Journal of Engineering and Sustainable Development
Vol. 29, No. 01, January 2025
https://jeasd.uomustansiriyah.edu.iq/index.php/jeasd
ISSN 2520-0917
Research Article
https://doi.org/10.31272/jeasd.2638

Characteristics of Port Scan Traffic: A Case Study Using Nmap

Zaid Al-Khazaali[1*], Ammar Al-Ghabban[2], Haneen Al-Musawi[3], Anwar Sabah[4], Noor Al Mahdi[5]

[1,2,3]Department of Construction and Projects, Mustansiriyah University, Iraq
[4]Department of Laser and Optoelectronics Engineering, University of Technology, Iraq
[5]Department of Computer Science and Electronic Engineering, University of Essex, UK

[*]Email: zaid_alkhazaali@uomustansiriyah.edu.iq

Article Info

Received	**30/04/2024**
Revised	**20/11/2024**
Accepted	**21/11/2024**

Abstract

Network ports, essential for communication, become susceptible to port scanning techniques employed by cybersecurity professionals, network administrators, and malicious hackers. The study digs into the specific characteristics of Nmap-generated port scan traffic, examining patterns, behaviors, and data relations throughout the packets. Also, researchers investigate the relationships between various port scan features and approaches to provide insightful information for developing more effective intrusion detection systems. The tool Nmap, which is widely employed for reconnaissance attacks in current network security, is the subject of this paper, and the Metasploit tool is also used to illustrate specific behavior and how it differs from the Nmap tool. The paper's contribution is summarized by introducing features like source ports, destination port distribution, statistics, and time-related attributes, which can be used as distinguishable features to detect the scan traffic. The term "Indicator of Scan" (IoS), as used by the authors, refers to a broad category that includes any useful indicators for scan detection. IoS can also be useful in determining which specific scanning tool is utilized in addition to scan detection.

Keywords: Cybersecurity; Intrusion detection; Nmap; Portscan; Reconnaissance; Wireshark

1. Introduction

Any digital system or device communicating with its counterparts inevitably has ports set aside to facilitate data transmission and reception. Ports are software abstractions used to help identify different communication channels. Ports are also used to identify individual applications running, similar to the way IP addresses are used to identify machines on networks [1].

Network administrators, cybersecurity experts, and immoral hackers use port scanning techniques to discover open ports on a computer or network. Port scanning covers a wide range of activities involving sending a stimulus to the Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) identifiers of specific services on specific computers [2], [3]. For network exploration and security auditing, the specialists use free and open-source tools called Network Mapper (Nmap), which is a free open-source common utility [2],[4]. Unfortunately, hackers and intruders also use this tool to gather information about the target they want to attack.

According to cyber kill chain methodology, which is a component of intelligence-driven defense for identifying and preventing malicious intrusion activities, cyberattacks can be divided into seven phases. They are (Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control, Actions on Objectives) respectively [5]-[7]. This paper aligns with the reconnaissance topic, the first phase of the cyber kill chain framework. An adversary gathers information about the target networks, websites, servers, systems, devices, etc., and performs scanning to identify open ports and services.

To detect port scanning activity on the network, the security expert needs to learn more about how it behaves and what aspects to consider to enhance the functionality of firewalls and intrusion detection systems. Nmap is, by far, the most popular port scanning tool for collecting information on remote hosts and systems [8],[9].

A set of traffic characteristics such as a source/destination IP address, ports, TCP flags, packet length, time to live (TTL), and protocols is called a signature [10]. Network traffic signatures are used to define the type of activity on a network classified

into two main categories depending on their behavior, as described below:

- Normal traffic signatures: These include normal network traffic and are defined based on an organization's normal traffic baseline.
- Attack Signatures: Traffic patterns that appear suspicious are generally treated as attack signatures.

This paper discusses the port scanning characteristics, where port scan attempts considered reconnaissance traffic consist of signatures indicating a try to scan the network for possible weaknesses. To analyze the network traffic, the authors have chosen the composite-signature-based analysis approach, which is one of four attack signature analysis techniques [9], and this technique is focused on analyzing a series of packets over a long period to detect composite attack signatures.

Nmap and Wireshark are the two main cybersecurity tools used in this study. As previously indicated, hackers can use Nmap to gather data and get ready to attack the system, or testers and auditors can utilize it as part of the evaluation process at the system security level by using its scripts and instructions. Wireshark is known as a packet sniffer tool that has the ability to capture and analyze traffic [11], [12]. The subsequent literature will outline cases in which authors have used similar tools for similar goals or describe how the authors select their features to use in their algorithms for scanning behavior detection.

Liao et al. addressed Nmap in [8], an infiltration reconnaissance attack tool in current network security. By examining the Nmap process, they identified specific characteristics that may serve as detection points and proposed and implemented detection rules for each scanning method. Researchers achieved a more accurate and with a lower false negative rate than the standard Suricata rule set (ET OPEN). The authors proposed the Comprehensive Nmap Detection Rules (CNDR). CNDR, designed to enhance precision and efficiency, removes Nmap's customizable fields and introduces rules for operating system scanning. The results indicate that CNDR achieves a 100% detection rate for normal Nmap scanning and a 91.7% detection accuracy for Nmap with IDS evasion on the dataset created for their work. The article depends on Context-based signature analysis.

Bagyalakshmi et al. emphasize the importance of packet analysis in [13]. According to the article, threats, malware, and attacks that may escape the notice of conventional security measures can be found through packet-level analysis. In particular, the research examines various sweep methods (e.g., Ping, TCP, and Null sweep) on widely used brain signal/image collections databases. The work involves scanning Nmap commands from a client to servers from the US, UK, and other countries. They included information on servers supporting Ping sweep and the outcomes of TCP sweeps on servers that do not support ICMP packets. Wireshark is used to illustrate TCP sweep results on specific databases.

In [14], Sharafaldin et al. introduced a new publicly available dataset that includes benign and various common attack network flows. The analysis involves extracting 80 traffic features using CICFlowMeter and evaluating the features using a random forest regressor to identify the best feature set for detecting specific attack categories. The evaluation identified the most relevant features for detecting port scanning attacks which are Initial Window Forward Bytes (Init Win F. Bytes), Bytes per Packet per second (B. Packet/s), and Push Flag count (PSH Flag), where the Nmap tool is used to generate port scanning attack to be examined.

2. Methodology

2.1. Environment

To simulate a real-world attack where an adversary initiates port scanning during reconnaissance to determine which services are running on the target system. The network protocol analyzer Wireshark and Nmap tools are installed in machine-1 (Zenmap can be used too, which is the graphical Nmap frontend and result viewer) while the virtual machine is deployed by using VMware Workstation 17 Pro and mentioned as machine-2. While Wireshark is in capture operation and the filter (host 192.168.15.1 and host 192.168.15.155) is applied as a capture filter to capture only the interested traffic between these two machines, Nmap from machine-1 (192.168.15.1) generates a scan toward machine-2 address (192.168.15.155). The collected traffic is saved as a CSV (Comma-Separated Values) file and inspected in Python using a Jupyter notebook.

The repository link is provided in [15] as part of the author's contribution to this significant field of research and scan datasets. Many studies use CICIDS2017, a network attack traffic dataset containing benign and malicious traffic[14], [16], but its limit is that there are no details about the scan attacks, such as which type of scan is used or which tool is used. The operating system of machine-1 is Windows 11, so Zenmap uses version 7.93, and the version of Wireshark is 4.0.10. In contrast, machine-2's operating system is Windows 7, the same version of Windows used in EC Council Certified Ethical Hacking (CEH) materials (version 11). Fig. 1 below shows the environment deployment.

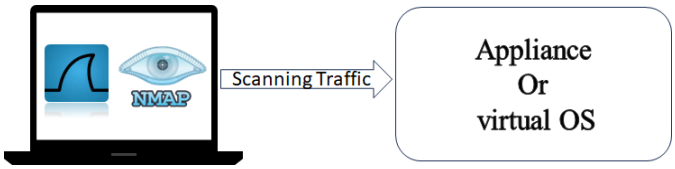


Figure 1. Environment Setup.

2.2 Utilities

Nmap tool scans the target system and detects the active services, while Wireshark captures the forward traffic from machine-1 towards machine-2 and backward traffic. Using the filter mentioned above, only the important packets are captured, and then the desired columns are added to the packet list pane; once the collected traffic has been saved as a CSV file, it can be analyzed. Python libraries such as Numpy, Pandas, and Matplotlib handle the analysis process.

Many port scanning techniques are available through the Nmap tool, as mentioned in the official website project (nmap.org) [17]. This paper deep dive into sS and sT scanning ('TCP SYN,'

which is the default scan and 'TCP connect' scan, respectively), other techniques like [1], [18], [19]:

- sX, sF, sN (TCP Xmas, FIN, and Null). These are special scans used to sneak firewalls to explore the system behind them
- sA (TCP ACK) helps understand whether firewall rules are stateful. Open and closed ports are not recognized by it.
- sW (TCP Window), also called Window scan, is like sA scan except it can detect open versus closed ports
- sM (TCP Maimon) is used to evade firewalls and works with fewer systems.

These techniques can be detected using context-based signatures. The analysis approach analyzes the packet's header and, specifically, the TCP flags, while the sS and sT techniques generate normal traffic that is hard to detect if it is a normal handshake, connection initiation, or adversary scan. Fig. 2 shows the difference between TCP connect and SYN scans for open ports. Nmap sends probes to the target machine during a TCP Connect scan. The scanner receives a SYN/ACK packet if the port is open, after which the scanner will send an ACK and use RST to close the connection. The SYN scans do not send an ACK packet; instead, an RST packet or no transmission will occur.

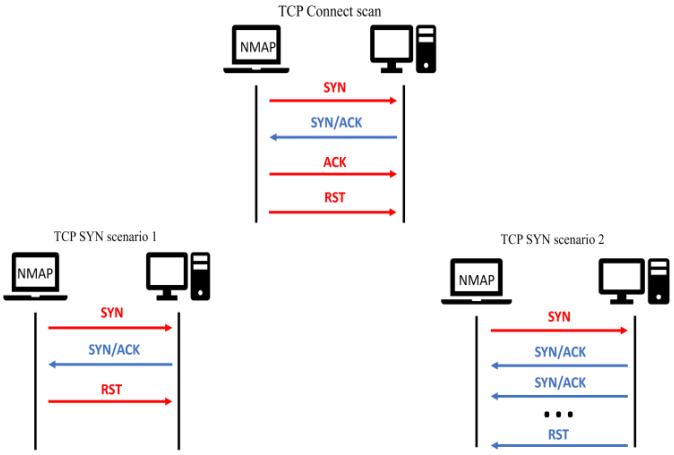


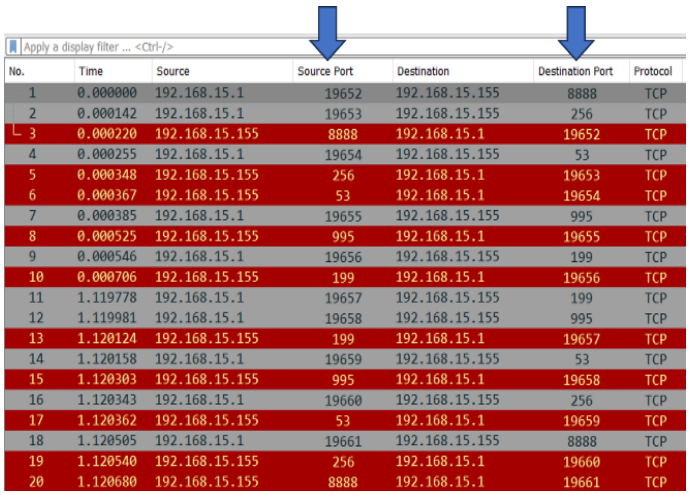
Figure 2. TCP port scan.

Important features that have been collected to classify the type of traffic will be mentioned in the results. The authors focus on features like:

- Destination port distribution: this feature illustrates the behavior of Nmap in choosing the default ports to be scanned and sending the Nmap props to it; this feature can help the intrusion detection systems (IDSs) and cybersecurity analysts to recognize which tools are used to run the scan inside the network and which commands are chosen to generate the port scan attack
- Source ports: This feature can help predict the type of scan and command used by Nmap to generate this scan, especially if the scan is running by script kiddies without modification.
- Size, duration, and number of packets: these features vary between papers due to research considerations; the size of generated traffic must be specified for forward, backward, or whole traffic. The number of packets is the same, and the attack duration is specified based on the attacker's perspective or the system itself. For attackers, the Nmap sends its props and waits for the first responses only, while the system keeps sending the responses until it reaches the threshold of time or number of packets sent without replying. This behavior exists in the TCP SYN scan, as illustrated in scenario 2 in Fig. 2.
- Statistical measures: This feature calculates the scan's statistical characteristics, such as the mean, standard deviation, and variance of destination or source ports chosen by a scan tool.

All these features can be considered an Indicator of Scan (IoS). IoS is used to determine the type of scan and even the tool of scan used for reconnaissance purposes.

When the scanning is running, the traffic between two machines will be displayed in the packet list pane of the Wireshark dashboard with the default columns, which are (No., Time, Source, Destination, Protocol, Length, and Info) and columns are not enough to analyze port scanning traffic so the source port and destination port will be added as columns in the packet list pane so that when the captured file saved as CSV the ports (source and destination) can be investigated to extract recognizable features from it, the authors retain the ability to further explore the aforementioned features. The Wireshark dashboard is shown in Fig. 3 after adding the two new columns.



No.	Time	Source	Source Port	Destination	Destination Port	Protocol
1	0.000000	192.168.15.1	19652	192.168.15.155	8888	TCP
2	0.000142	192.168.15.1	19653	192.168.15.155	256	TCP
3	0.000220	192.168.15.155	8888	192.168.15.1	19652	TCP
4	0.000255	192.168.15.1	19654	192.168.15.155	53	TCP
5	0.000348	192.168.15.155	256	192.168.15.1	19653	TCP
6	0.000367	192.168.15.155	53	192.168.15.1	19654	TCP
7	0.000385	192.168.15.1	19655	192.168.15.155	995	TCP
8	0.000525	192.168.15.155	995	192.168.15.1	19655	TCP
9	0.000546	192.168.15.1	19656	192.168.15.155	199	TCP
10	0.000706	192.168.15.155	199	192.168.15.1	19656	TCP
11	1.119778	192.168.15.1	19657	192.168.15.155	199	TCP
12	1.119981	192.168.15.1	19658	192.168.15.155	995	TCP
13	1.120124	192.168.15.155	199	192.168.15.1	19657	TCP
14	1.120158	192.168.15.1	19659	192.168.15.155	53	TCP
15	1.120303	192.168.15.155	995	192.168.15.1	19658	TCP
16	1.120343	192.168.15.1	19660	192.168.15.155	256	TCP
17	1.120362	192.168.15.155	53	192.168.15.1	19659	TCP
18	1.120505	192.168.15.1	19661	192.168.15.155	8888	TCP
19	1.120540	192.168.15.155	256	192.168.15.1	19660	TCP
20	1.120680	192.168.15.155	8888	192.168.15.1	19661	TCP

Figure 2. The source port and destination port are in the Wireshark packet list pane.

2.3 Commands

The most important commands that are used to generate scanning toward machine 2 are:

- nmap 192.168.15.155 -sS -F -v
- nmap 192.168.15.155 -sT -F -v
- nmap 192.168.15.155 -sS -v
- nmap 192.168.15.155 -sT -v

- nmap 192.168.15.155 -sS -v --top-ports 5000
- nmap 192.168.15.155 -sT -v --top-ports 5000
- nmap 192.168.15.155 -sS -p- -v
- nmap 192.168.15.155 -sT -p- -v

The switch -sS stands for SYN scan, which is the default scan, and the -F switch means the most 100 ports, while the -v switch is used to enable the verbose mode to track the scan and see the results. The switch -sT represents the TCP Connect scan, and --top-ports 5000 specifies the top 5000 ports in the Nmap database, while -p- means scans all ports. To scan a specific port, for example, port 445, the switch -p 445 is used to send the probe to only this port. In the result section the authors describe the destination port distribution, so the Nmap database will be understandable for all options (--top-ports, -F, or default, which is the most 1000 important ports), and the selection of the most important words can be used to determine which tool is used for scan cause. The destination ports the probes will send to it are chosen randomly by default unless the option -r is added to the instruction to select the destination ports in ascending order.

In contrast, other tools, such as Metasploit, which is considered an entire framework, provide the infrastructure needed for penetration testing. On identifying flaws within information security systems [20], [21], the port sequence is arranged in ascending order by default in many port scanning auxiliary tools. In real-world scenarios, there are best practice techniques for using Nmap in the scanning phase:

1. When the goal is to determine which hosts are online, the option (-sn) is used to skip the port scan.
2. Always limit the number of ports needed to scan using (-F, default, --top-ports, or create a customized file containing specific ports).
3. Skip advanced scan types (-sC, -sV, -O, --traceroute, and -A) It causes Nmap to do OS detection, version detection, script scanning (NSE), and traceroute as well as the default port scan.
4. When there is a confidence of being the target system is online, use (-Pn) to skip the host discovery phase.
5. Turn off the DNS resolution feature when the IP address of the hostname is already known by using (-n).
6. Don’t combine TCP and UDP port scans even though Nmap supports doing so with options -sSU.

Most of these recommendations are considered in this research, and some are not because the lab environment is not necessary for them.

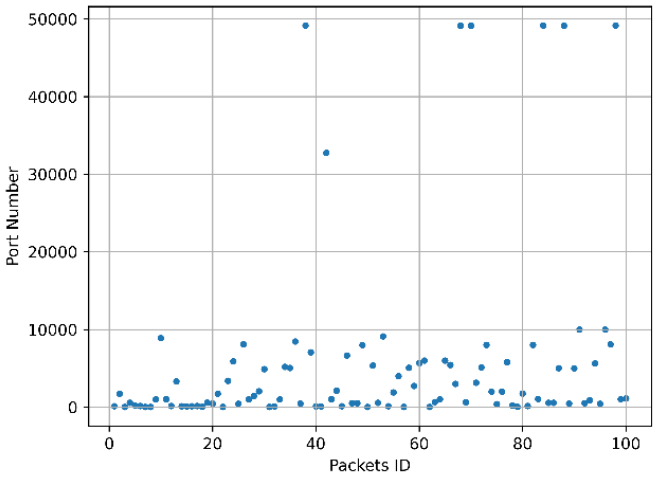
3. Experiments and Results

In this section, the authors investigate many features that are considered the footprint of the Nmap tool. The section is divided into three parts: port distribution (both source and distention ports), size, duration, and count of packets in scanning traffic, and statistical measures.

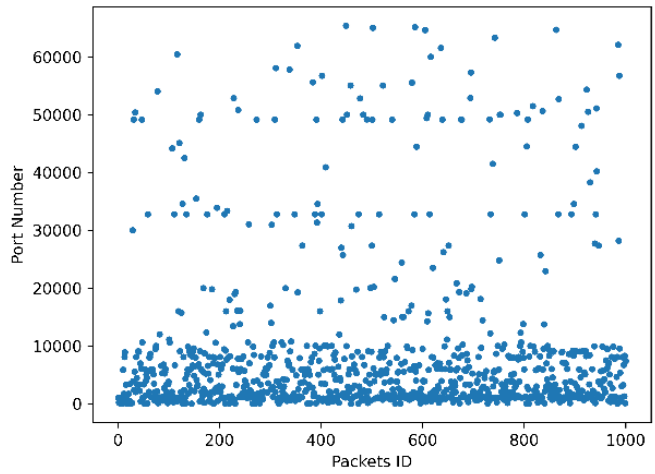
3.1 Port Distribution

One of the most crucial aspects is the port distribution. Certain scanning tools use sorted ports; others, like Nmap, select random ports from specific databases, yet others adopt an ascending pattern. The Nmap tool's distribution of the most significant destination ports is displayed in Fig. 4.

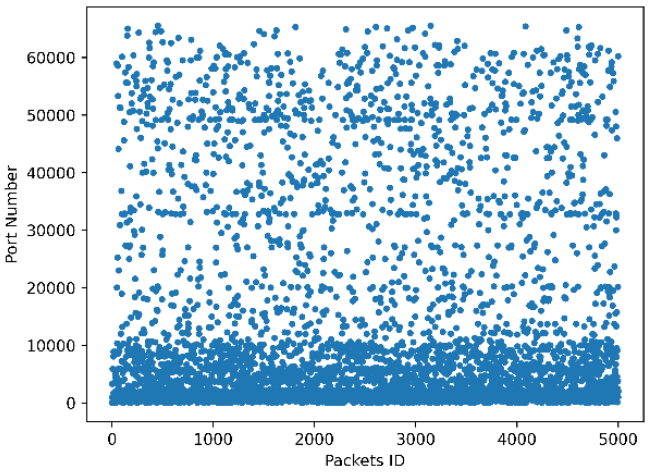
The figure shows that each packet has a different port number, and it is not arranged in an ascending or descending way like the scan pattern of the Metasploit tool. Also, it indicates that of the top 5000 ports scanned, the most important ports are found between 0 and 10,000.



(a) Distribution of the top 100 ports in Nmap



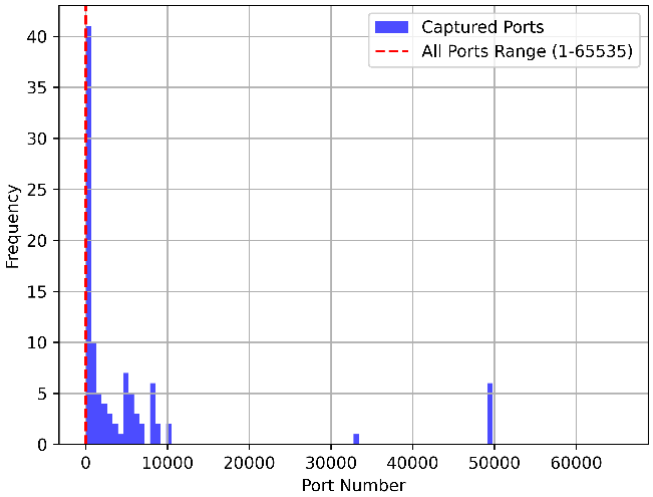
(b) Distribution of the top 1000 ports in Nmap



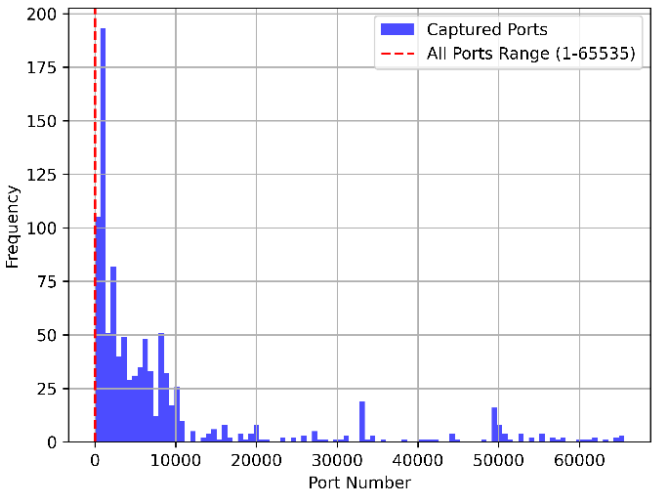
(c) Distribution of the top 5000 ports in Nmap

Figure 4. Distribution of top destination ports.

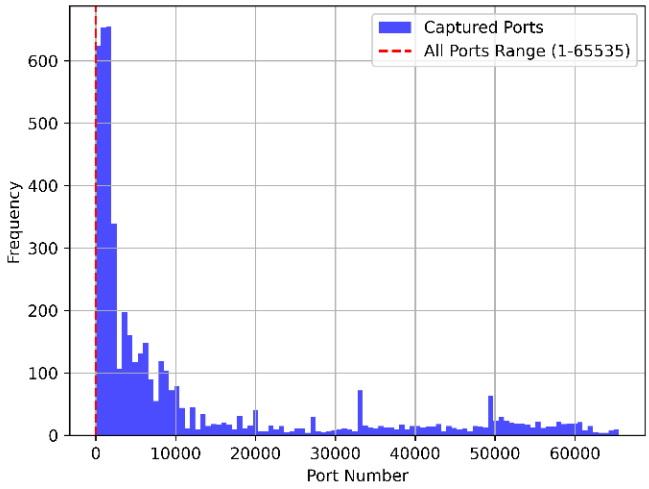
The histograms of 100, 1000, and 5000 are the most important ports displayed in Fig. 5.



(a) Histogram of top 100 distribution ports



(b) Histogram of top 1000 distribution ports



(c) Histogram of top 5000 distribution ports

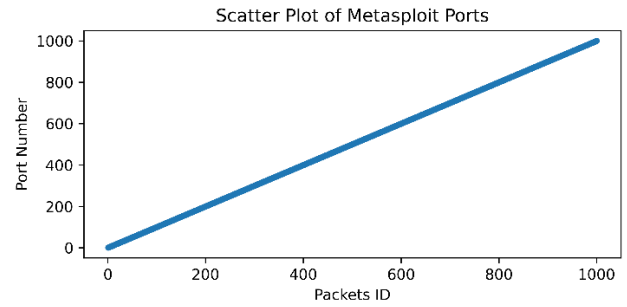
Figure 5. Histogram of Top distribution ports in Nmap.

The source ports of machine-1, which is the Nmap machine, are chosen randomly, but the count is changed based on the type of scan in Nmap; for the SYN scan and TCP Connect scan, the source port number is illustrated in Table. 1

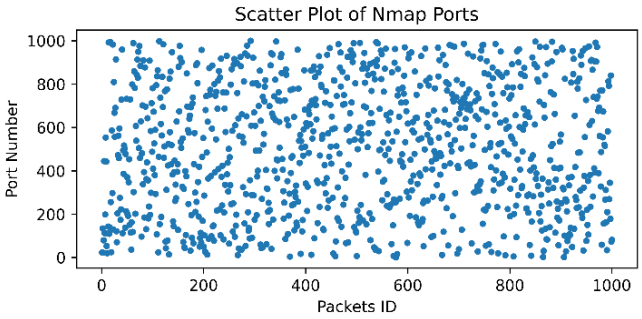
Table 1. Count of source ports in Nmap

Command	Number of source ports
nmap -sS -v -F	1
nmap -sS -v	2
nmap -sS -v --top-ports 5000	2
nmap -sT -v -F	293
nmap -sT -v	2020
nmap -sT -v --top-ports 5000	15229

As a result, the source ports can be considered footprints for the type of scan if it is an SYN or Connect scan within the Nmap tool. The count of source ports in -sT scan varies based on the response of machine-2 and the time that Nmap waits until sending another probe. Still, each destination port must connect to a unique source port, making the source port number equal to or greater than the destination ports. In -sS, the count of source ports varies between 1 and 2 ports, which are used to send probes to all destination ports. These two features, source, and destination port distribution, can also effectively determine which tools are used in the scan. The authors experiment with the characteristics of the Metasploit framework with SYN scan by using its integrated module (auxiliary /scanner/portscan/syn). The destination ports are set between 1 to 1000, and the behavior of ports selection is an ascending pattern from 1 to 1000, while in Nmap, the behavior for the same range of ports (-p1-1000) is random selection as shown in Fig. 6.



(a) Metasploit destination ports distribution



(b) Nmap destination ports distribution

Figure 6. Metasploit vs Nmap destination ports distribution**.**

3.2. Size, Duration and Number of Packets

When executing an Nmap command, time factors set by default determine many things like TCP scan delay, UDP scan delay, timeout, minimum rate of packets sent per second, and maximum rate of packets sent per second…etc. The authors in this work use the "Normal" time template, which is represented by the option (-T3) [1]. In this part, the authors evaluate the size, duration, and number of packets sent per scan. Table. 2 shows the approximate information for -sT and -sS scans.

The empirical results from Wireshark measurements are contained in the Table. The Nmap tool also gives results, and to illustrate the difference, Nmap gives the following result after -sS scan:

- Nmap done: 1 IP address (1 host up) scanned in 6.95 seconds
- Raw packets sent: 1062 (46.712KB) | Rcvd: 1001 (40.064KB)

By comparing the results of Nmap and the first row in the Wireshark table, we can see that Nmap needs about 7 seconds to determine which ports are open and which are closed, while machine-2 has not yet responded to all packets.

Table 2. Size and other traffic characteristics for Nmap scan by using Wireshark

Scan type	No. destination ports	Total Traffic	Total Packets	TX Traffic	RX Traffic	TX Packets	RX Packets	Duration
-sS	1000	117 kB	2093	62 kB	56 kB	1061	1032	22.3910 seconds
-sT	1000	247 kB	4116	137 kB	110 kB	2096	2020	44.2793 seconds
-sS	All	8 MB	135199	4 MB	4 MB	69635	65564	66.7 seconds
-sT	All	16 MB	270856	9 MB	7 MB	137620	133236	2858 seconds

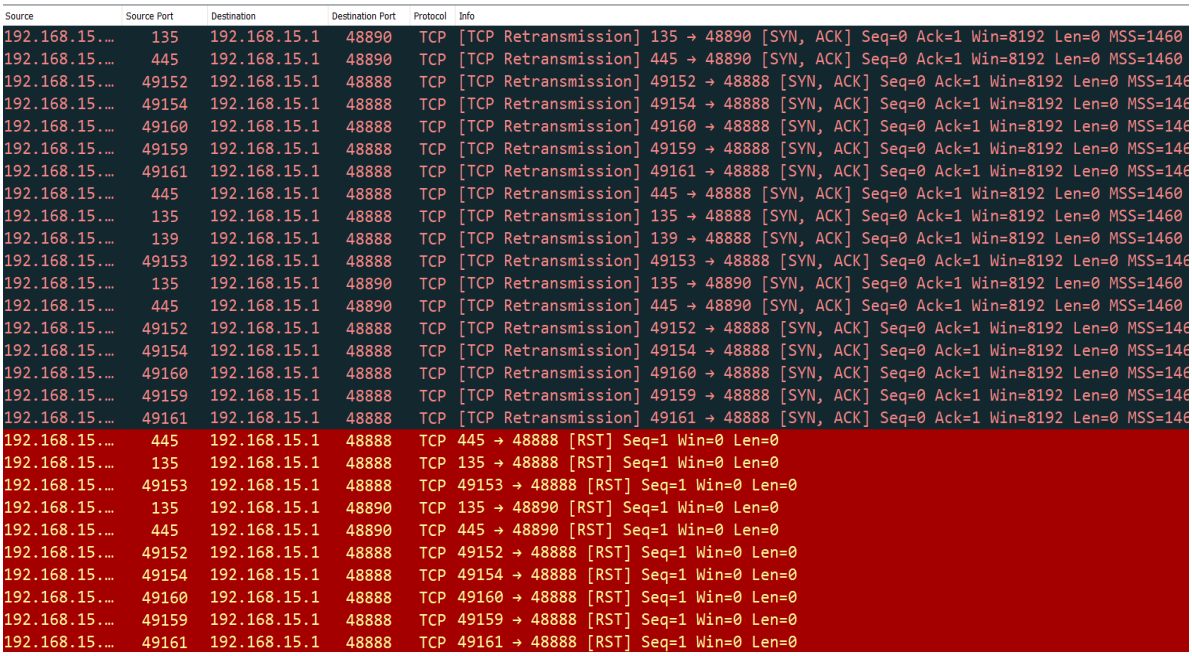


Figure 7. Open ports in captured traffic by Wireshark

Completely because it is still sending SYN-ACK packets to machine-1 from the open ports without any answer. After a number of packets, it will be considered that the connection has ended, and the RST will be sent. Fig. 7 shows that in the Wireshark dashboard, these are the last packets sent by open ports to machine-1 that runs Nmap, which already knows which port is open and does not need to reply. The whole time of the scan is not 7 seconds. The whole time represents all traffic that is captured due to scan operation, and it is about 23 seconds; some systems or firewalls do not send back SYN-ACK packets many times, and it will not take all this time.

The number of packets is approximately the same, 2093 in Wireshark and 2063 in Nmap, it depends on open ports and the timeout of probes, where less timeout means many packets will be sent. If the reply is delayed, the probes will be sent two times or more. Fig. 8 below shows the pattern of scan traffic in Metasploit and Nmap using Wireshark. It is important to mention that this behavior is due to the delay time of Metasploit being bigger than the normal time in Nmap and the rate of packets per second being lower. The red highlighted packets represent the RST flag from the target machine, and the gray represents the SYN flag from the Nmap machine; the default behavior of Metasploit scan is to send the packet and wait more time than Nmap so that the target machine can reply on the SYN packet before the Metasploit send the next probe. Metasploit machine with IP address (192.168.15.128) while the target is as mentioned before with the IP address (192.168.15.155). As shown in the figure, the Nmap traffic indicates that Nmap sends more probes during a specific period than Metasploit. The gray packets shown in (b) are continuously displayed on the Wireshark dashboard before receiving replies from the target machine.

3.3 Statistical Measures

The randomness selection of destination ports in Nmap can be detected by depending on statistical measures of some features in traffic, so if a machine runs a port scanning tool in the network, the behavior of the scan can be detected by taking a lot of targeted destination ports and compare it with the statistical database that will be created. Table 3 shows the TCP SYN scan for the top 100,1000, and 5000 ports and its statistical measures. The other types of scans, like TCP Connect, FIN, Xmas, etc., have the same statistics measures because Nmap depends on a specific port's database to select the top important ports to test if it's open.

Table 3. TCP SYN scan statistical measures

Statistics	Ports		
	Top 100 ports	**Top 1000 ports**	**Top 5000 ports**
Mean	5657.1	8634.848	12118.3094
Median	1069.5	3720	3798.5
Mode	7	1	1
Standard deviation	11794.444	13448.483	17368.530
Lower port	7	1	1
Higher Port	49157	65389	65514

No.	Time	Source	Source Port	Destination	Destination Port	Protocol	Info
18	2.502275	192.168.15.155	7	192.168.15.128	50029	TCP	7 → 50029 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
19	3.005906	192.168.15.128	43240	192.168.15.155	8	TCP	43240 → 8 [SYN] Seq=0 Win=3072 Len=0
20	3.006164	192.168.15.155	8	192.168.15.128	43240	TCP	8 → 43240 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
21	3.009084	192.168.15.128	4373	192.168.15.155	9	TCP	4373 → 9 [SYN] Seq=0 Win=3072 Len=0
22	3.009332	192.168.15.155	9	192.168.15.128	4373	TCP	9 → 4373 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
23	3.512060	192.168.15.128	62964	192.168.15.155	10	TCP	62964 → 10 [SYN] Seq=0 Win=3072 Len=0
24	3.512335	192.168.15.155	10	192.168.15.128	62964	TCP	10 → 62964 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
25	3.515582	192.168.15.128	51384	192.168.15.155	11	TCP	51384 → 11 [SYN] Seq=0 Win=3072 Len=0
26	3.515940	192.168.15.155	11	192.168.15.128	51384	TCP	11 → 51384 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
27	4.019765	192.168.15.128	21590	192.168.15.155	12	TCP	21590 → 12 [SYN] Seq=0 Win=3072 Len=0
28	4.020187	192.168.15.155	12	192.168.15.128	21590	TCP	12 → 21590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
29	4.523238	192.168.15.128	59747	192.168.15.155	13	TCP	59747 → 13 [SYN] Seq=0 Win=3072 Len=0
30	4.523606	192.168.15.155	13	192.168.15.128	59747	TCP	13 → 59747 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31	4.528668	192.168.15.128	41437	192.168.15.155	14	TCP	41437 → 14 [SYN] Seq=0 Win=3072 Len=0
32	4.528883	192.168.15.155	14	192.168.15.128	41437	TCP	14 → 41437 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
33	5.032354	192.168.15.128	47034	192.168.15.155	15	TCP	47034 → 15 [SYN] Seq=0 Win=3072 Len=0
34	5.032969	192.168.15.155	15	192.168.15.128	47034	TCP	15 → 47034 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35	5.035426	192.168.15.128	10355	192.168.15.155	16	TCP	10355 → 16 [SYN] Seq=0 Win=3072 Len=0
36	5.035655	192.168.15.155	16	192.168.15.128	10355	TCP	16 → 10355 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
37	5.538270	192.168.15.128	20110	192.168.15.155	17	TCP	20110 → 17 [SYN] Seq=0 Win=3072 Len=0
38	5.538462	192.168.15.155	17	192.168.15.128	20110	TCP	17 → 20110 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
39	5.542881	192.168.15.128	58818	192.168.15.155	18	TCP	58818 → 18 [SYN] Seq=0 Win=3072 Len=0
40	5.543152	192.168.15.155	18	192.168.15.128	58818	TCP	18 → 58818 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
41	6.046841	192.168.15.128	54145	192.168.15.155	19	TCP	54145 → 19 [SYN] Seq=0 Win=3072 Len=0
42	6.047091	192.168.15.155	19	192.168.15.128	54145	TCP	19 → 54145 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
43	6.550476	192.168.15.128	46777	192.168.15.155	20	TCP	46777 → 20 [SYN] Seq=0 Win=3072 Len=0
44	6.550650	192.168.15.155	20	192.168.15.128	46777	TCP	20 → 46777 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
45	6.555115	192.168.15.128	61572	192.168.15.155	21	TCP	61572 → 21 [SYN] Seq=0 Win=3072 Len=0
46	6.555375	192.168.15.155	21	192.168.15.128	61572	TCP	21 → 61572 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
47	7.058715	192.168.15.128	3046	192.168.15.155	22	TCP	3046 → 22 [SYN] Seq=0 Win=3072 Len=0
48	7.058963	192.168.15.155	22	192.168.15.128	3046	TCP	22 → 3046 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
49	7.062035	192.168.15.128	43818	192.168.15.155	23	TCP	43818 → 23 [SYN] Seq=0 Win=3072 Len=0
50	7.062286	192.168.15.155	23	192.168.15.128	43818	TCP	23 → 43818 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
51	7.565005	192.168.15.128	31988	192.168.15.155	24	TCP	31988 → 24 [SYN] Seq=0 Win=3072 Len=0
52	7.565214	192.168.15.155	24	192.168.15.128	31988	TCP	24 → 31988 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

(a) Metasploit pattern

No.	Time	Source	Source Port	Destination	Destination Port	Protocol	Info
44	0.441097	192.168.15.1	37549	192.168.15.155	668	TCP	37549 → 668 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
45	0.441155	192.168.15.1	37549	192.168.15.155	558	TCP	37549 → 558 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
46	0.441177	192.168.15.1	37549	192.168.15.155	691	TCP	37549 → 691 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
47	0.441196	192.168.15.1	37549	192.168.15.155	915	TCP	37549 → 915 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
48	0.441219	192.168.15.1	37549	192.168.15.155	122	TCP	37549 → 122 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
49	0.441260	192.168.15.1	37549	192.168.15.155	71	TCP	37549 → 71 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
50	0.441279	192.168.15.1	37549	192.168.15.155	204	TCP	37549 → 204 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
51	0.441298	192.168.15.1	37549	192.168.15.155	473	TCP	37549 → 473 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
52	0.441318	192.168.15.1	37549	192.168.15.155	563	TCP	37549 → 563 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
53	0.441338	192.168.15.1	37549	192.168.15.155	459	TCP	37549 → 459 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
54	0.441364	192.168.15.1	37549	192.168.15.155	596	TCP	37549 → 596 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
55	0.441383	192.168.15.1	37549	192.168.15.155	276	TCP	37549 → 276 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
56	0.441401	192.168.15.1	37549	192.168.15.155	174	TCP	37549 → 174 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
57	0.441421	192.168.15.1	37549	192.168.15.155	223	TCP	37549 → 223 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
58	0.441448	192.168.15.1	37549	192.168.15.155	729	TCP	37549 → 729 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
59	0.441467	192.168.15.1	37549	192.168.15.155	960	TCP	37549 → 960 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
60	0.441618	192.168.15.155	668	192.168.15.1	37549	TCP	668 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
61	0.441665	192.168.15.155	558	192.168.15.1	37549	TCP	558 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
62	0.441691	192.168.15.155	691	192.168.15.1	37549	TCP	691 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
63	0.441716	192.168.15.155	915	192.168.15.1	37549	TCP	915 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
64	0.441743	192.168.15.155	122	192.168.15.1	37549	TCP	122 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
65	0.441769	192.168.15.155	71	192.168.15.1	37549	TCP	71 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
66	0.441809	192.168.15.155	204	192.168.15.1	37549	TCP	204 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
67	0.441839	192.168.15.155	473	192.168.15.1	37549	TCP	473 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
68	0.441874	192.168.15.155	563	192.168.15.1	37549	TCP	563 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
69	0.441904	192.168.15.155	459	192.168.15.1	37549	TCP	459 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
70	0.441932	192.168.15.155	596	192.168.15.1	37549	TCP	596 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
71	0.441961	192.168.15.155	276	192.168.15.1	37549	TCP	276 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
72	0.442046	192.168.15.155	174	192.168.15.1	37549	TCP	174 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
73	0.442083	192.168.15.155	223	192.168.15.1	37549	TCP	223 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
74	0.442111	192.168.15.155	729	192.168.15.1	37549	TCP	729 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
75	0.442152	192.168.15.155	960	192.168.15.1	37549	TCP	960 → 37549 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
76	0.456447	192.168.15.1	37549	192.168.15.155	112	TCP	37549 → 112 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
77	0.456520	192.168.15.1	37549	192.168.15.155	495	TCP	37549 → 495 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
78	0.456542	192.168.15.1	37549	192.168.15.155	386	TCP	37549 → 386 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

(b) Nmap Pattern

Figure 8. A pattern of captured traffic generated by Nmap and Metasploit.

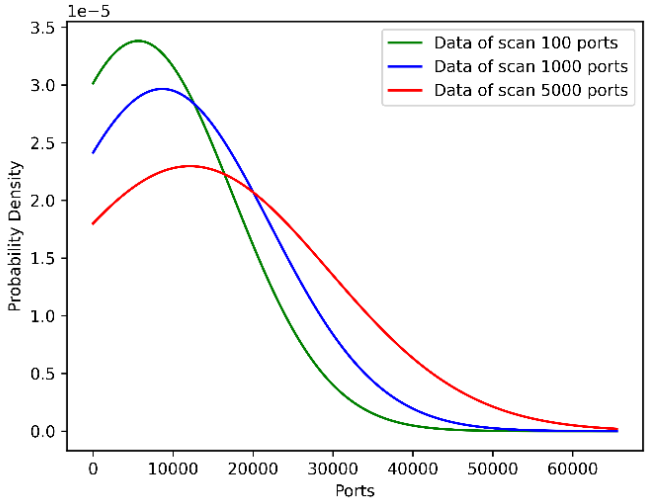
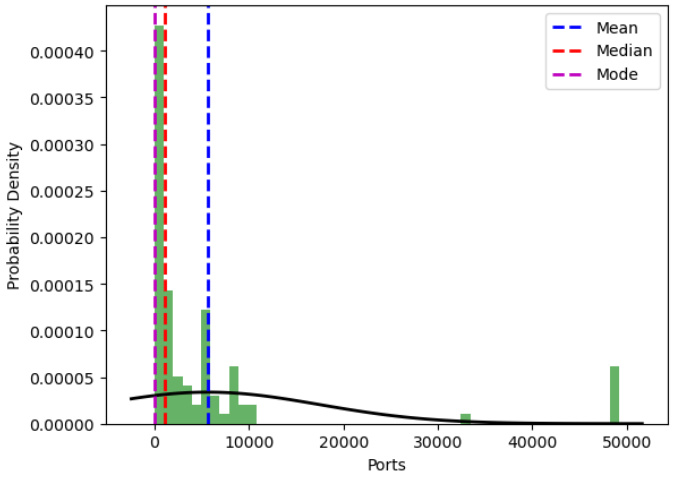
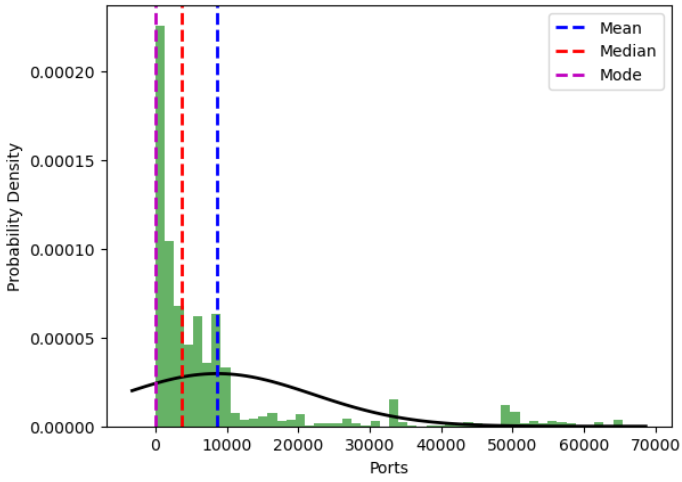


Figure 9. The probability density function for different scan types.

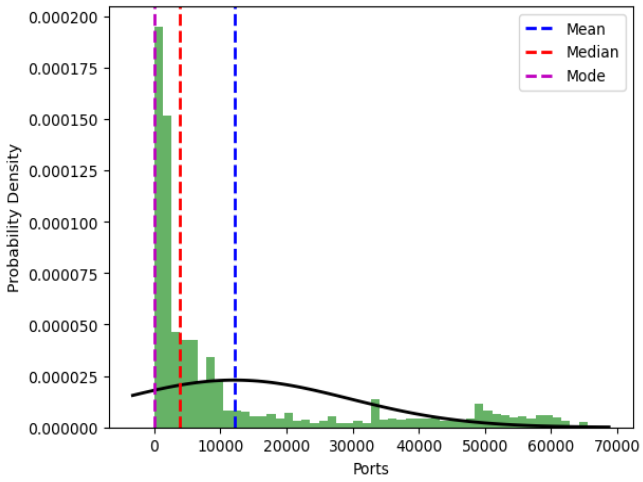
Fig. 9 shows the probability density function (PDF) for different scan types. The statistics of 100, 200, 1000, …etc. Top ports in Nmap can be calculated and stored in the database, so when the attacker initiates the scan on a specific number of top ports, this feature can be used as an indicator of the Nmap scan because the destination port statistics match the database. Fig. 10 shows the distribution ports with statistics information that can be used as recognizable features. The same thing can be done to the PDF of top ports.



(a) PDF of top 100 ports



(b) PDF of top 1000 port



(c) PDF of top 5000 ports

Figure 10. The PDF of 100,1000 and 5000 top ports with its statistics measures.

The green bars represent the histogram of each particular range of ports, where the ports from 0 to 65535 are divided into 50 bins or groups. As shown in Fig. 10, the most important destination ports or the prioritized ports for the Nmap tool are located between the mode and median of the distribution, and the count of significant ports decreases after the mean of the distribution.

4. Conclusion and Future Works

The paper's primary topic is Nmap, a tool for reconnaissance attacks utilized in current network security. Intrusion detection systems (IDS) and machine learning engineers can use Indicators of Scan (IoS) mentioned in this paper, such as count of source ports, destination port distribution, destination port statistics, and time-related features like scan duration, traffic generation size, and the total number of packets created from both the attacker and the target perspectives to build their detection model. Experiments indicate that the attributes are useful for scan detection and identifying the tool used during the scan process. The number of source ports a scan tool uses to initiate the scan against a specific number of ports can be considered a signature of using SYN scan from the Nmap tool, not Metasploit or another scanning tool, as shown in Table 1. The ascending destination port behavior is a Metasploit signature, not Nmap. The Nmap can be detected using statistical measures of destination ports as described in the results. For future work, a scan detection system will be built based on the mentioned features in this paper, and a comparison can be provided with recent related contributions in this field to indicate the importance of the introduced features of this work. Experiments will be made on more widely used targets with OS windows 10 and 11.

Acknowledgment

We gratefully acknowledge the University of Mustansiriyah's invaluable support. The university's resources and assistance have been essential to the success of this research.

Conflict of interest

The authors confirm that the publication of this article causes no conflict of interest.

Authors Contributions

Zaid Al-khazaali (Corresponding author) developed the study methodology, conducted the formal analysis and investigation, wrote the first manuscript, and participated in the funding acquisition.

Ammar Al-Ghabban and Haneen Al-Mousawi deployed the environment and generated the data for Nmap port scanning to work with it, and they participated in the funding acquisition.

Anwar Sabah and Noor Al Mahdi worked with Metasploit modules, generated Metasploit SYN scan traffic, and participated in funding acquisition.

References

[1] G. F. Lyon, *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Sunnyvale, CA, USA: Insecure, 2009. ISBN: 0979958717. url: https://nmap.org/book

[2] P. Calderon, *Nmap: Network Exploration and Security Auditing Cookbook*, 2nd Edition. Birmingham: Packt, 2017. ISBN: 1786467453. URL: https://www.packtpub.com/en-br/product/nmap-network-exploration-and-security-auditing-cookbook-9781786467454

[3] M. I. Kareem, M. Jawad Kadhim Abood, and K. Ibrahim, "Machine learning-based PortScan attacks detection using OneR classifier," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 6, pp. 3690–3696, Dec. 2023, doi: https://doi.org/10.11591/eei.v12i6.4142.

[4] W. M. Eid, S. Atawneh, and M. Al-Akhras, "Framework for Cybersecurity Centers to Mass Scan Networks," *Intelligent Automation & Soft Computing*, vol. 26, no. 4, pp. 1319–1334, 2020, doi: https://doi.org/10.32604/iasc.2020.013678.

[5] D. Kiwia, A. Dehghantanha, K.-K. R. Choo, and J. Slaughter, "A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence," *J Comput Sci*, vol. 27, pp. 394–409, Jul. 2018, doi: https://doi.org/10.1016/j.jocs.2017.10.020.

[6] M. Zaki Abdullah, A. Kalid Jassim, F. Noori Hummadi, and M. Majid M. Al Khalidy, "New Strategies for Improving Network Security Against Cyber Attack Based On Intelligent Algorithms*", J. eng. sustain. dev.*, vol. 28, no. 3, pp. 342–354, May 2024, doi: https:/doi.org/10.31272/jeasd.28.3.4.

[7] A. Villalon-Huerta, H. M. Gisbert, and I. Ripoll-Ripoll, "SOC Critical Path: A Defensive Kill Chain Model," *IEEE Access*, vol. 10, pp. 13570–13581, 2022, doi: https://doi.org/10.1109/ACCESS.2022.3145029.

[8] S., Liao, C., Zhou, Y., Zhao, Z., Zhang, C., Zhang, Y., Gao, and G., Zhong, "A Comprehensive Detection Approach of Nmap: Principles, Rules and Experiments," in *Proceedings - 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 64–71. doi https://doi.org/10.1109/CyberC49757.2020.00020.

[9] M. Aamir, S. S. H. Rizvi, M. A. Hashmani, M. Zubair, and J. A. Usman, "Machine Learning Classification of Port Scanning and DDoS Attacks: A Comparative Analysis," *Mehran University Research Journal of Engineering and Technology*, vol. 40, no. 1, pp. 215–229, Jan. 2021, doi: https://doi.org/10.22581/muet1982.2101.19.

[10] EC-Council, *Certified Cybersecurity Technician (CCT) v1 Professional Series*. EC-Council, 2021. ISBN: 9781635679564. URL: https://www.vitalsource.com/products/certified-cybersecurity-technician-cct-version-1-ec-council-v9781635679564

[11] C. Sanders, *Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems*, 3rd ed. USA: No Starch Press, 2017. ISBN: 1593278020. url: https://nostarch.com/packetanalysis3/

[12] I. Nedyalkov, "Study the Level of Network Security and Penetration Tests on Power Electronic Device," *Computers*, vol. 13, no. 3, p. 81, Mar. 2024, doi: https://doi.org/10.3390/computers13030081.

[13] G., Bagyalakshmi, G., Rajkumar, N., Arunkumar, M., Easwaran, K., Narasimhan, V., Elamaran, M., Solarte, I., Hernandez, and G., Ramirez-Gonzalez, "Network Vulnerability Analysis on Brain Signal/Image Databases Using Nmap and Wireshark Tools," *IEEE Access*, vol. 6, pp. 57144–57151, 2018, doi: https://doi.org/10.1109/ACCESS.2018.2872775.

[14] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, SCITEPRESS - Science and Technology Publications, 2018, pp. 108–116. doi: https://doi.org/10.5220/0006639801080116.

[15] GitHub "Z21Raven - Overview,". https://github.com/Z21Raven/nmap-scan-characteristics (accessed Aug. 03, 2024).

[16] G. Wang and Y. Gu, "Multi-Task Scenario Encrypted Traffic Classification and Parameter Analysis," *Sensors*, vol. 24, no. 10, p. 3078, May 2024, doi https://doi.org/10.3390/s24103078.

[17] Nmap, "Nmap Network Scanning. The Official Nmap Project Guide to Network Discovery and Security Scanning." *Nmap.org*, Accessed: Jul. 09, 2024. [Online]. Available: https://nmap.org/book/toc.html

[18] A. Upadhya and B. K. Srinivas "A Survey on different Port Scanning Methods and the Tools Used to perform them," *Int J Res Appl Sci Eng Technol*, vol. 8, no. 5, pp. 3018–3024, May 2020, doi: https://doi.org/10.22214/ijraset.2020.5505.

[19] F. H. Roslan, "A Comparative Performance of Port Scanning Techniques," *Journal of Soft Computing and Data Mining*, vol. 4, no. 2, Oct. 2023, doi: https://doi.org/10.30880/jscdm.2023.04.02.004.

[20] D. Kennedy, J. O'Gorman, D. Kearns, and M. Aharoni, *Metasploit: The Penetration Tester's Guide*, 1st ed. USA: No Starch Press, 2011. ISBN: 9781593272883. url: https://nostarch.com/metasploit.

[21] A., Makulova B., Sharipova, M.Othman, A., Pyrkova. & G. Ordabayeva "Methods Analyzing Network Traffic and Detecting Network Vulnerabilities," *Journal of Mathematics, Mechanics and Computer Science*, vol. 121, no. 1, Mar. 2024, doi: https://doi.org/10.26577/JMMCS2024121110.