© Universiti Tun Hussein Onn Malaysia Publisher's Office

**JSCDM**

http://penerbit.uthm.edu.my/ojs/index.php/jscdm

e-ISSN : 2716-621X

Journal of Soft Computing and Data Mining

# A Comparative Performance of Port Scanning Techniques

**Fatin Hazirah Roslan**[1*]

[1]Faculty of Computer Science and Information Technology,
University Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/jscdm.2023.04.02.004
Received 18 February 2023; Accepted 18 September 2000; Available online 04 October 2023

**Abstract:** Port scanning is the first step taken by attackers before an attack is deployed. It is employed to identify the targeted host's IP addresses, network devices and services running which later be used to determine the server locations and diagnose security levels of the victim by revealing the presence of security measures in place such as firewall between the server and the network devices. With different types of port scanning techniques and tools available, the impact on the targeted host's performance will vary. In this research, a comparative study of port scanning techniques is proposed to evaluate their impact on the scanned hosts performance. Three scanning techniques are compared namely TCP SYN, TCP Connect and UDP scan and several experiments have been conducted using NMAP, Unicornscan, Netcat, Apache2 web server and Zabbix running in virtual machine (VM) environment. Of the three port scanning techniques, TCP SYN scan has the least impact on the targeted scanned host with average response time of 0.69ms for a single scan and 0.421ms for 100 scans.

**Keywords:** Port scanning techniques, Nmap, Unicornscan, Netcat, UDP, TCP

## 1. Introduction

The internet is a vast interconnection that connects all people around the globe. In today's era, network devices are connected to the internet which allows us to access various kinds of services, daily. This leads to an increase of cyber-attacks that might expose confidential data of user, as cyber-crime continues to rise rapidly [1].

According to EC-Council Certified Ethical Hacker (CEH), the anatomy of a cyberattack can be divided into five phases. Figure 1 shows the methodology of penetration testing [2].

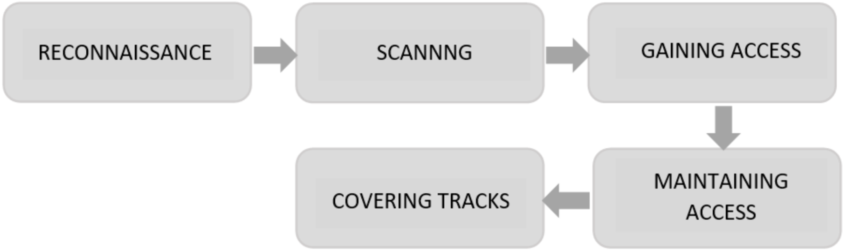


**Fig. 1 - Penetration testing methodology**

In the scanning phase, the attacker will do several scanning procedures in order to discover open doors or weaknesses in a network. The goal of this step is to get as much information as possible before launching an attack [3].

A tool such as port scanner is used to gather data by listening to open Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) ports to identify running services and operating systems (OS) on the targeted host [4]. Port numbers range from 0 to 65,536 and are ranked in terms of popularity. Ports numbered from 0 to 1023 are called "well-known" ports which are usually used for internet usage. These ports are assigned by the Internet Assigned Numbers Authority (IANA) [5]. Ports are generally managed by the Transmission Control Protocol (TCP) which defines how to establish and maintain a network communication between applications and User Datagram Protocol (UDP) which is mainly used for establishing low latency between applications. Table 1 shows some of the most famous and most frequently used TCP and UDP ports.

**Table 1 - Well-known TCP and UDP Ports**

| Port Number | Transport Protocol | Service Name |
|---|---|---|
| 20, 21 | TCP | File Transfer Protocol (FTP) |
| 22 | TCP and UDP | Secure Shell (SSH) |
| 23 | TCP | Telnet |
| 25 | TCP | Simple Mail Transfer Protocol (SMTP) |
| 53 | TCP and UDP | Domain Name Server (DNS) |
| 67, 68 | UDP | Dynamic Host Configuration Protocol (DHCP) |
| 69 | UDP | Trivial File Transfer Protocol (TFTP) |
| 80 | TCP | HyperText Transfer Protocol (HTTP) |
| 110 | TCP | Post Office Protocol (POP3) |
| 119 | TCP | Network News Transport Protocol (NNTP) |
| 123 | UDP | Network Time Protocol (NTP) |
| 135-139 | TCP and UDP | NetBIOS |
| 143 | TCP and UDP | Internet Message Access Protocol (IMAP4) |
| 161, 162 | TCP and UDP | Simple Network Management Protocol (SNMP) |
| 179 | TCP | Border Gateway Protocol (BGP) |
| 389 | TCP and UDP | Lightweight Directory Access Protocol |
| 443 | TCP and UDP | HTTP with Secure Sockets Layer (SSL) |
| 636 | TCP and UDP | Lightweight Directory Access Protocol over TLS/SSL (LDAPS |
| 989/990 | TCP | FTP over TLS/SSL |

## 2. Related Work

In port scanning, there are dozens of techniques that can be used by an attacker to verify open ports on the victim's server. As revealed by Muraleedharan [6], the most common port scanning techniques used are TCP SYN scan, TCP Connect scan, UDP scan and stealth scan. TCP SYN scan is the most commonly used because it does not establish a connection between the attacker and the victim's machine and is not logged by some of event tracking tools. However, TCP SYN scan requires superuser privileges in order to send requests.

TCP Connect scan is the alternative scan when SYN scan is not possible. It involves establishing a full connection with the target machine by completing a three-way handshake which can be time-consuming. Compared to other scans, the TCP Connect scan is slow and noisy which can be easily detected by IDS/IPS systems, and it does not require special rights to send requests.

UDP scan is different from previously discussed techniques as it is not able to determine open ports by analyzing the ports' responses because open ports do not react to received UDP packets. This excludes a statement about the port's status because the port scanner's UDP packet could have been lost without reaching the scan's target. Therefore, UDP scans work the other way around and only determine closed ports. Closed ports respond with an ICMP port unreachable message if an UDP packet is received. This allows us to determine ports that are closed.

As mentioned by Simon Bauer [7], different scan methods can assist the user to adapt scans to his needs. Nmap is the most widely used and famous port scanner due to its flexibility and ability to perform the greatest amount of different port scanning methods. Table 2 shows the overview of available scanning methods for each tool.

**Table 2 - Overview of available port scanning methods [7]**

| Tool | SYN | Connect | UDP | ACK | FIN | XMAS | NULL |
|---|---|---|---|---|---|---|---|
| Nmap | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Masscan | ✓ | X | X | X | X | X | X |
| ZMap | ✓ | X | ✓ | X | X | X | X |
| xProbe2 | ✓ | X | ✓ | X | X | X | X |

Abhinav and Srinivas [8] reviewed different port scanning methods using several port scanning tools. Nmap is the most reliable port scanning that offers a great range of port scanning methods, followed by Netcat and hping2. Table 3 shows the overview of the port scanning methods and their use cases and tools used to perform them.

**Table 3 - Overview of port scanning methods and tools [8]**

| Scanning Method | Details | Tool |
|---|---|---|
| TCP Connect | Uses 3-way handshake to make connection<br>Does not require special super user privilege | Nmap, hping2 |
| SYN scan | Does not complete 3-way handshake to make connection.<br>Can scan thousands of ports per seconds | Nmap, Strobe, TCP port scanner |
| FIN | A FIN flag set is sent within a packet. If the port is closed, the host returns RST flag whereas open port ignores the request. | Nmap, Netcat, hping2 |
| XMAS | Similar to FIN scan but have extra two flags within a packet. | Nmap, Netcat, hping2 |
| NULL | Similar to FIN and XMAS scan but differs in packet header flags. Instead of sending an invalid packet due to header is turning on flags, NULL turns off all header flags.<br>Only works with Unix based devices. | Nmap, Netcat, hping2 |
| UDP | Send UDP packet to the target port and can be time-consuming | Nmap, Netcat, ScanUDP |

Nmap works well in implementing all TCP port scanning attacks but unfortunately, not with UDP scanning [9]. As shown in Figure 2, the time taken to complete UDP scanning using Nmap was too long compared to other scanning. Unicornscan was observed as the best tool for UDP scanning.

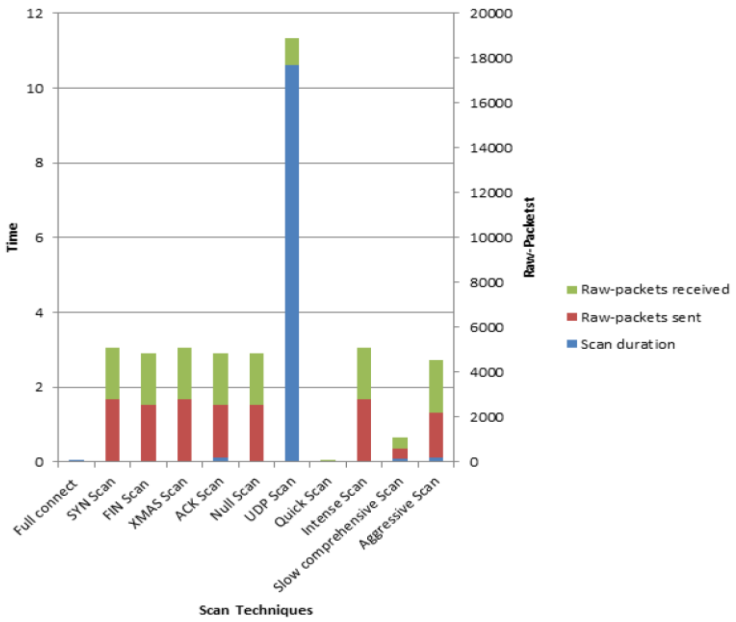


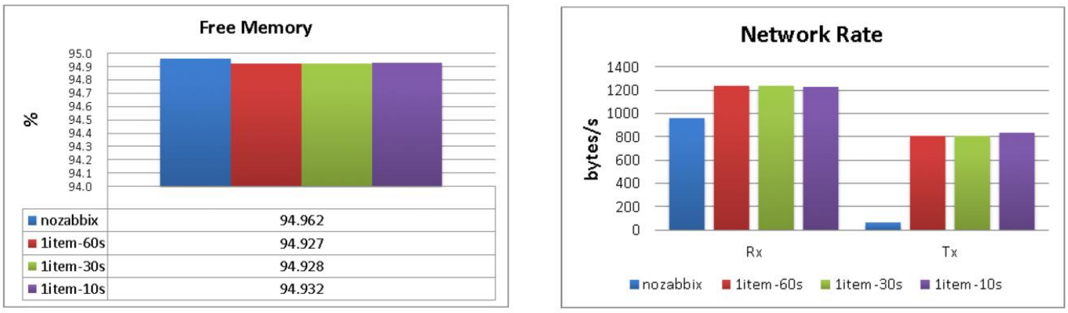
**Fig. 2 - Port scanning duration using Nmap [9]**

The performance of IIS 10.0 and Apache2 web server on the impact of TCP SYN flood attack has been done before. From the experimental results, it shows that IIS 10.0 has a better response time in a normal condition which is without any attack performed whereas Apache2 have a better response time performance with a TCP SYN attack performed [10]. Table 4 shows the response time performance for both IIS 10.0 and Apache2.

**Table 4 - Response time for IIS 10.0 and Apache2 web server [10]**

| Request | IIS 10.0 | | | | Apache2 | | | |
|---|---|---|---|---|---|---|---|---|
| | Without TCP SYN attack | | With TCP attack | | Without TCP SYN attack | | With TCP attack | |
| | 1st Server | 2nd server | 1st Server | 2nd server | 1st Server | 2nd server | 1st Server | 2nd server |
| 50000 | 4.09 | 4.1 | 12.74 | 12.15 | 4.09 | 4.1 | 12.74 | 12.15 |
| 100000 | 5.64 | 5.68 | 14.72 | 15.38 | 5.64 | 5.68 | 14.72 | 15.38 |
| 150000 | 7.16 | 7.75 | 16.65 | 16.92 | 7.16 | 7.75 | 16.65 | 16.92 |
| 200000 | 8.86 | 8.64 | 19.54 | 17.95 | 8.86 | 8.64 | 19.54 | 17.95 |
| 250000 | 16.48 | 16.8 | 20.05 | 19.22 | 10.41 | 10.07 | 20.05 | 19.22 |
| 300000 | 18.18 | 18.58 | 23.01 | 21.39 | 12.55 | 12.35 | 23.01 | 21.39 |

An extensive test to evaluate the impact of Zabbix clients on the system performance has been done before. The results show that when Zabbix has been connected to the monitored network device, the memory space and network rate of the device are decreased. However, in the worst cases, the memory space decreases only by 0.035% while the network rate increases by 800 bytes/s in output and by 300 bytes/s in input which is very acceptable in view of the foreseen use case [11]. Figure 3 shows the impact of Zabbix on the memory space and network rate on the monitored network device.



(a) Memory space (b) Network Rate

**Fig. 3 - The impact of Zabbix on monitored network device [11]**

## 3. Methodology

In this research, an experiment to evaluate the impact on a network device from each port scanning technique and tool is simulated. It is divided into 3 phases: design the proposed testbed, implement a port scanning attack on the proposed testbed and compare the impact on network performance.

## 3.1 Design Proposed Testbed

A testbed setup on a LAN network is designed for this research and it is fully run on VMware Workstation 17 Player and network devices are installed on Ubuntu 22.04 virtual machines guest OS for the purpose of this experiment is shown in Figure 4.

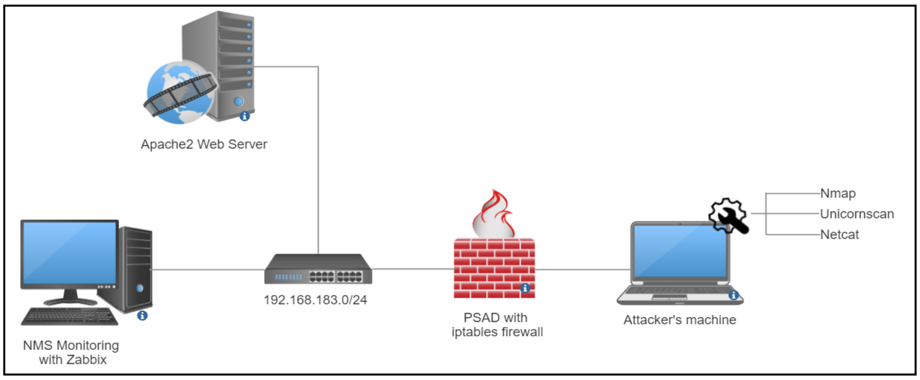


**Fig. 4 - Testbed setup**

The testbed setup consists of two virtual machines located in two different subnets. The right subnet is dedicated for attackers in which the machine is equipped with port scanning tools: Nmap, Unicornscan and Netcat. The left subnet is dedicated to the victim's network where a PC and an Apache2 web server are connected via LAN network. The PC will serve as a network monitoring device to evaluate the performance of the Apache2 web server.

## 3.2 Implement Port Scanning Attack

Three port scanning tools (Nmap, Unicornscan and Netcat) are used to simulate attacks toward the Apache2 web server using all the three port scanning techniques (TCP SYN, TCP Connect and UDP scan) so that, at the end of this research we will be able to decide which tool is the best to perform for which port scanning techniques. Table 5 shows the command line used to implement port scanning attack on Apache 2 web server (192.168.183.135) using Nmap, Unicornscan and Netcat for each port scanning technique.

**Table 5 - Port scanning implementation**

| Port Scanner Tools | Port Scanning Techniques | Command used |
|---|---|---|
| Nmap | TCP SYN scan | nmap -sS 192.168.183.135 |
| | TCP Connect scan | nmap -sT 192.168.183.135 |
| | UDP scan | nmap -sU 192.168.183.135 |
| Unicornscan | TCP SYN scan | unicorn -mT 192.168.183.135 |
| | TCP Connect scan | Not available |
| | UDP scan | unicorn -mU 192.168.183.135 |
| Netcat | TCP SYN scan | netcat -z 192.168.183.135 |
| | TCP Connect scan | Not available |
| | UDP scan | netcat -z -u 192.168.183.135 |

## 3.3 Compare Network Performance

In this research, Zabbix is used to monitor the performance of Apache2 web server. Zabbix is an open-source, real-time application, and network monitoring tool. It offers monitoring of thousands of metrics collected from physical machines or virtual machines and it offers a web-based management interface which is centralized through a database. Figure 5 below shows the architecture of Zabbix [12].

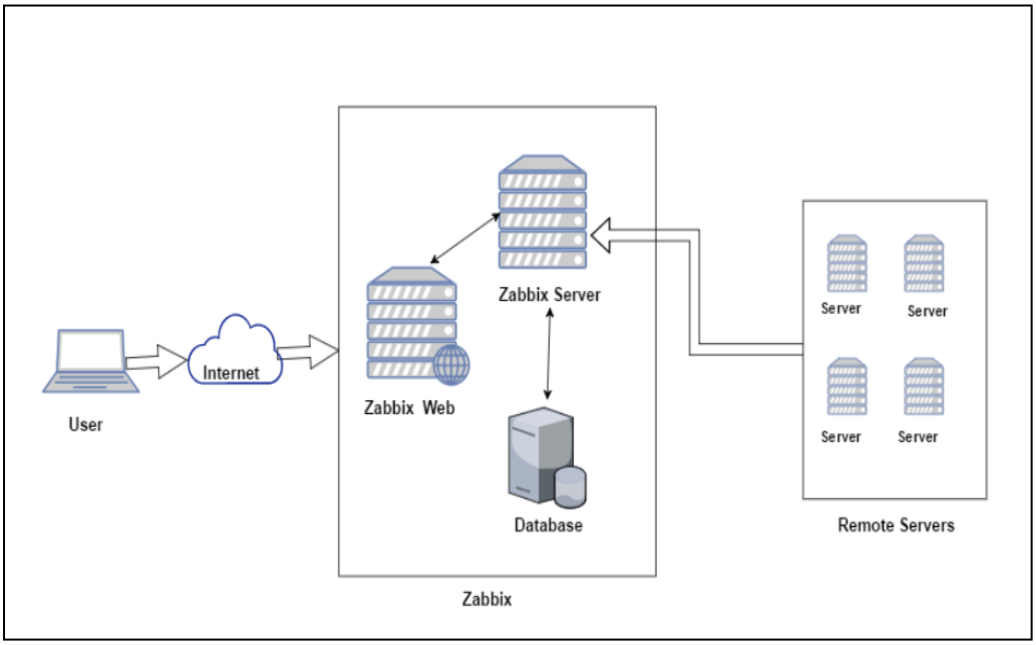


**Fig. 5 - Architecture of Zabbix**

Zabbix consists of three main major components: Zabbix server, database and Zabbix web [13]. The overview of the components is shown in Table 6.

**Table 6 - Overview of Zabbix components**

| Components | Description |
|---|---|
| Zabbix server | Zabbix server is the central component where all agents report availability and integrity information. It is the central repository where the data is being configured and stored. |
| Database | Database storage is where all the configuration as well as data collected by Zabbix is stored. |
| Zabbix web | Web interface is the platform used by user to have access to Zabbix. The interface is part of Zabbix server and mostly runs on the same physical machine as the Zabbix server. |

By default, there are three web scenarios item created for web monitoring on Zabbix server as shown in Table 7. From the web scenario, the web server speed and response time can be monitored as shown in Figure 6. The data collected from the Apache2 web server is kept in the database configured.

**Table 7 - Web scenario item created by default**

| Item | Description |
|---|---|
| Download speed for scenario | This item will collect information about the download speed (bytes per second) of the whole scenario |
| Response time | This item will collect information about the response in seconds. |
| Response code | This item will collect response codes of the step. |

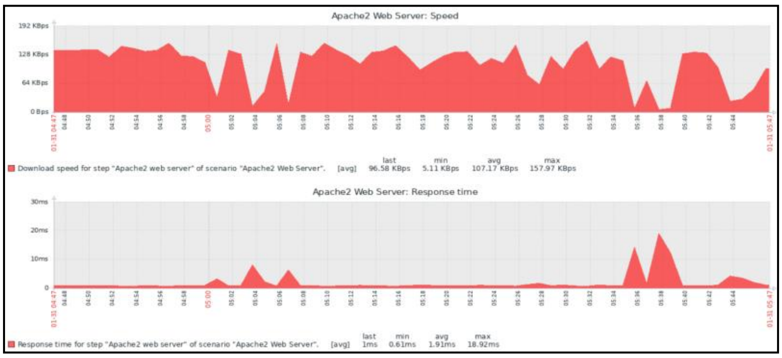
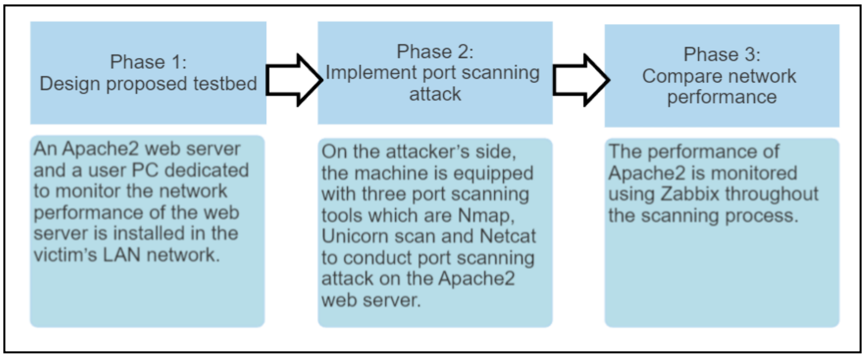


**Fig. 6 - Web monitoring from web scenario configured**

## 3.4 Research Framework

The research framework is divided into 3 phases: Design the proposed testbed, implement port scanning attack on the testbed designed and compare the impact on network performance. An Apache2 web server and a user PC dedicated to monitor the network performance of the web server is installed in the victim's LAN network. On the attacker's side, the machine is equipped with three port scanning tools which are Nmap, Unicorn scan and Netcat to conduct port scanning attack on the Apache2 web server. The performance of Apache2 is monitored using Zabbix. Figure 7 below shows the research framework used to summarize the research process.



An Apache2 web server and a user PC dedicated to monitor the network performance of the web server is installed in the victim's LAN network.

On the attacker's side, the machine is equipped with three port scanning tools which are Nmap, Unicorn scan and Netcat to conduct port scanning attack on the Apache2 web server.

The performance of Apache2 is monitored using Zabbix throughout the scanning process.

**Fig. 7 - Research framework**

## 4. Result and Discussions

The impact of port scanning techniques using different types of port scanning tools on the targeted host which is the Apache2 web server is evaluated and compared in different aspects and metrics.

### 4.1 Time Efficiency of Port Scanning Tools

The duration to complete each port scanning technique using different port scanning tools is recorded. The scan duration is the amount of time taken from when a port scanning request is made by the attacker to the time it takes for the response to get back to that attacker. The scan duration and packet bytes transmitted will affect the targeted host performance. Table 8 shows the scan duration and average packet bytes transmitted for each tool on port scanning techniques used.

**Table 8 - Scan duration for port scanning tools**

| Tool | Port Scanning Technique | Scan Duration (ms) | Average Packet Bytes Transmitted (bytes) |
|---|---|---|---|
| Nmap | TCP Connect | 27.74 | 148666 |
| | TCP SYN | 14.56 | 116058 |
| | UDP Scan | 688.63 | 89924 |
| Unicornscan | TCP Connect | Not available | Not available |
| | TCP SYN | 41.43 | 140862 |
| | UDP Scan | 187.80 | 201928 |
| Netcat | TCP Connect | Not available | Not available |
| | TCP SYN | 59.37 | 178820 |
| | UDP Scan | 350.99 | 177032 |

Based on Table 8, Nmap tool works well with all the port scanning techniques except UDP scan because it took the longest compared to Unicornscan and Netcat. The fastest scan for TCP SYN scan is with Nmap which is 64.85% faster than its closest competitor, Unicornscan and followed by Netcat, 75.48%. For TCP Connect scan which only Nmap can perform, it shows that the scan duration is slower compared to the TCP SYN scan as TCP SYN scan never complete the three-way handshake. The best tool for UDP scan is Unicornscan which is 46.49% faster than the speed of the second fastest tool which is Netcat and followed by Nmap, 72.73%.

### 4.2 The Impact of Port Scanning On Apache2 Web Server

The performance of the Apache2 web server is analysed with Zabbix after each port scanning activity is performed with multiple port scanning tools and the results are summarized in Table 9 below.

**Table 9 - Summary of result for 3 port scanning tools using 3 different port scanning techniques**

| Tool | Port Scanning Technique | Response Time (ms) |
|---|---|---|
| Nmap | TCP Connect | 0.85 |
| | TCP SYN | 0.69 |
| | UDP Scan | 10.84 |
| Unicornscan | TCP Connect | Not available |
| | TCP SYN | 0.79 |
| | UDP Scan | 1.68 |
| Netcat | TCP Connect | Not available |
| | TCP SYN | 0.95 |
| | UDP Scan | 1.37 |

Based on Table 9, port scanning using Nmap tool and UDP scan technique, has the lowest performance with average response time of 10.84ms. Nmap tool and TCP SYN scan technique have the best performance with an average response time of 0.69ms, which is 12.66% better than the average response time of the closest result using Unicornscan tool and 27.37% better than using Netcat.

The port scanning techniques were repeated for each tool and this time, each scanning is conducted 100 times to study the impact on a bigger scale of scanning towards the average response time of the Apache2 server. Table 10 shows the average response time for 100 scans.

**Table 10 - Summary of result for 3 port scanning tools using 3 different port scanning techniques for 100 scans**

| Tool | Port Scanning Technique | Average Response Time (ms) |
|---|---|---|
| Nmap | TCP Connect | 0.696 |
| | TCP SYN | 0.421 |
| | UDP Scan | 6.710 |
| Unicornscan | TCP Connect | Not available |
| | TCP SYN | 0.205 |
| | UDP Scan | 1.159 |
| Netcat | TCP Connect | Not available |
| | TCP SYN | 0.447 |
| | UDP Scan | 0.974 |

For 100 TCP Connect scans, Nmap shows a better result by having an average response time of 0.696ms (from Table 10), as compared to 0.85ms (from Table 9) for a single scan experiment which is 18.11% faster. For the TCP SYN scan, the average response time for 100 scans which is 0.421ms is 38.99% faster compared to the single scan in the previous experiment which is 0.69ms. For the UDP scan, the average response time for 100 scans is 6.710ms and it is 38.09% better than the single scan experiment which is 10.84ms.

For Unicornscan, the average response for 100 scans of TCP SYN scan is 0.205ms which is 74.05% better than in single scan which is 0.79ms and for UDP scan, for 100 scans, the average response time is 1.159ms and 31.01% better than in single scan which is 1.680ms.

For Netcat, the average response for 100 scans of TCP SYN scan is 0.447ms which is 52.95% better than in a single scan experiment which is 0.95ms. For 100 scans of UDP scan, the average response is 0.974ms which is 28.91% better than in a single scan experiment which is 1.37ms.

## 5. Conclusion

Throughout this research, a set of results has been obtained and has been discussed in detail. From the results obtained, it shows that the best tool to conduct port scanning using TCP SYN scan and TCP Connect scan techniques is Nmap. On the other hand, Unicornscan is the best tool for port scanning using UDP scan technique. The best port scanning technique is TCP SYN scan as it has the lowest response time and thus the least impact on the target host. TCP SYN scan is also well-known as the stealthiest port scanning technique. UDP scan also has the worst impact on the Apache2 web server as it took the longest response time.

## Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, for supporting this work.

## References

[1] Werner, G. Yang, S. and McConky, K. Leveraging Intra-Day Temporal Variations to Predict Daily Cyberattack Activity. 2018 IEEE International Conference on Intelligence and Security Informatics (ISI). Miami, Florida, USA. 2018. pp. 58-63.

[2] Messier, R. CEH v11: Certified Ethical Hacker. 11th Edition. Wiley. 2022. pp. 27-53.

[3] Y. J. Jia, Q. A. Chen, Y. Lin, C. Kong and Z. M. Mao. Open Doors for Bob and Mallory: Open Port Usage in Android Apps and Security Implications. 2017 IEEE European Symposium on Security and Privacy (EuroS&P). Paris, France. 2017. pp. 190-203.

[4] S. Lee, S. Y. Im, S. H. Shin, B. H. Roh and C. Lee. Implementation and vulnerability test of stealth port scanning attacks using ZMap of censys engine. 2016 International Conference on Information and Communication Technology Convergence (ICTC). Jeju, South Korea. 2016. pp. 681-683.

[5] ICANN (2019). The IANA Functions: An Introduction to the Internet Assigned Numbers Authority (IANA) Functions. Retrieved on June 12 2022, from https://www.iana.org/assignments/service-names-port-numbers

[6] Muraleedharan, N. Analysis of TCP flow data for traffic anomaly and scan detection. 2008 16th IEEE International Conference on Networks. New Delhi, India. 2008. pp. 1-4.

[7] Bauer, S. Evaluation of Port Scan and Port Scan Detection Tools. Bachelor’s Thesis. Technical University of Munich; 2015.

[8] Upadhya, A. and Srinivas, B. K. A Survey on different Port Scanning Methods and the Tools used to perform them. International Journal for Research. 2020. 8(5): 3018-3023.

[9] Mahdi, A. On Assessing the Impact of Ports Scanning on the Target Infrastructure. Ph.D. Thesis. University of Middlesex; 2018.

[10] Makwana, R. R. S. and Tomar, D. S. A Network Forensic Framework for Port Scan based on Efficient Packet Capturing. International Journal of Innovative Technology and Exploring Engineering (IJITEE). 2019. 8(12): 4632-4640.

[11] Telesca, A., Carena, F., Chapeland, S., Barroso, V. C., Costa, F., Denes, E., Divia, R., Fuchs, U., Grigore, A., Ionita, C., Delort, C., Simonetti, G., Soos, C., Vyvre, P. V., and Haller, B. V. System performance monitoring of the ALICE Data Acquisition System with Zabbix. 20th International Conference on Computing in High Energy and Nuclear Physics. 2014. 513(6): 1-7.

[12] Cloudthat (2017). Zabbix-A Simpler way of monitoring. Retrieved on November 17, 2020, from https://blog.cloudthat.com/zabbix/

[13] Nathan (2022). How to install ZABBIX 6.2 on Ubuntu 22. Retrieved on December 15, 2022, from https://www.layerstack.com/resources/tutorials/How-to-install-ZABBIX-on-Ubuntu22