Original Article

COMPARISON OF TCP SCANNING TECHNIQUES USING NMAP

Shoby Sunny[1], Anil George Christy[2]
[1]SCMS School of Technology and Management, Kochi, Kerala, India
2UST Global, Kochi,Kerala, India
Email: shobysunny@gmail.com
DOI: 10.47750/pnr.2022.13.S10.104

Abstract

A complex network of resources, people, and networks makes up the Internet. The bulk of users rely merely on the many services provided since they are ignorant of the design and components of the Internet. A significant technique for gathering technical data is port scanning. Based on scan statistics from a real-world network, network defense systems can spot malicious scans. The paper addresses Connect Scan and Stealth Scan, two of the current port scan detection techniques. The comparison is structured around the three well-known criteria for categorizing scan detection approaches: data source, data presentation, and detection mechanism (or display of data). For scan detection, research prototypes that combine data mining with threshold- based analysis have showed significant potential, according to the findings. The paper concludes that even though both TCP connect and stealth scan are effective methods of scanning a system's ports and port states, stealth scanning has the advantage of logging prevention due to the fact that it uses a half-open TCP connection with the target and thus detects the target's ports more quickly while also being less likely to be detected by Web Application Firewall.

Keywords: Connect Scan, Stealth Scan, Port Scanning, NMAP, Port Scan Detection.

1. INTRODUCTION

The Internet is a complicated network of networks, people, and resources. The majority of users are unaware of the Internet's design and its components and rely only on the numerous services offered. However, a tiny percentage of expert users employ their knowledge to investigate possible system vulnerabilities [1]. Hackers can exploit unprotected hosts in two ways: they either consume resources or utilize them as attack tools. Effective attacks frequently begin with an earlier and intentional process of assessing the hosts or networks of possible victims.

Port scanning is a critical tool for obtaining technical information. Network defenders can identify malicious scans based on scan statistics from a real-world network. A port scan is a technique for detecting the availability of certain services on a host or network by monitoring replies to connection attempts [2]. A port scan is consisting of "hostile Internet searches seeking open 'doors' or 'ports' that provide attackers access to systems." These approaches entail transmitting a message to a port and waiting for a response. The returned answer reveals the port's state and may be used to ascertain the host's operating system and other pertinent information before to launching a further assault. A vulnerability scan is similar, except that a positive response from the target initiates additional communication to ascertain whether the target is vulnerable to a specific attack. A majority of assaults are preceded by some type of scanning activity, most notably vulnerability scanning [3].

The purpose of port scanning is to check the availability of a network host for open ports and other services. System administrators and other network defenders often identify port scans as a viable method for spotting signs of impending major attacks. A port scan is helpful from the attacker's perspective for acquiring important information needed to launch a successful assault. Therefore, knowing whether or not a network's defense mechanisms frequently monitor ports is of great importance to attackers. Typically, attackers conceal their identity during port scanning whereas defenders do not. The two current port scan detection methods, Connect Scan and Stealth Scan, are discussed in this paper. The comparison is organized around the three widely used standards for classifying scan detection techniques namely data source, data presentation, and detection mechanism.

2. Review of Literature

In most cases, computers run a variety of services that interact with one another via TCP or UDP ports and are connected to a network. When starting an assault, an attacker typically follows the procedures depicted in Figure 1. On a computer, there are 65,536 standard ports [4]. They fall into three broad categories: well-known ports (0–1023), registered ports (10–49151), and dynamic and/or private ports (49152 - 65535).

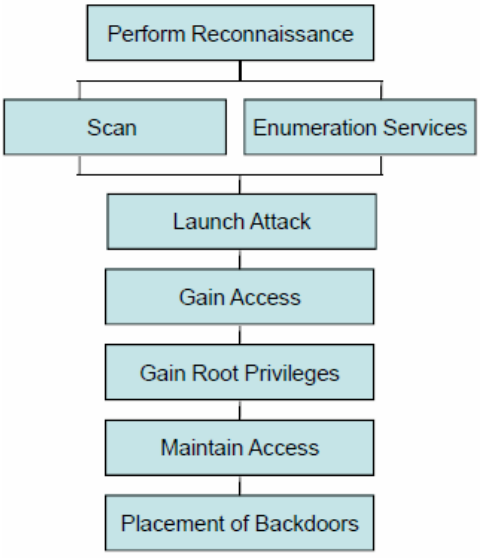

Figure 1. Steps in performing an attack

In most cases, a port scan does not pose a direct threat to the system, but it may aid the attacker in identifying the ports that are open for assault. Sending a message to each port individually and waiting for a response constitutes the core of a port scan. Depending on the response, it may be determined whether the port is in use and subsequently explored for vulnerabilities in order to launch additional attacks. TCP ports, i.e; ports that use a connection-oriented protocol, are the ones that are typically used for port scanning since they provide the attacker with useful information. It also occurs on UDP ports, but since they deliver connectionless services, they might not immediately reveal vital information to attackers. Additionally, network administrators may quickly restrict a UDP port[5]. Figure 2 depicts the various port scan types[6].

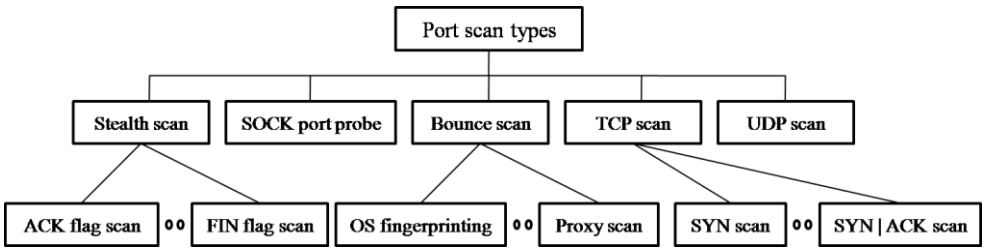

Fig. 2. Types of Port Scans

These various port scanning techniques are detailed in more detail below.

(a) Stealth Scan: The purpose of a stealth scan is to evade auditing technologies. To the destination host, it sends TCP packets marked with stealth flags. Among the flags are SYN, FIN, and NULL, to name a few.

(b) SOCKS Port Probe: Sharing Internet connections among multiple machines is made possible using SOCKS ports. Because a sizable portion of users incorrectly configure SOCKS ports, allowing communication between randomly selected sources and destinations, attackers scan these ports. An attacker may be able to connect to other Internet hosts through a system's SOCKS port while hiding his or her true location.

(c) Bounce Scan: This enables a user to establish a connection to one FTP server and then request that files be delivered to a third-party server. Given how easily such a function may be manipulated on numerous levels, the majority of servers no longer support it. This feature can be exploited by making the FTP server port scan other hosts. Email servers and HTTP proxies are two examples of apps that may support bounce scans.

(d) TCP Scanning: This type of scanning never results in a fully established TCP connection. It is therefore used by skilled attackers. An attacker can launch an attack right away if they are certain that a remote port is allowing connections. Since the server's logging system does not record this kind of connection attempt, it is far more challenging for network defences to recognise it. Some of the TCP scans are TCP Connect (), Reverse Identification, IP Header Dump scan, SYN, FIN, ACK, XMAS, NULL, and TCP Fragment.

(e) UDP Scanning: The UDP protocol-related open ports are looked for by this functionality. Attackers, however, hardly ever employ UDP because it is easily blocked since it is a connectionless protocol

In his article, Gordon Lyon[7] covers a variety of methods for figuring out which hosts' ports (or comparable protocol abstractions) are open and accepting connections. Potential communication paths are represented visually by these ports. For anyone interested in exploring their networked surroundings, including hackers, mapping their presence makes it easier to share information with the host.

TCP port scanners are distinct software applications that can be used to check if TCP ports on a host are actively accepting connections, according to research by Marco de Vivo et al. [8]. Network and/or security administrators need to be aware of these ports since they help to describe how vulnerable hosts are to outside attacks.

3. CONCEPTS & METHODOLOGY

Almost all operating systems that support TCP/IP networking come with a variety of port scanning programs. The most feature-rich tool is probably the Network Mapper[9]. While certain scanning tools, like Hping2, take advantage of various TCP/IP implementation flaws, the great majority are devoid of any built-in stealth technology. Since they will be easily identified by logging tools and even the most basic intrusion detection system, the system administrator benefits the most from this.

One of the best port scanner and stealth scanning tools currently available is NMAP, which is globally recognized. NMAP was developed to give system administrators and analysts the ability to scan huge networks to find out which hosts are online and what services they offer. All major stealth scanning techniques are built upon NMAP, which also offers new opportunities when they are found. It also makes it possible to perform useful tasks like fragmentation, decoy, and IP spoofing.

In order to perform intrusion detection and port scanning detection tests, Network Mapper was developed with the security auditor in mind. Because the attacker has access to the same information that the system administrator does, a watchful administrator must be equally aware of what NMAP can do for them as they are of how attackers can utilize it. In addition to the common scanning method, network mapper can be used to determine an OS system remotely. A technique known as TCP/IP fingerprinting is used to achieve this. Although TCP/IP is a standard, it has allegedly been implemented differently by different software manufacturers, despite being a standard. When given accurate data, all variants behave exactly the same; nevertheless, when given inaccurate data, all implementations behave substantially differently. A fingerprint can be generated by comparing these differences to those of other operating systems. This is helpful while looking for distinctive services.

The NMAP security scanner's official graphical user interface is called Zenmap[10]. It is compatible with several operating systems, including Linux, Windows, and Mac OS X. Zenmap, an open source and cost-free tool, aims to make NMAP usage simpler for beginners while yet providing extensive features for seasoned network mappers. For easy reuse, frequently used scans may be saved as profiles. You can interactively generate NMAP command lines using a command designer. Scan results can be seen and saved for later viewing. You can compare saved scan results to one another to observe how they differ. A searchable database is kept with recent scan findings.

FULL TCP CONNECTION AND SYN SCANNERS

Full TCP Connection:

The most fundamental approach to TCP port scanning has long been this one[11]. The scanning host tries to connect normally (i.e., using the full three-way handshake) to the target machine through the ports it has chosen.

To start connections, use the system call connect (). Every listening port will have success with Connect (); if not, a -1 will be given, indicating that the port is unreachable. Since connect () is frequently accessible to all users (even on multi-user systems) without requiring special rights, this method is easy to implement. This scanning method can easily be identified (logs will show a fast succession of connections and errors).Monitoring programmes like Courtney, Gabriel, and TCP Wrappers is frequently used for detection. Additionally, wrappers can be used to (try to) prevent full connection probing from known hosts since they have some control over incoming requests.

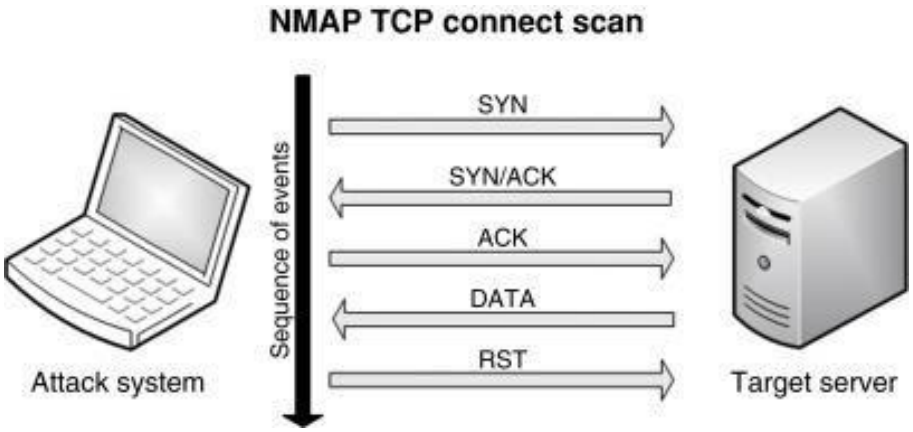

Fig. 3. NMAP TCP connect scan

TCP SYN Scanning:

By sending a SYN segment to a specific port on the target machine, this technique simulates an active open. If the answer is a RST, the target port is genuinely listening; otherwise, if a SYN | ACK segment is received, the target port is closed and a new port is probed if scheduled. A RST is sent back to the target to end the connection establishment because this information is adequate for the scanning host. This technique is frequently referred to as "half open scanning" because no full connections are established during SYN scanning. Although some loggers (like tcplogger, for example) may still be able to recognise this technique, the main advantage of SYN scanning is that incomplete connections are recorded more frequently than SYN connection attempts. For this type of scanning under most operating systems, the trade-off is that the sender must custom build the whole IP packet, and frequently super-user or privileged group access to particular system functions is necessary to generate these custom SYN packets [12]. NMAP, a very powerful and full TCP port scanner, implements both techniques (SYN and complete connection scanning), as well as nearly all of the strategies discussed in this paper.

STEALTH AND INDIRECT SCANNING

Stealth Scanning

Half-open or "SYN" scans are other names for stealth scanning. With closed and filtered ports, SYN scan operates similarly to TCP Connect scan in that it receives a RST message for closed ports and no answer for filtered ports. The way they handle open ports is the only distinction. The common techniques under Stealth scan includes FIN, Xmas Tree, and Null scans[13].

Sending a packet with only the FIN flag enabled is known as a FIN scan. The FIN packet indicates that the attacker wants the connection to be terminated yet there is no connection to close if the attacker sends it to the target. The target would become confused as a result. The port is open if the destination doesn't reply. Target ports are closed if they respond with a RST packet. This process is depicted in Fig 4 and Fig 5:

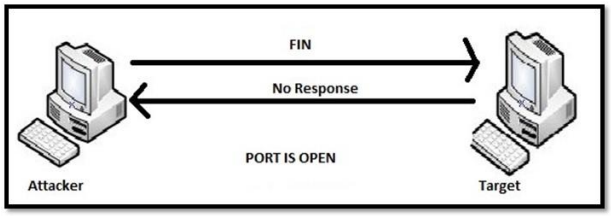

Fig. 4. FIN scan for open port

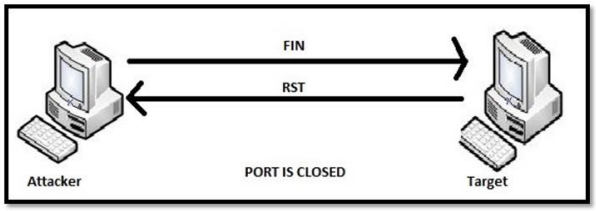

Fig. 5. FIN scan for closed port

XMAS scan, on the other hand, transmits the packet with the FIN, URG, and PUSH flags set. The target machine responds with RST if the port is closed. The packet will be disregarded if the port is open. The process is depicted in Fig 6 and Fig 7.

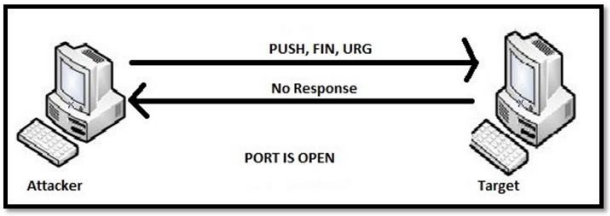

Fig. 6. XMAS scan for open port

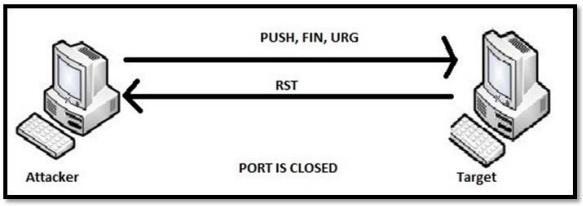

Fig. 7. XMAS scan for closed port

The TCP segment that the Null Scan sends has no packet header flags. Therefore, in Null Scan, if a port is open, we won't receive a response as shown in Fig 8. If no flags are set during Null Scan, the target will not understand how to handle the request. As a result, the target will throw away the packet and send no response. The target will reply with a RST message if the port is closed as shown in Fig 9.

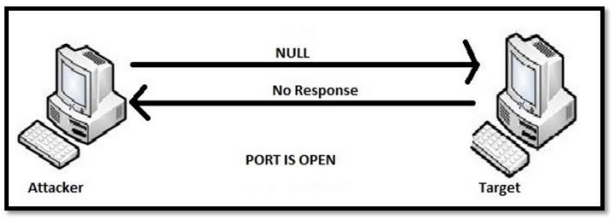

Fig. 8. NULL scan for open port

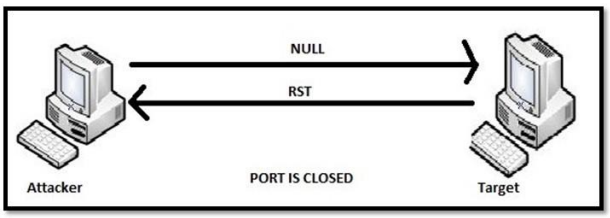

Fig. 9. NULL scan for closed port

Indirect Scanning

Recently, a unique approach to anonymous scanning has been proposed. Known as idle or zombie scanning, hackers use TCP port scanning to map a victim's system and identify its vulnerabilities [14].The actual scanning device will be hidden behind a third server's (spoofed) IP address. Because the scanned host will reply by sending or not sending specific segments to the spoof host, all that is necessary to determine the outcomes of the initial scan is to observe the spoof host's IP activity. This attack is one of the most advanced hacking techniques [15].An idle scan attack begins with a hacker taking control of a zombie computer. A zombie computer may belong to a normal user and that user may have no idea that their computer is being used for malicious attacks. The hacker doesn't use their computer to scan, so the victim can only block the zombie, not the hacker.

The paper has compared the performance of TCP Connect Scan and TCP Stealth Scan at packet flow level. Both techniques were implemented on Scapy platform with Python3 libraries.

4. RESULT

These approaches are used to visualize network traffic to identify whether network packet flow is an attack or normal behavior. Results are expressed both in terms of final test accuracy and total scan time. Table 1 below shows the results of scan time while employing TCP Connect Scan and TCP Stealth Scan.

Table 1. Performance comparison of TCP Connect and TCP Stealth Scan

Scan Type	TCP Connect Scan	TCP Stealth Scan
Port Number	22	22
Total Scan Time	0.738s	0.714s

From the above table, it can be seen that TCP Stealth Scan takes lesser time compared to TCP Connect Scan. The TCP Connect Scan takes longer time to complete as it establishes a full connection, sending a SYN packet, waiting for a SYN-ACK response, and then sending an ACK packet to close the connection. On the other hand, TCP Stealth Scan sends a SYN packet and waits for a SYN-ACK or RST response, which makes it quicker, more difficult to detect, and produces less network traffic.

5. CONCLUSION

In this paper, we have examined the status of two commonly used modern port scan detection methods at network packet floe level. Experiments show that different anomaly detection systems can be more successful for different types of port scanning attacks.

TCP connect scan is a type of full connect scan, which establishes a full connection with the target host and port, and then terminates the connection. This type of scan is relatively slow and easy to detect, as it generates a large amount of network traffic and can be easily identified by firewalls and intrusion detection systems. In addition, the scan time is longer as the scan establishes a full connection, sending a SYN packet, waiting for a SYN-ACK response, and then sending an ACK packet to close the connection.

On the other hand, TCP stealth scan is a type of half-open scan, which does not establish a full connection with the target host and port. Instead, it sends a SYN packet and waits for a SYN-ACK or RST response. This type of scan is faster and harder to detect, as it generates less network traffic and can evade detection by firewalls and intrusion detection systems. The scan time is shorter as the scan only sends the SYN packet and wait for the response.

Although both TCP Connect and Stealth scanning are effective methods for monitoring system ports and port states, Stealth scanning has the advantage of avoiding logging because it uses a half-open TCP connection to the target and thus detects the target's ports faster, while Web Application Firewalls detects fewer of them.

In summary, TCP connect scan is a slower and more detectable method of scanning, while TCP stealth scan is a faster and stealthier method of scanning. The choice of scan method depends on the purpose of the scan and the level of security required.

REFERENCES

[1] Abomhara, Mohamed, and Geir M. Køien. "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks." Journal of Cyber Security and Mobility (2015): 65-88.

[2] Lee, C. B., Roedel, C., and Elena, S. (2003) Detection and characterization of port scan attacks. Technical report. University of California, San Diego, CA. http://cseweb.ucsd.edu/users/clbailey/PortScans.pdf.

[3] Panjwani, S., Tan, S., and Jarrin, K. M. (2005) An experimental evaluation to determine if port scans are precursors to an attack. Proceedings of DSN'05, Washington, DC, USA, June 28–July 1, pp. 602–611. IEEE Computer Society.

[4] Zaripova, D. A. "Network security issues and effective protection against network attacks." International Journal on Integrated Education 4.2 (2021): 79-85.

[5] Choi, Hyunsang, Heejo Lee, and Hyogon Kim. "Fast detection and visualization of network attacks on parallel coordinates." computers & security 28.5

(2009): 276-288.
[6] Bhuyan, Monowar H., Dhruba Kr Bhattacharyya, and Jugal K. Kalita. "Surveying port scans and their detection methodologies." The Computer Journal 54.10 (2011): 1565-1581.
[7] Lyon, Gordon Fyodor. Nmap network scanning: The official Nmap project guide to network discovery and security scanning. Insecure. Com LLC (US), 2008.
[8] De Vivo, Marco, Gabriela O. de Vivo, and Germinal Isern. "Internet security attacks at the basic levels." ACM SIGOPS operating systems review 32.2 (1998): 4-15.
[9] Lyon, Gordon. "Nmap security scanner." línea] URL: http://nmap. org/[Consulta: 8 de junio de 2012] (2014).
[10 Samantha, B. Surya, and M. V. Phanindra. "An Overview on the Utilization of Kali Linux Tools." International Journal of Research and Analytical Reviews 5.2 (2018).
[11] Ali, Fakariah Hani Mohd, Rozita Yunos, and Mohd Azuan Mohamad Alias. "Simple port knocking method: Against TCP replay attack and port scanning." Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec). IEEE, 2012.
[12] Balram, Soniya, and M. Wiscy. "Detection of TCP SYN scanning using packet counts and neural network." 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems. IEEE, 2008.
[13] Staniford, Stuart, James A. Hoagland, and Joseph M. McAlerney. "Practical automated detection of stealthy portscans." Journal of Computer Security 10.1-2 (2002): 105-136.
[14] Ensafi, Roya, et al. "Idle port scanning and non-interference analysis of network protocol stacks using model checking." 19th USENIX Security Symposium (USENIX Security 10). 2010.
[15]Bou-Harb, Elias, Mourad Debbabi, and Chadi Assi. "Cyber scanning: a comprehensive survey." Ieee communications surveys & tutorials 16.3 (2013): 1496-1519.