Performance of Kismet Wireless Intrusion Detection System on Raspberry Pi

Gede Arna Jude Saskara[1], I Made Edy Listarta[2], Gede Saindra Santyadiputra[3], Putu Bayu Megawanta, Putu Agus Wahyu Adi Perdana Giri

{jude.saskara@undiksha.ac.id[1], listarta@undiksha.ac.id[2], gsaindras@undiksha.ac.id[3]}

Department of Information System[1,2], Department of Informatics Education[3], Faculty of Engineering and Vocational, Universitas Pendidikan Ganesha

Abstract. Attacks on wireless networks can disconnect and slow down the network by flooding the network with junk packets or Dos and scanning. To solve these problems, a system that is able to monitor and detect security problems is needed. The system that can detect and monitor security problems is known as the Intrusion Detection System. One of the Intrusion Detection System software that is widely used is Kismet. In this research, the Kismet Intrusion Detection System software used to secure wireless networks was installed on a Raspberry Pi to measure the performance of the Kismet Intrusion Detection System. The method used in this study was a literature review that proceeded with a system design that included designing the topology and tests. The next step was implementing the system, followed by testing the performance of the Intrusion Detection System so that conclusions can be drawn and made into a report. Based on research on Intrusion Detection System Performance using Kismet software installed on the Raspberry Pi, it was concluded that Kismet installed on the Raspberry Pi could detect 10 Denial of Service attacks from 10 attacks with an average detection rate of attacks sent until detected by Kismet was 3.42 seconds. Therefore, it can be said that the performance of the Kismet intrusion detection system installed on the Raspberry Pi was accurate and could detect attacks quickly.

Keywords: Wireless, Intrusion Detection System, Kismet, Raspberry Pi

1 Introduction

Currently, every home, restaurant, office, cafe, school, campus, or public place has its own internet network [1]. Wireless network is the easiest network to connect to the internet. A device can be connected to wireless in just a matter of minutes [2]. In addition, it does not use cables so that the device from the user can directly be connected to the internet network, as long as it is still within the coverage area of the wireless network. Wi-Fi is a common term that refers to Wireless Local Area Network (WLAN) systems. WLAN uses the IEEE 802.11 communication standard. IEEE 802.11 was first introduced in 1999 and developed for home and office devices for WLAN connectivity. When it was first developed, the maximum data transfer speed was 2 Mbps, which then continued to grow, and now the data transfer reaches the speed of 540Mbps. To maintain wireless network security, network managers usually add authentication

IConVET 2021, November 27, Singaraja, Indonesia

DOI 10.4108/eai.27-11-2021.2315535

configurations to the wireless devices used. Several types of authentications are available, but the most commonly used are Wired Equivalent Privacy (WEP) and Wi-Fi Protection Access (WPA). There are two versions of WPA, namely WPA and WPA2, which will be encrypted to secure authentication from Wi-Fi [3].

As technology grows rapidly, software developers begin to develop specific software that can be used to break the authentication of Wi-Fi to connect to the network illegally. If vicious users manage to enter the network, the network is now vulnerable to cyber crimes such as finding server locations or carrying out DDoS attacks. Apart from the threat of DDoS attacks, the wireless networks are at risk of Eavesdropping Attack, Node Capture attack, Sybil attack, Byzantine attack [4]. S and Pavithran, 2015 conducted research and carried out a brute-force attack by using aircrack software to enter a wireless network that has been given authentication [5]. It is then important to have additional security to protect the Wi-Fi network as there is software capable of cracking authentication. On that Wi-Fi network, there may be a server that stores classified data, or it can be used to transmit important data.

To strengthen the Wi-Fi security, the network needs to have additional security. As stated in previous research, adding the captive portal to Wi-Fi network disables attackers to penetrate the network [6]. In addition to the previous attempt, the network also needs to have an additional server used to monitor and provide information if an attack happens. The process of monitoring and providing the information is known as the Intrusion Detection System [7][8]. Various studies related to the Intrusion Detection System have been carried out, such as by Agarwal et al. 2013 they added an Intrusion Detection System on Wi-Fi networks to detect de-authentication attacks [9]. Sobari, 2015 added an intrusion detection system on wireless networks. The intrusion detection system used is Snort that uses the Hierarchical Clustering method so that it can detect new attacks on the network [10]. Nizam et al. made a RaspyAir, a Raspberry Pi that had an Intrusion Detection System installed, which was then tested by carrying out all kinds of attacks that could be carried out over a Wi-Fi network [2]. Wibowo, Triyono, and Sutanta, 2017 tested the security of wireless networks owned by the Yogyakarta Communications and Informatics Service. The attacks carried out were Aircrack, ARP Spoofing, Cracking WPA/WPA2 [11]. Pranata, Kunang, and Saputri, 2019 researched on improving wireless network security with an attack detector based on Kismet DD-WRT. The result was that Kismet software could detect attacks on the network [12]. Haninda and Swari, 2020 used a Raspberry Pi 3 paired with the Snort Intrusion Detection System to secure the Wi-Fi network at the STMIK STIKOM Indonesia Campus [13].

Considering that the WEP or WPA2 equipped wireless networks are still penetrable and are supported by previous studies, it is necessary to study the Performance of Intrusion Detection System using Kismet Software Installed on the Raspberry Pi. The Intrusion Detection System used is Kismet as it supports wireless networks monitoring. It is installed on a Raspberry so that network managers can provide cheap network security compared to buying a server computer. The output of the study is to use the Intrusion Detection System Kismet which in turn is able to detect DoS Attacks. Besides, its performance on a mini-computer, namely raspberry pi, has not been implemented yet.

2 Method and procedures

The present research has been done in consecutive steps, shown in **Figure 1**.

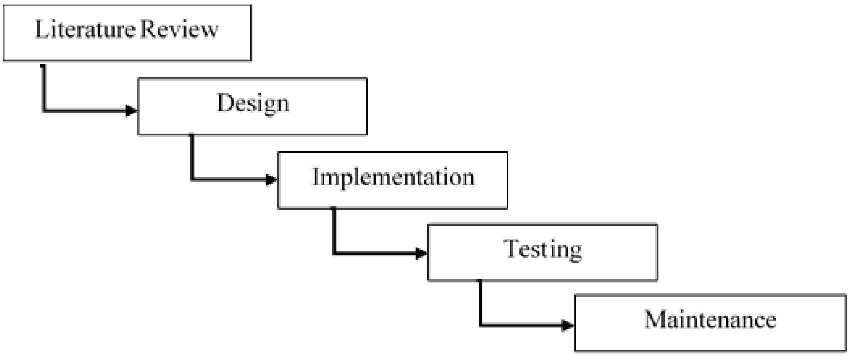


Fig. 1. Research process flowcart.

The explanation of the previous steps is provided below.

2.1 Literature review

The literature review is the stage of conducting a search for various written sources in the form of books, archives, magazines, articles, and journals, or documents relevant to the problems studied. This stage aimed to obtain information about things that previous researchers have done, the object that has been studied, the techniques applied, the results and problems encountered in conducting research, and the differences between the research problem and the problem that the previous researcher has solved. The information obtained from this literature review is used as a reference to strengthen the existing arguments. This research reviewed literature related to Intrusion Detection System, Wireless Network, and Kismet.

2.2 Design

The next step was to design the network topology that would be used. The simulation scenario used an Access point that has WPA2/PSK Authentication connected to a switch, an Intrusion Detection System, and the internet. The configuration used on the Raspberry Pi was the Intrusion Detection System Application, namely Kismet, in which it can provide notifications if an attempt is made to penetrate the network by DDoS attack.

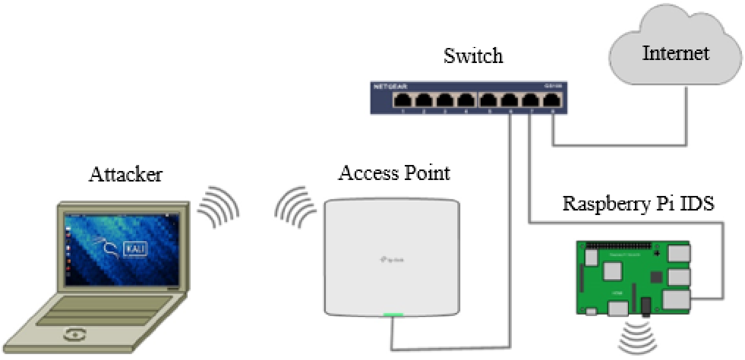


Fig. 2. Network topology.

2.3 Implementation

At this stage, the Raspberry Pi and the Intrusion Detection System application were configured to provide information when an attack occurred.

2.4 Testing

The testing stage was done by conducting Dos attack on the wireless SSID was also carried out ten times and the amount of time spent by the Kismet Intrusion Detection System to respond to an attack was also recorded and calculated.

2.5 Maintenance

This stage was the stage of repairing and maintaining the system so that it could run well. The results of the testing stage were used as the basis for the maintenance.

3 Result and discussion

This section explains the results of the implementation and discussion of assembling prototypes.

3.1 Implementation

The research started by making a topology design then implemented the topology. The implementation was based on the design that had been made. There was a switch connected to the internet, access point and raspberry pi. Raspberry Pi was connected to the switch and the internet was equipped with Kismet application. The access point was set specifically to broaden the internet network using a password, namely WPA2/PSK. An overview of the implementation design system can be seen in **Figure 3**.

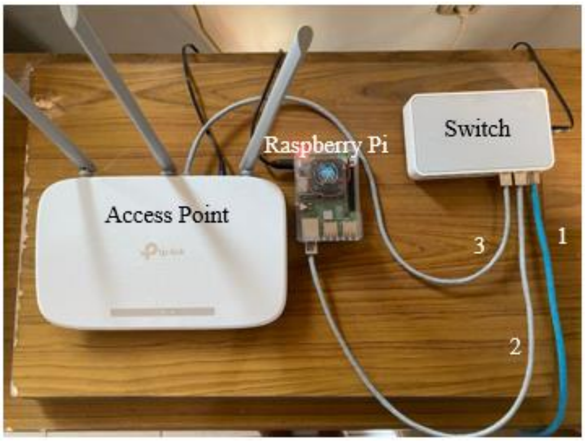
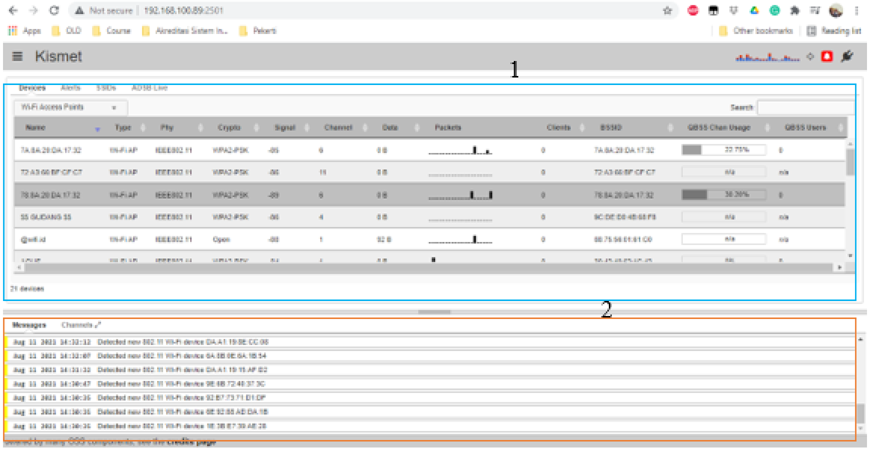


Fig. 3. System design implementation.

Figure 3 shows three UTP cables connected to the switch. The UTP cable numbered 1 was connected to the internet, then the UTP cable numbered 2 was connected to the Raspberry Pi paired with Kismet to detect attacks experienced by the Access Point. Finally, the UTP cable numbered 3 was connected to the Access Point which would transmit a Wi-Fi signal is connected to the internet and already has a WPA2-AES security password and WPA Authentication: PSK and with the name SSID Kemuning4_A.

After every device was connected and in accordance with the designed topology, the Operating System on the Raspberry Pi was installed. In this present study, the operating system used on the Raspberry Pi was Kali Linux. It is because Kali Linux has installed the Kismet Intrusion Detection System application. To run Kismet on Kali Linux, the root user must be accessed first, then run the command Kismet -c wlan0. After that, the Kismet intrusion detection system could be accessed via a browser with the address http://localhost:2501/ or through the IP address of the Raspberry Pi.



Kismet

Wi-Fi Access Points

Name	Type	Phy	Crypto	Signal	Channel	Data	Packets	Clients	BSSID	QBSS Chan Usage	QBSS Users
7A:8A:20:DA:17:32	Wi-Fi AP	IEEE802.11	WPA2-PSK	-85	6	0 B		0	7A:8A:20:DA:17:32	22.75%	0
72:A3:60:BF:CF:C7	Wi-Fi AP	IEEE802.11	WPA2-PSK	-86	11	0 B		0	72:A3:60:BF:CF:C7	n/a	n/a
78:8A:20:DA:17:32	Wi-Fi AP	IEEE802.11	WPA2-PSK	-89	6	0 B		0	78:8A:20:DA:17:32	38.39%	0
SS GUDANG SS	Wi-Fi AP	IEEE802.11	WPA2-PSK	-86	4	0 B		0	8C:DE:E0:4B:68:F8	n/a	n/a
@wifi.id	Wi-Fi AP	IEEE802.11	Open	-88	1	52 B		0	88:75:56:01:61:C0	n/a	n/a

21 devices

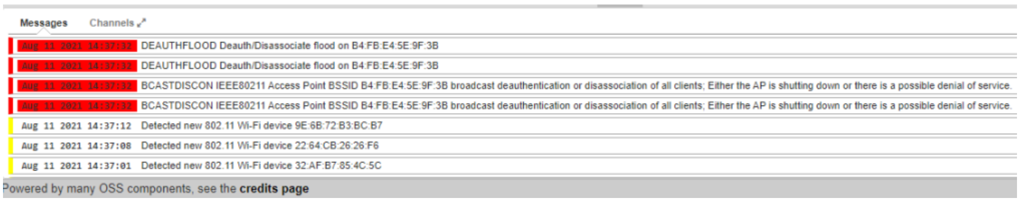
Messages Channels

```
Aug 11 2021 14:32:12  Detected new 802.11 Wi-Fi device DA:A1:19:8E:CC:08
Aug 11 2021 14:32:07  Detected new 802.11 Wi-Fi device 6A:8B:0E:6A:1B:54
Aug 11 2021 14:31:32  Detected new 802.11 Wi-Fi device DA:A1:19:15:AF:E2
Aug 11 2021 14:30:47  Detected new 802.11 Wi-Fi device 9E:6B:72:40:37:3C
Aug 11 2021 14:30:35  Detected new 802.11 Wi-Fi device 92:B7:73:71:D1:DF
Aug 11 2021 14:30:35  Detected new 802.11 Wi-Fi device 0E:52:88:AD:DA:1B
Aug 11 2021 14:30:35  Detected new 802.11 Wi-Fi device 1E:3B:E7:39:AE:28
```

Powered by many OSS components, see the credits page

Fig. 4. Wi-Fi monitoring by kismet intrusion detection system.

Figure 4, part 1 displays all Access Points and devices connected to Access points within reach of the Raspberry Pi. Part 2 displays the alerts (notifications of newly connected devices to the access point and devices that try to connect illegally). When an attack on the Wi-Fi on one of the access points commenced, Kismet would notify the attack by giving red color in column number 2 in **Figure 4** or shown in **Figure 5**. The notification can also be seen in the notification in the upper right corner of the Kismet web. In addition, Kismet would also immediately disconnect the devices that attack from the network.



Messages Channels

```
Aug 11 2021 14:37:12  DEAUTHFLOOD Deauth/Disassociate flood on B4:FB:E4:5E:9F:3B
Aug 11 2021 14:37:12  DEAUTHFLOOD Deauth/Disassociate flood on B4:FB:E4:5E:9F:3B
Aug 11 2021 14:37:12  BCASTDISCON IEEE80211 Access Point BSSID B4:FB:E4:5E:9F:3B broadcast deauthentication or disassociation of all clients; Either the AP is shutting down or there is a possible denial of service.
Aug 11 2021 14:37:12  BCASTDISCON IEEE80211 Access Point BSSID B4:FB:E4:5E:9F:3B broadcast deauthentication or disassociation of all clients; Either the AP is shutting down or there is a possible denial of service.
Aug 11 2021 14:37:12  Detected new 802.11 Wi-Fi device 9E:6B:72:B3:BC:B7
Aug 11 2021 14:37:08  Detected new 802.11 Wi-Fi device 22:64:CB:26:26:F6
Aug 11 2021 14:37:01  Detected new 802.11 Wi-Fi device 32:AF:B7:85:4C:5C
```

Powered by many OSS components, see the **credits page**

Fig. 5. Alert on kismet during the attack.

3.2 Testing

After designing the topology, devices were then arranged and connected. Kismet Intrusion Detection System Application was installed and run on the Raspberry Pi. The next step was to test the performance of the Kismet Intrusion Detection System on Raspberry. The tests were done by performing DoS (Denial of Service) attack on the Wi-Fi. These attacks was carried out ten times and the time when the attacks commenced until it was detected was then recorded.

In this study, the computer carrying out the attack had Kali Linux operating system. Denial of Service (DoS) attack was made by running the command sudo aireplay-ng --deauth 0 -a

B4:FB:E4:5E:9F:3B wlan0mon. The following is a description of the command --deauth 0 was sending infinite de-authentication packets, -a B4:FB:E4:5E:9F:3B was the mac address of the target access point, and at the end, there was wlan0mon which is the device used to carry out the attack, namely wireless adapter on Raspberry Pi. The command was repeated ten times. The following results were obtained.

Table 1. Detection result of denial of service attack.

Number of Attempts	Succeeded/Failed	Time (in second)
1	Succeeded	3,26 s
2	Succeeded	3,95 s
3	Succeeded	2,93 s
4	Succeeded	3,32 s
5	Succeeded	4,05 s
6	Succeeded	2.97 s
7	Succeeded	3,44 s
8	Succeeded	3,24 s
9	Succeeded	3,99 s
10	Succeeded	3,05 s

In addition to testing the Denial of Service attack, testing the strength of the Raspberry Pi is running the Kismet Intrusion Detection System by running Kismet to monitor Wi-Fi around the Raspberry was also done. The results obtained that the Raspberry was only able to run the Kismet Intrusion Detection System in less than 1 hour. After that time, Kismet could not do monitoring activity and must be restarted.

3.3 Maintenance

After conducting the last test, maintenance, it can be stated that Kismet could detect Denial of Service attacks. Detecting and anticipating these attacks reduced the Raspberry Pi performance, therefore Kismet would stop running every 1 hour. Considering this problem, maintenances were done by moving Kismet logs to the cloud, deleting them from raspberries, rebooting the operating system, and then running Kismet immediately.

3.4 Discussion

The results of the testing showed that Kismet was effective in detecting Denial of Service attacks. From the Denial of Service attacks, from 10 attacks, Kismet was able to detect and anticipate all attacks with an average detection time from the attack sent to detected was 3.42 seconds. However, when Kismet was started, the Raspberry Pi was only able to run for approximately 1 hour. The Raspberry Pi needed to be restarted and run again after 1 hour period of running time. Kismet was only able to run for approximately 1 hour on the Raspberry pi due to the small processor and storage memory, causing Kismet to stop after running for approximately 1 hour. Therefore, it is necessary to transfer log data from Kismet to the cloud and delete it from the device. After all the log data was transferred to the cloud and deleted from the device, restart the Raspberry to monitor with Kismet.

4 Conclusion

Based on the research on Intrusion Detection System Performance using Kismet software installed on the Raspberry Pi, it can be concluded that Kismet installed on the Raspberry Pi could detect 10 Denial of Service attacks from 10 attacks attempts with an average detection rate of attacks commenced until detected by Kismet was 3.42 seconds. Therefore, it can be said that the performance of the Kismet intrusion detection system installed on the Raspberry Pi was extremely accurate, and could detect attacks incredibly quickly. However, the problem was found when the Kismet Intrusion Detection System was running for approximately 1 hour. The application was not able to detect attacks, and raspberry pi needed restarting and running Kismet gain. This is because the Raspberry processor and storage memory were small. The processor and memory were stuffed with Kismet logs. Therefore, it is necessary to move logs periodically every hour.

It is recommended to run Kismet on clustered raspberries. Clustered arrangement increases the power to run Kismet and reduces the need to restart the Raspberry Pi every hour. It is also recommended to develop a system that provides notifications to network administrators via over the top (OTT) such as Whatsapp or Telegram. In addition, the system must also be able to perform other Intrusion Detection measurements on the Raspberry Pi.

References

[1] R. Haryunarendra, M. N. Al-Azam, and D. Rizaluddin, “Performa Jaringan Free Wireless di Taman Kota Surabaya,” *J. Link*, vol. 26, no. 2, pp. 25–29, 2017.

[2] M. Nizam *et al.*, “RaspyAir : Self-Monitoring System for Wireless Intrusion Detection using Raspberry Pi RaspyAir : Self-Monitoring System for Wireless,” vol. 1, no. February, pp. 20–31, 2017.

[3] S. Banerji and R. S. Chowdhury, "On IEEE 802 . 11 : Wireless LAN Technology," no. July 2013, 2015, DOI: 10.5121/ijmnct.2013.3405.

[4] S. D. Kanawat and P. S. Parihar, “Attacks in Wireless Networks,” *Int. J. Smart Sens. Adhoc Network.*, no. 1, pp. 113–116, 2011, doi: 10.47893/ijssan.2011.1033.

[5] M. P. S and S. Pavithran, "Advanced Attack Against Wireless Networks Wep , Wpa / Wpa2-Personal And Wpa / Wpa2- Enterprise," no. August, 2015, DOI: 10.13140/RG.2.2.35107.91686.

[6] G. Arna, J. Saskara, I. P. O. Indrawan, and P. M. Putra, “KEAMANAN JARINGAN KOMPUTER NIRKABEL DENGAN CAPTIVE PORTAL DAN WPA / WPA2 DI POLITEKNIK GANESHA GURU,” *J. Pendidik. Teknol. dan Kejuru.*, vol. 16, no. 2, pp. 236–247, 2019, doi: 10.23887/jptk-undiksha.v16i2.18559.

[7] A. Hamdan, M. N. Rafidah, B. Bahaa, and A. A. Zaidan, “Intrusion Detection System: Overview,” *J. Comput.*, vol. 2, no. 2, pp. 130–133, 2010.

[8] S. Saini Yogesh Kumar Sharma, “A Research Study of Wireless Network Security: A Case Study,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 6, no. 3, p. 2277, 2016, [Online]. Available: www.ijarcsse.com.

[9] M. Agarwal, S. Biswas, and S. Nandi, “Detection of De-authentication Denial of Service,” 2013.

[10] I. A. Sobari, “Rancangan Wireless Intrusion Detection System Menggunakan Snort,” *J. Techno Nusa Mandiri*, vol. 12, no. 1, pp. 1–9, 2015.

[11] M. G. H. Wibowo, J. Triyono, and E. Sutanta, “Keamanan Jaringan Wlan Terhadap Serangan Wireless Hacking Pada Dinas Komunikasi & Informatika Diy,” *Semin. Nas. Call Pap. Pengemb. Smart*

City menuju Pembang. Kota yang Cerdas dan Berkelanjutan, vol. 1, no. 1, pp. 2–9, 2017.

[12] D. Pranata, Y. N. Kunang, and N. A. O. Saputri, “Peningkatan Keamanan Jaringan Nirkabel Dengan Pendeteksi Serangan Berbasis Kismet DD-WRT,” *Bina Darma Conf. Comput. Sci.*, vol. 1, no. 5, pp. 1126–1132, 2019.

[13] M. Hanindia and P. Swari, “Intrusion Detection System (Ids) Menggunakan Raspberry Pi 3 Berbasis Snort Studi Kasus : Stmik Stikom Indonesia,” *J. SCAN*, vol. XV, pp. 2–7, 2020.