Journal of Computing Research and Innovation (JCRINN) Vol. 8 No.2 (2023)
https://jcrinn.com : eISSN: 2600-8793 https://dx.doi.org/10.24191/jcrinn.v8i2.375

Performance Analysis of Open-Source Network Monitoring Software in Wireless Network

Mohd Faris Mohd Fuzi[1]*, Mohd Firdaus Mohd Mahdzir[2], Iman Hazwam Abd Halim[3], Rafiza Ruslan[4]
[1,2,3,4] *College of Computing, Informatics, and Mathematics, Universiti Teknologi MARA Perlis Branch, Arau Campus, 02600 Arau, Perlis, Malaysia.*

*Corresponding author: * farisfuzi@uitm.edu.my*

Received Date: 27 July 2023
Accepted Date: 30 July 2023
Revised Date: 10 August 2023
Published Date: 1 September 2023

HIGHLIGHTS

- Identify open sources of network monitoring software that demonstrates a significant level of configurability.
- TCP attacks for response time and UDP attacks for packet loss were launched in the testing phase.
- The attacks include four different scenarios, each involving varying numbers of threads: 15 threads, 30 threads, 45 threads, and 60 threads.

ABSTRACT

Nowadays, computer networks are used in education, sports, business, transportation, manufacturing, and more. Thus, it makes the network more complex and harder to monitor, especially in terms of performance. The implementation of network monitoring software presents a viable solution for resolving this issue. It is because network monitoring software can monitor behaviours, provide alertness for any failing component of a computer network, and measure performance such as CPU usage, bandwidth, throughput, response time, and packet loss. However, network monitoring software has common problems, such as being hard to set up and making it hard to find the best network monitoring that fits the requirements. Thus, this study aims to identify network monitoring software that is easy to configure and analyse the performance of network monitoring software in terms of response time and packet loss. The study comprises the implementation of open-source network monitoring software, including PRTG, OpManager, Zabbix, and LibreNMS. The two experiments were conducted by launching TCP attacks for response time and UDP attacks for packet loss. Both attacks consisted of four scenarios: 15 threads, 30 threads, 45 threads, and 60 threads. As a result, PRTG offers a simpler installation procedure and easier configuration options, with a straightforward and attractive Graphical User Interface (GUI) that allows for all configurations to be conveniently performed directly through the interface. For monitoring websites, users need to add the device that was targeted by the website and choose an HTTP sensor for response time and PING for packet loss. The evaluation was based on the lowest average response time and consistency in the detection of packet loss. Consequently, Zabbix demonstrates the highest level of responsiveness about web services, which have the lowest average response time and consistently exhibit the lowest occurrence of packet loss.

Keywords: *network monitoring software, performance, response time, packet loss, TCP attack, UDP attack, threads.*

INTRODUCTION

The Internet has become a part of human needs and is commonly used for business, education, banking, conferences, entertainment, news, and others. Without it, all human activities might be delayed. The latest survey of Malaysian Internet users in 2020 shows that 88.7% of the Malaysian population are Internet users, which equates to 29.77 million users (MCMC, 2020). This finding shows that the Internet is important to our lives and keeps growing. Since the Internet has become a human need, it is important to monitor the performance of hardware or software that was used to forward and maintain the Internet connection to the users such as routers, switches, and hubs. In 2018, almost 200,000 Cisco switches globally which included Iran, the United States, India, China, and Europe were being attacked by hackers (Geetha Nandikotkur, 2018). This is due to the vulnerability in the software of the Cisco router which is known as the Cisco Smart Install Client. The attackers are capable of resetting the routers and blocking it from being reconfigured. These are some of the main reasons why network monitoring software was required.

There are numerous network monitoring tools with different features, either open source such as The Dude, Wireshark, Nagios, Observium, and Zabbix, or in paid versions like Datadog, Pulseway, GFI Lan Guard, and others. Network monitoring software is getting more advanced and more complex to manage and handle. Therefore, the management of the network requires great attention, and this task belongs to the network administrator. Failure to manage the network could lead to huge problems for small businesses or large enterprises. Network monitoring is the best answer to having good network management, but to implement it, a higher level of understanding of network monitoring is required. However, the available network monitoring software and solutions are not only costly but also hard to set up, configure, administer, and maintain (Ghafir et al., 2015).

The selection of appropriate network monitoring software presents difficulties when considering both paid and open-source platforms, as it necessitates meeting the network's specific requirements (Hernantes, 2015). This issue generates a common debate on network communities and forums, but which one is the best? For example, if a company wants to choose free network monitoring, it will face the challenge of deciding on the best network monitoring software that suits its requirements since there are several free network monitoring options. Based on the current problems, two objectives have been determined to overcome the problem: to identify the network monitoring software that is easy to configure and to analyse the performance of the network monitoring software in terms of response time and packet loss between PRTG, OpManger, Zabbix, and LibreNMS.

LITERATURE REVIEW

This session discusses categories of network monitoring, types of attacks, a list of network monitoring tools, and any related work based on network monitoring software.

Categories of Network Monitoring

Network monitoring categories can be divided into several parts, which are security fault management, configuration management, accounting management, performance management, and security management. Each category has different working principles. However, this project will only focus on performance management. Performance management functions ensure the network's performance is maintained in its best state (Jiang et al., 2023). There is a strong relationship management component with traffic management and Quality of Services (QoS) because it helps manage the Service Level Agreements (SLAs) between the service provider and the client. Generally, the functionality of performance management can be divided into several parts, like collecting statistical information, preserving and examining logs of system state histories, and changing system modes of operation to manage performance activities (Tsapardakis, 2018).

Types of Attacks

Many ways can be used to test the network environment. One of the methods is to launch an attack. Attacks that were launched in this study were TCP-SYN flood attacks and UDP flood attacks. The TCP-SYN flood attack is one of the most popular and basic attacks in the denial-of-service category. It takes advantage of the TCP three-way handshake, which will give the effect of unavailable access to network resources for end users. The attacker takes advantage of the TCP handshake process. The high volume of TCP SYN packets is sent to the server, and it usually comes with a spoofed IP address. Once the server accepts the TCP SYN packet, it will respond by sending a SYN-ACK to the sender. Since the SYN packet making the request has a high rate and comes with a fake IP address, there will be no connection or response from the sender. Therefore, the connection between client and server is half-open, and this blocks legitimate users from accessing the network resources.

In a UDP flood, the attack has been launched by sending continuous UDP packets at a very high volume using a large source IP range, either a random or fixed IP address. This will make the network devices such as firewalls, servers, routers, and IPS/IDS become overwhelmed with the spamming of UDP packets, which will lead to a service crash. Usually, network resources and bandwidth will be affected by this attack.

Open-Source Network Monitoring Software

Generally, there are a variety of network monitoring software, either paid versions or open source. However, this research will focus on four types of free network monitoring software, which are PRTG, Zabbix, OpManager, and LibreNMS due to their ability to monitor websites and measure response time and packet loss.

PRTG provides friendly interfaces that come with live-time statistics on every aspect of the network. It provides monitoring of resource availability, bandwidth, and network usage. The PRTG Traffic Grapher is a Windows application that can be easily used for monitoring and classifying bandwidth usage (Rahman, 2019). It is free for personal use and may be downloaded from their website. Real-time readings, a trouble-free operation, and an interesting interface are additional features that PRTG offers.

Zabbix is software that functions to monitor the network's performance, network equipment, and availability. Zabbix can be synchronised with Open Stack (Rahman, 2019). In cases where failure occurs, the network administrator will receive notification by phone or mail. There are no limits in terms of its monitored devices and capabilities. Besides, modifications are permitted on the source code level for those who are familiar with coding. In addition, installation of Zabbix can be done either on a small network or a large network.

OpManager is software that was developed by Zoho Corporation. It supports Windows and Linux platforms and can run on 32-bit and 64-bit operating systems. For this research, the free version is selected.

OpManager can manage network health monitoring, VoIP monitoring, network mapping, server monitoring, VMware monitoring, and others.

LibreNMS is an open-source network monitoring software that is based on auto-discovering PHP, MySQL, and SNMP and comes with the GNU General Public Licence. It can monitor the operating system and a wide range of network hardware. LibreNMS provides several features such as syslog, NetFlow, sFlow, alerting, URL monitoring, and others. Besides, LibreNMS has its community, which functions to help its users with problems that occur on LibreNMS.

METHODOLOGY

Design and Implementation Phase

Figure 1 shows the network topology with IP addresses for each device. This diagram consists of two laptops, a computer, and a mobile, which will work as the access point. Each laptop and computer will be connected to a mobile hotspot. To achieve the first objective, the configuration phase of each network monitoring software has been set up. Laptop A is used to set up LibreNMS and Zabbix, while Laptop B will act as an attacker. The computer was used to set up OpManager and PRTG. The flow will start with the attacker launching two types of attacks, such as UDP Flood and TCP SYN Flood, using a Low Orbit Ion Cannon (LOIC). The attack will target the web server as a victim. Once the attack is launched, the computer and Laptop A will capture the data based on the network monitoring tools that have been configured.

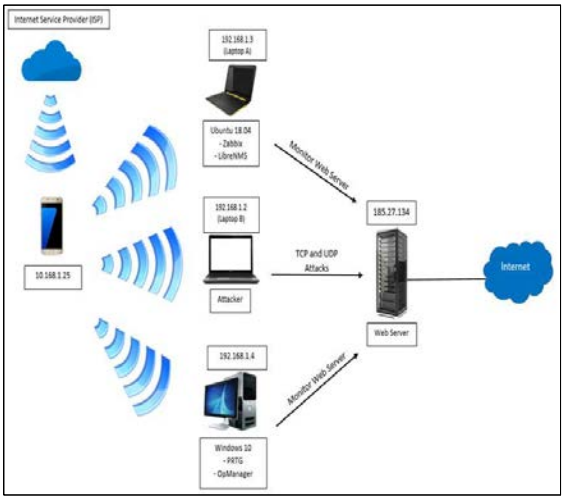


Figure 1: Network Topology

Experimentation Phase

Figure 2 shows the two experiments that have been conducted to measure the network monitoring software's performance. For Experiment 1, a TCP-SYN flood attack was launched to measure the performance of network monitoring software. Response time can be defined as the amount of time required to load the website from the access attempted (Peeva, 2018), while for Experiment 2, testing on packet loss with a

UDP attack has been examined. Packet loss can be described as data transmitted that was unsuccessful in reaching its destination (Rivenes, 2016).

Based on Experiment 1 and Experiment 2, four readings have been taken from response time and packet loss, which consist of 15 threads, 30 threads, 45 threads, and 60 threads. Then, this data will be analysed and compared by plotting the graph to determine the best network monitoring software in terms of response time and packet loss.

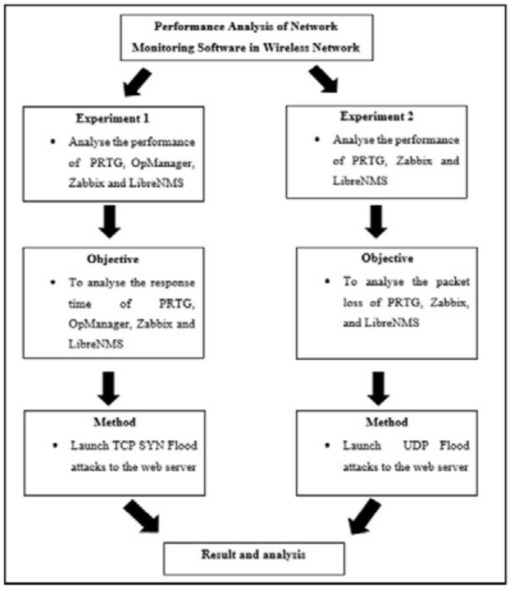

Performance Analysis of Network Monitoring Software in Wireless Network
Experiment 1
• Analyse the performance of PRTG, OpManager, Zabbix and LibreNMS
Experiment 2
• Analyse the performance of PRTG, Zabbix and LibreNMS
Objective
• To analyse the response time of PRTG, OpManager, Zabbix and LibreNMS
Objective
• To analyse the packet loss of PRTG, Zabbix, and LibreNMS
Method
• Launch TCP SYN Flood attacks to the web server
Method
• Launch UDP Flood attacks to the web server
Result and analysis

Figure 2: Experiment Framework

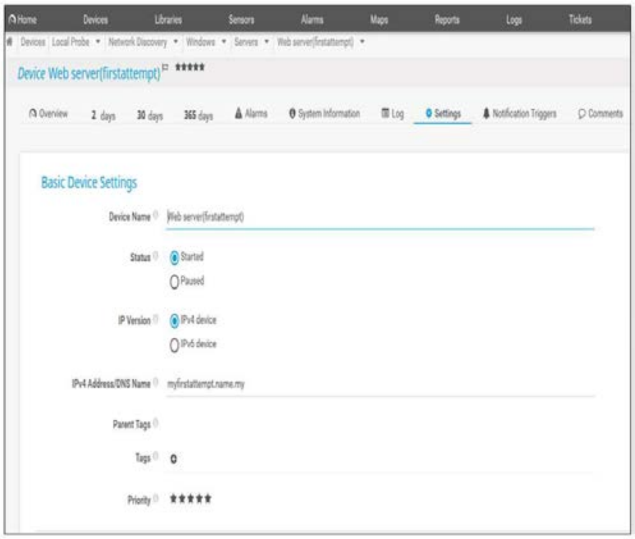
RESULTS AND DISCUSSIONS

The results were separated into two sections: the first section is the configuration of the network monitoring software, and the second section is the results of response time and packet loss. The configuration of the network monitoring software was divided into two categories: configuration on Windows and configuration on the Ubuntu platform. Each configuration consists of two parts, which are the configuration on response time and the configuration on packet loss. Configuration on Windows involved PRTG and OpManager, while configuration for Ubuntu consists of Zabbix and LibreNMS.

Configuration of PRTG

The device was configured as a Web Server (first attempt). Next, the target IP address that needs to be monitored was entered, which is myfirstattempt.name.my.

Once adding a device is completed, as shown in Figure 3, the HTTP sensor is added to monitor the device's response time. Figure 4 shows the configuration of the HTTP sensor. The timeout refers to a value, and if the response time takes longer than this value, it will trigger an error message. The same URL as the device's IP address needs to be entered for it to properly function. For the request method, GET needs to be selected to obtain the request data from myfirstattempt.name.my.



Device Web server(firstattempt) ★★★★★

Basic Device Settings

Device Name: Web server(firstattempt)

Status: ● Started ○ Paused

IP Version: ● IPv4 device ○ IPv6 device

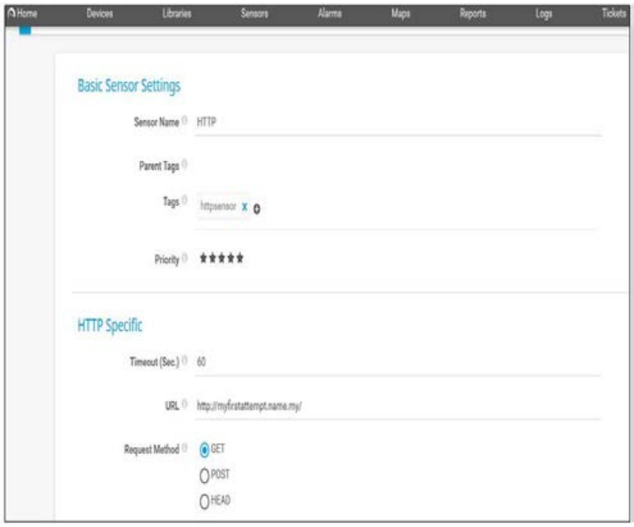
IPv4 Address/DNS Name: myfirstattempt.name.my

Parent Tags:

Tags:

Priority: ★★★★★

Figure 3: Adding device to be monitored.



Basic Sensor Settings

Sensor Name: HTTP

Parent Tags:

Tags: httpsensor ✕

Priority: ★★★★★

HTTP Specific

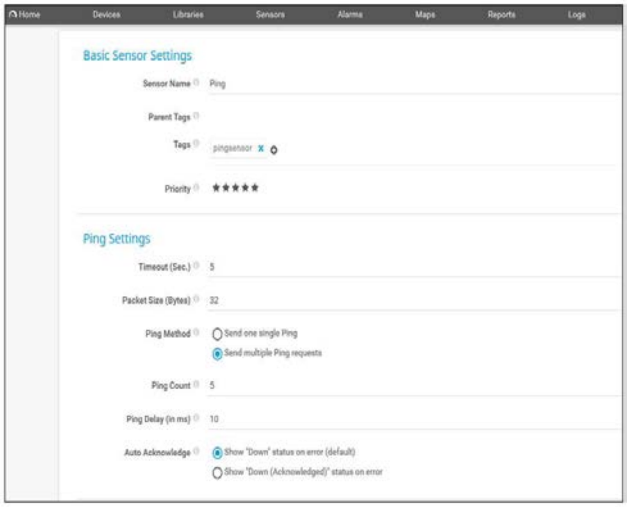
Timeout (Sec.): 60

URL: http://myfirstattempt.name.my/

Request Method: ● GET ○ POST ○ HEAD

Figure 4: HTTP Sensor

The PRTG ping sensor consists of downtime, ping time, and packet loss. Figure 5 shows the configuration of the ping sensor. Timeout refers to a value whereby, if the ping reply takes longer than this value, the ping sensor will generate an error message. Normally, the packet size of the ping is set to the default value, which is 32. For the ping method, sending multiple ping requests was selected, which means multiple pings will be sent within each scanning interval. Ping count and ping delay can only be enabled if sending multiple ping requests is selected.



Basic Sensor Settings

Sensor Name: Ping

Parent Tags:

Tags: pingsensor

Priority: ★★★★★

Ping Settings

Timeout (Sec.): 5

Packet Size (Bytes): 32

Ping Method: ○ Send one single Ping ● Send multiple Ping requests

Ping Count: 5

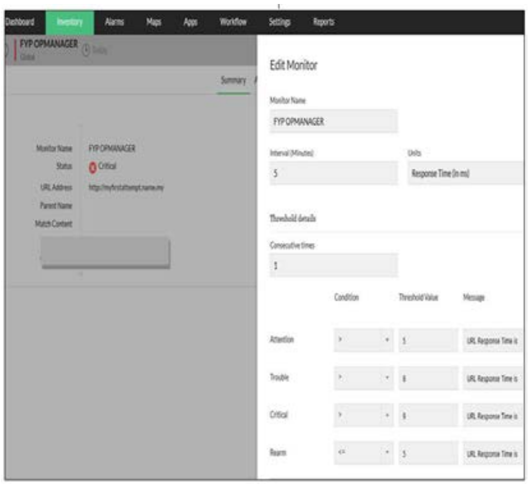
Ping Delay (in ms): 10

Auto Acknowledge: ● Show "Down" status on error (default) ○ Show "Down (Acknowledged)" status on error

Figure 5: Ping Sensor

Configuration of OpManager

OpManager only provides configuration on response time. The interval time used was the default configuration, which was 5 minutes. It means that every 5 minutes, the graph will be updated. Threshold can be defined as an action, such as an alert message, that will be generated based on the threshold value that has been set up, as shown in Figure 6. For example, if the response time exceeds the threshold value of 9, an alert message will be generated under critical conditions.



FYP OPMANAGER

Monitor Name: FYP OPMANAGER

Status: Critical

URL Address: http://myfirstattempt.name.my

Parent Name

Match Content

Edit Monitor

Monitor Name: FYP OPMANAGER

Interval (Minutes): 5

Units: Response Time (in ms)

Threshold details

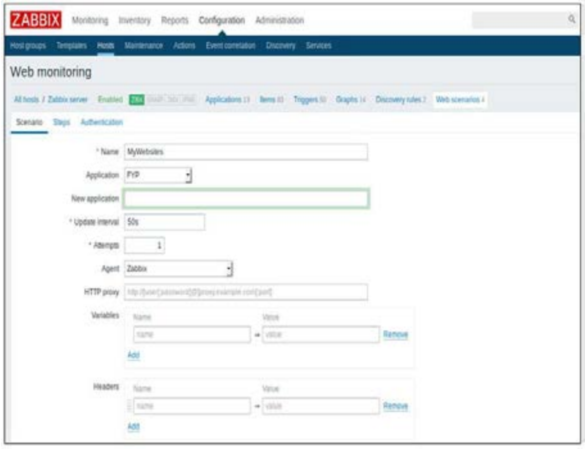
Consecutive times: 1

	Condition	Threshold Value	Message
Attention	>	5	URL Response Time is
Trouble	>	8	URL Response Time is
Critical	>	8	URL Response Time is
Rearm	<=	5	URL Response Time is

Figure 6: Configuration of response time

Configuration of Zabbix

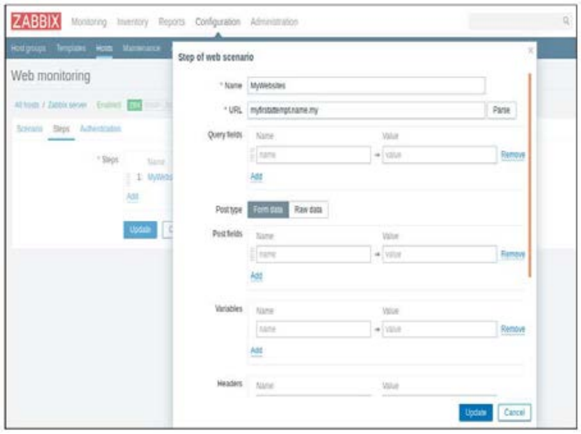
The first step is to create the web scenario, as shown in Figure 7. The web scenario was entered, and either an existing application or creating a new application was selected. This application was used to define the group of host items. For example, the application 'FYP' might contain items such as CPU idle time, available memory, system uptime, used disc space, and others. Any number on the update interval was filled in, and the number of attempts, which will be used in case there is any error, was entered. For the agent, ‘Zabbix’ was selected as the default configuration.



* Name: MyWebsites
Application: FYP
New application:
* Update interval: 50s
* Attempts: 1
Agent: Zabbix
HTTP proxy: http://[user[:password]@]proxy.example.com[:port]
Variables: Name name → Value value
Headers: Name name → Value value

Figure 7: Configuration of web scenario

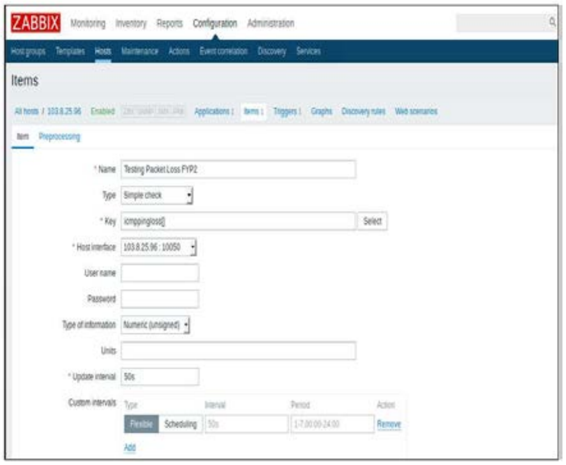
Figure 8 shows the steps for monitoring the targeted website. Any suitable name was written in the name textbox, the URL that needs to be monitored was entered in the URL textbox, and the Update’ button was clicked. Once completed, users need to wait for several minutes and keep refreshing Zabbix to generate the graph of download speed and response time for the targeted website.



* Name: MyWebsites
* URL: myfirstattempt.name.my
Query fields: Name name → Value value
Post type: Form data
Post fields: Name name → Value value
Variables: Name name → Value value
Headers: Name → Value

Figure 8: Steps of web scenario

Firstly, the new host needs to be created, and the targeted website needs to be specified. Then a new application was created, and the application was renamed. Next, the new item was created, as shown in Figure 9. Any suitable name, such as Testing Packet Loss FYP2, was chosen as the same. For the type, ‘Simple Check was selected, which is used to monitor simple things such as icmpping, icmppingloss, icmppingsec, and net. tcp.service, and others. Icmppingloss was chosen for monitoring the packet loss. Once all steps are completed, users need to wait a few minutes for Zabbix to generate the packet loss graph.


ZABBIX
Monitoring Inventory Reports Configuration Administration
Host groups Templates Hosts Maintenance Actions Event correlation Discovery Services
All hosts / 103.8.25.96 Enabled Applications 1 Items 1 Triggers 1 Graphs Discovery rules Web scenarios
Item Preprocessing
Select
Flexible Scheduling Remove
Add

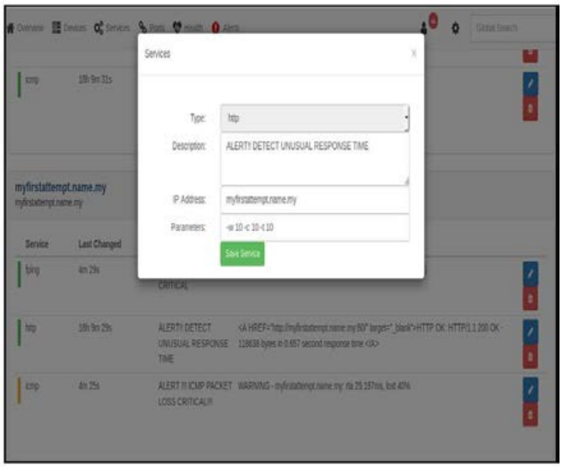
Items

* Name: Testing Packet Loss FYP2
Type: Simple check
* Key: icmppingloss[]
* Host interface: 103.8.25.96 : 10050
User name:
Password:
Type of information: Numeric (unsigned)
Units:
* Update interval: 50s
Custom intervals: Type: Flexible; Interval: 50s; Period: 1-7,00:00-24:00

Figure 9: Configuration of packet loss

Configuration of LibreNMS

Adding a device needs to be done first before configuration on response time can be done. Then, the 'Service' tab was selected, the device that had been created was chosen, and the 'Add Service' tab was chosen. For this research, the device was named myfirstattempt.name.my. To monitor the response time of the website, HTTP was chosen, as shown in Figure 10. The description and IP address of the targeted website were entered. The parameter is an optional option. The -w describes warning time, the -c is a critical time, and the –t -t is timeout.



Services

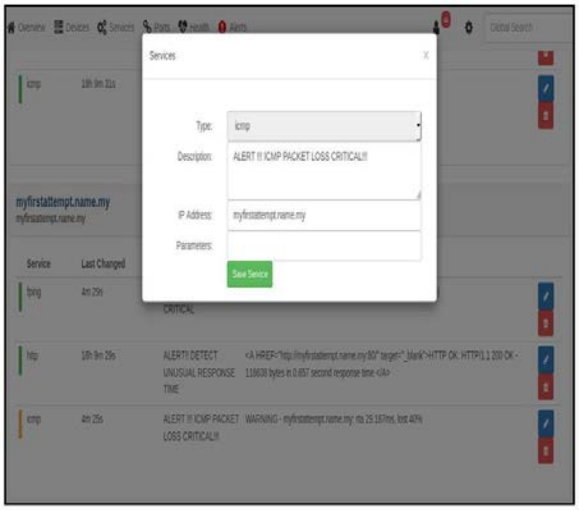
Type: http
Description: ALERT! DETECT UNUSUAL RESPONSE TIME
IP Address: myfirstattempt.name.my
Parameters: -w 10 -c 10 -t 10

myfirstattempt.name.my
myfirstattempt.name.my

Service	Last Changed		
icmp	18h 9m 31s		
fping	4m 29s	CRITICAL	
http	18h 9m 29s	ALERT! DETECT UNUSUAL RESPONSE TIME	<A HREF="http://myfirstattempt.name.my:80" target="_blank">HTTP OK: HTTP/1.1 200 OK - 118638 bytes in 0.657 second response time </A>
icmp	4m 25s	ALERT !!! ICMP PACKET LOSS CRITICAL!!!	WARNING - myfirstattempt.name.my: rta 25.157ms, lost 40%

Figure 10: Configuration of response time

The same step for adding the service was repeated, and ICMP was chosen. ICMP is used to monitor the packet loss of websites, as shown in Figure 11. The description and IP address of the targeted website were entered. For specific configurations, the parameter text box was either filled in or left blank for the default configuration.



Type: icmp
Description: ALERT !!! ICMP PACKET LOSS CRITICAL!!
IP Address: myfirstattempt.name.my
Parameters:

icmp 18h 9m 31s

myfirstattempt.name.my
myfirstattempt.name.my

Service	Last Changed		
fping	4m 29s	CRITICAL	
http	18h 9m 29s	ALERT!! DETECT UNUSUAL RESPONSE TIME	<A HREF="http://myfirstattempt.name.my:80" target="_blank">HTTP OK: HTTP/1.1 200 OK - 116638 bytes in 0.657 second response time </A>
icmp	4m 25s	ALERT !!! ICMP PACKET LOSS CRITICAL!!	WARNING - myfirstattempt.name.my: rta 29.167ms, lost 40%

Figure 11: Configuration of packet loss

Based on the explanation above, PRTG provided the easiest configuration for response time and packet loss. The user needs to create the device name, which consists of the targeted website the chosen HTTP sensor for response time, and the PING sensor for packet loss. OpManager also provided a simple configuration on response time, but configuration on packet loss was not available.

For monitoring response time using Zabbix, users need to create the web scenario. A web scenario consists of two main configurations: a scenario and steps. The scenario means the configuration of response time parameters such as interval, application, and others, while the steps are the configuration of the targeted websites. In terms of packet loss, a new host and application need to be created, followed by the configuration of the item. LibreNMS does not provide URL monitoring features. Thus, to monitor the website, the Nagios plugin needs to be installed, and several changes need to be made to config.php and the cron job. Once completed, users need to add the device and choose HTTP for response time and ICMP packet loss.

Results of the Response Time

Figure 12 shows the overall analysis of the average response time for each network monitoring software. The attacks were conducted in an hour, but at different times for each thread. For the 15 thread attacks, PRTG initially detected 1901 ms of response time. Then, the time gradually increased to 8596 ms when 30 threads were launched and slightly increased to 9284 ms for 40 threads. However, when 60 threads were launched, PRTG was not able to detect the response time. Thus, the dotted lines on the graph represent the unreadable response time of myfirstattempt.name.my.

OpManager dominated the average response time for the first attacks among other network monitoring software with a value of 2246 ms, and the time continued to increase significantly when there were 30 threads. During the 45-thread attacks, the time recorded in the OpManager graph increased rapidly to 9776 ms, which was the highest average response time among the threads. When 60 threads were launched, the data on the response time was not able to be retrieved.

The red bar in Figure 12 represents Zabbix. The average response time of myfirstattempt.name.my during the first scenario was only 1785 msec. It slightly increased to 2372 ms when the second scenario was conducted. Then, it continued to increase drastically until it reached 5889 ms for the 45 threads. For the 60 threads, the result was similar to PRTG and OpManager due to the crashed website.

LibreNMS recorded the lowest average response time for the 15 threads. LibreNMS was only capable of detecting 1470 ms. The response time continued to rise until it reached 3230 ms for the 30 threads and started to increase when 45 threads were launched, with a value of 6380 ms. The result for the 60 threads was undetectable, which was similar to other network monitoring software.

As a conclusion, the comparison of the performance analysis on the normal average response time and the attacked average response time graph can be clearly distinguished. Each network monitoring software shows an increase in response time from 15 threads to 45 threads, and the response time started to decrease when 60 threads were launched. Based on Figure 12, the best network monitoring software for monitoring the response time is Zabbix. The evaluation was based on the lowest average response time. Zabbix took less time responding to myfirstattempt.name.my. Zabbix had the lowest average response time for the second and third scenarios. For example, in the third scenario, Zabbix detected 5889 ms and OpManager detected 9776 ms. At this phase, Zabbix already knows that when the response time reaches 5889 ms, something will happen in the next phase. Compared to OpManager, when the value reached 9776 ms, it only knew that the website would not be able to be reached for the next scenario.

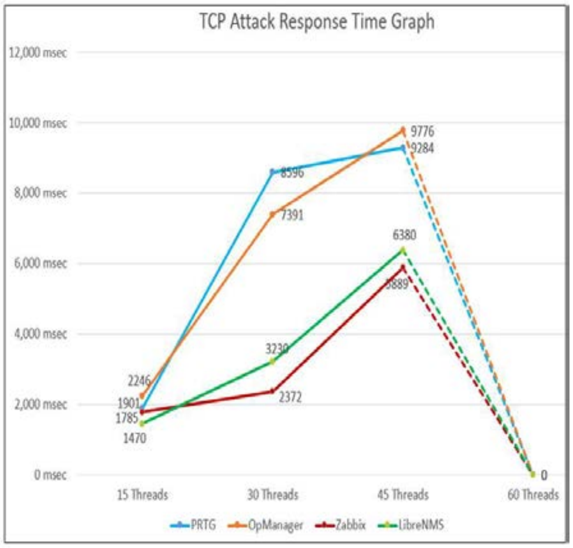


Figure 12: Analysis of response time on TCP attack

Results of the packet loss

Figure 13 shows the overall result of the UDP attacks, which involved four different types of scenarios: 15 threads, 30 threads, 45 threads, and 60 threads. For the first scenario, the average percentage of packet loss detected by PRTG was 5% only which was the lowest packet loss among the 15 threads. However, when 30 threads were launched, the percentage increased drastically to 48%. The third scenario shows an increase of 28% from the previous threads, and for the last scenario, there was a slight increase in percentage captured by PRTG with a value of 64%.

The red line represents Zabbix. Only 7.43% of the average packet loss for the 15 threads was detected by Zabbix. Then, Zabbix detected 51.39% of the average packet loss, which was the highest percentage for the overall 30 threads. The percentage kept rising until 64.44 % for 45 threads and reached 80 % for the last experiment.

Theoretically, the percentage of packet loss keeps increasing from the first scenario until the last scenario. The best network monitoring tool for average packet loss was Zabbix. This is because of the consistency that shows up in Zabbix for each scenario. For example, in the first scenario between Zabbix and PRTG, which are 7.43% and 5%, LibreNMS was at 14.43%. The range of differences between Zabbix and PRTG is only 2.43%, while Zabbix and LibreNMS are 7%. The second scenario shows the difference between each network monitoring software is close. Zabbix and PRTG still show the consistency values, which are 51.39% and 48% with only 3.39% differences, while 45.07% was detected by LibreNMS and shows the difference only at 6.43% with Zabbix.

The third scenario still shows the closed range of packet loss. The average percentage of packet loss for Zabbix, PRTG, and LibreNMS is 64.444%, 60%, and 66.66%, respectively. Zabbix and PRTG show a difference with values of 4.444%, while Zabbix and LibreNMS show only 2.16% of average packet loss. However, the differences between Zabbix and PRTG for the fourth scenario created a huge range, which was at 16% with a value of 80% and 64% of average packet loss, while Zabbix and LibreNMS provided a close range of differences with only 6.72% of average packet loss.

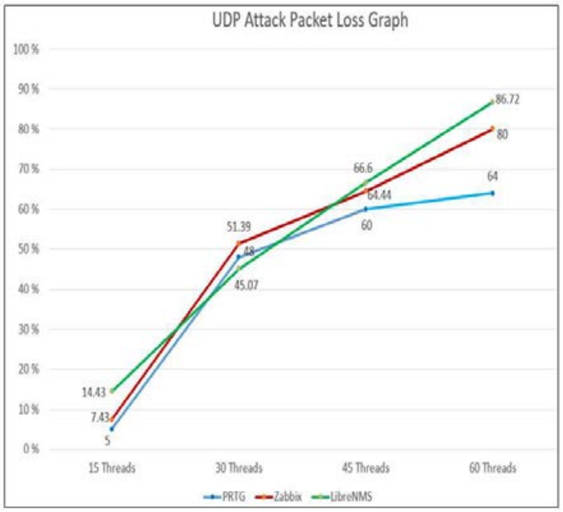


Figure 13: Analysis of packet loss on UDP attack

CONCLUSION AND RECOMMENDATIONS

In conclusion, based on the installation and configuration processes between the four open-source network monitoring software, PRTG offers a simpler installation procedure and easier configuration setup. It has a straightforward installation process and an attractive Graphical User Interface (GUI). All the configurations can be conveniently performed directly through the GUI. For example, to monitor the website, the user needs to create the device name and select any sensor that is suitable to be used for monitoring the website, such as HTTP, ping, FTP, IMAP, and others. Monitoring websites using OpManager also provided such a simple configuration. However, the configuration was limited to the response time only, while for Zabbix, the user needed to create the web scenario to monitor the website. Under the web scenario, there are three categories to be configured: scenario, steps, and authentication. For LibreNMS, the installation of the

Nagios plugin is required to monitor the website. Once the plugin is installed, some changes need to be made to config.php and the crob job files. After that, launch LibreNMS and select the services that need to be added.

The results obtained from the experiments conducted on response time indicate that Zabbix can identify the most efficient response time for web services. Zabbix was shown to have the lowest average response time during experimentation. For the second experiment on packet loss, Zabbix shows a consistent graph of packet loss from the first scenario until the last scenario. For example, the first scenario shows that Zabbix and PRTG were very close with values of 7.43% and 5%, while LibreNMS was far away with 14.43%. The second scenario shows that the ranges were very close to each other. The same goes for the third scenario; the range was very close. Zabbix captured 64.44 % while PRTG and LibreNMS captured 60 % and 66.6 %. However, for the fourth scenario, the closed range was between Zabbix and LibreNMS, which is 80% and 86.72 %, while PRTG was left behind with only 64 %.

There are numerous recommendations for future research. Firstly, all the network monitoring software used in this research was set up on a computer and laptop. In other words, the server would only work if the computer and laptop were turned on. Future research is recommended to implement all this network monitoring on a real server such as Digital Ocean, Exabytes, SiteGround, and others that can get more accurate data.

In terms of packet loss monitoring, OpManager might not be a suitable network monitoring tool. This is because OpManager does not provide features for monitoring packet loss; rather, it only provides URL monitoring to measure the response time and the availability of the websites.

CONFLICT OF INTEREST DISCLOSURE

The authors declared that they have no conflicts of interest to disclose.

REFERENCES

Geetha Nandikotkur (2018). 200,000 Cisco Network Switches Reportedly Hacked. Retrieved May 2023, from https://www.bankinfosecurity.com/200000-cisco-network-switches-reportedly-hacked-a-10788

Ghafir, I., Svoboda, J., & Prenosil, V. (2015). Network Monitoring Approaches An Overview. *Third International Conference on Advances in Computing, Communication and Information Technology-CCIT 2015*. DOI: 10.15224/978-1-63248-061-3-72

Hernantes, J., Gallardo, G. & Serrano, N. (2015). IT Infrastructure-Monitoring Tools. *The IEEE Computer Society*, 88-93

Jiang,M., Jianfeng, F. & Xiaodeng, Z. (2023). Research on Key Technology and System Design of Network Performance Monitoring System. *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*, Jilin, China, 210-214, doi: 10.1109/ICCECT57938.2023.10140534.

MCMC (2020). Internet Users Survey 2020. Retrieved May 2023, from https://www.mcmc.gov.my/skmmgovmy/media/General/pdf/IUS-2020-Report.pdf

Peeva, B. (2018). Website Response Time Explained by WebSitePulse. Retrieved May 2023, from https://www.websitepulse.com/blog/website-response-time-main-factor-for-your-online-business-revenue

Rahman, W., Phong, T. N., Rusliyadi, M., Laxmi., L. E. & Shankar, K. (2019). Network Monitoring Tools and Techniques Used in the Network Traffic Management System. International Journal of Recent Technology and Engineering (IJRTE), Volume-8, Issue-2S11,4182-4188. DOI: 10.35940/ijrte.B1603.0982S1119

Rivenes, L. (2016). What are the Causes of Packet Loss? - Datapath.io. Retrieved May 2023, from https://datapath.io/resources/blog/causes-of-packet-loss/

Tsapardakis,E., Ojo, M., Chatzimisios, P. & Giordano. (2018). Performance Evaluation of SDN and RPL in Wireless Sensor Networks. *2018 Global Information Infrastructure and Networking Symposium (GIIS),* Thessaloniki, Greece, 1-5, doi: 10.1109/GIIS.2018.8635599.