JURNAL UPI

Journal of Computers for Society

Journal homepage: https://ejournal.upi.edu/index.php/JCS

Journal of Computers for Society

Implementation of Signature based Intrusion Detection System with Snort Rule on E-Voting System

Muhammad Adnan Khairi A.S., Eddy Prasetyo Nugrohob, Rizky Rachman J.*

Computer Science Education, Universitas Pendidikan Indonesia, Indonesia
*Correspondence: E-mail: dnankhairi@student.upi.edu

ABSTRACT

Security is an important thing for everyone, including network security, which everyone needs, including the security at web server, there are problems encountered on the server one of which is on the E-voting site server, this server serves to store all the data storage of votes in an election between registered candidates. In this paper we propose a solution to detect these attacks using SNORT IDS. snort will detect an attack by adding a special rule to handle the attack. We tested the proposed solution by comparing the system against four different attacks, the result was that DDoS attacks had the greatest number of data packets compared to other attacks.

ARTICLE INFO

Article History:
Submitted/Received 04 Feb 2023
First Revised 28 Mar 2023
Accepted 12 May 2023
First Available Online 13 May 2023
Publication Date 15 Jun 2023

Keyword:
Computer security,
E-voting,
Intrusion detection system,
Snort.

1. INTRODUCTION

Security is an important thing for everyone, including network security, which everyone needs. According to Abomhara and Koien (2015), there are various types of vulnerabilities to system resources that will be a threat. A threat is something that represents a potential security hazard to an asset or system. There are also attacks that are threats that are carried out (threatening actions) and if the attack is successfully carried out, the results will lead to unwanted security breaches, or the consequences of the threat. Someone who attacks is called an Attacker or Threat Agent. This attack can occur anytime and anywhere, both in a computer network and a computer system on a server which will later harm those who own the system; therefore, efforts are made to prevent this from happening by using an attack detection tool or Intrusion Detection System (IDS). This IDS serves to detect an attack that occurs in network traffic then this detection is reported to the user or server admin to make further countermeasures for attacks that have been detected.

Problems encountered on the server, one of which is on the E-voting site server, this server serves to store all the data storage of votes in an election between registered candidates. This e-voting server will be very useful for selecting a candidate quickly, accurately, and accurately, and access to vote will be easier than using manual election. However, the more voters and the greater the area that takes part in the election, the more e-voting sites will be vulnerable to attacks by attackers, whether there is data leakage, data theft, server performance problems that are not functioning properly, and problems others that will endanger the server or server manager.

For example e-voting can be done in a school, some of which are to elect the student council president or something, and the election requires a sufficient number of votes from students or teachers who are in the school environment so that it takes a long time to collect votes if With manual voting, however, if you use the e-voting system, this time limitation can be overcome very well, and when it happens at the same time it will allow an unwanted party attack both inside and outside the school so that There will be problems that will be research in all of this paper.

2. RELATED WORKS

2.1. Computer Security

Security is an ongoing process to protect objects from unauthorized access. Security is a state of feeling protected from danger. A person, an entity like a company, or a piece of property like a computer system or file can all be the subject of security (Kizza & Kizza, 2013).

To be considered sufficiently advanced across the security spectrum, a system must adequately handle identification, authentication, access control or authorization, availability, confidentiality, integrity, accountability, and non-repudiation (Abomhara & Koien, 2015).

Identification is the process of identifying someone to another entity or determining the identity of the individual or entity that you communicate with.

Authentication serves as evidence of a user in an existing system. Authentication is very important if there is trust between the parties. Authentication is required when communicating through the network or entering the network.

Authorization refers to the ability to control the level of access an individual or entity has to a network or system and how much information they can receive.

The ability of the network, systems, hardware, and software to promptly and fully recover in the case of a service interruption is referred to as availability. These components ought to be impervious to denial-of-service attacks in theory.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2723-4088

Confidentiality refers to protecting information from unauthorized disclosure. This is usually achieved by restricting access to information or by encrypting information so that it does not mean unauthorized individuals or entities.

Integrity can be considered as accuracy. This refers to the ability to protect information, data, or transmission from unauthorized, uncontrolled, or accidental changes. The term integrity can also be used in connection with the functioning of a network, system, or application.

Accountability refers to the ability to track or audit what individuals or entities do on a network or system.

Non-repudiation is the ability to prevent individuals or entities from denying (rejecting) that information, data, or files are sent or received or that information or files are accessed or changed, when in fact. This capability is very important for e- commerce. Without it an individual or entity can deny that he, he, or is responsible for a transaction and that he, he, or that, therefore, is not financially responsible.

There is also other related terminology, namely the presence of Threats and Vulnerabilities, which consists of Hardware and Software Vulnerabilities, Media Vulnerabilities, Transmission and Emanation Vulnerabilities, and Human Vulnerabilities. What will be discussed in this section is about Threat and Vulnerability in general.

Threat is anything that can interfere with the operation, function, integrity, or availability of the network or system. This can be in any form and can be evil, unintentional, or just a natural act.

Vulnerability is a weakness inherent in the design, configuration, implementation, or management of a network or system that makes it vulnerable to threats. Vulnerability is what makes a network vulnerable to information loss and downtime. Every network and system have some kind of vulnerability.

2.2. Intrusion Detection System

Intrusion detection system is another tool for security staff to use to protect organizations from attacks. Intrusion detection is a reactive concept that tries to identify hackers when penetration is carried out (Li *et al*., 2001). According to Sobh (2006), IDS can be in the form of software or hardware or a combination of both that detects intrusion into a system or network.

There are two basic detection methods on IDS according to Maseer *et al*. (2021) namely Anomaly Based and Signature Based.

Anomaly Based: Anomaly based techniques are able to detect unknown attacks due to the ability to learn.

Signature Based: Signature based techniques depend on predetermined attack signature rules that enable them to achieve very high accuracy in detecting known attacks. When new attacks are identified, experts or programs must identify typical patterns in such attacks, which can be used as signatures. Because this process takes time, there will be a gap between the new threats found and the signatures applied in IDS to detect threats (Vuppala & Farik, 2016). Usually this method is called misuse-based detection (Le Jeune *et al*., 2021), constantly monitoring packets on the network and comparing them with database signatures or attributes of known dangerous threats.

In signature-based detection, the pattern of attack that is predetermined in the form of a signature and signature is further used to determine network attacks (Kumar & Sangwan, 2012). They usually check network traffic with a predetermined signature and every time the database is updated. Examples of Signature- based intrusion detection systems are Snort.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2774-7891

2.3. Intrusion Detection System Types

There are two types of IDS according to Li *et al*. (2001) namely host- based (H-IDS) and network-based (N-IDS). H-IDS resides on a particular host and looks for indications of attacks on that host. N-IDS is a separate system that monitors network traffic, looking for indications of attacks that cross that part of the network.

Networks intrusion prevention systems (NIPS) and network intrusion detection systems (NIDS) known as Snort were developed by Martin Roesch in 1998. Sekhar *et al*. (2015) explain that Snort is now developed by Sourcefire, where Roesch is the founder and CTO. In 2009, Snort entered the Open-Source Hall of Fame InfoWorld as one of the "greatest open-source software of all time". The open-source network-based (NIDS) Snort intrusion detection system has the ability to perform real-time traffic analysis and packet logging on an Internet Protocol (IP) network. Snort is a flexible tool with a variety of uses. This is intended to be used in the most classical sense of network intrusion detection systems (Olanrewaju *et al*., 2018).

Snort is logically divided into several components. These components work together to detect certain attacks and to produce output in the required format of the detection system.

Snort Architecture categorized into five basic modules (Olanrewaju *et al*., 2018) namely Libcap, Packet Decoder, Preprocessors, Detection Engine and Output plugins. Initially, traffic originating from the Internet is received by the router and forwarded to be routed. The switch then sends this data traffic to the firewall for the first level evaluation. After that the firewall forwards it to the Ethernet server adapter. Here Snort is the focus for all types of evaluation of the data package. The arrangement of these parts is depicted in **Figure 1**. The packet decoder receives all Internet-sourced data packets. It is either dropped, recorded, or a warning is created en route to the output module.

Packet Capture Library captures packets from various network interfaces in a network. On Linux and UNIX using the libpcap library and on Windows using WinPcap (Bhosale & Mane, 2015).

Mahmoud *et al*. (2016) and Olanrewaju *et al*. (2018) suggested that Packet decoders function to take packets from various types of network interfaces and prepare packets to be processed or sent to the detection machine. Packets are decoded here by the way each packet is diagnosed in terms of implementing the right and correct protocol to follow TCP / IP protocol with the help of several decoders (Goel & Varistha, 2017).

Packet Capture Library captures packets from various network interfaces in a network. On Linux and UNIX using the libpcap library and on Windows using WinPcap.

A preprocessor is a component that can be used to manage or modify data packages before the detection machine performs several operations to find out whether the package is used by intruders. The detection machine is responsible for detecting intrusion activity in a packet. The detection engine uses the Snort rule for this purpose.

Detection Engine is a time-critical part of Snort. Depending on how strong a computer machine is and how many rules have been set, it might take a different time to respond to different packages. If traffic on the network is too high when Snort works in NIDS mode, some packets might be canceled to get a real-time response.

Logging and Alerting systems depend on what detection machines are found in a packet, the package can be used to record activities or generate warnings. The log is stored in a simple text file.

Output modules can carry out operations to save the output generated by the Snort logging and warning system. Basically, these modules control the type of output produced by logging and alerting systems (Mahmoud *et al*., 2016).

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2723-4088

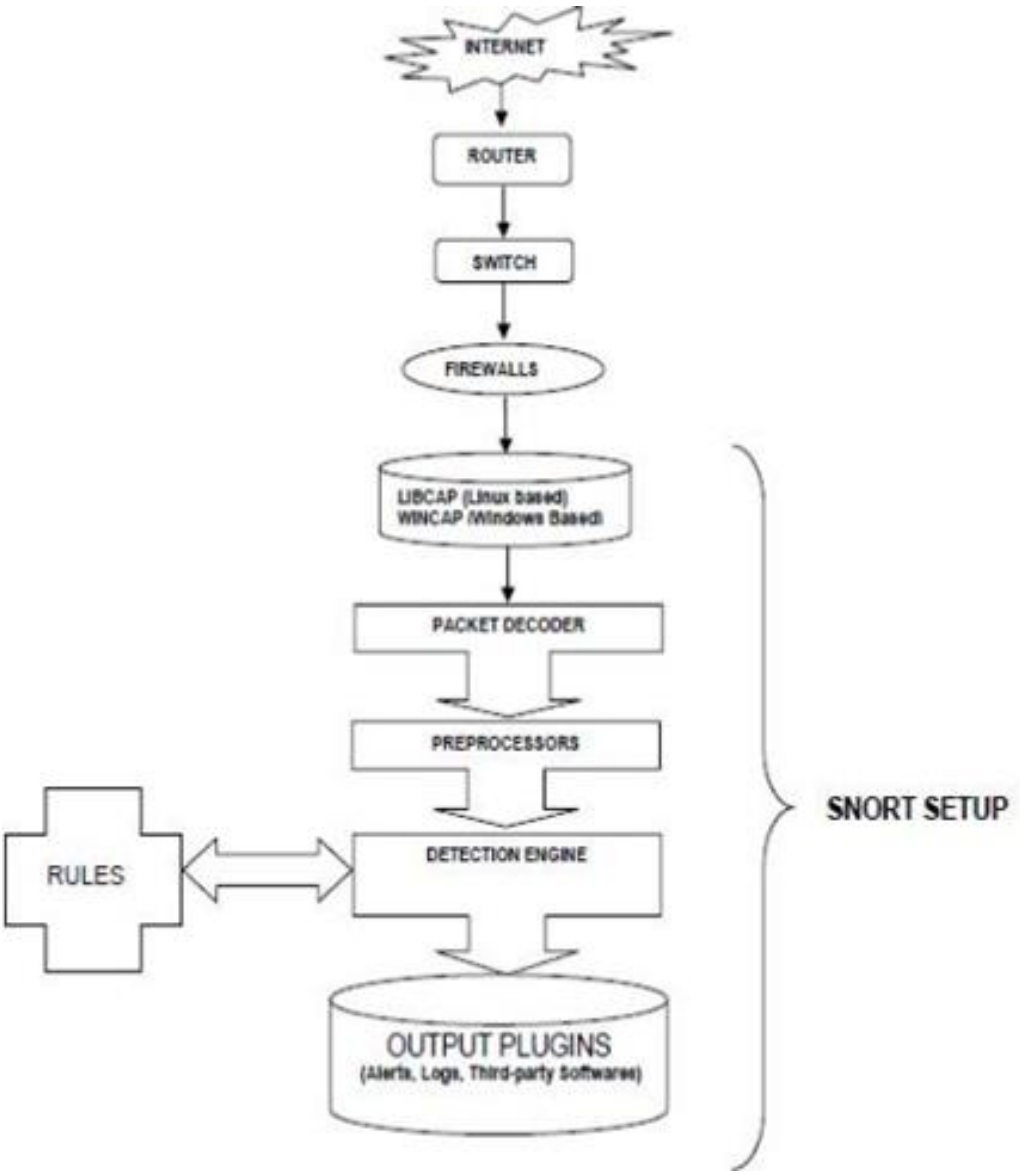

Figure 1. Architectural components compiled in Snort.

2.4. E-Voting

An understanding of E-voting refers more to the process of utilizing electronic devices to better support the smooth process and the automation model that allows minimal interference from individuals in all processes (Smith & Clark, 2005). Kohno *et* al. (2004) assert that the electronic election system cannot be separated from the importance of confidentiality and security. That is, if confidentiality and security are met, then E-voting is very appropriate to use. In addition, Abhipraya *et al*. (2023) emphasized the existence of several aspects of the benefits of implementing E- voting, namely:

(i) Costs: related to resources and investments that are more efficient compared to traditional systems that are complicated, complex, and inefficient.
(ii) Timing: related to the timing of the election being faster and calculating the results more precisely than the traditional system.
(iii) Results: related to the calculation of more precise and accurate results and minimization of the occurrence of human error cases as long as the system built is guaranteed from various threats of crime.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2774-7891

(iv) Transparency: related to the transparency of all processes because all is done by an automated system and in real time online.

2.5. Previous Works

There are several studies that have been carried out previously related to, among others Moloja and Mpekoa (2017) conducting research on how IDS can be implemented for detection and prevention of attacks that exist in an m-voting application that offers e- voting functions that can be carried out where course with the level of data collection speed that is very fast and fast and easy to use for the elderly and disabled. Zissis and Lekkas (2011) conduct research related to how to secure e-voting and e-Government systems by using open cloud computing architecture, which means there are various kinds of security tools used to protect attacks that will attack both parties, both from the system side and the voters participating in the election, or in certain government activities that require electronic devices. Chavaro *et al*. (2018) conducted research on how to implement several methods to obtain confidentiality and anonymity in E-Voting system services for educational institutions, including by creating e- voting machines specifically designed to have security features such as Encryption, anonymous access, and machines special authentication for voters to make the voting votes made legal and original without any engineering.

There are also several studies that examine how IDS is applied by Gupta *et al*. (2017) conducted research on how IDS can work to detect Cross Site Scripting (XSS) attacks on a web server. The method used is to try to attack the server using various tags and scripts that are common on the server side then capture the attack using a sniffer for detection, then Goel and Vasistha (2017) conduct research on the use of IDS by using Snort by implementing an existing Signature with an example detecting a virus in the data packet received, and Olanrewaju *et al*. (2018) implements IDS by creating a Signature rule that can be combined using the Machine Learning method, namely Artificial Neural Network with Snort and MATLAB. Yamamoto and Yamguchi (2023) also applies the Snort IDS rule to the honeypot log. The log obtained from the Honeypot IDS server is successfully sent to the server, and then based on the log obtained by the IDS server the rules are made. Then Jim *et al*. (2022) improvised IDS Snort with the Clonal Selection method to improve Signature based detection of unknown attacks. Saied *et al*. (2024) also implements IDS to protect the server from Brute force attacks with a Snort signature rule that prevents http-brute from being forwarded to the server. Khamphakdee *et al*. (2015) and Chanthakoummane *et al*. (2016) improvised the IDS rules on Snort in the detection of Probing Attacks with the Association Rules Technique in Data Mining. and develop intrusion detection on Snort rules for botnet attack detection with a signature rule. And Dewi *et al*. (2017) Implementing IDS with digital Forensic methods, to analyze network traffic that exists in Information Systems at a University.

3. METHODS

This chapter will explain the research methodology, starting from the research design, tools and research, and research methods.

3.1. Research Design

Problem Formulation after the results of the literature study that the author has done, is to contain the stages of the conclusion of the problem which will later be the main topic of this research, in **Figure 2** it is explained that the formulation of the problem contains several

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2723-4088

things that will be discussed in this study, namely the design of IDS with snort , Snort integration with web server services, and logging when an incoming attack occurs.

There is also a research need that will later become the main point of the research needed, for the needs themselves, namely the system to attack servers that have been made before, the attacks to be carried out in this study consist of Denial of Service (DoS), Port Scanning, Cross Site Scripting (XSS), and Structured Query Language Injection (SQLI).

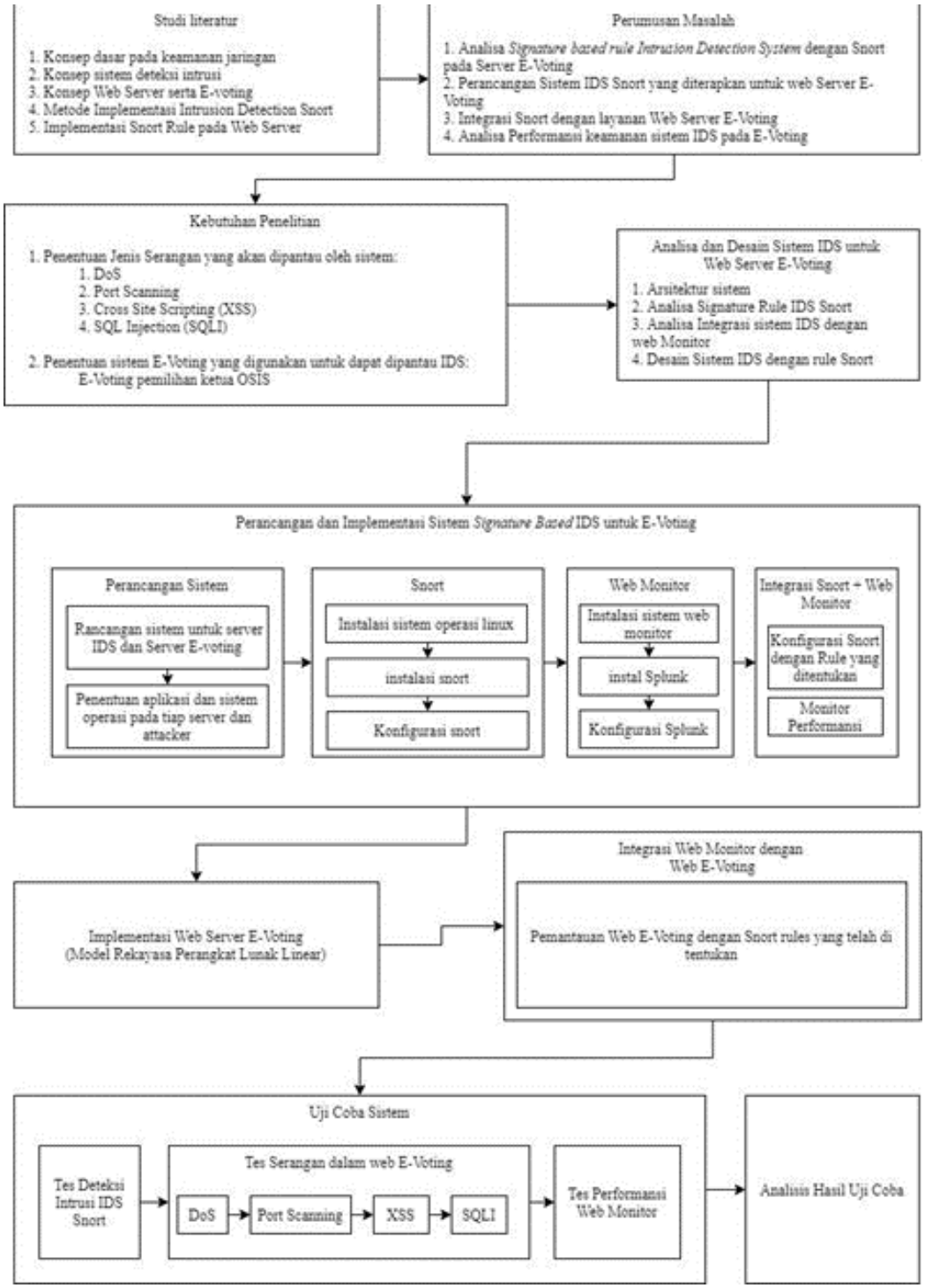

Figure 2. Formulation research diagram.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2774-7891

Web installation service is the preparation made by the author to do this research by installing Linux, Apache, MySQL, and PHP (LAMP) which will later be used to create a web monitor and web server itself.

Installing a Signature Based IDS system is the most important part of this study, Snort which will be used as an intrusion detection system that will be installed later, then configured later will be made together with a web monitor to facilitate checking for attacks on the server, by integrating between Snort with web monitor, configuration rules and then monitor performance and record all the logs that exist when a server attack is detected.

At the software development stage, a linear software model (waterfall) is used, this method consists of the analysis, design, implementation, and testing stages.

At the web monitor integration stage with the web server, some monitoring will be carried out by the web monitor of the activities contained on the web server, the web monitor will detect IDS attacks with predetermined rules for certain attacks.

After the system integration phase is carried out, the system is tested by 3 stages, namely 1) intrusion detection tests conducted by snort, namely testing the predetermined rules, then subsequently carried out 2) testing of attacks on the web server, this involves system that will be installed outside the server, which at this stage will do DoS, Port Scanning, XSS, and SQLi attacks into the server, and finally, by 3) performing a performance test by a web monitor, does this IDS system affect the system computer or not.

In the analysis phase of the trial results, all research results will be re-analyzed to get conclusions about the research that has been done then finally conducted documentation in accordance with applicable rules, at this stage is the last stage in the research to be conducted.

3.2. Research Limitation

There are limitations to the implementation of this research, namely:

(i) Hardware (Hardware), namely a computer with specifications:
   a. AMD Ryzen 5 3400G processor with Radeon Vega @ 3.7GHz
   b. 8 GB DDR4 Random Access Memory (RAM)
   c. AMD Radeon RX570 4GB GDDR5

(ii) Software (Software) as follows:
   a. Snort
   b. Text editor (nano, vim, Sublime Text 3, notepad)
   c. VMWare Workstation 12
   d. Ubuntu Linux (Ubuntu Server 18.04 LTS)
   e. Kali Linux 2019.4
   f. Windows 10 1903
   g. Opera GX web browser
   h. PhpMyAdmin

(iii) Attack carried out with Kali Linux
   a. Denial of Service (DoS)
   b. Port Scanning
   c. Cross Site Scripting (XSS)
   d. SQL Injection (SQLI)

(iv) IDS detection conducted by researchers comes from the detection generated by Rule snort, which has been integrated with Web Monitor.

(v) The research material that will be used comes from journals, textbooks, and some other documentation available on the internet.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2723-4088

(vi) The research will use the Signature Based Rule Network Intrusion Detection system (NIDS) method in Snort.

(vii) This research does not discuss in detail about making E-voting applications in the realm of software engineering (such as software requirements specification documents, data flow diagrams, etc. relating to detailed application system specifications). However, this research will focus on how the integration of IDS with a web server can be done along with how IDS can monitor the web server and detect attacks on the web server.

At the testing stage carried out in accordance with the attack and handling scenarios that have been determined by researchers.

4. RESULTS AND DISCUSSION

On the student council election e-voting website at high school, these attacks can be carried out and can occur both within the school environment and attacks carried out by outsiders who want to disrupt the process of conducting elections at the school, these attacks will include several the part of this chapter will be explained.

4.1. IDS System Design

This IDS system is designed and created with the aim of detecting various attacks carried out by others against the E- voting server, attacks that will be carried out include port scanning, SQL Injection, Cross Site Scripting, and Distributed Denial of Service (DDoS).

The existence of this system allows network administrators in the E-voting server to know what the characteristics of each predetermined attacks.

At the initial stage of making this IDS system, it is needed that the name of the network architecture, in **Figure 3** is a concrete network architecture design that has been designed. **Figure 3** shows that the Web Server and IDS server are made in different devices so that the IDS can become a Network Intrusion Detection System (NIDS) in the network's scope. The computer designs that have been provided later will access the E-voting web server for the testing and use of the web server, including the Attacker laptop device that can carry out penetration attacks to the web server section.

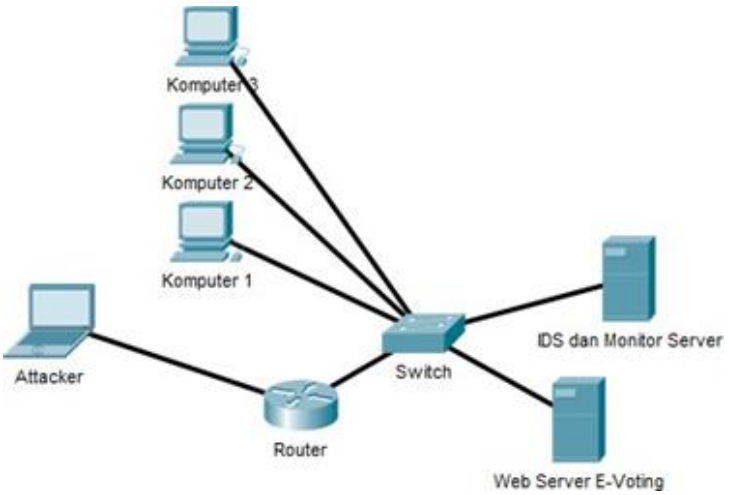

Figure 3. IDS architecture design.

With the installed Snort system, separate the rules that will be filled in by the signature rules of the four attacks (see **Figure 4**).

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2774-7891

```
sudo mkdir /usr/local/etc/snort/rules
sudo nano /usr/local/etc/snort/rules/local.rules
```

Figure 4. Installed Snort system.

The rule signature used by researchers is the rules relating to DDoS attacks, SQL Injection, Port Scanning, and Cross Site Scripting (XSS) (see **Figure 5**).

```
# DDoS Attacks

# 1. SYN FLOOD ATTACK
alert tcp any any -> 192.168.75.128 any (flags: S; msg:"Possible DoS
Attack Type : SYN
flood";flow:stateless;sid:10000001;detection_filter:track by_dst,
count 1000, seconds 10; rev:2)

# 2. UDP FLOOD ATTACK
alert udp any any -> 192.168.75.128 any (msg:"Possible DoS Attack Type
: UDP flood";flow:stateless;sid:10000002;detection_filter:track
by_dst, count 10000, seconds 10; rev:3;)

# 3. FIN-SYN FLOOD ATTACK
alert tcp any any -> 192.168.75.128 80 (flags: FS; msg:"Possible DoS
Attack Type : FIN-SYN flood"; flow:stateless;sid:10000003;
detection_filter:track by_dst, count 10000, seconds 10; rev:2;)

# 4. PSH-ACK FLOOD ATTACK
alert tcp any any -> 192.168.75.128 any (flags: PA; msg:"Possible DoS
Attack Type : PSH-ACK flood"; flow:stateless; sid:10000004;
detection_filter:track by_dst, count 10000, seconds 10; rev:2;)

# 5. RES/RST DOS
alert tcp any any -> 192.168.75.128 any (flags: R; msg:"Possible DoS
Attack Type : RST DoS"; flow:stateless; sid:10000005;
detection_filter:track by_dst, count 10000, seconds 10; rev:2;)

# 6. FIN Flood
alert tcp any any -> 192.168.75.128 any (flags: F; msg:"Possible DoS
Attack Type : FIN Flood"; flow:stateless; sid:10000006;
detection_filter:track by_dst, count 10000, seconds 10; rev:2;)

# 7. ICMP Smurf flood
alert icmp 192.168.75.128 any -> any any (msg:"Possible DoS Attack Type
: Smurf Attack Flood"; itype:8; sid:10000007; detection_filter:track
by_dst, count 10000, seconds 10; rev: 2;)

# 8. Slowloris DDoS
alert tcp any any -> 192.168.75.128 80 (msg:"Terjadi Serangan
Slowloris"; flow:to_server,established; pcre:"/X-a|3a| \d{4}../";
detection_filter: track by_dst, count 3, seconds 30; classtype:denial-
of-service; sid:10000098; rev:3;)

# SQL Injection Attacks

# 9. SERANGAN IDENTIFIKASI ', ", --, DAN #
alert tcp any any -> 192.168.75.128 80 (msg: "SQL Injection Attack
detected"; pcre: "/(\%27)|(\')|(\-\-)|(%23)|(#)/i" ; sid:10000009; )

# 10. SERANGAN IDENTIFIKASI OR
alert tcp any any -> 192.168.75.128 80 (msg: "OR SQL Injection Attack
detected"; pcre: "
/\w*((\%27)|(\'))((\%6F)|o|(\%4F))((\%72)|r|(\%52))/ix" ;
sid:10000010; )
```

Figure 5. Rules relating to DDoS attacks, SQL Injection, Port Scanning, and Cross Site Scripting (XSS).

There are various ways to make a web monitor, one of the ways used to make a web monitor is to use Splunk which is a tool that can function to view data and log information that has been obtained and then processed into statistical information that can facilitate network administrators through analysis data related to certain attacks on the E- voting site. Therefore, researchers installed a splunk to create a web monitoring (see **Figure 6**).

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2723-4088

```
# XSS Attacks

# 11. SERANGAN <script> (menggunakan pcre/regex)
alert tcp any any -> 192.168.75.128 80 (msg: "XSS Attack detected";
pcre: "/((\%3C)|<)((\%2F)|\/)*[a-z0-9\%]+((\%3E)|>)/i"; sid:10000016;
rev:1; )

# 12. SERANGAN <img src=> (menggunakan pcre/regex)
alert tcp any any -> 192.168.75.128 80 (msg: "XSS img src Attack
detected";                                                    pcre:
"/((\%3C)|<)((\%69)|i|((\%49)((\%6D)|m|(\%4D)((\%67)|g|(\%47)[^\n]+((
\%3E)|>)/i"; sid:10000017; rev:1; )
```

```
# Port scanning Attacks

# 13. no Ping probe attack
alert icmp any any -> 192.168.75.128 any (msg: "NMAP ping sweep Scan";
dsize:0;sid:10000018; rev: 1;)

# 14. Port tcp scan
alert tcp any any -> 192.168.75.128 22 (msg: "NMAP TCP
Scan";sid:1000018; rev:2;)
alert tcp any any -> 192.168.75.128 22 (flags: FPU; msg: "NMAP TCP
XMAS Scan";sid:10000019; rev:2;)
alert tcp any any -> 192.168.75.128 22 (flags: F; msg: "NMAP TCP FIN
Scan";sid:10000020; rev:1;)
alert tcp any any -> 192.168.75.128 22 (flags: 0; msg: "NMAP TCP NULL
Scan";sid:10000021; rev:1;)
```

Figure 6. Installed a splunk to create a web monitoring.

After all the rules are set then what is done is to make automation on snort3 so that detection can be easily carried out without having to run it manually using the command, this automation will run without a root system and has its own user to run it for security reasons (see **Figure 7**).

```
# /lib/systemd/system/snort3.service

[Unit]
Description=Snort3 NIDS Daemon
After=syslog.target network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/snort -c /usr/local/etc/snort/snort.lua -s
65535 -k none -l /var/log/snort -D -u Snort -g Snort -i ens33 -m 0x1b

[Install]
WantedBy=multi-user.target
```

Figure 7. Make automation on snort3.

Snort uses the alert_json feature to allow JSON files to be sent to various places depending on their needs. This feature can be done by configuring the snort.lua file (see **Figure 8**).

```
appid =
{
    -- appid requires this to use appIDS in rules
    app_detector_dir = '/usr/local/lib',
    log_stats = true,
}

alert_json =
{
       file = true,
       limit = 10,
       fields = 'seconds action class b64_data dir \
       dst_addr dst_ap dst_port eth_dst eth_len \
       eth_src eth_type gid icmp_code icmp_id \
       icmp_seq icmp_type iface ip_id ip_len msg \
       mpls pkt_gen pkt_len pkt_num priority proto \
       rev rule service sid src_addr src_ap \
       src_port target tcp_ack tcp_flags \
       tcp_len tcp_seq tcp_win tos ttl udp_len \
       vlan timestamp',
}
```

Figure 8. Configuring the snort.lua file.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2774-7891

4.2. E-Voting Site Implementation

The implementation of the E-voting site carried out for this research aims to be able to carry out testing which will later involve the IDS system as a security device.

There are several pages that can be accessed on this E- voting site, the first is the login page for all users, both admin and ordinary visitors, in **Figures 9 and 10** this shows the first login page on the e-voting site and the login page for registered users in the database.

Figure 9. E-voting site main menu.

Figure 10. User login menu.

Furthermore, in **Figure 11** after the user has logged in, a candidate will be presented, and the user must select one of the several options for the menu.

Figure 11. Candidate selection.

After the user chooses it will display a message finished selecting, in **Figure 12** a simple message is displayed which later this page will automatically log out and redirect to the main page for a new login.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2723-4088

Figure 12. Redirect page after selection of candidates.

4.3. Testing the IDS System on the E-Voting Website

After all the implementations that have been prepared, what is done in this study is to test the system that has been made, this test will be carried out by carrying out the following steps:

(i) IDS system detection test is a test to ensure that the IDS system is functioning properly detecting several common signatures such as ping server or identification of certain IP addresses.

(ii) The attack test is a continuation of the IDS system detection test, but this test uses certain attacks that have been predetermined in this test, attacks that will be carried out include, DDoS, Port Scanning, SQL Injection, and finally XSS.

(iii) Web monitor performance test is a continuation of the test after the attack test, by looking at the web monitor's response speed in detecting attacks that appear and then the attack is identified in the alert signature that is changed into the form of JSON which will be automatically detected by Splunk directly.

Testing this IDS detection test is to ensure that the IDS Snort system can function properly after installation. The rule used for testing this system is the icmp ping rule (see **Figure 13**).

```
#/usr/local/etc/snort/rules/local.rules

alert icmp any any -> 192.168.75.128 any (msg:"Terdeteksi Ping pada
server";sid:10000001;)
```

Figure 13. ICMP ping rule.

At the same time when the E-voting server is active, do the ping command to send an ICMP signal to the E-voting server. In **Figure 14** is how to ping the destination server 192.168.75.128.

```
adnan@ubuntu:~$ ping 192.168.75.128
PING 192.168.75.128 (192.168.75.128) 56(84) bytes of data.
64 bytes from 192.168.75.128: icmp_seq=1 ttl=64 time=0.514 ms
64 bytes from 192.168.75.128: icmp_seq=2 ttl=64 time=0.260 ms
64 bytes from 192.168.75.128: icmp_seq=3 ttl=64 time=0.342 ms
64 bytes from 192.168.75.128: icmp_seq=4 ttl=64 time=0.229 ms
64 bytes from 192.168.75.128: icmp_seq=5 ttl=64 time=0.256 ms
64 bytes from 192.168.75.128: icmp_seq=6 ttl=64 time=0.466 ms
64 bytes from 192.168.75.128: icmp_seq=7 ttl=64 time=0.296 ms
64 bytes from 192.168.75.128: icmp_seq=8 ttl=64 time=0.288 ms
^C
--- 192.168.75.128 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7163ms
rtt min/avg/max/mdev = 0.229/0.331/0.514/0.098 ms
```

Figure 14. Ping on the E-voting server.

So that later in **Figure 15** the results will be seen that a ping detection has occurred on the server generated by the Snort command that is active at that time.

```
05/14-06:01:01.511971 [**] [1:10000001:0] "Terdeteksi Ping pada server" [**] [P
riority: 0] [AppID: ICMP] {ICMP} 192.168.75.133 -> 192.168.75.128
```

Figure 15. The results of Snort ICMP ping detection.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2774-7891

Attack Test and attack detection is to conduct a predetermined attack on an existing E-voting site, attacks that will be carried out include DDoS, Port Scanning, SQL Injection, and XSS.

Figures 16-23 is the result of detection of a SYN Flood attack to a Slowloris attack by Snort.

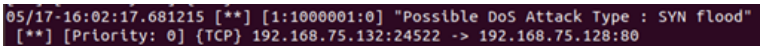

Figure 16. The results of the detection of SYN Flood attacks.

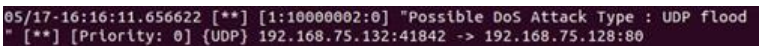

Figure 17. The results of the detection of UDP Flood attacks.

Figure 18. The results of the detection of FIN Flood attacks.

Figure 19. The results of the detection of PSH-ACK Flood attacks.

Figure 20. The results of the detection of RST Flood attacks.

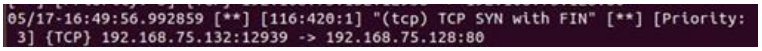

Figure 21. The results of the detection of FIN- SYN Flood attacks.

Figure 22. The results of the detection of Smurf ICMP Flood attacks.

Figure 23. The results of the detection of Slowloris attacks, it displays as PSH-ACK and SYN Flood.

Port scanning attacks are carried out so that the attacker can see how and where active and open ports appear, testing port scanning can use nmap as a tool to perform several types of scanning, such as fast scanning, TCP scan, UDP scan, Operating system (OS Scan), and others. **Figures 24-25** is the result of detection of a Port Scanning attack.

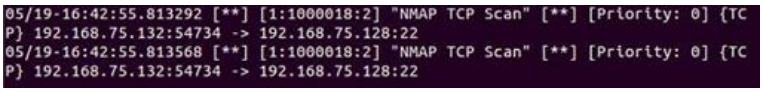

Figure 24. The results of the detection of TCP Port Scan attacks.

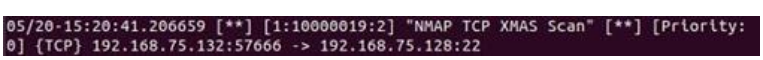

Figure 25. The results of the detection of TCP XMAS Port Scan attacks.

Cross Site Scripting (XSS) testing is a web attack carried out by code injection. A special Snort signature rule is created to detect this XSS attack, primarily this attack will definitely involve HTML injection and then marked by writing "<script> payload </script>" code in the URL tab as well as other XSS tools. **Figure 26** is the result of XSS Attack detection.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2723-4088

Figure 26. The result of XSS attack detection.

Testing SQL injection attack is done by entering the code of execution code in the browser URL, SQL injection itself consists of error-based attacks (the use of quotation marks 'and', the use of AND or OR both symbolic or not (&& or ||), and the use of UNION accompanied by the ORDER syntax. login, login here is to add SQL 'OR 1 = 1 injection to the password field. **Figures 27-28** is the result of SQLi Attack detection.

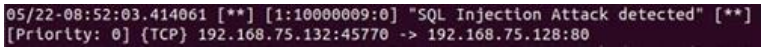

Figure 27. The result of Error-based SQL Injection attack detection.

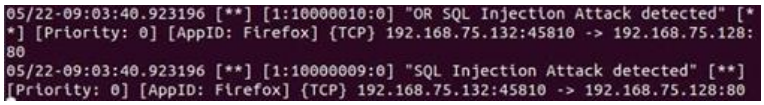

Figure 28. The result of OR SQL Injection attack detection.

The performance test is the final test that will be carried out in this paper, The first test is to carry out various attacks on all DDoS attacks carried out once with a time span of 30 seconds. Port scanning, XSS, and SQLi attacks will be carried out using cheat sheets that have been determined by the researchers. Data that has been collected will be entered into the snort log which then splunk will process this log. In **Table 1** is the result data of all DDoS attacks sent and detected when this attack runs for 30 seconds.

Table 1. DDoS Attack Detection Results are based on the packet sent.

	Attack Type	Number of trial / packages sent	number of packages detected	detection rate (%)
DDoS	SYN	2317310	160275	6.916
	UDP	2542351	154358	6.071
	FIN	2636511	199938	7.583
	SYN-FIN	2630071	173676	6.603
	PSH-ACK	2481514	91408	3.684
	ICMP	2405729	200240	8.323
	Slowloris	150	92	61.33

Basically, cks are attacks that require a lot of resources and computer computing so that in this case, all attacks that use hping will send as many packets as possible, at the same time snort detects the packet using a detection filter, for detection of SYN floods, carried out restrictions when incoming packets are more than 1000 packets and occur for less than 10 seconds it will activate the warning, while for other TCP flag attacks, UDP, and ICMP smurf flood, is limited to up to 10000 warnings in 10 seconds, because basically attacks It has a data packet that is quite clearly visible in packet capture, after that DDoS detection only logs less than 9% of the packet sent by the attacker, this happens partly because Snort IDS discards all packets it receives and only does logging according to its computational capabilities. For

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2774-7891

Slowloris attacks, the packets sent will be determined by the attacker so that the data packets that arrive will match those sent, but each HTTP data packet sent will cause an HTTP request that is so slow that the E-Voting server must serve the HTTP packet, detection at Snort to identify this attack is to look for headers sent, headers sent have "X-a:" with 4 random numbers in it. At the time of the 30-second attack, Snort gave 92 warnings against this attack. In port scanning attacks using nmap, the number of packet shipments carried out will differ based on the commands sent, so that when Snort detects this attack, there is a doubling of detection for each one of its attacks, proven that in **Table 2** is the result of detection of port scanning attacks carried out.

Table 2. Detection of Port Scanning Attack Results based on the packet sent.

	Attack Type	Number of trial / packages sent	number of packages detected
Port Scan	Ping Sweep (No. port)	5	22
	TCP Scan	4	168
	XMAS	4	8
	FIN	5	23
	NULL	6	32

In the TCP scan there is a very large increase in detection with other special Nmap attacks, because basically for XMAS, FIN, and NULL attacks still use TCP as an intermediary to detect these attacks, and moreover there is a Decoy (IP feed installed) so that it makes one Nmap command can be detected as much as n Decoy, besides that the operating system search command also makes TCP detection becomes more numerous, but with the same message by IDS. XSS attacks are then carried out, namely attacks that involve injection of the browser client which generally sends manipulated packages that allow the attacker to directly enter the system, this attack will involve JavaScript as an intermediary, and this e-voting server has JavaScript to modify its appearance, especially on when the login page, the attack will be carried out by manipulating the URL tab in the browser, **Table 3** is the result of the detection attempt carried out.

Table 3. Detection of XSS Attack Results based on the packet sent.

	Attack Type	Number of trial/packages sent	Number of packages detected
XSS	Script attack	5	5
	Script img=src	5	5

In XSS attacks detection will be easier to detect because basically this attack uses HTML code sections, namely <script> and <img src ... as a start for the injection, snort detection is in accordance with the experiments conducted by the attacker.

Next is an attack attempt and recording detection for SQL injection attacks, in this attack the signature rule used is a rule that detects all characters based on Error (', ", -, #, / *) and OR so that this signature rule will be easier to detect These characters are basically because these characters are used to attack the login page and to enter the system and change the code snippet that is in the e-voting server, in **Table 4** is an attempted attack carried out and the detection obtained.

Performance on the E-Voting site that runs will greatly affect the performance of the use of e-voting itself, thus the IDS system that has been made has an influence on the system itself, one of which is affecting the receipt of information and data attacks on the site,

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2723-4088

administrators can see and check for what information is on the web monitor, in **Table 5** is the difference in performance on the E-Voting Server.

Table 4. Detection of SQL Injection Attack Results based on the packet sent.

	Attack Type	Number of trial / packages sent	number of packages detected
SQL Injection	error based & comment based	8	22
	OR SQL injection	4	2

Table 5. Comparison of IDS and non-IDS system performance.

	Performance	IDS	non-IDS
Idle	CPU	1.80%	1.02%
	Load Average	0.07, 0.08, 0.10	0.03, 0.05, 0.07
	Memory	735/952 MB	736/952 MB
	Swap Memory	633/947 MB	632/947 MB
Being Attacked (DDoS)	CPU	46.10%	42.60%
	Load Average	0.55, 0.36, 0.29	0.48, 0.40, 0.31
	Memory	694/952 MB	696/952 MB

From the results of calculations and performance tests this is distinguished when in a state of silence and when attacked, which of course when attacked using Denial of Service (DoS) because this attack has a significant impact on system performance both on the IDS and the e-voting server itself.

After testing it can be concluded that the IDS system installed on the e-voting web server will slightly affect CPU performance and computing, i.e. by a difference of 0.78% at rest and 3.5% under conditions when attacked, the load average obtained has a difference of 0.02 to 0.07 so that in minutes first, fifth, and fifteenth shows a comparison of performance that is not too significant when this IDS system is installed or not.

5. CONCLUSION

Based on the results of research conducted by researchers regarding the Implementation of Intrusion Detection System (IDS) on E-Voting Servers with Signature Based Rule on Snort, the conclusion is that IDS installed on the E-Voting web server will make it easier for network administrators to monitor attacks carried out by monitoring the condition of the server by using a web monitor that is installed on the IDS server itself. Web monitor installed will immediately retrieve logs from snort in real-time, so the data captured will be in accordance with what snort retrieves when an attack occurs. There are several ways to process log data that is displayed by a web monitor, using Search Processing Language (SPL) so that the data obtained by the administrator will be as desired. The IDS system that is running will be able to function again if a system is reset or restarted, because the IDS system used is using a boot start. The E-Voting Server that uses the IDS system has a comparative performance that is almost the same as that that does not have an IDS System, therefore the IDS system installed will not affect the performance of the web server either when it is attacked or not, because basically this IDS system has a server separately so that it will not interfere with the activities contained in the web server. The number of warning logs received by the IDS system will vary depending on what attacks are received by the E-Voting web server, the most warnings are dominated by Denial of Service (DoS) attacks because this attack sends very many data

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2774-7891

packets so that it is good both the IDS system and e-voting server experienced a very significant decrease in performance.

Researchers also provide suggestions to researchers who are out there for those who are interested in continuing this research, so there are several recommendations directly given by the researchers themselves, the Snort IDS system with web monitoring implemented on this e-voting website can also be configured as an Intrusion Prevention System (IPS) as an additional layer of protection that will automatically prevent attacks. The signature rules contained in the IDS system that researchers have are still lacking, namely the lack of rules contained in this system making it quite large and complex attacks (especially XSS and SQLi attacks) difficult to identify. Therefore, adding a rule signature for more complex attacks can deal with this problem. Adding features that can make e-voting web server administrators more aware of attacks that have been obtained by web monitors such as notifications or others.

6. AUTHORS' NOTE

The authors declare that there is no conflict of interest regarding the publication of this article. The authors confirmed that the paper was free of plagiarism.

7. REFERENCES

Abhipraya, F. A., Yogar, B. N. A., and Prasetyo, S. I. (2023). Toward Effective Electoral Affairs: The Implementation of E-Voting in the Village Chief Executive Election 2021. *Indonesian Governance Journal: Kajian Politik-Pemerintahan, 6*(1), 28-36.

Abomhara, M., and Køien, G. M. (2015). Cyber security and the internet of things: Vulnerabilities, threats, intruders, and attacks. *Journal of Cyber Security*, *4*(1), 65-88.

Bhosale, D. A., and Mane, V. M. (2015, October). Comparative study and analysis of network intrusion detection tools. In *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, *2016*, 312-315.

Chanthakoummane, Y., Saiyod, S., Benjamas, N., and Khamphakdee, N. (2016). Improving intrusion detection on snort rules for botnets detection. In *Information Science and Applications (ICISA) 2016,* 765-779.

Chavarro, F. A. C., Homes, C. D. C., and Mora, D. C. F. (2018). Implementation of confidentiality and anonymity as services in an e-voting system for educational institutions. *International Journal of Applied Engineering Research*, *13*(18), 13555-13565.

Dewi, E. K., Harini, D., and Miftachurohmah, N. (2017, February). Snort IDS sebagai tools forensik jaringan universitas Nusantara PGRI Kediri. *Seminar Nasional Inovasi Teknologi*, *1*(1), 397-404.

Goel, A., and Vasistha, A. K. (2017). The implementation and assessment of snort capabilities. *International Journal of Computer Applications*, *167*(13), 15-23.

Gupta, K., Singh, R. R., and Dixit, M. (2017, June). Cross site scripting (XSS) attack detection using intrustion detection system. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018, 199-203.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2723-4088

Jim, L. E., Islam, N., and Gregory, M. A. (2022). Enhanced MANET security using artificial immune system-based danger theory to detect selfish nodes. *Computers and Security*, *113*, 102538.

Khamphakdee, N., Benjamas, N., and Saiyod, S. (2015). Improving intrusion detection system based on snort rules for network probe attacks detection with association rules technique of data mining. *Journal of ICT Research and Applications*, *8*(3), 234-250.

Kizza, J. M., and Kizza, J. M. (2013). Security in wireless networks. *Guide to Computer Network Security*, 387-411.

Kohno, T., Stubblefield, A., Rubin, A. D., and Wallach, D. S. (2004, May). Analysis of an electronic voting system. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, 27-40.

Kumar, V., and Sangwan, O. P. (2012). Signature based intrusion detection system using SNORT. *International Journal of Computer Applications and Information Technology*, *1*(3), 35-41.

Le Jeune, L., Goedeme, T., and Mentens, N. (2021). Machine learning for misuse-based network intrusion detection: overview, unified evaluation, and feature choice comparison framework. *IEEE Access*, *9*, 63995-64015.

Li, S., Dai, Y., and Chen, Y. (2001). Intrusion detection system. *Computer Engineering*, *27*, 7-9.

Mahmoud, T. M., Ali, A. A., and Elshafie, H. M. (2016). A hybrid snort-negative selection network intrusion detection technique. *International Journal of Computer Applications*, *146*(5), 24-31.

Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., and Foozy, C. F. M. (2021). Benchmarking of machine learning for anomaly-based intrusion detection systems in the CICIDS2017 dataset. *IEEE Access*, *9*, 22351-22370.

Moloja, D., and Mpekoa, N. (2017, July). Towards a cloud intrusion detection and prevention system for M-voting in South Africa. In *2017 International Conference on Information Society (i-Society)*, *2018*, 34-39.

Olanrewaju, R. F., Khan, B. U. I., Najeeb, A. R., Zahir, K. N. A. K., and Hussain, S. (2018). Snort-based smart and swift intrusion detection system. *Indian Journal of Science and Technology, 11*(4), 1-9.

Roesch, M. (1999, November). Snort: Lightweight intrusion detection for networks. In *Lisa*, 99(1), 229-238.

Saied, M., Guirguis, S., and Madbouly, M. (2024). Review of artificial intelligence for enhancing intrusion detection in the internet of things. *Engineering Applications of Artificial Intelligence*, *127*, 107231.

Sekhar, M., Tulasi, K., Amulya, V., Teja, D., and Kumar, M. (2015). Implementation of IDS using Snort on bayesian network. *International Journal of Computer Science and Mobile Computing*, *4*(4), 790-795.

Smith, A. D., and Clark, J. S. (2005). Revolutionising the voting process through online strategies. *Online Information Review*, *29*(5), 513-530.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2774-7891

Sobh, T. S. (2006). Wired and wireless intrusion detection system: Classifications, good characteristics, and state-of-the-art. *Computer Standards and Interfaces*, *28*(6), 670-694.

Vuppala, R., and Farik, M. (2016). Intrusion detection and prevention systems-sourcefire snort. *International Journal of Scientific and Technology Research*, *5*(7), 5-8.

Yamamoto, Y., and Yamaguchi, S. (2023). Defense mechanism to generate ips rules from honeypot logs and its application to log4shell attack and its variants. *Electronics*, *12*(14), 3177.

Zissis, D., and Lekkas, D. (2011). Securing e-Government and e-Voting with an open cloud computing architecture. *Government Information Quarterly*, *28*(2), 239-251.

DOI: http://dx.doi.org/10.17509/xxxx.xxx
p- ISSN xxxx-xxxx e- ISSN 2723-4088