Mobile and Forensics
Vol. 6, No. 1, March 2024, pp. 1-9
ISSN: 2714-6685 (online), 2656-6257 (print)
DOI: http://dx.doi.org/10.12928/mf.v6i1.10317

NETWORK SECURITY MONITORING SYSTEM VIA ANDROID MOBILE APPLICATION WITH IDS

1,*Hamas Ardyan Prasetyo, 2Nuril Anwar
1,2Departement of Informatics, Universitas Ahmad Dahlan Yogyakarta, Indonesia
1,*hamasadiyanprasetyo@gmail.com, 2 nuril.anwar@tif.uad.ac.id
*correspondence email

Abstract

Network security is an important factor in securing data on a server, so a server needs to be kept safe from things that could threaten the validity and integrity of stored data. One way that can be used to detect threats on a server is implementing an Intrusion detection system on the server. A literature study conducted on research that implemented intrusion detection systems, found that there was a lack of intrusion detection system research that could detect one type of network security attack with a variety of attack variables and it was also found in research that had successfully implemented an intrusion detection system to detect network security attacks but still incorrectly identifying the type of attack. This research uses the Snort intrusion detection system method with an experimental model of an attack detection system and an Android application which is applied to monitor the statistics of attacks detected on the Xyz University network. The research results showed that the rules created on the IDS can detect network security attacks, especially DoS/DDoS and PortScan attacks. Then an IDS was created that can send application alert notifications and SMS with a response time that is quite responsive based on the NIST Cybersecurity reference with an average of 22 seconds for DoS/DDoS attacks and 21 seconds for Port Scanning attacks. For the percentage results from 3 times testing the rule by sending DoS/DDoS attack packets of 309,462 to 1,459,548, getting a high level of accuracy with an average of 92.1% on first test, 91.7% on the second test and 91.5% on the third test. In the results of testing the PortScan rule by sending 1,001 to 10,564 attack packets, a high level of accuracy was obtained with an average result of 92.2% in the first test, 94.2% in the second test and 93.4% in the third test.

Keywords: Android App, Intrusion Detection System (IDS), Network Security, Monitoring System, Snort

INTRODUCTION

A computer network is a group of two or more computers that are connected and interconnected, so that they can exchange information and communicate between one device and another network device [1], [2].

A computer network must be able to provide a sense of security regarding access made by a user, this is related to the three pillars of network security, namely CIA, which stands for Confidentiality [3], namely Confidentiality, Integrity, namely Integrity, and Availability, namely Availability, by implementing the basic pillars, it has become a reference for guarantees the security of information or personal data from illegal access by unauthorized users such as intruders (Attackers). Network security is one of the factors in a system to ensure the validity and integrity of data is maintained as well as the availability of services for users. The network security system must be protected from various types of unauthorized network access, especially from external networks (the Internet) [4].

Article History: *Received Maret 9, 2018; Revised Maret 13, 2024; Accepted Maret 30, 2024*

The main goal of network security is to anticipate threats that can be physical threats or non-physical threats that can disrupt network traffic, performance and configuration of devices on the network. One method for maintaining network security is to use an intrusion detection system (IDS) which was developed by Martin Roesch [5].

IDS can be used to monitor network traffic, especially traffic entering from the outside network to the local network. Network monitoring can be the first action for an administrator to find out what types of attacks are being launched by an intruder if an attack incident occurs, so that the administrator can immediately take quick and appropriate preventive action to overcome the attack [6]. A literature study conducted in several journals with research on network security attack detection systems [7], found that there is still a lack of research that focuses on detecting one type of network security attack with various variations in its variables. and it was also found that the results of the network security attack detection system configuration created showed a false positive warning message when a network security attack incident occurred, meaning the system was able to detect attacks but incorrectly identified the type of network security attack that occurred [8].

Based on the problems above, the title that can be raised is "Design of a Network Security Monitoring System through an Android-Based Mobile Application Using the Intrusion Detection System Method" [9], [10]. This research focuses on developing a system that can detect accurately according to the type of attack especially DoS/DDoS & Port Scanning attacks and will send a warning message to the network administrator in real time if an attack incident occurs [11], [12]. One way is to create a system using the Snort intrusion detection system (IDS) in collaboration with a mobile application [13], [14]. Android-based to display attack information in statistical form. So that when an attack incident occurs by an intruder (attacker), the administrator will receive a warning notification from the managed network and can see statistics on attacks that occur on the application. Armed with this attack information notification, administrators can: determine appropriate and rapid preventive action.

METHODS

The method used in this research is an intrusion detection system with experimental rules to be able to detect network security attacks. Experimental rules on IDS were carried out to obtain IDS rules that could detect network security attacks with a high level of accuracy. The research stages used are described in Fig. 1

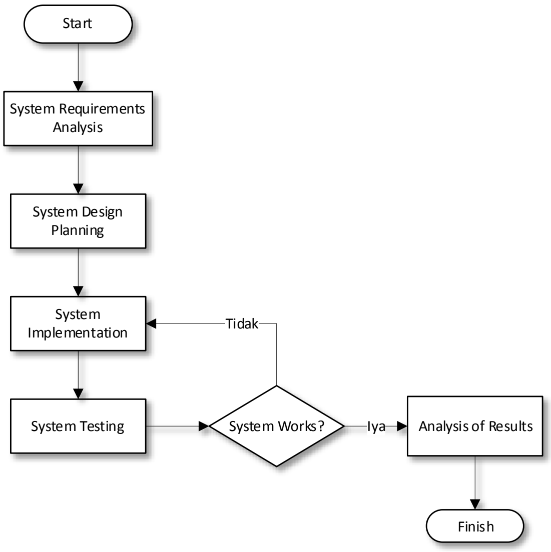


Fig. 1. Research stages

As mentioned before and depicted in Fig. 1:

1. **System Requirements Analysis**: At this stage, an analysis of system requirements is carried out which is determined based on data from observations, literature studies and interviews. The analysis stage becomes a reference in the system design and development process.
2. **System Design Planning**: At this stage, the system design process is carried out. The following system design plans are divided into:
   a. Network Topology Design: In this research, there is an intrusion detection system installed on a server or device in the network. The network topology design in this research becomes a reference in developing and testing the finished system.
   b. Application Design: The application design process includes designing business processes, activity diagrams and application interfaces.
   c. Database Design: Includes designing the database used by the application to store data.
   d. Web API Design: Web API design to send data from the intrusion detection system on the server to the application installed on the mobile device.
3. **System Implementation:** At this stage, the results from the previous stage are implemented into a system, starting from configuring the intrusion detection system, application development and web API (application programming interface) development.
4. **System Testing:** At the system testing stage, several testing processes are carried out which are explained as follows:
   a. Testing attack detection rules on IDS against various attack variables
   b. Testing rules against high traffic loads due to user requests.
   c. Black box testing on the application system.
5. **Analysis of Results:** At the results analysis stage, analysis is carried out from the results of the research that has been carried out, to determine the percentage level of success and performance of the system that has been created.

Hardware and Software Used

1. **Hardware**

Table 1. Specifications Of The Device Used

	Server Device	IDS Device	Attacker Device	Traffic Generate Device	Admin Device
Processor	Virtual Machine Vbox	Virtual Machine Vbox	Virtual Machine Vbox	Virtual Machine Vbox	Octa-core (4x2.0 GHz Kryo 260 Gold & 4x1.8 GHz Kryo 260 Silver)
Memory	1 GB DDR3	4 GB DDR3	4 GB DDR3	1 GB DDR3	4 GB
Storage	20 GB SSD	25 GB SSD	80 GB SSD	20 GB SSD	64 GB
Operating System	Ubuntu Server 20.04.2 64 Bit	Linux Mint 20.3	Kali Linux 2022.1 64 Bit	Ubuntu CLI 20.04.2 32 Bit	Android 11 Red Velvet Cake

2. **Software:** The following software or tools will be used, Android Studio 2020.3.1.24, Figma 2023, WebSMS Gateway, Apache Benchmark, Xampp Win 7.4.2, Visual Studio Code 1.8.0, Hping3, Wireshark, IDS Snort 2.9.7.0, Barnyard2 2-1.14.

RESULT AND DISCUSSIONS

A. System Requirements Analysis

Table 2 . System Functional Requirements

No	Code	Description
1	SKPL-F1	The system must be able to display attack statistical information
2	SKPL-F2	The system must be able to display a log of detected attacks
3	SKPL-F3	The system must be able to display attack alert notifications
4	SKPL-F4	The system must be able to update contact data and network admin names
5	SKPL-F5	The system must be able to display attack alert notifications via SMS

Table 3. System Non-Functional Requirements

No	Code	Description
1	SKPL-NF1	The system must be able to display attack alert notifications less than 5 minutes after the attack incident
2	SKPL-NF2	The system must be usable by Android users

B. Planning

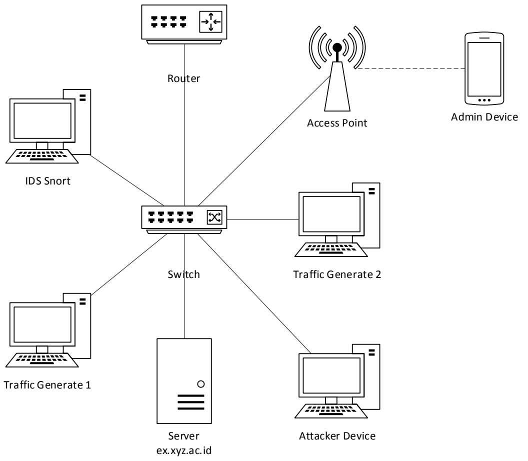


Fig. 1. *IDS Network Topology*

Fig. 2. is the topology used in the application of this research, the system is applied to the XYZ local network.

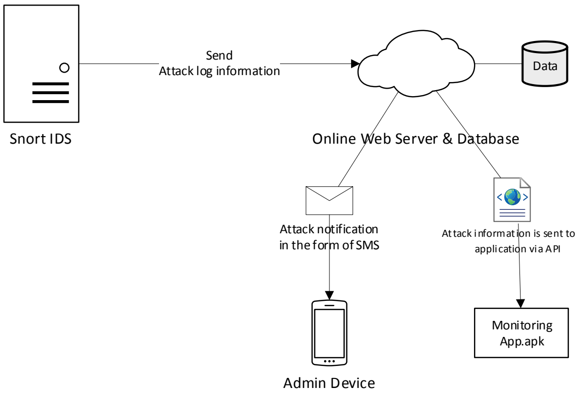


Fig. 2. *Network Security Monitoring System Business Process*

C. Implementation & Testing

1. API Web

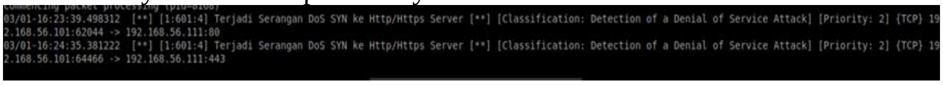
2. Mobile Application
3. IDS snort rule

1) DoS Attack TCP Flags SYN to Web Server Http and Https

```
alert tcp   $EXTERNAL_NET any -> $HOME_NET 80,443  (msg: "Terjadi
Serangan DoS SYN ke Http/Https Server";   classtype: denial-of-service;
flags: S; threshold: type threshold, track by_dst, count 10000, seconds
20; dsize: >0; sid: 601; rev: 4;)
```

Testing

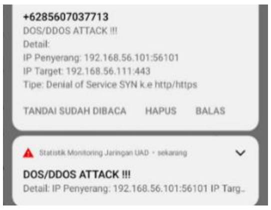
The attack was carried out using the hping3 tool with the command "hping3 --flood ex.xyz.ac.id -d 1 -p 443 —syn"



```
Commencing packet processing (pid=8168)
03/01-16:23:39.498312  [**] [1:601:4] Terjadi Serangan DoS SYN ke Http/Https Server [**] [Classification: Detection of a Denial of Service Attack] [Priority: 2] {TCP} 19
2.168.56.101:62044 -> 192.168.56.111:80
03/01-16:24:35.381222  [**] [1:601:4] Terjadi Serangan DoS SYN ke Http/Https Server [**] [Classification: Detection of a Denial of Service Attack] [Priority: 2] {TCP} 19
2.168.56.101:64466 -> 192.168.56.111:443
```

__Fig. 3.__ DoS TCP SYN successfully detected by IDS

Notification Alert



+6285607037713
DOS/DDOS ATTACK !!!
Detail:
IP Penyerang: 192.168.56.101:56101
IP Target: 192.168.56.111:443
Tipe: Denial of Service SYN k.e http/https

DOS/DDOS ATTACK !!!
Detail: IP Penyerang: 192.168.56.101:56101 IP Targ..

__Fig. 4.__ App and SMS notifications from DoS TCP SYN

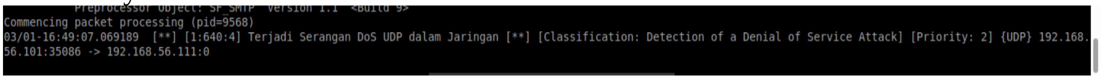
2) UDP DoS Attacks on the Network

Snort rule for UDP DoS detection:

```
alert udp   $EXTERNAL_NET any -> $HOME_NET any  (msg: "Terjadi Serangan DoS
UDP dalam Jaringan"; classtype: denial-of-service; dsize: >0; threshold: type
threshold, track by_dst, count 100000, seconds 20; sid: 640; rev: 4;)
```

Testing

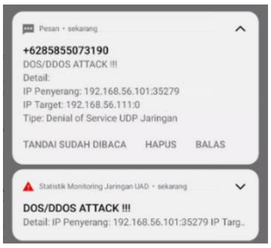
The attack was carried out using the hping3 tool with the command "hping3 --flood ex.xyz.ac.id -d 1 -2 "



```
        Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
Commencing packet processing (pid=9568)
03/01-16:49:07.069189  [**] [1:640:4] Terjadi Serangan DoS UDP dalam Jaringan [**] [Classification: Detection of a Denial of Service Attack] [Priority: 2] {UDP} 192.168.
56.101:35086 -> 192.168.56.111:0
```

__Fig. 5.__ DoS UDP successfully detected by IDS

Notification Alert



+6285855073190
DOS/DDOS ATTACK !!!
Detail:
IP Penyerang: 192.168.56.101:35279
IP Target: 192.168.56.111:0
Tipe: Denial of Service UDP Jaringan

DOS/DDOS ATTACK !!!
Detail: IP Penyerang: 192.168.56.101:35279 IP Targ..

__Fig. 6.__ App and SMS notifications from DoS UDP

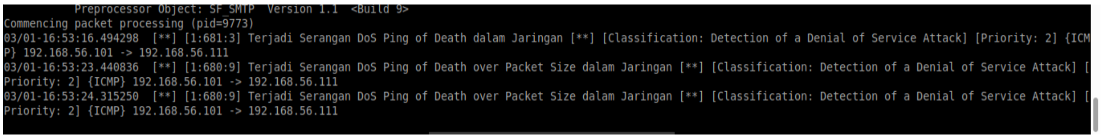
3) ICMP DoS Attack on the Network

Snort rules for small and large size ICMP DoS detection:

```
alert icmp  $EXTERNAL_NET any -> $HOME_NET any  (msg: "Terjadi Serangan DoS
Ping of Death over Packet Size dalam Jaringan";     classtype: denial-of-
service; threshold: type threshold, track by_dst, count 10000, seconds 20;
itype: 8; dsize:100<>1000; sid: 680; rev: 9;)
alert icmp  $EXTERNAL_NET any -> $HOME_NET any  (msg: "Terjadi Serangan DoS
Ping of Death dalam Jaringan";                      classtype: denial-of-
service; threshold: type threshold, track by_dst, count 10000, seconds 20;
itype: 8;dsize:<100; sid: 681; rev: 3;)
```

Testing

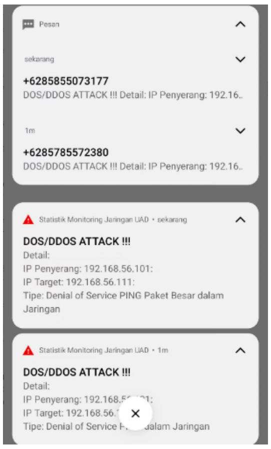
The attack was carried out using the hping3 tool with the command " hping3 --flood ex.xyz.ac.id -d 1 -1" and “hping3 —-flood ex.xyz.ac.id -d 200 -1”



```
        Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
Commencing packet processing (pid=9773)
03/01-16:53:16.494298  [**] [1:681:3] Terjadi Serangan DoS Ping of Death dalam Jaringan [**] [Classification: Detection of a Denial of Service Attack] [Priority: 2] {ICM
P} 192.168.56.101 -> 192.168.56.111
03/01-16:53:23.440836  [**] [1:680:9] Terjadi Serangan DoS Ping of Death over Packet Size dalam Jaringan [**] [Classification: Detection of a Denial of Service Attack] [
Priority: 2] {ICMP} 192.168.56.101 -> 192.168.56.111
03/01-16:53:24.315250  [**] [1:680:9] Terjadi Serangan DoS Ping of Death over Packet Size dalam Jaringan [**] [Classification: Detection of a Denial of Service Attack] [
Priority: 2] {ICMP} 192.168.56.101 -> 192.168.56.111
```

Fig. 7. *DoS ICMP small and large package sizes successfully detected by IDS*

Notification Alert



Pesan

sekarang

+6285855073177
DOS/DDOS ATTACK !!! Detail: IP Penyerang: 192.16..

1m

+6285785572380
DOS/DDOS ATTACK !!! Detail: IP Penyerang: 192.16..

Statistik Monitoring Jaringan UAD • sekarang

DOS/DDOS ATTACK !!!
Detail:
IP Penyerang: 192.168.56.101:
IP Target: 192.168.56.111:
Tipe: Denial of Service PING Paket Besar dalam Jaringan

Statistik Monitoring Jaringan UAD • 1m

DOS/DDOS ATTACK !!!
Detail:
IP Penyerang: 192.168.5   1:
IP Target: 192.168.56.1
Tipe: Denial of Service P   dalam Jaringan

Fig. 8. *App and SMS notifications from DoS ICMP small and large package sizes*

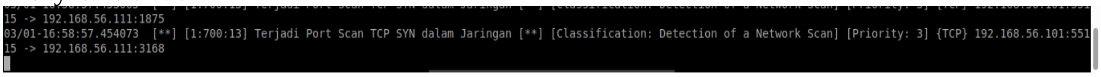
4) Port Scanning TCP Flags SYN to Network

Snort rule for TCP SYN PortScanning detection:

```
alert tcp   $EXTERNAL_NET any -> $HOME_NET any  (msg: "Terjadi Port Scan TCP
SYN dalam Jaringan"; flags: S; classtype: network-scan; dsize: 0; threshold:
type threshold, track by_dst, count 5, seconds 20; sid: 700; rev: 13; flow:
not_established, from_client; ack: 0; window: 1024;)
```

Testing

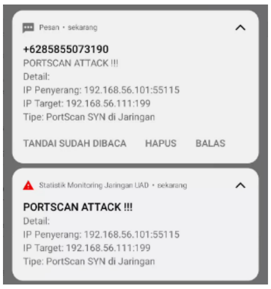
The attack was carried out using the hping3 tool with the command "nmap ex.xyz.ac.id -sS "



```
15 -> 192.168.56.111:1875
03/01-16:58:57.454073  [**] [1:700:13] Terjadi Port Scan TCP SYN dalam Jaringan [**] [Classification: Detection of a Network Scan] [Priority: 3] {TCP} 192.168.56.101:551
15 -> 192.168.56.111:3168
```

Fig. 9*. PortScanning TCP SYN successfully detected by IDS*

Notification Alert



Pesan • sekarang

+6285855073190
PORTSCAN ATTACK !!!
Detail:
IP Penyerang: 192.168.56.101:55115
IP Target: 192.168.56.111:199
Tipe: PortScan SYN di Jaringan

Statistik Monitoring Jaringan UAD • sekarang

PORTSCAN ATTACK !!!
Detail:
IP Penyerang: 192.168.56.101:55115
IP Target: 192.168.56.111:199
Tipe: PortScan SYN di Jaringan

Fig. 10. *App and SMS notifications from PortScanning TCP SYN*

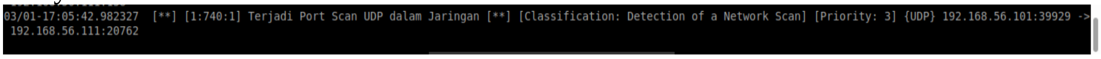
5) Port Scanning UDP to Network

Snort rule for UDP PortScanning detection:

```
alert udp   $EXTERNAL_NET any -> $HOME_NET any  (msg: "Terjadi Port Scan UDP
dalam Jaringan"; classtype: network-scan; dsize: 0; threshold: type threshold,
track by_dst, count 5, seconds 20; sid: 740; rev: 1;)
```

Testing

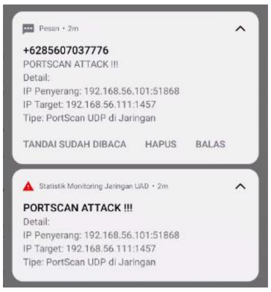
The attack was carried out using the hping3 tool with the command "nmap ex.xyz.ac.id -sU "



```
03/01-17:05:42.982327  [**] [1:740:1] Terjadi Port Scan UDP dalam Jaringan [**] [Classification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.56.101:39929 ->
 192.168.56.111:20762
```

Fig. 11. PortScanning UDP successfully detected by IDS

Notification Alert



Pesan • 2m

+6285607037776
PORTSCAN ATTACK !!!
Detail:
IP Penyerang: 192.168.56.101:51868
IP Target: 192.168.56.111:1457
Tipe: PortScan UDP di Jaringan

TANDAI SUDAH DIBACA HAPUS BALAS

Statistik Monitoring Jaringan UAD • 2m

PORTSCAN ATTACK !!!
Detail:
IP Penyerang: 192.168.56.101:51868
IP Target: 192.168.56.111:1457
Tipe: PortScan UDP di Jaringan

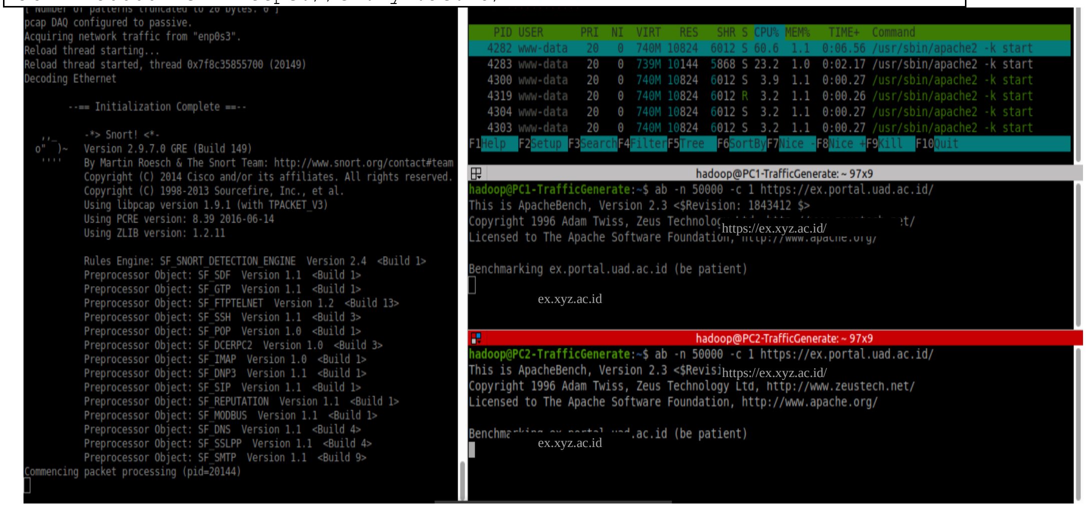
Fig. 12. *App and SMS notifications from PortScanning UDP*

4. Testing

High Traffic Load Testing on IDS

Command used to generate traffic to the server:

```
ab -n 50000 -c 1 https://ex.xyz.ac.id/
```



```
[ Number of patterns truncated to 20 bytes: 0 ]
pcap DAQ configured to passive.
Acquiring network traffic from "enp0s3".
Reload thread starting...
Reload thread started, thread 0x7f8c35855700 (20149)
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 2.4  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
Commencing packet processing (pid=20144)
```

```
  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 4282 www-data   20   0  740M 10824  6012 S 60.6  1.1  0:06.56 /usr/sbin/apache2 -k start
 4283 www-data   20   0  739M 10144  5868 S 23.2  1.0  0:02.17 /usr/sbin/apache2 -k start
 4300 www-data   20   0  740M 10824  6012 S  3.9  1.1  0:00.27 /usr/sbin/apache2 -k start
 4319 www-data   20   0  740M 10824  6012 R  3.2  1.1  0:00.26 /usr/sbin/apache2 -k start
 4304 www-data   20   0  740M 10824  6012 S  3.2  1.1  0:00.27 /usr/sbin/apache2 -k start
 4303 www-data   20   0  740M 10824  6012 S  3.2  1.1  0:00.27 /usr/sbin/apache2 -k start
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

hadoop@PC1-TrafficGenerate: ~ 97x9

```
hadoop@PC1-TrafficGenerate:~$ ab -n 50000 -c 1 https://ex.portal.uad.ac.id/
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technolog https://ex.xyz.ac.id/ et/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ex.portal.uad.ac.id (be patient)

        ex.xyz.ac.id
```

hadoop@PC2-TrafficGenerate: ~ 97x9

```
hadoop@PC2-TrafficGenerate:~$ ab -n 50000 -c 1 https://ex.portal.uad.ac.id/
This is ApacheBench, Version 2.3 <$Revisi https://ex.xyz.ac.id/
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchma        .ac.id (be patient)
        ex.xyz.ac.id
```

Fig. 13. *Test high traffic loads on the server*

5. Analysis Of Results

Percentage of IDS Rules in Detecting DoS/DDoS Attacks and PortScan

Table 1. Percentage of IDS Rules in Detecting Attacks

Rule Type	**Number of Attacks (Packets)**			**Detected Attacks (Packets)**			**Percentage**		
	Test 1	***Test 2***	***Test 3***	***Test 1***	***Test 2***	***Test 3***	***Test 1***	***Test 2***	***Test 3***
DoS TCP SYN	500457	1100289	1459548	318636	717801	1026039	63,7%	65,2%	70,3%
DoS TCP ACK	371396	615663	1090752	*308833*	539755	1008970	83,2%	87,7%	92,5%
DoS TCP FIN	522350	793255	1047292	522346	760351	1041621	100,0%	95,9%	99,5%
DoS TCP RST	555946	712803	1057601	547677	712797	1011819	98,5%	100,0%	95,7%
DoS TCP PUSH	327021	722313	1231834	327019	722308	1141917	100,0%	100,0%	92,7%
DoS TCP tanpa *Flags*	360982	736253	1135910	306897	731485	1025052	85,0%	99,4%	90,2%
DoS UDP	309462	736183	1014942	309460	736178	1014935	100,0%	100,0%	100,0%
DoS ICMP Small Packet Size	322449	714279	1026937	322448	714272	1026942	100,0%	100,0%	100,0%

DoS ICMP Over Packet Size	322192	722260	1025934	322191	722254	1025925	100,0%	100,0%	100,0%
PortScan TCP SYN	1001	2002	3003	900	1800	2700	89,9%	89,9%	89,9%
PortScan TCP ACK	1001	2002	3003	900	1801	2700	89,9%	90,0%	89,9%
PortScan TCP FIN	1004	2008	3012	903	1806	2710	89,9%	89,9%	90,0%
PortScan UDP	1148	4813	10564	1135	4669	10159	98,9%	97,0%	96,2%

Response Time in Displaying alerts in the Application

Table 2. IDS Alert Notification Response Time

Attack Type	**Start Attack (Time)**	**Detected by Snort IDS (*Time*)**	**Application Alert Notification (Time)**	**Alert Response Time (Time)**
DoS TCP SYN	16:03:43	16:03:50	16:04:15	33 second
DoS TCP ACK	16:05:57	16:06:01	16:06:21	24 second
DoS TCP FIN	16:06:59	16:07:03	16:07:27	28 second
DoS TCP RST	16:08:11	16:08:15	16:08:33	22 second
DoS TCP PUSH	16:09:23	16:09:27	16:09:40	17 second
DoS TCP without Flags	16:10:55	16:11:00	16:11:18	23 second
DoS UDP	16:11:49	16:11:53	16:12:07	18 second
DoS ICMP Small Packet Size	16:12:39	16:12:44	16:12:55	16 second
DoS ICMP Over Packet Size	16:13:28	16:13:33	16:13:45	17 second
PortScan TCP SYN	16:14:50	16:14:54	16:15:07	17 second
PortScan TCP ACK	16:48:04	14:48:05	16:48:37	33 second
PortScan TCP FIN	16:16:29	16:16:35	16:16:48	19 second
PortScan UDP	16:17:13	16:17:18	16:17:31	18 second

CONCLUSIONS

In conclusion, this research into network security monitoring system design demonstrates several key findings. Firstly, leveraging Snort's rules within an IDS implementation effectively detects various attacks, particularly DoS/DDoS and PortScan attacks. Secondly, the developed IDS system exhibits commendable responsiveness, aligning with NIST Cybersecurity guidelines, with an average response time of 22 seconds for DoS attacks and 21 seconds for PortScan attacks. Thirdly, rigorous rule testing confirms the system's robustness in detecting PortScan attacks with an average accuracy of 92.1% to 91.5% across multiple tests, and DoS/DDoS attacks with an average accuracy ranging from 92.2% to 94.2%. These results underscore the system's capability to detect and respond to network threats swiftly and accurately, validating its effectiveness in enhancing network security.

REFERENCES

[1] M. Ardhiansyah, S. Noris, and R. Andrianto, Jaringan Komputer Jaringan Komputer, 1st ed. Banten: Unpam Press, 2020. [Online]. Available: http://eprints.unpam.ac.id/8869/1/TPL0183_JARINGAN KOMPUTER-ok.pdf

[2] I. Sofana, "No Title," Membangun Jaringan Komputer: Mudah membuat Jaringan Komputer (Wire & Wireless) untuk pengguna Windows dan Linux, 2013.

[3] M. Wills, "Information Security Fundamentals," in (ISC)2 SSCP Systems Security Certified Practitioner Official Study Guide, Wiley, 2019, pp. 25–49. doi: 10.1002/9781119547921.ch2.

[4] A. Sadiqui, "Fundamentals of Network Security," in Computer Network Security, Wiley, 2020, pp. 1–14. doi: 10.1002/9781119706762.ch1.

[5] R. Widodo and I. Riadi, "Sistem Deteksi Penyusup Pada Jaringan Komputer Menggunakan Teknik Host Based Intrusion Detection System," Buletin Ilmiah Sarjana Teknik Elektro, 2019.

[6] J. Wang and Z. A. Kissel, "Intrusion Detections," in Introduction to Network Security: Theory and Practice, Wiley, 2015, pp. 309–336. doi: 10.1002/9781119113102.ch9.

[7] H. Yanto and F. Hadi, "Intruder Detection Monitoring System in Computer Networks Using Snort Based Sms Alert (Sistem Monitoring Deteksi Penyusup Dalam Jaringan Komputer Menggunakan Snort Berbasis Sms Alert)," Jurnal KomtekInfo, vol. 7, no. 2, pp. 159–170, 2020, doi: 10.35134/komtekinfo.v7i2.76.

[8] N. Christianto and W. Sulistyo, "Model Pemantauan Keamanan Jaringan Melalui Aplikasi Telegram Dengan Snort," Jurnal Teknik Informatika dan Sistem Informasi, vol. 7, pp. 2443–2229, 2021, doi: 10.28932/jutisi.v7i1.4088.

[9] J. V. M. Edy Irwansyah, Pengantar teknologi informasi, 1st ed. Yogyakarta, 2014.

[10] M. Irsan, T. Jl, H. Hadari, and N. Pontianak, "Rancang Bangun Aplikasi Mobile Notifikasi Berbasis Android Untuk Mendukung Kinerja Di Instansi Pemerintahan."

[11] M. Dooley and T. Rooney, "Service Denial Attacks," in DNS Security Management, 1st ed., Wiley, 2017, pp. 139–141. doi: 10.1002/9781119328292.ch7.

[12] V. Trola, "Manual Network Exploration," in Hunting Cyber Criminals, 1st ed., Wiley, 2020, pp. 45–65. doi: 10.1002/9781119541004.ch3.

[13] W. W. Purba and R. Efendi, "Perancangan dan analisis sistem keamanan jaringan komputer menggunakan SNORT," AITI: Jurnal Teknologi Informasi, vol. 17, no. Agustus, pp. 143–158, 2020.

[14] A. F. Mutaqin, "Rancang Bangung Sistem Monitoring Jaringan Prodi Teknik Informatika Melalui SMS Alert dengan Snort," Jurnal Sistem dan Teknologi Informasi (JUSTIN), vol. 1, No. 1, 2016.