Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 3309 – 3325

doi:10.3906/elk-1812-86

SoftSwitch: a centralized honeypot-based security approach using software-defined switching for secure management of VLAN networks

Muhammet BAYKARA, Resul DAŞ*
Department of Software Engineering, Faculty of Technology, Fırat University, Elazığ, Turkey

Received: 13.12.2018 • **Accepted/Published Online:** 08.07.2019 • **Final Version:** 18.09.2019

Abstract: Honeypot systems are traps for intruders which simulate real systems such as web, application, and database servers used in information systems. Using these systems, unauthorized and malicious access can be efficiently detected. Honeypot is an entity which acts as a source of valued information and its behavior can be monitored. The inability or difficulty of intrusion detection is a serious security problem in networks including virtual local area network (VLAN). According to the literature, the use of honeypots for intrusion detection and prevention in networks including VLAN is strongly recommended.

In this paper, in order to provide security and to detect unauthorized and malicious access to the VLAN, a centralized honeypot-based approach with a software-defined switching is proposed. With the developed and proposed honeypot-based intrusion detection and prevention approach, reduction in false alarm, network traffic, and cybersecurity cost, as well as centralized control, was provided. The proposed system has been run in GNS3 simulation software and successful results have been obtained by reducing false alarm level, network traffic, and cybersecurity cost. The numerical results of the attacks that were detected based on the port and protocol using SoftSwitch are detailed in the performance evaluation subsection.

Key words: Intrusion detection and prevention systems, honeypots, network security, system security, VLAN security

1. Introduction

Information security has been one of the most important areas of research in the world of information in recent years. Many tools or software are used in computer network systems to provide corporate or personal information security. Among the tools used for this purpose are systems such as intrusion detection and prevention systems, firewalls, antivirus programs, and vulnerability scanners. However, in computer network system, these tools are insufficient in most cases when used alone. For this reason, the security scenarios where these tools are used together are recommended [1, 2].

Honeypot systems are security tools that have recently been used in vulnerable information systems [3]. They gain their security advantage as they are being attackable [4]. These honeypot systems simulate real systems such as web, application, and database servers used in information systems, and they are trapping systems for attackers. With the use of these systems, unauthorized or malicious access can be detected. A honeypot is an isolated resource which looks like a target with valuable information to the attackers; however, the traffic movement to this target is particularly monitored. These systems attract the attackers to them allowing the investigators to analyze any pattern of aggressive or assault behavior [5].

*Correspondence: rdas@firat.edu.tr

When the studies in the literature are examined, three basic structures related to the positioning of honeypot systems can be found. Specifically, it is possible to position honeypot systems on local area network (LAN), Internet, and demilitarized zone (DMZ). Each of these positioning scenarios has advantages and disadvantages. Honeypots are structured in ways and shapes that do not risk network security, depending on where they are located [6]. The following design criteria are crucial for successful honeypot traps:

- The content of honeypots information to be presented to the attackers must be well-defined in terms of value and level.
- If the honeypots are accessed by an attacker, precautions must be taken to avoid the probable damage due to reaching out classified information.
- The location that serves on the network of honeypots should be correctly selected.
- The appropriate infrastructure must be effectively set up so that honeypot-trapped attackers can be monitored, tracked, and intervened if necessary.

Honeypots can be evaluated in three groups according to their interaction levels: low, medium, and high interaction [7].

Low- and medium-interaction honeypots emulate services that specifically harbor security holes to attract attackers. However, they do not have any real or important information on them. The risk of compromising the honeypot is very low, as the attacker cannot directly access the real system in such honeypot systems. Nevertheless, they collect less information about attacks because attackers cannot have one-to-one communication with the real system. It is important to imitate the services in low- and medium-interaction honeypots. Any unintentional mistakes made during this process can cause the attacker to detect the honeypot, which is undesirable. A detected honeypot will not be able to achieve its goal since it loses its attractiveness to attackers [7–9]. In high-interaction honeypot systems, it is desired to attract the attacker by running real services [9]. In addition, external software is used to monitor the activities of the attacker [7]. Unlike low- and medium-interaction honeypots, high-interaction honeypots are more susceptible to access because the attackers have a direct interaction with the real system; some measures can be taken in the firewalls to prevent the malicious event. As long as the attacker is in communication with a real system, more detailed information about the attack can be obtained. It is more difficult for an attacker to detect a honeypot because real services are running. High-interaction honeypots are more costly and require more attention. Furthermore, they can also cause new security weaknesses. The networks where high-interaction honeypots are completely used must be isolated and all safety precautions must be taken. Otherwise, the attacker is in contact with the real system which may allow the infiltration of the honeypot, and thus compromising the system and create new security threats.

2. Literature review

Honeypots are not used for solving any network security problem on their own; they are rather used as a part of the system security, which means that honeypots are designed and positioned according to the problems they are to solve. An extensive literature review reveals that there are many security applications where intrusion detection and prevention systems (IDPS) are used in conjunction with honeypot systems [10–17].

Malanik et al. demonstrated the possible applications of honeypot systems in LANs in their work. In their study, it was reported that honeypot systems can be used to detect zero-day malignancies. In this study, it is stated that in case of network-defined VLANs, honeypot systems must be configured in each VLAN [6]. In

the work of Li Li et al., honeypot applications in a LAN was developed where virtual and physical honeypots were placed at a specific location. The focus of the work has been described as the provision of a combination of defense systems such as firewalls, intrusion detection systems (IDSs), and honeypot systems to increase safety in LANs [10]. Song Li et al. sought to establish a complex interaction honeypot-based IDS mechanism in their work. The aim of the system they built is to increase the durability and security of the network. In their work, they performed several studies to increase the trapping capacity of the honeypot system and to improve the network security [11]. Chawda and Patel proposed a distributed honeypot system for studying new weaknesses and vulnerabilities. They used low-interaction honeypot systems as the initial stage (front-end) content filter to further expose vulnerabilities in the system they have developed. This study is one of the studies that focus on the idea that honeypot systems can be used especially for detecting unknown attacks [12].

Xiangfeng et al. discussed how honeypot technologies could be implemented within IDS. Their study suggests that honeypot systems can be used to solve problems that IDSs have [13]. Paul and Mishra have developed a honeypot-based signature generator to provide computer network security. The developed system is mainly used for protection against polymorphic worm attacks. The developed system has the ability to isolate suspicious traffic and collect much useful information about malicious traffic and worm attacks. In case signature-based systems are unable to detect new attacks, the proposed system has the ability to generate signatures for unknown worm attacks [14]. Beham et al. have benefited from the advantages of virtualization technologies to develop their own work. In their study, they explored the use of intrusion detection and honeypot systems on nested virtualization environments. Here, existing nested virtualization technologies and virtual machine introspection-based intrusion detection and honeypot systems have been studied comparatively [15]. Liu et al. developed a honeypot technology-based IDS that uses IP traceback technology in their work. An intrusion detection scheme combined with honeypot systems has been proposed by setting the limits of conventional IDS [16]. In the work of Pomsathit, IDSs were used with honeypot systems on distributed networks. In this study, the efficiency of these systems was measured by analyzing a system first with IDS, then reanalyzing the same system with a honeypot IDS [17]. In their work, Jiang and Liu emphasized the importance of implementing honeypot systems in systems that enterprise business networks. By examining existing honeypot systems, they have combined the methods used in IDS structures with a new honeypot system [18]. Akiyama et al. designed a highly scalable client honeypot system that is highly interactive and efficient. In this regard, it has been stated that it is aimed for in-depth analysis and efficient attack detection [19]. Buvaneswari and Subha proposed an approach named IHoneycol to prevent distributed service blocking attacks in their work. They showed that distributed denial of service (DDoS) attacks can be prevented by honeypot and IDS-based implementations [20]. Vishal et al. focused on a honeypot system based on open-source software such as SNORT [21], OSSEC [22], and Honeyd [8], and exploiting machine learning algorithms to predict attacks [23]. As it is evident from the literature, IDS, IPS, firewalls, or honeypots alone are not sufficient in providing information systems and network security. Studies have shown that hybrid systems are used where these security systems complement each other [24, 25]. Thanks to the software switch application developed in the present study, honeypot systems and IDS/IPS can be used as complementary to each other without any bottleneck in network communication. The developed software provides a centralized control, and reduces installation and maintenance costs. Controls and security of VLAN-containing networks have been achieved. In addition, the use of honeypot systems and IDS together to detect unknown new attacks and to reduce the false alarm level is provided.

3. The proposed approach for intrusion detection systems

Controlling VLANs installed in a network system is very difficult and costly. IDS and honeypots are not capable of monitoring VLANs. Thanks to the approach proposed in this study, a solution to this problem has been provided. With this solution, a reduced false alarm level in IDPS, lower cost, and a central point network traffic control are achieved. The overall map of the proposed honeypot-based approach is presented in detail in Figure 1.

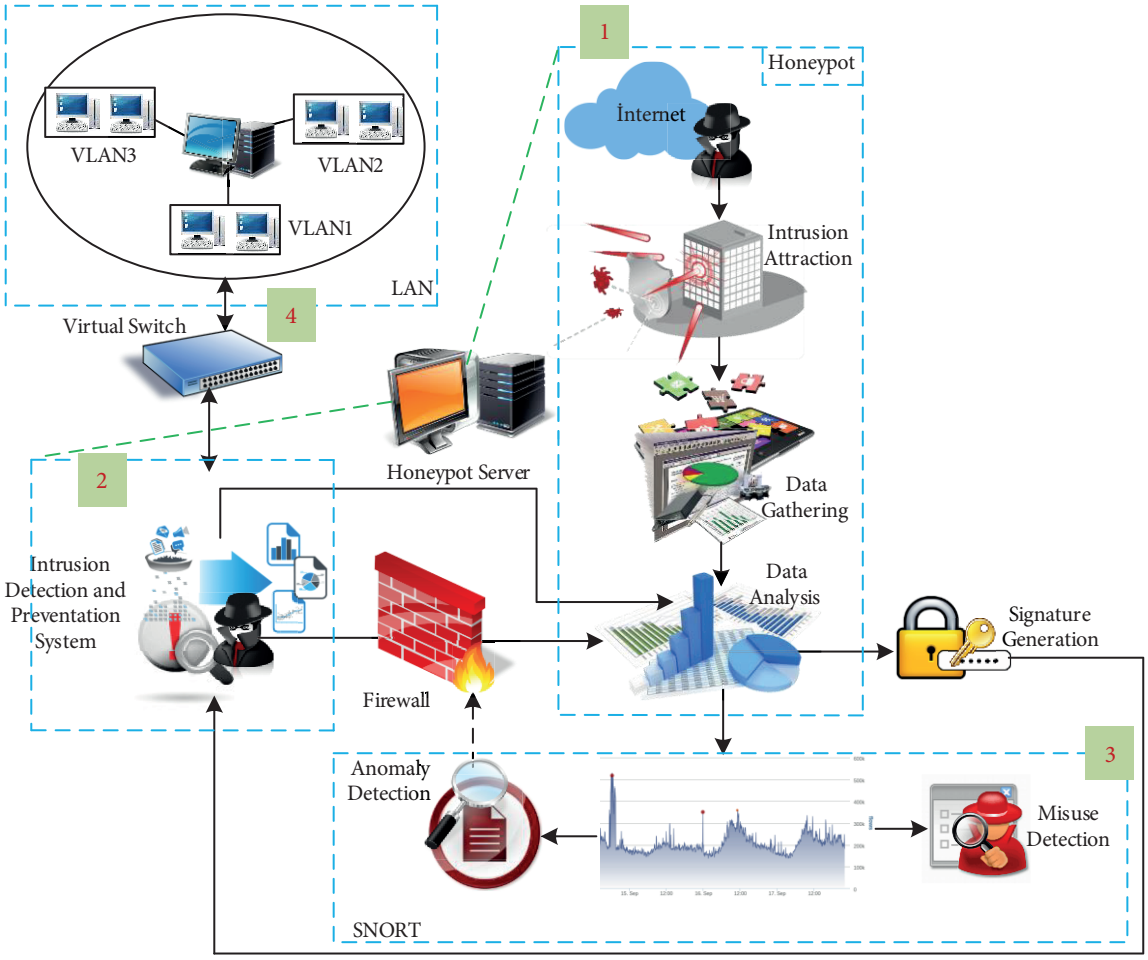

Figure 1. Architecture of the proposed honeypot-based system.

The architecture of the developed system consists of four main modules as shown in Figure 1; these include: honeypot system module (1), intrusion detection and prevention module (2), anomaly and misuse detection module (3), and software switch (4).

3.1. Honeypot system module

Honeypot systems consist of components of attack, data collection, and data analysis as shown in Figure 1 [4, 25, 26]. The honeypot interaction level determines the type of attack component. Honeypot systems can operate in all areas of the network when they are located in the LAN area on the network. The seizure of the honeypot located in the LAN area is causing great security risks. For this reason, if the honeypot is designed to

have a low interaction with the attack component, this problem can be avoided. Honeypot data collection and data analysis components are provided with external software, such as in highly interactive systems, providing a more flexible monitoring and detailed analysis.

3.2. Intrusion detection and prevention module

This module is located between the switch and the router to physically divide the network. It provides the control function between the honeypot module and VLAN communication.

3.3. Anomaly detection and misuse detection module

This module is a software that provides an interface service to integrate IDS structures that can operate according to abuse detection and anomaly detection methods into the system. Here, signature database and anomaly detection structure of IDS structure called SNORT has been used in particular.

3.4. Software switch module

This module is a software that can operate like a hardware switch that runs between the trunk port of the physical switch and the router. This software ensures that all VLANs connected to the network hardware switch are monitored. Thus, it is possible to monitor each VLAN from a central point.

In this study, a novel method is developed to reduce the installation and maintenance cost when honeypots are deployed in large corporate networks, including VLAN networks. In this method, a software switch is designed to enable honeypots to be located on a single server in systems with VLANs. Since the VLAN configuration can be done on switches that can operate in the 2nd and 3rd OSI layers, the developed software switch has been examined in different scenarios in terms of being applicable to both layers. With the implemented design, on all subnets of VLANs in layer 2 and layer 3 switches, honeypot systems can be represented by a single honeypot server, providing centralized control, thus increasing the level of security. Therefore, the advantages of the honeypot systems have been combined with the capabilities of the IDSs, and an environment has been created to capture the zero-day attacks. In this section, the proposed and implemented software switch approach will be introduced in three parts. Since the VLAN configuration can be implemented in OSI layers 2 and 3, Subsections 3.1 and 3.2 explain which scenarios are valid for both the 2nd layer and 3rd layer switch devices. Under these different scenarios, it is explained how the software switch application performs on a centrally located IPS application that performs network communication.

3.5. Localization of honeypots on layer 2 switch

Second layer switches direct the packets arriving at their ports to other ports looking at the table, where the previously created media access control (MAC) address information is located. In some cases, it may be desirable to create a physically disjoint network by isolating a group of ports from other ports in the network, VLAN technology is used for such situations. A switch with a VLAN configuration will not send an incoming packet to its destination port even though this switch knows that the destination MAC address is in a different port belonging to another VLAN. In this way, a complete isolation of the network between VLANs will be ensured. However, it is not possible for VLANs created in this way to communicate with each other. If VLANs also need to communicate, this will require routers running on layer 3. Routers are network devices that connect two networks which are physically separated. The switches can also combine isolated networks with the VLAN configuration configured on them. For this, the subinterfaces of the routers can be used. The port to be

connected to the router on the switch where the VLAN configuration is made is set as trunk. In addition, packets arriving at the switch are encapsulated with the "Dot1Q protocol", containing VLAN information, to be sent to the routers. The router pulls out this capsule and then it encapsulates the packet with the VLAN information to be routed and sends it to the switch. The switch reads the VLAN information of the Dot1Q header and it extracts the Dot1Q packet and sends it to the port where the destination MAC address is located. In this way, the communication between VLANs occurs. In order for honeypots to operate on all VLANs, each VLAN must have a server to represent the honeypot. It is possible for a honeypot server running in a VLAN to operate by representing the IP addresses within the subnet of its VLAN network. In addition, users in other VLANs can access to this honeypot. However, it is not possible to represent an IP address outside the VLAN's subnetwork. Hence, a honeypot server is required to be configured on each VLAN and its subnet. In this study, a software switch and a honeypot running between the switch's trunk-configured port and the router are included in all VLANs. Figure 2 shows a scenario created and tested with the GNS3 network emulator application. In Figure 2, the IPS computer has two physical network interfaces. The interface connected to the

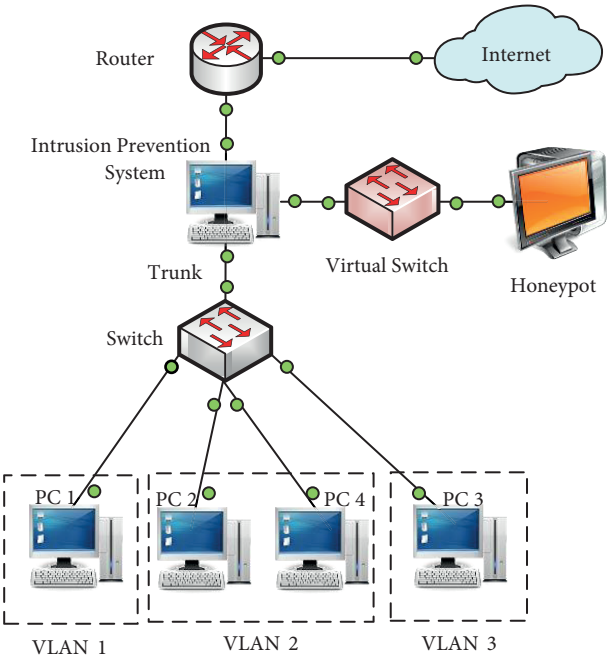

Figure 2. Localization of IPS on layer-2 switch.

virtual switch is a virtual interface created on the IPS and honeypot uses this interface to communicate with the virtual server. IPS is located between the switch and the router to physically divide the network. It uses the 2nd layer and 3rd layer protocols to guide the virtual honeypot server connected to the virtual network interface. In this way, IP addresses of subnets of all VLANs on the switch are represented by a central honeypot server. In Figure 3, the network flow diagram of the scenario in Figure 2 is given. In Figure 3, it is shown how the software switch operating on the IPS system proposed on the 2nd layer switches provides the communication. Accordingly, the software switch application needs to perform the routing for three different interfaces. First, it checks whether the packets coming from the switch match the IP addresses represented in the honeypot server. If the incoming packet belongs to the honeypot server, the packet is sent to the virtual switch that is bound to the honeypot. In this way, the 3rd layer will be redirected. In addition, incoming packets are encapsulated with

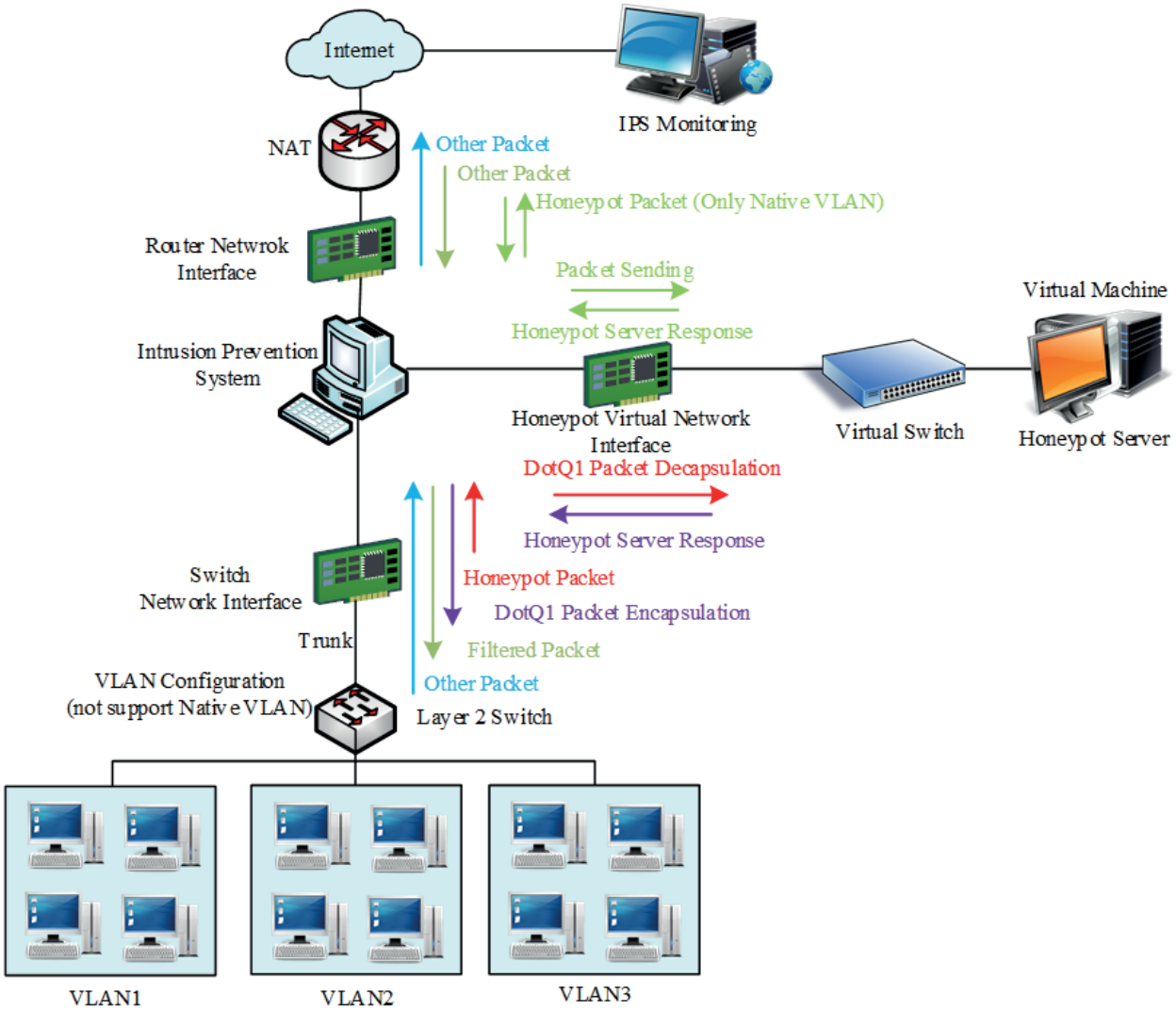

Figure 3. Localization of IPS on layer 2 switch.

the Dot1Q protocol by the switch. The source MAC address and VLAN information of the packet from this capsule are being saved in a table, and then the packet is terminated. Packets that come to the honeypot server are packets with the Dot1Q protocol removed. Consequently, packets from the native VLAN feature coming from the outside will also be sent to the honeypot server to represent the external IP zones. However, in order to use this feature, the switch must not have a VLAN set as native. External network monitoring can be provided by including network administrators in the native VLAN. If the IP addresses of the packets coming from the switch do not belong to the honeypot server, the VLAN information in the Dot1Q header and the source IP address are added to the table, and they are forwarded to the interface of the connected router. The packets from the router are first checked to see if they match the IP addresses belonging to the honeypot server. If the incoming packet belongs to the honeypot server, the software switch sends the packet to the virtual switch that it is bound to the honeypot. In order for external IP addresses to be represented by honeypots, ports must be routed over the routers to the honeypots, and the IP address that the honeypot will use for forwarding should be selected from the native VLAN subnet. In this way, the incoming packets will not be encapsulated with the

Dot1Q protocol and will be sent directly to the honeypot. If the packet does not belong to the honeypot, it will be filtered by the intrusion detection and prevention system, and then sent to the switch. If the destination MAC address of packets coming from the honeypot belongs to the router, the packet will be directly forwarded to the router. In this way, routing will be done in the 2nd layer. If the destination MAC address does not belong to the router, the VLAN address to which the packet should go is found by looking at the table in which the previously created MAC address and VLAN information are stored. Then, the source MAC address will be changed to the router's MAC address and sent to the switch. This way, the switch will route the packet as if it were requesting, assuming that the router sent packets to the appropriate VLAN. Apart from these, exceptions are also made for cases where packets are sent to broadcast addresses using layer 2 protocols. For these cases, new scenarios were created and added to the software switch design. The control of these exceptions is examined in the flow diagram of the software switch application.

3.6. Localization of honeypots on layer 3 switches

Layer 3 switches, unlike layer 2 switches, can also do layer 3 routing among the ports. There is no need to use an external router to communicate VLANs created on this vault. Another purpose of configuring the switches' ports as trunks is to connect the VLANs on different switches together. If a switch cannot find a record of the destination MAC address in its VLANs it can forward, it encapsulates the packet with Dot1Q and sends it to the other switches from the trunk-configured interface. In this way, VLANs on multiple switches are connected together.

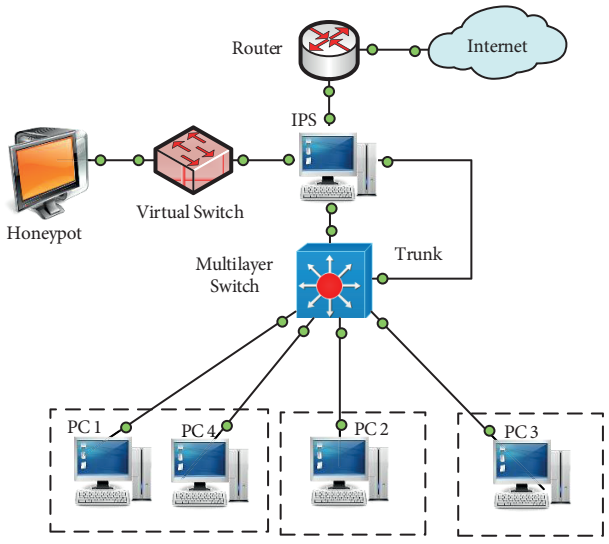

Figure 4. Localization of IPS on layer 2 switch.

The application of honeypots on layer 3 switches is also based on the ability to communicate with the software switch that the switch thinks is a different switch. In Figure 4, a scenario created and tested with the GNS3 network emulator application is depicted. The IPS computer has three physical network interfaces as shown in Figure 4. The interface that connects the virtual switch is a virtual interface created on the IPS and HoneyPot uses this interface to communicate with the virtual server. The IPS is positioned between the switch and the router, and the network is physically partitioned. IPS is operated the related directions to the virtual honeypot server to which the virtual network interface is connected using layer 2 and layer 3 protocols. By

connecting to the switch a second time with the port configured as a trunk of the switch, the IP addresses of the subnets of all the VLANs on the switch are represented by a central honeypot server. Figure 5 presents the network flow diagram given in Figure 4. Figure 5 shows how the software switch running on the proposed IPS

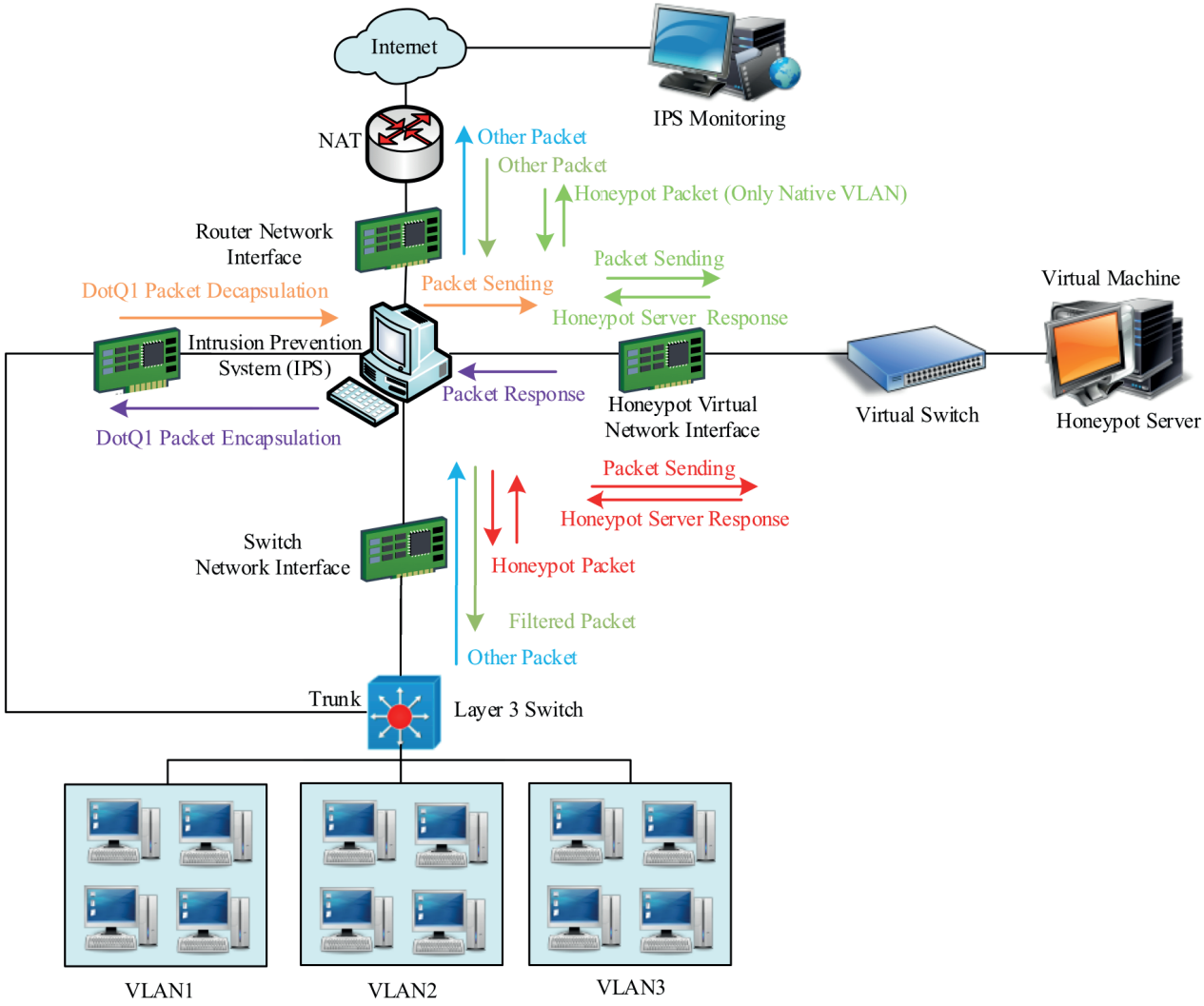

Figure 5. Flowchart of IPS network flow on layer 3 switch.

system communicates with the layer 3 switches. Accordingly, the software switch application needs to perform routing using four different interfaces. Switches and router-connected interfaces use the same algorithm that was previously designed for layer 2 switches. Native VLANs are also supported in this design because the port set as the trunk is excluded from the port used for communication with the external network. Native VLANs follow the path shown with red arrows while other VLANs will reach the IPS from the trunk-configured port. Unlike the previous application, the software switch on the IPS will also decapsulate and encapsulate packets from the port-based interface, which is set as the trunk of the switch.

3.7. Adding SoftSwitch to the application interface

The MAC address of the router is the interface of the switch, which is used in the network and the relevant network interfaces, have been accomplished using a simple and understandable design as shown in Figure 6.

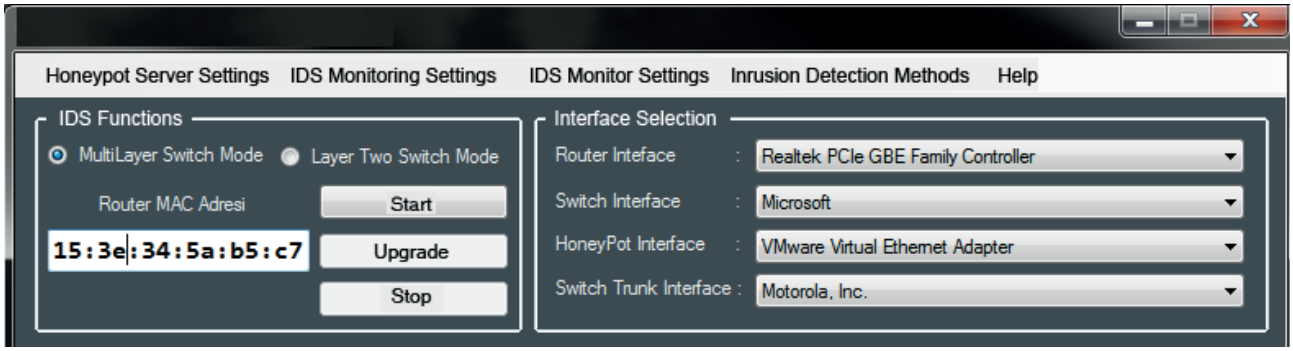

Figure 6. The application interface.

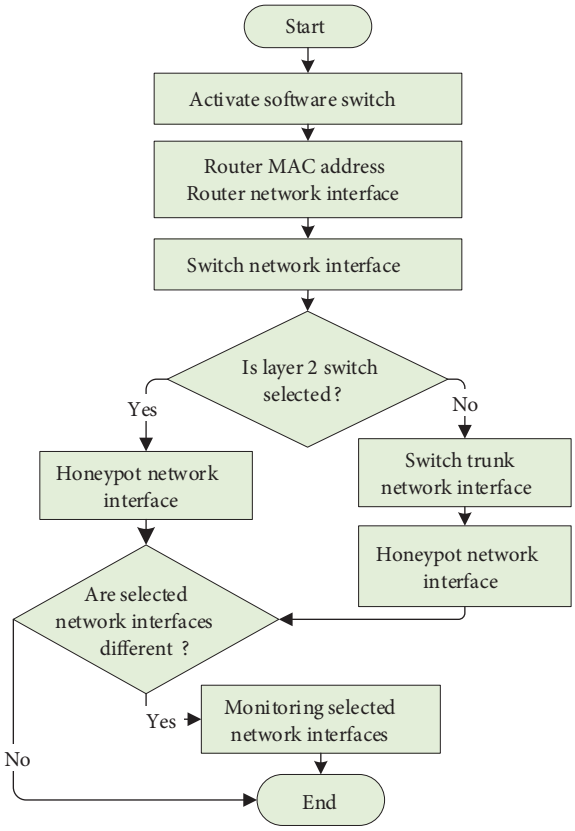

Figure 7. Software switch algorithm scheme.

Figure 7 shows the algorithm diagram when the 2nd and 3rd layers switches are in the selected positions. After the start button is pressed, this flow is performed and the asynchronously selected interfaces receive the traffic; thus, the routing process gets executed. Figure 7 presents the algorithm diagram of the developed software switch. Based on the configuration of layer 2 and layer 3 switch devices in any enterprise campus networks with VLANs, the software switch capable of listening to selected network interfaces is run on a centrally located IPS and as shown in Figure 7. This configuration provides the network flow according to the possible scenarios. In Figure 8, the network flow for the routers connected network interface is schematically shown. Figure 9 shows the flow diagram for the switched network interface. The flow shown in Figure 9 occurs only when the third

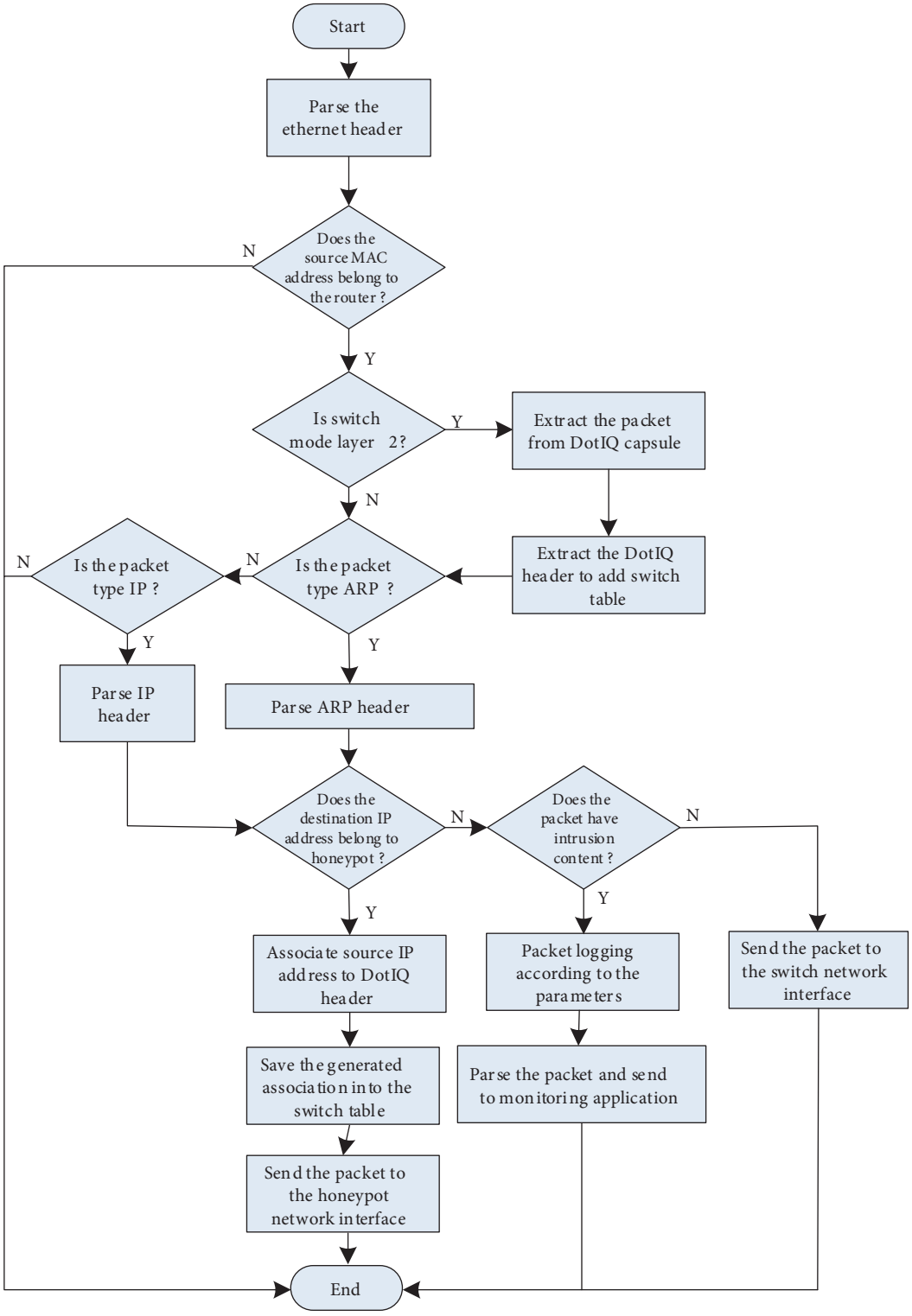

Figure 8. Flowchart of network interface connected to router.

layer switch type is selected. In this way, honeypots on all subnets of the VLANs on the 2nd layer and 3rd layer switches are represented by a single honeypot server to achieve the purpose of this work.

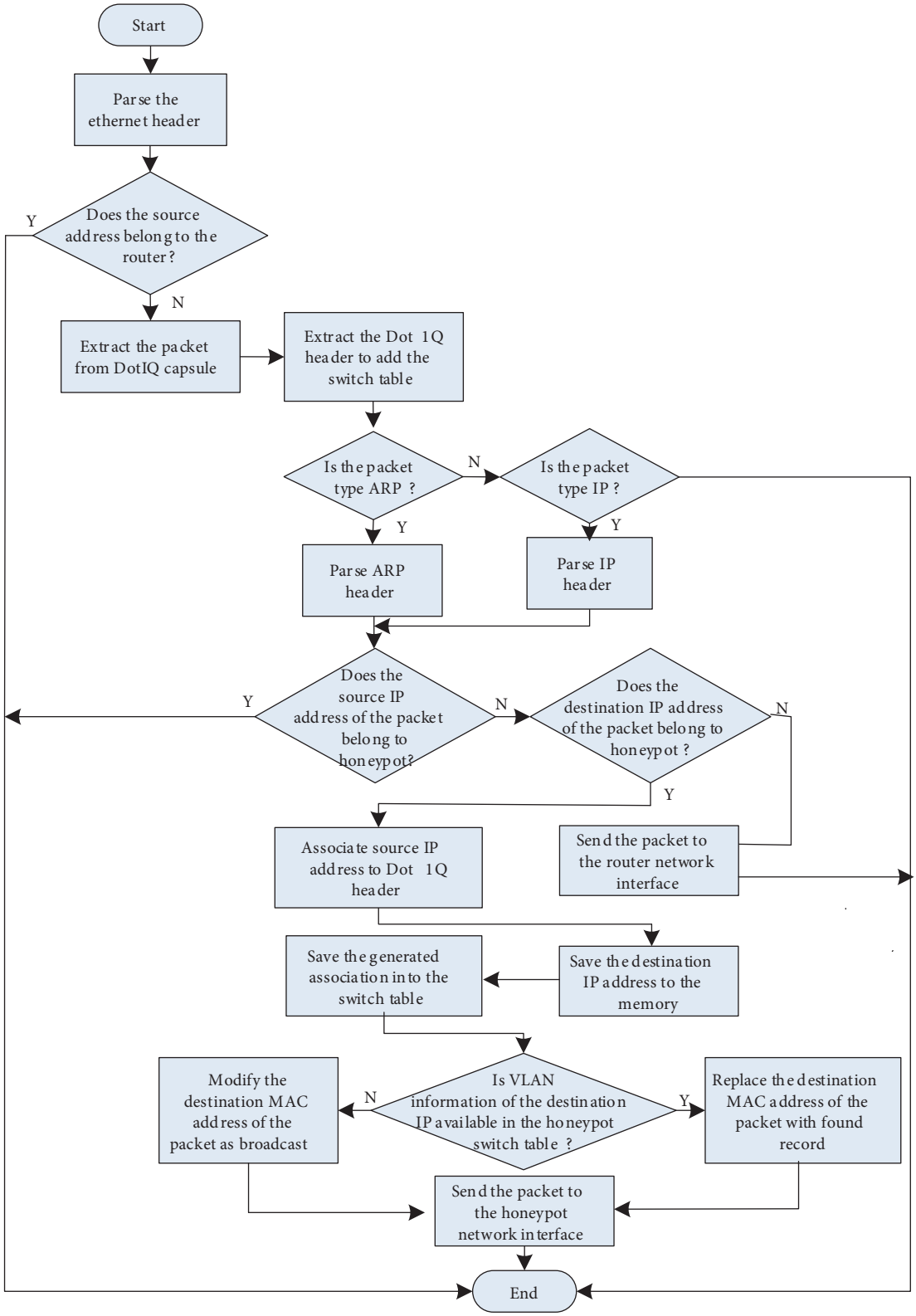

Figure 9. Flowchart of network interface connected to switch.

3.8. Performance evaluation and benchmarking

In this study, control of the VLAN networks in institutional networks with a central honeypot server in a single point with the software switch developed is provided. In this sense, a unique contribution has been made to the capabilities of classical intrusion detection systems. The soft-switch approach eliminates the need to configure a honeypot for each VLAN, which reduces costs. To our knowledge, there is no institutional honeypot-based intrusion detection and prevention system that shows both internal and external attacks together in the literature. Soft-switch can be used to overcome this problem. To evaluate our study with the relevant studies, a comparison is given in Table 3.8. To determine the performance of the developed system, it is needed to measure

Table. A comparison of the relevant studies on honeypot-based IDSs.

Ref.	Specification	Honeypot level	Application environment
[6]	0-day malware detection	Low-, middle-, and high-level honeypot	LAN, DMZ, and internet
[7]	Simulation and virtualization at the service, operating system, and network level	Low-, middle-, and high-level honeypot	IPv6 networks
[12]	Front-end content filters	Low- and high-level honeypot	Network and host components
[18]	Detailed research	High-level honeypot	Employment network
[19]	Multiprocess approach creates a virtually isolated environment	High-level honeypot	High-interaction system
[27]	Rule-based	Low-level honeypot	Web-based monitoring interface
[28]	Experiences with a low- and high-interaction-honeypot	Low- and high-level honeypot	University network
[29]	Capture and analysis of malicious traffic	Low-level honeypot	VoIP environments
[30]	Preventing XSS and SQL injection attacks	Low-level honeypot	Web application
[31]	Classification and identification of unwanted traffic	Low-level honeypot	IP networks
[26] Our study	Honeypot IDPS, low cyber-security cost, centralized control, low false alarm, protocol-based analysis	Low-level and configurable honeypot	Institutional network (Adaptable to other network systems)

the transaction rate of Honeyd and SNORT. The developed honeypot IDPS can simulate thousands of machines and address space that can be arbitrarily large, because of using Honeyd. When honeypot IDPS applications are used, the important benchmarking features are bandwidth rate, transaction rate, and packet sending/receiving performance. To see the bandwidth that Honeyd can support, the number of returned ICMP packets are checked. The number of TCP transactions per second that Honeyd supports for different configurations is measured to determine TCP performance. Performance decreases slightly when each of the 65K honeypots is configured individually. On a system that has 1 GHz Pentium II processor, Honeyd supports about 2000 transactions per second. In our system (i7-3770 3.4 GHz) honeypot IDPS support approximately 5000 transactions per second. With the developed soft switch, internal and external network traffic was made traceable together. So package, port, and protocol-based values can be tracked and logged.

Figures 10 and 11 show the number of attacks, detected attacks, and performance rates in a VLAN network, taking into account protocol and port numbers. Figure 10 shows the results obtained without SoftSwitch, while Figure 11 shows the results detected with SoftSwitch. As it is seen from these figures, more successful results were obtained in the detection of attacks for each different protocol while SoftSwitch

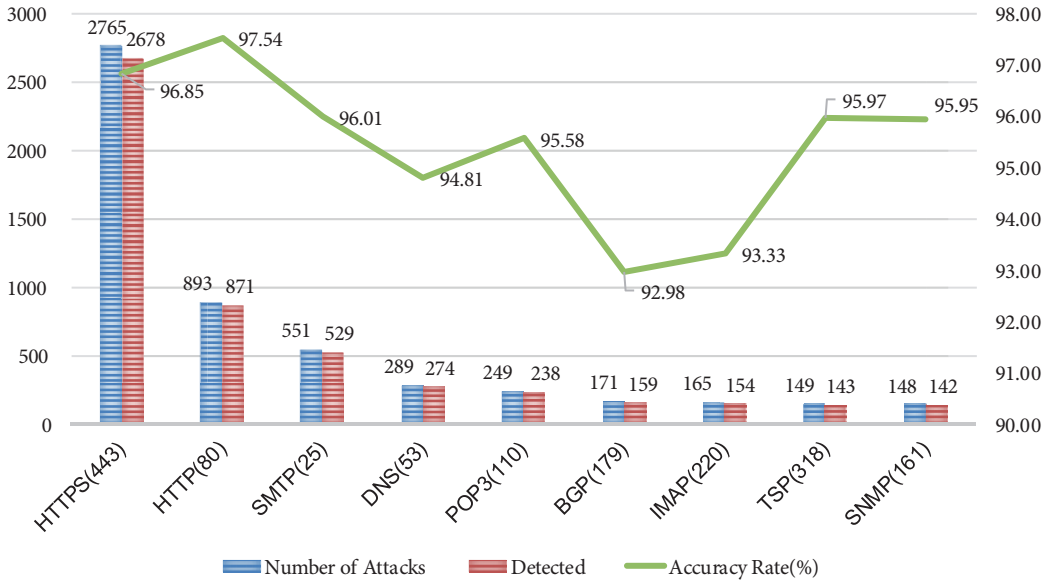

Figure 10. Accuracy rate of honeypot IDPs without SoftSwitch approach.

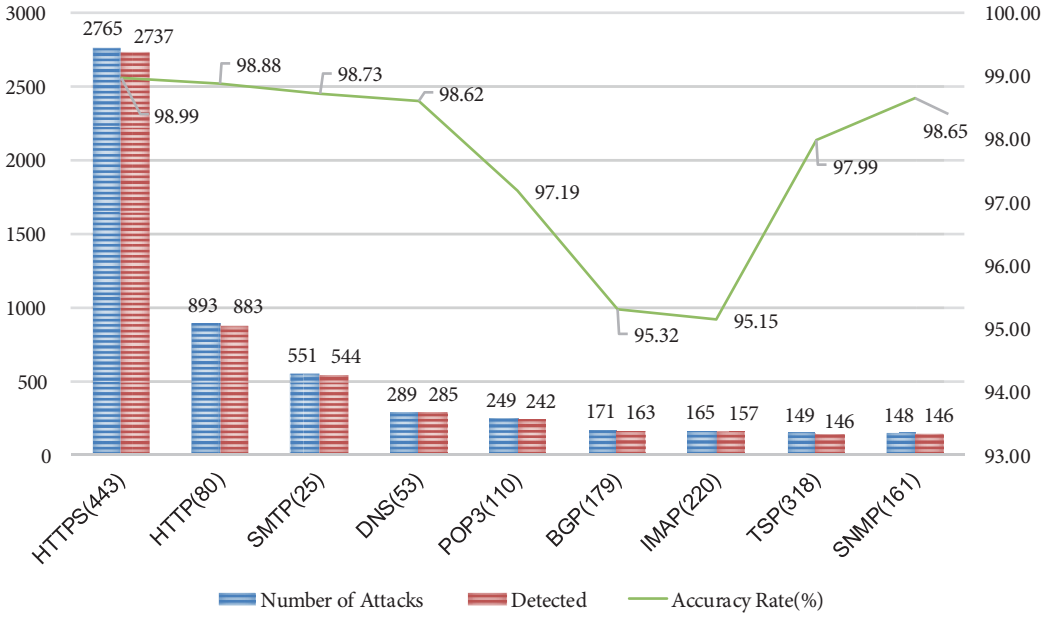

Figure 11. Accuracy rate of honeypot IDPs with SoftSwitch approach

was active. The most attacked services are HTTP, HTTPS, and SMTP. Studies on the improvement of false alarm level in the literature provide improvements between 7 per thousand and 1 per hundred. When new attack patterns and signatures are generated using honeypot systems, a significant contribution is made to the reduction of false alarm rate in rule-based systems. However, especially in anomaly-based systems, performance can be less because of the difficulty of detecting zero-day attacks in real time scenarios. In this study, an improvement of 1.34% to 3.81% was observed as shown in figures. It should be noted that these rates may be different in different time periods. Developed honeypot-based IDPS can receive, drop, analyze and filter

the network traffic. The system uses the SNORT IDS basically; hence, the limitation of the developed system depends on SNORT limitation. Honeypot-based IDPS can process the network traffic to 600 Mbps without highly dropping so it can be used for the large corporate network.

4. Conclusion and discussion

Information and computer systems are becoming increasingly prevalent along with rapidly evolving technology. Therefore, the necessity for the measures to be taken in terms of system security has reached the highest level. Although the tools used in information security systems are very diverse and powerful, they can be successful in finding new security vulnerabilities arising from emerging new technologies. In terms of security, there is never a hundred percent safe system. Detecting how attackers can attack before malicious attempts and ensuring security accordingly will bring the effectiveness of the measures taken to the highest level. Honeypot systems, which are preferred for predicting attempts by attackers and taking precautions accordingly, provide a serious advantage at this point. A malicious intruder looking for a victim in a network will try to find a computer with vulnerabilities that will work on the network.

Honeypot systems draw the attacker's attention and traps them. The intention of these systems, which are deliberately weakened, is to detect and record attacks. Honeypot systems, such as IDS and other security tools, play an important role in the detection of new vulnerabilities and in the discovery of new methods of attack. As a result of examining corporate networks and honeypot designs, it has been determined that in a network configuration where a VLAN configuration is used, at least a machine with a different network interface must be used for each VLAN. Using a real machine for each VLAN, the application of honeypots throughout the network increases the installation and maintenance costs especially in large corporate networks, including campus networks. For this reason, a centralized software switch application has been developed. This application enables honeypots to operate throughout the network with a network interface to reduce installation, maintenance, and management costs in large-scale enterprise networks. In enterprise networks with VLANs, VLAN networks can be restored by designing a unique software switch that can listen to layer 2 and layer 3.

In this study, IPS and honeypot servers can easily interact with each other thanks to the implemented software switch application and operations such as configuration, maintenance, network monitoring, and network analysis can be done easily with an easy-to-use interface. The implemented software switch application was applied in a hybrid scenario. The originally developed IDS/IPS with a honeypot application that uses a low-interaction Honeyd software as an attack traversal as well as the SNORT rule bases is used to provide network security. As a result of the successful application of the software switch developed within the scope of the study, the problem of not being able to detect new attack patterns (zero-day), which is one of the disadvantages of signature-based attack detection methods, has been reduced to a minimum level by analyzing log records collected from honeypot servers. Because log records collected from honeypot servers are mostly attack content, new attack patterns can be added to the signature database by generating signatures from log records. Extracting logs collected from honeypot servers from anomaly detection methods and non-attack log records increases the performance. The attack log records collected from honeypot servers reduce the high false alarm level, which is a disadvantage for anomaly detection methods, yet the developed hybrid attack detection method provides a higher performance.

Acknowledgments

This paper was produced from the PhD dissertation entitled "Designing and implementing attack detection and prevention approaches for information systems" presented at Fırat University, Graduate School of Natural

and Applied Sciences, Department of Software Engineering. In addition, this study was supported by grants given to PhD dissertation project by the Scientific Research Projects Administration Unit of Fırat University, Elazığ, Turkey [Grant number: TEKF.15.04]. The authors would like to thank the Scientific Research Projects Administration Unit of Fırat University for its financial support during their studies.

References

[1] Baykara M, Daş R, Karadoğan İ. Bilgi güvenliği sistemlerinde kullanılan araçlarin incelenmesi. In: 1st International Symposium on Digital Forensics and Security (ISDFS-13); Elazığ, Turkey; 2013. pp. 231-239 (in Turkish with an English abstract).

[2] Sagiroglu S, Yolacan E N, Yavanoglu U. Designing and developing an intelligent intrusion detection system. Journal of the Faculty of Engineering and Architecture of Gazi University 2011; 26 (2): 325-340 (in Turkish).

[3] Silnov DS, Prokofiev AO, Berezovskaya G, Perevozchikov VA, Troitskiy S S et al. A method of detecting a malicious actions using HTTP and FTP protocols. In: Intelligent Systems Conference 17 (IntelliSys); London, UK; 2017. pp. 1083-1088.

[4] Saadi C, Chaoui H. Cloud computing security using IDS-AM-Clust, Honeyd, honeywall and Honeycomb. Procedia Computer Science 2016; 85 (1): 433–442. doi: 10.1016/j.procs.2016.05.189

[5] Wang J, Zeng J. Construction of large-scale honeynet based on Honeyd. Procedia Engineering 2011; 15 (1): 3260-3264. doi:10.1016/j.proeng.2011.08.612

[6] Malanik D, Kouril L. Honeypot as the intruder detection system. In: Recent Advances in Computer Science; Kos, Greece; 2013. pp. 96-101.

[7] Gökırmak Y, Bektaş O, Soysal M, Yiğit S. Sanal IPv6 balküpü ağı altyapısı: kovan. In: Ulusal IPv6 Konferansı; Ankara, Turkey; 2011. pp. 49-55 (in Turkish).

[8] Schindler S, Schnor B, Kiertscher S, SchefflerT, Zack E. HoneydV6: A low-interaction IPv6 honeypot. In: International Conference on Security and Cryptography (SECRYPT); Reykjavik, Iceland; 2013. pp. 1-12.

[9] Kaur S, Singh M. Automatic attack signature generation systems: a review. IEEE Security & Privacy 2013; 11 (6): 54-61. doi: 10.1109/MSP.2013.51

[10] Li L, Sun H, Zhang Z. The research and design of honeypot system applied in the LAN security. In: IEEE 2nd International Conference on Software Engineering and Service Science; Beijing, China; 2011. pp. 360-363.

[11] Li S, Zou Q, Huang W. A new type of intrusion prevention system. In: International Conference on Information Science, Electronics and Electrical Engineering; Sapporo, Japan; 2014. pp. 361-364.

[12] Chawda K, Patel AD. Dynamic and hybrid honeypot model for scalable network monitoring. In: International Conference on Information Communication and Embedded Systems (ICICES2014); Chennai, India; 2014. pp. 1-5.

[13] Suo X, Han X, Gao Y. Research on the application of honeypot technology in intrusion detection system. In: IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA); Ottawa, ON, Canada; 2014. pp. 1030-1032.

[14] Paul S, Mishra B K. Honeypot based signature generation for defense against polymorphic worm attacks in networks. In: 3rd IEEE International Advanced Computing Conference (IACC); Ghaziabad, India; 2013. pp. 159-163.

[15] Beham M, Vlad M, Reiser H P. Intrusion detection and honeypots in nested virtualization environments. In: 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN); Budapest, Hungary; 2013. pp. 1-6.

[16] Dongxia L, Yongbo Z. An intrusion detection system based on honeypot technology. In: International Conference on Computer Science and Electronics Engineering; Hangzhou, China; 2012. pp. 451-454.

[17] Pomsathit A. Effective of unicast and multicast IP address attack over intrusion detection system with honeypot. In: Spring Congress on Engineering and Technology; Xian, China; 2012. pp. 1-4.

[18] Zhen J, Liu Z. New honeypot system and its application in security of employment network. In: IEEE 6 Symposium on Robotics and Applications (ISRA); Kuala Lumpur, Malaysia; 2012. pp. 627-629.

[19] Akiyama M, Kawakoya Y, Hariu T. Scalable and performance-efficient client honeypot on high interaction system. In: IEEE/IPSJ 12th International Symposium on Applications and the Internet; İzmir, Turkey; 2012. pp. 40-50.

[20] Buvaneswari M, Subha T. IHoneycol: a distributed collaborative approach for mitigation of DDoS attack. In: International Conference on Information Communication and Embedded Systems (ICICES); Chennai, India; 2013. pp. 340-345.

[21] Alnabulsi H, Islam MR, Mamun Q. Detecting SQL injection attacks using SNORT IDS. In: Asia-Pacific World Congress on Computer Science and Engineering; Nadi, Fiji; 2014. pp. 1-7.

[22] Vukalović J, Delija D. Advanced persistent threats - detection and defense. In: 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO); Opatija, Croatia; 2015. pp. 1324-1330.

[23] Mehta V, Bahadur P, Kapoor M, Singh P, Rajpoot S. Threat prediction using honeypot and machine learning. In: International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE); Noida, India; 2015. pp. 278-282.

[24] Baykara M, Das R. A novel hybrid approach for detection of web-based attacks in intrusion detection systems. International Journal of Computer Networks And Applications 2017, 4 (2): 62-76. doi: 10.22247/ijcna/2017/48968

[25] Baykara M. Design and implementation of intrusion detection and prevention approaches for information systems. PhD, Fırat University, Elazığ, Turkey, 2016.

[26] Baykara M, Das R. A novel honeypot based security approach for real-time intrusion detection and prevention systems. Journal of Information Security and Applications 2018, 41 (1): 103-116. doi: 10.1016/j.jisa.2018.06.004

[27] Mai Y, Upadrashta R, Su X. J-Honeypot: a java-based network deception tool with monitoring and intrusion detection. In: International Conference on Information Technology: Coding and Computing; Las Vegas, NV, USA; 2004. pp. 804-808.

[28] McGrew R, Vaughn JR. Experiences with honeypot systems: development, deployment, and analysis. In: Proceedings of the 39th Hawaii International Conference on System Sciences; Kauia, HI, USA; 2006. pp. 1-9.

[29] Vargas IRJdS, Kleinschmidt JH. Capture and analysis of malicious traffic in VoIP environments using a low interaction honeypot. IEEE Latin America Transactions 2015, 13 (3): 777-783. doi: 10.1109/TLA.2015.7069104

[30] Djanali S, Arunanto F, Pratomo BA, Baihaqi A, Studiawan H et al. Aggressive web application honeypot for exposing attacker's identity. In: 1st International Conference on Information Technology, Computer, and Electrical Engineering; Semarang, Indonesia; 2014. pp. 212-216.

[31] Puska A, Nogueira M, Santos A. Unwanted traffic characterization on IP networks by low interactive honeypot. In: 10th International Conference on Network and Service Management (CNSM) and Workshop; Rio de Janeiro, Brazil; 2014. pp. 284-287.