to communicate with a lower-level component, while "northbound" denotes communication with a higher-level component.

The application layer consists of end users and applications that use SDN network services. The SDN application may request certain changes to the controller in the configuration and operation of the network. Network infrastructure management requests to take place via the "northbound" API interface to the controller and use these interfaces to provide an abstract view of the network. One of the most used APIs is the REST API.

The SDN controller in the control layer is mainly responsible for two tasks. One is to translate the application layer requirements into the infrastructure layer, and the other is to give an abstract physical network model to the application layer. The control layer is often referred to as a network operating system because it supports network management logic and provides the application layer with an abstract view of the global network. In an SDN

environment, the controller uses APIs to communicate with the application layer, infrastructure layer, and other controllers. In distributed controller architecture, they communicate with each other using the so-called "eastbound"/"westbound" APIs, which are not used as often as the "northbound" and "southbound" APIs.

The infrastructure layer consists of packet forwarding devices, which is its main function-providing efficient forwarding mechanisms. Communication between the control and infrastructure layer takes place via the so-called "southbound" API interface, such as OpenFlow.

APIs are the key components of SDN. They make it a powerful tool for network management and working with features such as programmability, protocol independence, the ability to change network parameters as needed, elasticity. The control layer uses APIs to monitor, manage, and facilitate the communication of all other SDN layers. One of the advantages of SDN is the fact that the API is used in an open, neutral, and interoperable way [11-15].

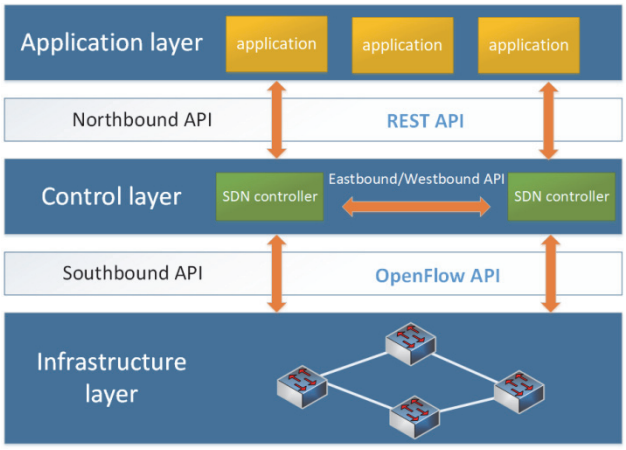

Figure 2 Software-defined network architecture

2.3 Hybrid Software-defined Network

A network that is a combination of SDN and traditional network devices is commonly referred to as a hybrid SDN network. It offers numerous advantages and represents a transitional step towards the full adoption of SDN. A hybrid SDN network combines traditional networking and SDN protocols that operate in the same environment and allows the introduction of new SDN technologies such as OpenFlow protocols into traditional environments without a complete reconfiguration of the network architecture. A complete change of network to SDN without any testing poses high risk in terms of performance and security. In addition, large financial resources are required for SDN network components, and upgrading relatively new traditional network devices is considered unprofitable [16]. Hybrid SDN networks offer a number of advantages [17-20]:

a. They reduce financial cost because the implementation of a complete SDN network is very expensive and additional investment is needed in education, design, configuration, and work on the SDN network.

b. They can be used to take advantage of some of the SDN paradigms without implementing a full SDN network. The access network can use the legacy, traditional devices while the distribution network uses SDN devices. Therefore, a hybrid SDN network can be used to process most forwarding packets in the access network via legacy devices while SDN devices are used in the distribution network to take advantage of SDN. To ensure a smooth and controlled transition, it is recommended to initially implement SDN for only a small portion of non-critical traffic.

c. SDN allows fine granulation of data flow control. If such control is required for only a small part of the network, a hybrid SDN network can be implemented, while the rest of the network uses traditional networking.

d. Traditional routing protocols are very effective for some tasks, such as connecting SDN controllers to control different parts of the network. Thus, a hybrid SDN network can be applied to void the SDN controller of tasks that can be efficiently performed by traditional routing protocols.

e. SDN devices are not as mature as traditional network devices. A hybrid SDN network facilitates the transition from legacy to SDN network devices. With the help of a hybrid SDN network, it is possible to gradually deploy more and more SDN devices and evaluate SDN performance.

f. A hybrid network solves the connection of two separate SDN networks via traditional network devices.

Fig. 3 depicts SDN network devices connected with traditional devices and functionally belonging to both the control and infrastructure layer.

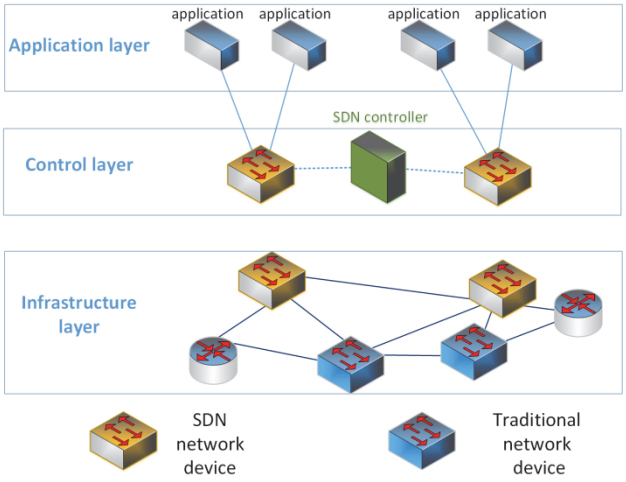

Figure 3 The layers of hybrid SDN network

There are several possible hybrid SDN models described in [17, 21]:

a. topology-based model where the network is divided into zones so that each node belongs to only one zone, traditional or SDN.

b. a service-based model where services are divided into traditional and SDN part of the network. In order to implement some services, such as network-wide forwarding, certain nodes may belong to both paradigms.

c. a model based on the classification and division of traffic into traditional and SDN controlled traffic.

d. an integrated model where SDN is responsible for all network services and uses traditional protocols as the packet forwarding interface.

Table 1 The comparison of traditional, hybrid and SDN network [12]

Characteristic	Traditional network	Hybrid network	SDN network
Protocols	OSPF, IGP, BGP, etc.	OpenFlow, OSPF, IGP, BGP etc.	OpenFlow
Stability	high	high	high
Scalability	low	depends on the implementation	high
Robustness	low	high	high
Programmability	low	depends on the implementation	high
Technical complexity	low	high	high
Forwarding function	local	depends on the implementation	Program-controlled
Cost-effectiveness	low	moderate	high

Hybrid SDN network is a possible way of migrating the traditional network to complete SDN network architecture. Tab. 1 shows the main differences among the three types of networks observed in the previous parts of this paper.

3 SECURITY FEATURES AND ATTACK VECTORS IN SOFTWARE DEFINED NETWORK

Numerous security issues related to traditional network architecture are also found in SDN architecture which is exposed to various security risks from a network architecture design perspective as it includes application, control, and infrastructure layer. One of the most significant security risks is the attack on the SDN controller in the control layer, which is very sensitive to attacks of denial of service, the so-called DoS attack. Compromised SDN switches can cause a large number of queries to SDN controller and can potentially cause delays or non-execution of queries. Unprotected applications in the application layer are in high risk of manipulation and reprogramming network traffic flow. Communication between the control and infrastructure layer is susceptible to the so-called "man in the middle" attacks, where it is possible to modify the rules sent from the SDN controller to switches to take control of the packet forwarding function [22]. For effective security, overall visibility needs to be catered for, which includes:

a. Information on each system user.
b. An overview of each digital conversation.
c. Knowing which condition is normal.
d. Information on each change in the system.
e. Quick response to security threats.

A hybrid SDN is a transitional type from a traditional to a fully software-defined network that allows for numerous benefits to the existing traditional network mentioned in Chapter 2.3.

The controller is a particularly attractive target for security attacks, as it is an indispensable part of the SDN architecture. Unauthorized access and exploitation of network resources in the absence of a robust, secure controller platform allow an attacker to take control of the controller and carry out malicious activities. In the past, such attacks targeted DNS servers, but the attack on the SDN controller could cause much more damage. In [23], the authors demonstrated the feasibility of attacking controllers from a data layer, by implementing and testing the "fingerprint" technique of the SDN controller, with the primary goal of emphasizing the need for high controller security. By introducing open source SDN interfaces and known protocols to simplify programming, the network functionality allows attacker location detection presented in [24], where in a hybrid SDN the attacker is detected by analyzing the ARP request from the source. A graph-based switching mechanism is also used to detect the location of the attacker by checking legitimate users. The good side of the SDN architecture in terms of security is that it supports and enables a very fast system of reaction, monitoring, analysis, and response to a security attack. From a security perspective, SDN enables:

a. Network forensics: facilitates fast and (pre-set) adaptive identification and management of security threats through a cycle of gathering information from the network, analysis and security policy updates, after which it is easy to reprogram and optimize network functionality.
b. Security policy change: allows security policy to be defined and implemented on all elements of the network infrastructure, reducing the frequency of misconfigurations and conflicting policies throughout the infrastructure.
c. New security services implementation: facilitates the implementation of security services where applications such as firewalls and intrusion detection systems (IDS) can be applied to specific network traffic according to the organization's rules.

Bearing all of the above in mind, SDN security will be as good as a well-defined security policy. The implementation of existing authentication and authorization verification mechanisms may address some aspects of the security challenge, but new threat detection and protection techniques need to be further developed [25].

4 SURVEY OF THE PREVIOUS RESEARCH

This section is a survey of the research conducted so far on software-defined networks, implementation, troubleshooting, and applied solutions. This paper analyzes various aspects of security attack mitigation and compares some of the approaches proposed to increase the security of SDN architecture in previous research. When the advantages of SDN and layered architecture are considered, the main strength of SDN architecture, i.e., programmability, is simultaneously the main vulnerable aspect exploited for security attacks. In addition, this basic feature of SDN cannot be completely removed as it can undo the fundamental function of SDN. This paper also analyzes and compares the approaches proposed in previous research to increase security and address specific security issues using SDN architecture. Access control, intrusion detection and prevention systems for network elements and network-connected systems are some of the observed attack vectors.

4.1 Access Control

The business environment requires traffic management established on the role of the user, such as limited access to some resources for users with limited

privileges. Traditional policy management requires constant maintenance of the configuration of many network nodes. This calls for a solution that will simplify the configuration based on the abstract characteristics of the network architecture with the help of SDN. User identities should not only be considered at the application level, but also at the network level. Martinez-Julia et al. described the identity issue in their research on SDN [26-27], which describes an identity-based network architecture that sets digital identity in the middle of communication. This architecture adds new features to the network, such as user identification, management and authentication, and encryption. It is implemented in a higher layer of the network that allows entities to connect without the need for IP addresses. Alsmadi et al. [28] suggest a global central access control system that uses SDN that can provide all legitimate users with the exact levels of access they should have but will also prevent an illegitimate user or request to access internal resources. Their proposal reduces inconsistencies in decision-making between different decision points of the access control. Paladi et al. [29] suggest an SDN infrastructure that allows applications to execute a range of resource access requests. Jager [30] proposes an access control system and limits applications and the SDN controller to access only a reduced set of critical operations, so that the security of end-user SDN traffic can be significantly improved. The most common standard for access control of users and authorization on interfaces in access networks is applicable in SDN with 802.1X framework transformation on traditional switches. The FlowIdentity protocol presented by Yakasai et al. [31] and AuthFlow [32] by Mattos et al. use 802.1X framework in the SDN architecture. FlowIdentity is a network access control solution that uses 802.1X framework in the SDN architecture combined with a novel authorization method through a stateful role-based firewall on OpenFlow switches implemented by the separation of the authentication. The interface entity is transferred and centralized on the SDN controller, while interface controls (and logical interfaces) are maintained on the switches. The main concept of AuthFlow is authentication using infrastructure layer protocols and pairing user identities with data streams they created in the network. Therefore, the proposed mechanism applies the IEEE 802.1X standard and EAP (Extensible Authentication Protocol). Nayak et al. [33] suggested a model for determining dynamic rules for network control Resonance. Their research indicated that managing dynamic access controls in SDN is easier than in traditional networks. Access control management is implemented and based on real-time data flow information and alerts. Monitoring subsystems are integrated with the SDN controller for easier access control. Allouzi et al. [34] proposed a SafeFlow protocol designed to support authentication between the SDN switch and the controller each time the switch requests access to a classified resource. Casado et al. [35] introduced a new network architecture called Ethane that manages the network without allowing any communication between end devices without explicit permission.

4.2 Intrusion Detection Systems

The survey of the existing research offers new proposals for the use of SDN and new possibilities for security intrusion detection mechanisms. The detection of intrusion and attacks on controllers and the control layer of the SDN is specific to the SDN environment. Despite these novel security vulnerabilities, SDN creates new opportunities to implement more effective intrusion detection methods. Due to the openness of platforms that support SDN technologies, it is possible to use existing data collection mechanisms and protocols as a data source for SDN intrusion detection algorithms.

Jankowski et al. [36] describe an intrusion detection method integrated with an SDN controller where unauthorized activities performed in an SDN environment are classified. Some IDSs are designed as a service that seeks to detect and prevent the breach of malicious traffic and keep it away from gateways and compromising network elements. Such systems are designed based on centralized functions to increase the ease of control [37] [38]. Ajaeiya et al. [39] developed a method to detect different attacks using the advantages of SDN to measure traffic flow statistics that periodically collects flow statistics from OpenFlow switches and analyzes the obtained data. The proposed IDS for SDN was able to detect malicious traffic with high accuracy. Latah et al. [40] achieved a higher accuracy and precision of intrusion detection by combining IDS approach based on flow and data packets. Some research aims to increase the security of the SDN environment by building IDS using machine learning principles as proposed by Vetreslevi et al. [41]. Real-time traffic is monitored for intrusion detection, and IDS is divided into two phases, the first for attack detection and the second for categorization. This approach reduces the dependence and workload of the controller, as well as the high rate of attack detection. Honeypots are a type of active defensive security technology and they are expected to be attacked. Some SDN approaches with honeypots were presented in studies by Wang et al. 42] and Fan et al. [43]. These systems can simulate a large and realistic network to attract attackers and redirect intrusions to honeypots and provide further analysis. The SDN approach improves the shortcomings of the existing honeypot technologies whose mechanisms are noticeable and can be easily detected by attackers. The SDN controller allows users to configure their own network data management rules, which will forward or redirect traffic to appropriate honeypots depending on the type of alert.

4.3 Intrusion Prevention Systems

Similar to IDS, IPS monitors networks and systems for malicious activity or security policy violations and takes certain steps to mitigate such activities. IPS represents reliability and security in the network system and is considered one of the most popular security devices. In traditional network architecture, IPSs must be deployed at the input and output of each branch of the internal network, even at the input and output of each subnet of the internal network to protect data and devices in the internal network. The high cost of such an implementation and the low usability of an individual IPS are the reasons for discarding

this approach. SDN-based IPS implementation can reduce the cost of implementing IPS, improve usability, and provide a higher level of security. In order to reduce implementation and maintenance costs compared to the traditional design and number of IPS systems, Zhang et al. [44] suggest the SDN/OpenFlow architecture-based IPS implementation. The implementation in the SDN environment leads to the improvement in the response (ping) of network devices as well as the usability of individual IPSs. SDN network with the so-called adaptive IPS [45] has the ability to detect attacks and can block advanced persistent threats based on the frequency and type of attack using fuzzy logic. Ammar et al. suggest an IPS system [46] that integrates with host security software and can use any security device that supports a remote system log. The control unit of the proposed framework consists of an agent and a log server integrated with the SDN controller. The security solution is independent of the SDN controller and allows greater scalability. The proposed framework demonstrates the ability to detect security threats and block an attacker at the network edge. One of the most common network attacks are denial-of-service (DoS) attacks and distributed DoS (DDoS) attacks. To prevent these attacks, the authors in [47] suggest a distributed firewall and IPS modules on SDN switches that inspect packets entering the framework. In [48] authors use the existing Snort IDS for attacker information, and show that denial-of-service attacks can be monitored and mitigated by combining SDN security mechanisms, while the authors in [49] use a hybrid model of two types of machine learning (SVM and SOM) to improve the accuracy of DDoS attack detection relative to a separate machine learning approach. Neu et al. [50] present an SDN solution to prevent port scan attacks. They used the statistics collected on SDN networks and updated the OpenFlow routing rules when a port scan was observed. The results of the experimental evaluation indicated the method is effective in detecting malicious data flows with the help of statistics, which resulted in a decrease in false positive results. Some improvements to the existing IDS/IPS system were presented in a study by Xing et al. [51]. A complete SDN-based attack detection and prevention solution was called SDNIPS. It utilizes Snort and its detection capabilities and flexible SDN network configuration. The evaluation indicated better performance and efficiency compared to traditional approaches. In their research, Nam et al. [52] propose a structure to improve the security function of the SDN that performs intrusion detection and the automatic blocking function by monitoring the intrusion detection results of existing open source IDS/IPS software. When an attack is detected, the controller sends OpenFlow commands to the network device, the firewall function is activated, and the intrusion is automatically blocked. Birkinshaw et al. [53] showed in their study that it is possible to instantly reject a packet when an attack is detected with the help of SDN. The system was designed, implemented, and tested, and based on traffic anomalies. Two types of algorithms were used: a random pass threshold (CB-TRW) and a rate limit (RL). They introduced the Port Bingo (PB) port scan detection technique. The results of the experiment showed that port scans and DoS attacks can be detected and prevented in real time. The rate of false-positive results can be kept low enough by adjusting the threshold parameters of the attack detection algorithms.

5 CONCLUSION

The SDN paradigm introduces some advantages and improves network security through dynamic and centralized data flow control, broad network view, network programmability, data layer simplification, etc. Software-defined network is a next-generation network technology with innovations that open extensive research topics on network security. Nowadays networks presuppose a traditional topology with a logical network boundary and a single exit/entry point. In this approach, various security devices such as firewall, IDS, IPS, SIEM systems are usually implemented immediately at the entrance to the network, but network security also depends on the access control mechanisms and user authentication. Software-defined networks (SDNs) are broadly accepted in enterprise networks, so the gradual placement of several SDN devices among devices on a traditional network creates a hybrid SDN network and adds new features supported by SDN devices. A survey of the available literature focuses on the set of security aspects applied to SDN.

Access control, firewalls connected to the intrusion detection and prevention systems in the SDN environment can improve overall security. In addition, the security mechanisms of SDN have demonstrated the ability of SDN as a technology that can successfully overcome the existing flaws in a traditional network such as programmability, conditioned real-time response, central monitoring, and network view, etc. Even though firewall and IPS still play an important role in protecting the network and network systems, novel threats require a solution that can protect the network in as many layers as possible. Inadequate hardware and scalability issues with traditional approaches can be overcome by a hybrid SDN architecture. Each new major security implementation will probably include SDN performances. Based on existing research and analysis, it can be concluded that network security can be improved with appropriate SDN mechanisms. From the control point, the entire network can be monitored, as well as applications, data, user and device identities, and overall network behavior. SDN analysis tools can use information coming from all devices on the network, not just security devices, to find security threats and respond better to them.

The novelty of this paper is the conformation that the efficient use of SDN devices improves network security. The programmability as one of the main advantages of this network architecture allows changes in real time according to user requirements or by changing the environment. Device manufacturer independence, freedom in programming mode, standardized API queries, and centralized control are also the disadvantages and benefits of a hybrid software-defined network. This does not change the function of the network but introduces a great novelty at the device's programmability level because, as the name suggests, the network is defined by software from a central location and with an overall view of the network. The current way of configuring network devices is highly error-prone and can be fully automated and minimized.

Based on the surveyed research and results, future research will be based on troubleshooting of monitoring user actions in the network, primarily network scanning and flexible real-time firewall rules to make access control more effective. Port scanning (TCP, UDP) is commonly used as a preparation for security attacks by identifying available and potentially vulnerable devices in the network. The port scanning process itself is not designed to cause damage but as a preparation for a security attack that will allow more damage to occur. The IPS/IDS model with SDN performance will try to overcome the shortcomings of current solutions in detecting anomalies of common network traffic such as port scanning and improve existing algorithms whose performance and comparison will be based on publicly available test data. Each algorithm for detecting traffic anomalies will be reviewed to find which of them give the best result in the hybrid infrastructure of SDN. The best ones will be implemented in controlled conditions of the network infrastructure.

6 REFERENCES

[1] Dixit, V. H., Kyung, S., Zhao, Z., Doupé, A., Shoshitaishvili, Y., & Ahn, G. J. (2018). Challenges and preparedness of SDN-based firewalls. *SDN-NFVSec 2018 - Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks and Network Function Virtualization, Co-Located with CODASPY 2018*.

[2] Lorenz, C., Hock, D., Scherer, J., Durner, R., Kellerer, W., Gebert, S., & Tran-Gia, P. (2017). An SDN/NFV-Enabled Enterprise Network Architecture Offering Fine-Grained Security Policy Enforcement. *IEEE Communications Magazine*, *55*(3), 217-223. https://doi.org/10.1109/MCOM.2017.1600414CM

[3] Cox, J. H., Chung, J., Donovan, S., Ivey, J., Clark, R. J., Riley, G., & Owen, H. L. (2017). Advancing software-defined networks: A survey. *IEEE Access*, *5*, 25487-25526. https://doi.org/10.1109/ACCESS.2017.2762291

[4] Humayun, K. (2013). Software Defined Networking (SDN): A Revolution in Computer Network. *IOSR Journal of Computer Engineering*, *15*(5), 103-106. https://doi.org/10.9790/0661-155103106

[5] Alsmadi, I. (2016). The integration of access control levels based on SDN. *International Journal of High Performance Computing and Networking*, *9*(4), 281-290. https://doi.org/10.1504/IJHPCN.2016.077820

[6] Ghosh, U., Chatterjee, P., Shetty, S. S., Kamhoua, C., & Njilla, L. (2019). Towards Secure Software-Defined Networking Integrated Cyber-Physical Systems: Attacks and Countermeasures. *Cybersecurity and Privacy in Cyber-Physical Systems*, 103-132. https://doi.org/10.1201/9780429263897-6

[7] Khan, F. N., Saleem, Y., & Bashir, M. K. (2019). Migration of Multiplatform Legacy Network to Single Software-Defined-Network (SDN). *Proceedings - 2018 International Conference on Computing, Electronics and Communications Engineering, ICCECE 2018*, 333-338. https://doi.org/10.1109/iCCECOME.2018.8658652

[8] Amin, R., Reisslein, M., & Shah, N. (2018). Hybrid SDN networks: A survey of existing approaches. *IEEE Communications Surveys and Tutorials*, *20*, 3259-3306. https://doi.org/10.1109/COMST.2018.2837161

[9] Canini, M., Feldmann, A., Levin, D., Schaffert, F., & Schmid, S. (2014). Software-defined networks: Incremental deployment with panopticon. *Computer*, *47*(11), 56-60. https://doi.org/10.1109/MC.2014.330

[10] Software-Defined Networking (SDN) Definition. (n.d.). Retrieved from https://www.opennetworking.org/sdn-definition/

[11] Solution, O. N. F., September, B., & ONF. (2013). SDN in the Campus Environment. *ONF Workshpop*.

[12] Huang, X., Cheng, S., Cao, K., Cong, P., Wei, T., & Hu, S. (2018). A Survey of Deployment Solutions and Optimization Strategies for Hybrid SDN Networks. *IEEE Communications Surveys & Tutorials*, *21*(2), 1-25.

[13] Nkosi, M., Lysko, A., Ravhuanzwo, L., Nandeni, T., & Engelberencht, A. (2017). Classification of SDN distributed controller approaches: A brief overview. *Proceedings - 2016 3rd International Conference on Advances in Computing, Communication and Engineering, ICACCE 2016*, 342-344. https://doi.org/10.1109/ICACCE.2016.8073772

[14] Vijay Tijare, P. & Vasudevan, D. (2019). The Northbound Apis Of Software Defined Networks. *International Journal of Engineering Sciences & Research Technology*, *501*. https://doi.org/10.5281/zenodo.160891

[15] Bailey, S., Bansal, D., Dunbar, L., Hood, D., Kis, Z. L., Mack- Crane, B., & Varma, E. (2013). SDN Architecture Overview. *EPC and 4G Packet Networks*, 17-64. https://doi.org/10.1016/b978-0-12-394595-2.00002-5

[16] Binlun, J. N., Chin, T. S., Kwang, L. C., Yusoff, Z., & Kaspin, R. (2018). Challenges and Direction of Hybrid SDN Migration in ISP networks. *2018 IEEE International Conference on Electronics and Communication Engineering, ICECE*, 60-64. https://doi.org/10.1109/ICECOME.2018.8644812

[17] Vissicchio, S., Vanbever, L., & Bonaventure, O. (2014). Opportunities and research challenges of hybrid software defined networks. *Computer Communication Review*, *44*(2), 70-75. https://doi.org/10.1145/2602204.2602216

[18] Galán-Jiménez, J. (2018). Exploiting the control power of SDN during the transition from IP to SDN networks. *International Journal of Communication Systems*, *31*(5), 1-19. https://doi.org/10.1002/dac.3504

[19] Vissicchio, S., Vanbever, L., Cittadini, L., Xie, G. G., & Bonaventure, O. (2017). Safe update of hybrid SDN networks. *IEEE/ACM Transactions on Networking*, *25*(3), 1649-1662. https://doi.org/10.1109/TNET.2016.2642586

[20] Wang, W., He, W., & Su, J. (2017). Boosting the Benefits of Hybrid SDN. *Proceedings - International Conference on Distributed Computing Systems*, 2165-2170. https://doi.org/10.1109/ICDCS.2017.302

[21] Sandhya, Sinha, Y., & Haribabu, K. (2017). A survey: Hybrid SDN. *Journal of Network and Computer Applications*, *100*, 35-55. https://doi.org/10.1016/j.jnca.2017.10.003

[22] Wang, T. (2016). Benefits and the Security Risk of Software-defined Networking. *ISACA Journal*, *4*, 59.

[23] Azzouni, A., Braham, O., Nguyen, T. M. T., Pujolle, G. & Boutaba, R. (2016). Fingerprinting OpenFlow Controllers: The First Step to Attack an SDN Control Plane. *IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA.* https://doi.org/10.1109/GLOCOM.2016.7841843

[24] Ubaid, F., Rashid, A., Faisal, B. U. & Iqbal, M.M. (2017). Mitigating Address Spoofing Attacks in Hybrid SDN. *International Journal of Advanced Computer Science and Applications.* https://doi.org/10.14569/IJACSA.2017.080474

[25] Sezer, S., Scott-Hayward, S.,Kaur Chouhan, P., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M. & Navneet R. (2013). Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Magazine, 51*(7), 36-43. https://doi.org/10.1109/MCOM.2013.6553676.

[26] Martinez-Julia, P. & Gomez-Skarmeta, A. F. (2012). A novel identity-based network architecture for next generation internet. *Journal of Universal Computer Science*, *18*(12), 1643-1661. https://doi.org/10.3217/jucs-018-12-1643

[27] Martinez-Julia, P. & Skarmeta, A. F. (2015). Using an Identity Plane for Adapting Network Behavior to User and Service Requirements. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, *158*, 253-265. https://doi.org/10.1007/978-3-319-26925-2

[28] Alsmadi, I. & Xu, D. (2015). Security of Software Defined Networks: A survey. *Computers and Security*, *53*, 79-108. https://doi.org/10.1016/j.cose.2015.05.006

[29] Paladi, N. & Gehrmann, C. (2019). SDN Access Control for the Masses. *Computers and Security*, *80*, 155-172. https://doi.org/10.1016/j.cose.2018.10.003

[30] Buchegger, S. & Dam, M. (2015). Secure IT Systems: 20th Nordic Conference, NordSec 2015 Stockholm, Sweden, October 19-21, 2015 Proceedings. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9417*, 197-204. https://doi.org/10.1007/978-3-319-26502-5

[31] Yakasai, S. T. & Guy, C. G. (2016). FlowIdentity: Software-defined network access control. *2015 IEEE Conference on Network Function Virtualization and Software Defined Network, NFV-SDN 2015*, *5*, 115-120. https://doi.org/10.1109/NFV-SDN.2015.7387415

[32] Ferrazani Mattos, D. M. & Duarte, O. C. M. B. (2016). AuthFlow: authentication and access control mechanism for software defined networking. *Annales Des Telecommunications/Annals of Telecommunications*, *71*(11-12), 607-615. https://doi.org/10.1007/s12243-016-0505-z

[33] Nayak, A., Reimers, A., Feamster, N., & Clark, R. (2009). Resonance: Dynamic access control for enterprise networks. *Computer Communication Review*, 11-18. https://doi.org/10.1145/1592681.1592684

[34] Allouzi, M. & Khan, J. (2018). SafeFlow: Authentication Protocol for Software Defined Networks. *Proceedings - 12th IEEE International Conference on Semantic Computing, ICSC 2018*, 374-376. https://doi.org/10.1109/ICSC.2018.00076

[35] Casado, M., Freedman, M. J., Pettit, J., Luo, J., Gude, N., McKeown, N., & Shenker, S. (2009). Rethinking enterprise network control. *IEEE/ACM Transactions on Networking*, *17*(4), 1270-1283. https://doi.org/10.1109/TNET.2009.2026415

[36] Jankowski, D. & Amanowicz, M. (2015). Intrusion detection in software defined networks with self-organized maps. *Journal of Telecommunications and Information Technology*, *2015*(4), 3-9.

[37] Chukwu, J., Osamudiamen, O., & Matrawy, A. (2017). IDSaaS in SDN: Intrusion Detection System as a service in software defined networks. *2016 IEEE Conference on Communications and Network Security, CNS 2016*, 356-357. https://doi.org/10.1109/CNS.2016.7860509

[38] Monshizadeh, M., Khatri, V., & Kantola, R. (2017). Detection as a service: An SDN application. *International Conference on Advanced Communication Technology, ICACT*, 285-290. https://doi.org/10.23919/ICACT.2017.7890099

[39] Ajaeiya, G. A., Adalian, N., Elhajj, I. H., Kayssi, A., & Chehab, A. (2017). Flow-based Intrusion Detection System for SDN. *Proceedings - IEEE Symposium on Computers and Communications*, 787-793. https://doi.org/10.1109/ISCC.2017.8024623

[40] Latah, M. & Toker, L. (2019). Minimizing false positive rate for DoS attack detection: A hybrid SDN-based approach. *ICT Express*, 10-12. https://doi.org/10.1016/j.icte.2019.11.002

[41] Harish, M., Karthick, R., Rajan, R. M., & Vetriselvi, V. (2019). Two-Level Intrusion Detection System in SDN Using Machine Learning. In *Proceedings of the International Conference on Communications and Cyber Physical Engineering*, *500*. https://doi.org/10.1007/978-981-13-0212-1

[42] Wang, H. & Wu, B. (2019). SDN-based hybrid honeypot for attack capture. *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019*, 1602-1606. https://doi.org/10.1109/ITNEC.2019.8729425

[43] Fan, W. & Fernandez, D. (2017). A novel SDN based stealthy TCP connection handover mechanism for hybrid honeypot systems. *2017 IEEE Conference on Network Softwarization: Softwarization Sustaining a Hyper-Connected World: En Route to 5G, NetSoft 2017*. https://doi.org/10.1109/NETSOFT.2017.8004194

[44] Zhang, L., Shou, G., Hu, Y., & Guo, Z. (2013). Deployment of Intrusion Prevention System based on Software Defined Networking. *International Conference on Communication Technology Proceedings, ICCT*, 26-31. https://doi.org/10.1109/ICCT.2013.6820345

[45] Pratama, R. F., Suwastika, N. A., & Nugroho, M. A. (2018). Design and implementation adaptive Intrusion Prevention System (IPS) for attack prevention in software-defined network (SDN) architecture. *2018 6th International Conference on Information and Communication Technology, ICoICT 2018*, *0*(c), 299-304. https://doi.org/10.1109/ICoICT.2018.8528735

[46] Ammar, M., Rizk, M., Abdel-Hamid, A., & Aboul-Seoud, A. K. (2016). A framework for security enhancement in SDN-based datacenters. *2016 8th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2016*, 3-6. https://doi.org/10.1109/NTMS.2016.7792427

[47] Rengaraju, P., Ramanan, V. R., & Lung, C. H. (2017). Detection and prevention of DoS attacks in Software-Defined Cloud networks. *2017 IEEE Conference on Dependable and Secure Computing*, 217-223. https://doi.org/10.1109/DESEC.2017.8073810

[48] Fares, A. A. Y. R., De Caldas Filho, F. L., Giozza, W. F., Canedo, E. D., De Mendonca, F. L. L., & Nze, G. D. A. (2019). DoS attack prevention on IPS SDN networks. *WCNPS 2019 - Workshop on Communication Networks and Power Systems*. https://doi.org/10.1109/WCNPS.2019.8896233

[49] Deepa, V., Muthamil Sudar, K., & Deepalakshmi, P. (2018). Detection of DDoS attack on SDN control plane using hybrid machine learning techniques. *Proceedings of the International Conference on Smart Systems and Inventive Technology, ICSSIT 2018*, 299-303. https://doi.org/10.1109/ICSSIT.2018.8748836

[50] Neu, C. V., Tatsch, C. G., Lunardi, R. C., Michelin, R. A., Orozco, A. M. S., & Zorzo, A. F. (2018). Lightweight IPS for port scan in OpenFlow SDN networks. *IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018*, 1-6. https://doi.org/10.1109/NOMS.2018.8406313

[51] Xing, T., Xiong, Z., Huang, D., & Medhi, D. (2014). SDNIPS: Enabling Software-Defined Networking based intrusion prevention system in clouds. *Proceedings of the 10th International Conference on Network and Service Management, CNSM 2014*, 308-311. https://doi.org/10.1109/CNSM.2014.7014181

[52] Nam, K. & Kim, K. (2018). A Study on SDN security enhancement using open source IDS/IPS Suricata. *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, 1124-1126. https://doi.org/10.1109/ICTC.2018.8539455

[53] Birkinshaw, C., Rouka, E., & Vassilakis, V. G. (2019). Implementing an intrusion detection and prevention system using software-defined networking: Defending against port-scanning and denial-of-service attacks. *Journal of Network and Computer Applications*, *136*, 71-85. https://doi.org/10.1016/j.jnca.2019.03.005

Contact information:

Igor FOSIĆ, PhD student
(Corresponding author)
HEP-Telekomunikacije d.o.o.,
M. Divalta 199, HR-31000 Osijek
E-mail: igor.fosic@hep.hr

Drago ŽAGAR, PhD, Full Professor
Josip Juraj Strossmayer University of Osijek,
Faculty of Electrical Engineering, Computer Science and Information Technology Osijek,
Kneza Trpimira 2B, HR-31000 Osijek
E-mail: drago.zagar@ferit.hr