Review

Port Scanning Techniques Tools and Detection

Sam Coyle *

Posted Date: 5 March 2024

doi: 10.20944/preprints202403.0225.v1

Keywords: Port Scanning; TCP; TCP SYN; UDP; Stealth Scan; ZMap; NMap; MASSCAN; Anomaly based detection; Signature based detection

Review

Port Scanning: Techniques, Tools and Detection

Sam Coyle

University of Bradford, Bradford, United Kingdom s.d.coyle@bradford.ac.uk

Advisor Name: Ibrahim Ghafir

Abstract: This review aims to consolidate varied information regarding Port Scanning, and examine the tools, techniques, and detection algorithms used. It explores TCP SYN, Full TCP, UDP, and stealth scans, and shows their functions and abilities. Paired with these techniques, it also shows the tools that they can be used by, such as NMap, Zmap, and MASSCAN. It discusses the various detection techniques such as Signature and Anomaly based detection, and discusses the real world impact of this technology, while commenting on the legal and ethical issues posed.

Keywords: Port Scanning; TCP; TCP SYN; UDP; Stealth Scan; ZMap; NMap; MASSCAN; Anomaly based detection; Signature based detection

I. Introduction

Network security is becoming more important, as times goes on. Cyber attack damage has increased by 50% from 2018 to 2022 [12]. Port scanning is the most prevalent technique used by attackers, to gain information about a potential target. It has led to many cyber attacks over time, one of which was the deadly ransomware Wannacry in 2017 [23], and it has the potential to cause a huge amount of damage to key infrastructure if misused [27].

Port Scanning can be key evidence that a cyber attack is likely to attack your network soon. It is a technique that any network administrator must understand, and must have a counter to. I will lay out techniques used by attackers, and methods that network administrators can use to defend against these types of attacks. This review aims to analyze the existing port scanning techniques, their counterparts in the detection industry, and any other tools that are used in Port Scanning. As well as understanding the impact of Port Scanning in Cyber Attacks, and their impact in the industry.

Figure 1. Stages of a Cyber Attack [13].

Port scanning is the main component of the "Reconnais- sance" segment of a Cyberattack [13]. In this stage of a Cyber Attack, Attackers gather information about the target system(s), and aim to gather enough information to strike. Port Scanning is the most prevalent tool for information gathering. With the information from one, Attackers can know which segments of a network have vulnerabilities [5] and use this information to target weaknesses and gain access to a system [10].

 doi:10.20944/preprints202403.0225.v1

II. Techniques

A. Full/Control TCP

Transmission Control Protocol is the main manager of services within a system. Full TCP port scanning is the process of connecting to the desired TCP ports on a system, using the standard TCP connection protocol [1]. This method works by the port scanning application sending its TCP sequence num- ber and maximum segment size, to the system it is scanning. The system would respond with the same information but in reverse, then the port scanning software would acknowledge that it has received this information [14]. Due to there being 3 steps to this, this is commonly referred to as the "TCP three- way handshake [1]".

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

Figure 2. Ports and their uses and transfer protocols [13].

This is one of the most common scans, due to being able to send requests as a general user, with no permissions needed [5]. This is important, as since this is the recon stage, you usually have no permissions on the system you are trying to access. This method works by simply connecting to the desired ports, and closing the connection if the scan succeeds [28]. If it fails then you simply get an error returned [7]. This allows you to categorize the port in question as Open, Closed or Filtered.

The main issue with this type of scan, is that if you use it on multiple ports at once the system in question will be able to see the large amount of connections that were opened, just to be instantly closed [8]. A secondary issue is that various modern operating systems disallow using this type of port scanning with just general permissions, rendering this method hampered on modern systems [7]. Along with these 2 drawbacks, it also takes longer than alternative methods due to having to complete the three-way handshake.

B. TCP SYN

TCP SYN works by sending a SYN segment to the target port. Once the port receives this SYN segment, it will either return an RST meaning the port is closed, or a SYNACK seg- ment meaning that the port is open and can be communicated with. Once a SYNACK segment is received the port scanner can close the connection and mark that port as open [1].

TCP SYN differs from Connect TCP by the fact that it never makes a full connection [6]. This helps it avoid detection methods on the system's end. This is referred to as a "Half- Open" scan [1]. This means it is faster to operate.

You also have to custom build the IP packets in question for this method to work. This is intensive to do, takes time and also requires Superuser permissions [1]. SYN is one of the most common port scanning methods, due to being logged less. It is frequently used across different applications, such as NMAP, ZMap, and MASSCAN [6].

 doi:10.20944/preprints202403.0225.v1

C. UDP

User Datagram Protocol Scanning is more niche than Full TCP and TCP SYN. This is due to the fact that UDP is simply less used than TCP for the main services attackers would want to take advantage of. UDP Port scanning works by abusing the fact that any closed UDP port responds with an error [1]. Using this error response, you can quickly determine which ports are closed [29]. You can also use this to infer which ports are open by the lack of response, however this method is not foolproof and can result in inaccuracies [10].

UDP is less useful than TCP, however it still has uses such as detecting unauthorized services or detecting some important services such as HTTPS, DNS or DHCP [1]. A study found [6] that using NMAP, UDP scans take on, on average, 688MS while TCP SYN scans take only 14.56 MS on average. This is the main disadvantage of UDP scans, along with its unreliability due to no handshake taking place [30].

D. Stealth Scans: NULL, XMAS, FIN

FIN scans are a type of TCP scans that trade being more hidden for more ambiguity in the results. They work by sending a packet to the port containing just a FIN flag. Once the port receives this it will either respond with a reset flag, meaning the port is closed, or no response [7]. This method allows you to confirm that ports are closed, and similar to UDP it allows you to infer if ports are open. This method only work for Unix type machines, and any Microsoft machines will only return reset flags [17].

XMAS Scans are incredibly similar, and only differ by including an URG and PSH flag along with the FIN flag during the initial port scan. The results are largely the same, and they are mainly used for determining if ports are closed [31]. NULL Scans also exist which return the same results, but function in practise by not sending any flags whatsoever, thus the NULL name [17]. These are a less common type of scan, and are less of a target for firewalls. Collectively they are referred to as “Stealth Scans [1]”.

They are harder to detect, commonly are not logged due to never having a connection made, and are looked for less since they provide less information.

III Tools

A. NMap

Network Mapper, AKA NMap, is the leading piece of software for port scanning. It is an open source tool that allows for advanced port scanning, network scanning, and can determine operating systems [4]. It allows for Connect TCP, SYN, FIN, XMAS, NULL and UDP scanning [6].

This example scan returns any non closed ports on the network. NMap classifies ports by type, being Open, Closed, Filtered, and Unfiltered. Open means that the port is open and can communicate, and Closed means that the port is closed for the moment and will not accept any communication requests. Filtered means that a firewall/filter is involved and blocking NMap from any further inspections, while Unfiltered is effectively the blanket term NMap uses for when it cannot classify a port accurately [4].

Along with all this information, as the string to start contains

-A, NMap also returns the operating system of the system, and any versions it can find for specific ports [32]. In Figure 3, we can see that the example operating system is Linux 2.6.39, and we can see that, where possible, NMap has provided the specific version for the ports that are not closed.

NMap also allows a large degree of customization. You can change what ports to scan, the order of the ports, the speed vs accuracy of a scan, and also which scan method to use [4]. This allows for NMap to be a truly versatile piece of software that can provide a large amount of information at once. NMap being open source, having a scripting engine al- lowing for custom scripts, means that it is very flexible. There exists datasets of scripts within NMap which allow for searching of specific vulnerable port versions [33], or specific vulnerable operating systems. All this

 doi:10.20944/preprints202403.0225.v1

customizability, with the fact it has all major port scanning techniques implemented by default, along with new cutting edge versions, has made it the de facto port scanning tool of many. The only issue with NMap for the moment, is that in some cases the UDP scans can take 4x as long as alternatives [6].

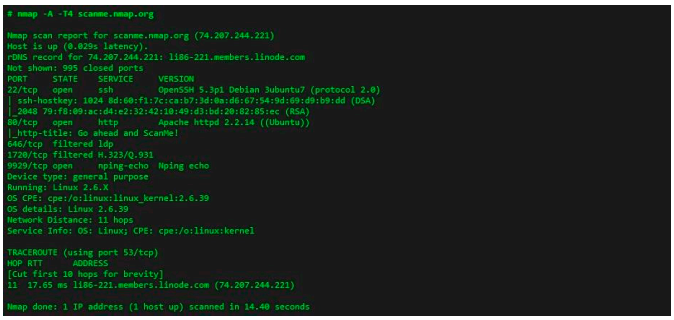

Figure 3. NMAP example scan [19].

B. ZMap

ZMap differs from NMap by having a focus on speed over accuracy. ZMap only allows for SYN and UDP scanning. This would be a large factor for anyone choosing NMap instead. ZMaps main function is for wide fast scans [34], a type of brute force method that values quantity over quality.

This is ideal for anyone doing wide area scans, for instance, if you are penetration testing a large organization with many ports available this would be much more efficient than NMap. The downside to this is that it's very obvious what is going on, with a lot of subtlety lost.

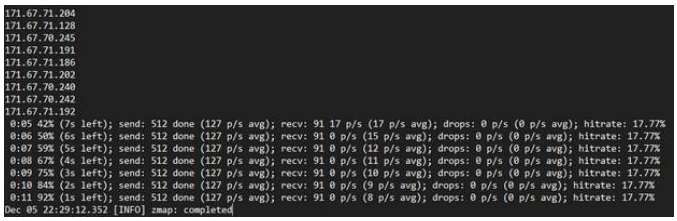

Figure 4. ZMAP example scan [9].

ZMap allows mass data collection from its port scans, which is very convenient for scanning any large area network. However, these data dumps are still a lot less accurate than NMap and do not allow for as much customizability, or precision.

Finally, due to ZMap choosing only SYN and UDP scan- ning, it loses a lot of the information that you can gain via FULL TCP. Things such as versions, operating systems, or vulnerabilities are all absent compared to NMap. Overall, you generally want to use NMap for accuracy, and ZMap for a quick but summarized approach.

C. MASSCAN

MASSCAN is the most extreme out of all the options. It's similar to NMAP in functionality, but completely focused on speed [18]. It was designed to "Scan the entire internet [18]", and can go up to 1.6million packets per second.

MASSCAN can only send SYN packets, and this means it has the lack of information this method provides [6]. It also treats all ports as open and just attempts to scan anything provided, its outputs are much less detailed[18], it has less customizability, and its less customizable compared to ZMap, and NMap.

It can do some simple banner gathering, and can return whether a port is setup for FTP, HTTP, etc [18]. MASSCAN is essentially the program where you want to go to the absolute extreme, you

 doi:10.20944/preprints202403.0225.v1

don't mind how obvious what you're doing is, and you have a huge amount of ports to scan. It's quite niche compared to ZMap or NMap, due to what it sacrifices to achieve the speeds it can.

IV. Detection

A. Signature Based

Signature based detection is one of the most common forms of detection. It works by building up a dataset of well defined port scanning attacks, and how they appear [3].

With this dataset it can then be compared to any suspicious activity on the network. If you see any of the signatures reappear you can confirm that port scanning is happening, and deal with it [19]. Snort is a common Intrusion Detection System (IDS) that uses a signature based approach [20]. This works by manually configuring the system to alert to any signatures of port scans in the database of Snort [20]. Snort reads the scan and if it matches a signature in its database, it will discard this packet and send an alert. If it does not it will allow the process to proceed [19].

The main advantage of port scanning is that it's easy to use [19]. You simply build up a dataset of port scanning attacks, and how they appear, and then you cross reference that with suspicious behavior on your system(s) in the future [3]. IDS such as Snort mentioned above already have existing databases of previous attacks, allowing for this to be an easy security feature to include on your system. Due to signatures having to match attacks exactly, false positives are rare [19].

The main disadvantage is that any port scan attacks that are not in the database will be completely left alone and have free reign until the database is manually updated [19]. It also requires intimate knowledge of the attack types, which can vary over time and can vary operating system to operating system [19].

B. Anomaly Based

Anomaly based detection is set up by setting a baseline for "normal" activity on a system [21]. This is usually done by monitoring and recording the system for a period of time. Once this normal baseline has been established, port scanning attacks that vary from the normal drastically would result in an alert being pushed [22].

The main advantage to Anomaly based IDS is that any novel attacks that would go undetected by a Signature based IDS, could be triggered on an Anomaly based IDS [22] as the actual signature of the attack is irrelevant. The only valuable metric is how different the traffic from the port scanning attack is compared to the default. This can make Anomaly based IDS the best method for detecting new port scanning vulnerabilities.

The biggest disadvantage is that if the port scanning attacks do not exceed the default thresholds set on the Anomaly based IDS, nothing will be done [21]. This is where a Signature based IDS would easily prevail, as it can detect a very small amount of anomalous traffic. A common strategy is to spread out your port scanning attacks over a longer period of time with less scans per day [21], if given enough time this can allow every port to be scanned without being detected.

V. Case Study

EternalBlue was an exploit usable on Microsoft Windows systems discovered by the NSA [24]. It was the foundation for the malware WannaCry that spread across much of the world in 2017. EternalBlue allowed for an attacker to inject malware via port 445 the Server Message Block [24] port.

This exploit was patched by Microsoft 2 months earlier, but many systems had yet to update [23]. This port vulnerability allowed for Wannacry to be spread to other devices within the same network of the first infected device [23]. This drastically increased the amount of damage caused by Wannacry, as it was common for entire networks to have not been updated to a version that fixed this vulnerability. It also required no user interaction for this to spread, which made the spread even worse [24].

The main lesson to be learned from this case study, is that updating your system's operating system every time possible, is the easiest way to reduce the risk of being infected. This also highlights

 doi:10.20944/preprints202403.0225.v1

the need for firewalls, if this port was blocked on the devices that did not need it open then it's likely the damage would have been much less severe [23]. It is also good practise to ensure that only users that need higher privileges have access, and that lower level users only receive basic permissions [23]. Legal issues:

VI. Legal and Ethical Concerns

The legality of port scanning varies by country and location. Even in some of the places where port scanning is legal, specific internet service providers expressly ban the act from taking place when you're using their services [25]. The UK law is rather vague on this matter simply saying it is illegal to "supply or offer to supply a program, believing that it is likely to be used to commit, or to assist in the commission of a Computer Misuse Act violation" [26].

The main issue with having port scanning itself be illegal, is that port scanning can be used as a very accurate method to perform penetration testing on willing participants networks. This is why port scanning is usually determined to be legal or not based on intent; if someone performs it while having permission with the goal to enhance a businesses security practices, this would be fine. But if you start scanning for vulnerabilities within a business without consent then this would be acting maliciously, and you would likely be guilty of committing a computer misuse act violation.

The ethical view of port scanning is similar to the legal interpretation, that it depends on your intent while doing it. The creator of NMap himself says: "Nmap was designed to help secure the Internet", and NMap and the other tools like it can help patch vulnerabilities that would otherwise go unnoticed. It's when people use these tools to start exploiting them with malicious intent that port scanning becomes a very unethical act to commit.

VII. Conclusion

Port scanning is a key technology in Cybersecurity. Attack- ers can use it to cause massive damage, while administrators are constantly playing catch up to patch any new exploits that come about. Techniques such as Full TCP, TCP SYN and UDP, are the current favorites of this branch of Cybersecurity, and are used both by penetration testers testing for clients, and attackers testing for their own profit. Programs such as NMap, ZMap and MASSCAN allow for attackers or testers to set their own parameters and requirements for their intended purpose, while admins are constantly trying to develop on top of the existing detection algorithms. Port scanning is a technology that will always be around, and this review hopes to provide some insight into the intricacies of this technique.

References

1. Evgeny V Ananin, Arina V Nikishova, and Irina S Kozhevnikova. Port scanning detection based on anomalies. 2017 Dynamics of Systems, Mechanisms and Machines (Dynamics), pages 1–5, 2017.
2. Soniya Balram and M Wiscy. Detection of tcp syn scanning using packet counts and neural network. In 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems, pages 646–649. IEEE, 2008.
3. S Chakrabarti, Mohuya Chakraborty, and Indraneel Mukhopadhyay. Study of snort-based ids. In Proceedings of the International Conference and Workshop on Emerging Trends in Technology, pages 43–47, 2010.
4. Laura Chappell. Inside the tcp handshake. NetWare Connection, 2000.
5. Diab, D. M., AsSadhan, B., Binsalleeh, H., Lambotharan, S., Kyriakopoulos, K. G., & Ghafir, I. (2019, August). Anomaly detection using dynamic time warping. In 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (pp. 193-198). IEEE.
6. Roger Christopher. Port scanning techniques and the defense against them, 2002.
7. Frank Cremer, Barry Sheehan, Michael Fortmann, Arash N Kia, Martin Mullins, Finbarr Murphy, and Stefan Materne. Cyber risk and cyberse- curity: a systematic review of data availability. The Geneva Papers on risk and insurance-Issues and practice, 47(3):698–736, 2022.

 doi:10.20944/preprints202403.0225.v1

8. Lefoane, M., Ghafir, I., Kabir, S. and Awan, I.U., 2021, December. Machine learning for botnet detection: An optimized feature selection approach. In The 5th International Conference on Future Networks & Distributed Systems (pp. 195-200).
9. Mehiar Dabbagh, Ali J Ghandour, Kassem Fawaz, Wassim El Hajj, and Hazem Hajj. Slow port scanning detection. In 2011 7th International Conference on Information Assurance and Security (IAS), pages 228–233. IEEE, 2011.
10. Marco De Vivo, Eddy Carrasco, Germinal Isern, and Gabriela O De Vivo. A review of port scanning techniques. ACM SIGCOMM Computer Communication Review, 29(2):41–48, 1999.
11. Zakir Durumeric. Getting started guide, 2023. Accessed: 12 12, 2023.
12. Jayant Gadge and Anish Anand Patil. Port scan detection. In 2008 16th IEEE international conference on networks, pages 1–6. IEEE, 2008.
13. Robert David Graham. Masscan: Mass ip port scanner, 2023. Accessed: 12 13, 2023.
14. Eltanani, S. and Ghafir, I., 2020, November. Coverage Optimisation for Aerial Wireless Networks. In 2020 14th International Conference on Innovations in Information Technology (IIT) (pp. 233-238). IEEE.
15. Shaun Jamieson. The ethics and legality of port scanning. SANS, 1(1):1, 2021.
16. Sairam Jetty. Network Scanning Cookbook: Practical Network Security Using Nmap and Nessus 7. Packt Publishing Ltd, 2018.
17. Yunhan Jack Jia, Qi Alfred Chen, Yikai Lin, Chao Kong, and Z Morley Mao. Open doors for bob and mallory: Open port usage in android apps and security implications. In 2017 IEEE European Symposium on Security and Privacy (EuroS&P), pages 190–203. IEEE, 2017.
18. Zhang, Y., Yang, Q., Lambotharan, S., Kyriakopoulos, K., Ghafir, I. and AsSadhan, B., 2019, October. Anomaly-based network intrusion detection using SVM. In 2019 11th International conference on wireless communications and signal processing (WCSP) (pp. 1-6). IEEE.
19. VVRPV Jyothsna, Rama Prasad, and K Munivara Prasad. A review of anomaly based intrusion detection systems. International Journal of Computer Applications, 28(7):26–35, 2011.
20. Vinod Kumar and Om Prakash Sangwan. Signature based intrusion detection system using snort. International Journal of Computer Applications & Information Technology, 1(3):35–41, 2012.
21. Wentao Liu. Design and implement of common network security scanning system. In 2009 International Symposium on Intelligent Ubiquitous Computing and Education, pages 148–151. IEEE, 2009.
22. Gordon Lyon. Nmap legal issues, 2023. Accessed: 12 13, 2023.
23. Lefoane, M., Ghafir, I., Kabir, S. and Awan, I.U., 2022. Unsupervised learning for feature selection: A proposed solution for botnet detection in 5g networks. IEEE Transactions on Industrial Informatics, 19(1), pp.921- 929.
24. Gordon Lyon. Nmap reference guide, 2023. Accessed: 12 13, 2023.
25. Jiefei Ma, Franck Le, Alessandra Russo, and Jorge Lobo. Detecting distributed signature-based intrusion: The case of multi-path routing attacks. In 2015 IEEE Conference on Computer Communications (INFOCOM), pages 558–566. IEEE, 2015.
26. Eltanani, S. and Ghafir, I., 2021, May. Aerial Wireless Networks: Proposed Solution for Coverage Optimisation. In IEEE INFOCOM 2021- IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 1-6). IEEE.
27. Thabiso M. Maupong Maurice Dawson, Oteng Tabona. Cybersecurity capabilities in developing nations and its impact on global security. IGI Global, page 206, 2022.).
28. MS-ISAC. Eternalblue security primer. MS-ISAC, 1(1):1, 2019.
29. Angela Orebaugh and Becky Pinkard. Nmap in the enterprise: your guide to network scanning. Elsevier, 2011.
30. Susmit Panjwani, Stephanie Tan, Keith M Jarrin, and Michel Cukier. An experimental evaluation to determine if port scans are precursors to an attack. In 2005 International Conference on Dependable Systems and Networks (DSN'05), pages 602–611. IEEE, 2005.
31. Aparicio-Navarro, F.J., Kyriakopoulos, K.G., Ghafir, I., Lambotharan, S. and Chambers, J.A., 2018, October. Multi-stage attack detection using contextual information. In MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM) (pp. 1-9). IEEE.
32. Rajni Ranjan Singh and Deepak Singh Tomar. Network forensics: detection and analysis of stealth port scanning attack. International Journal of Computer Networks and Communications Security, 3(2):33– 42, 2015.

 doi:10.20944/preprints202403.0225.v1

33. Noah Stone. New high-severity vulnerability (cve-2023-29552) discov- ered in the service location protocol (slp), 2023. Accessed: 12 13, 2023.
34. Fatin Hazirah Roslan. A comparative performance of port scanning techniques. Journal of Soft Computing and Data Mining, 4(2):43–51, 2023.