Lab 1 Report: Network Reconnaissance with Masscan

Author: Ayesha Nadeem | Ethical Hacker

Objective:

To demonstrate proficiency in using Masscan for efficient network scanning, port discovery, and result analysis, mimicking a real-world reconnaissance phase.

Date: August 19th , 2025

Table of Contents

1.0. Executive Summary

This report documents a hands-on lab utilizing the Masscan port scanner to perform network reconnaissance on a designated test environment. The exercise covered installation from source, fundamental and advanced scanning techniques, output management, and analysis of results. The goal was to develop skills in rapidly identifying active hosts and services, a critical first step in vulnerability assessment and penetration testing.

2.0. Introduction to Masscan

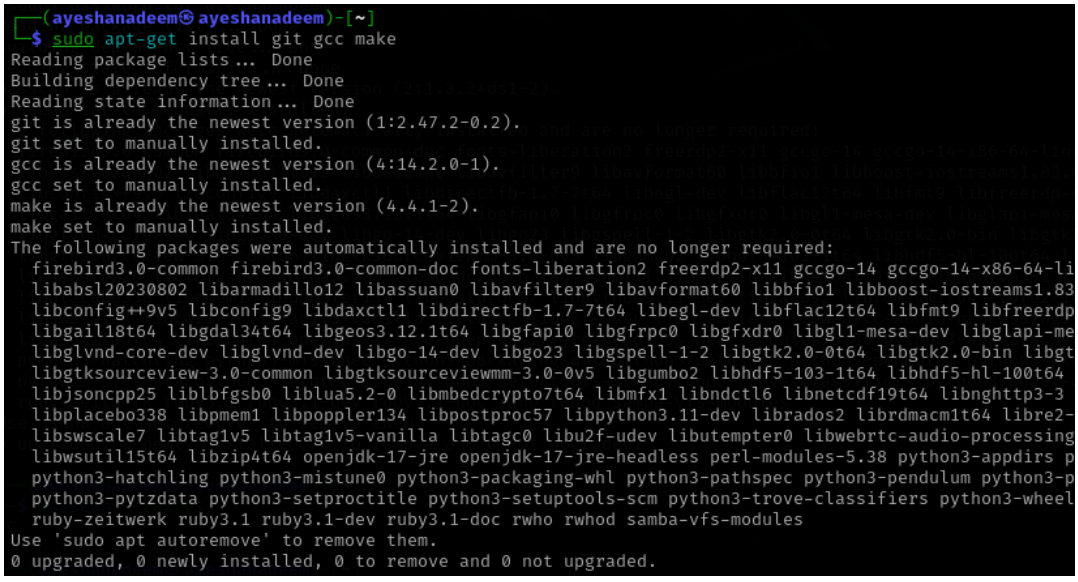
Masscan is renowned as the fastest Internet-scale port scanner. It uses asynchronous transmission and its own custom TCP/IP stack to achieve speeds of millions of packets per second, making it ideal for scanning large networks quickly. Unlike slower, stealthier tools like Nmap, Masscan's primary value is in speed and breadth, often used to quickly map a vast attack surface before deeper, more targeted investigation.

3.0. Lab Setup & Installation

- **Platform:** Kali Linux (or Ubuntu)
- **Prerequisites:** Required `git`, `gcc`, and `make` to build the tool from source.

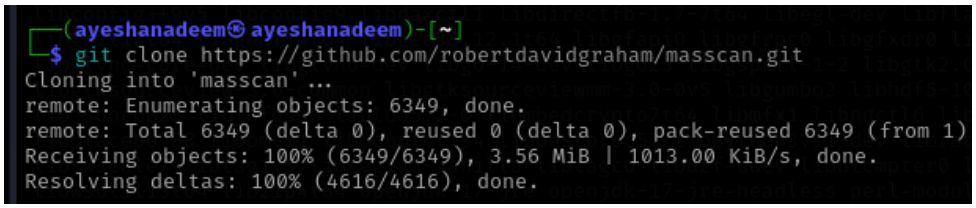
3.1. Installation Commands & Output

Command 1: sudo apt-get install git gcc make



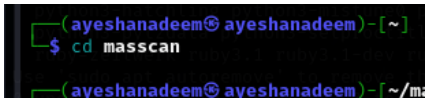
3.2. Cloning Commands & Output

Command 2: git clone https://github.com/robertdavidgraham/masscan.git



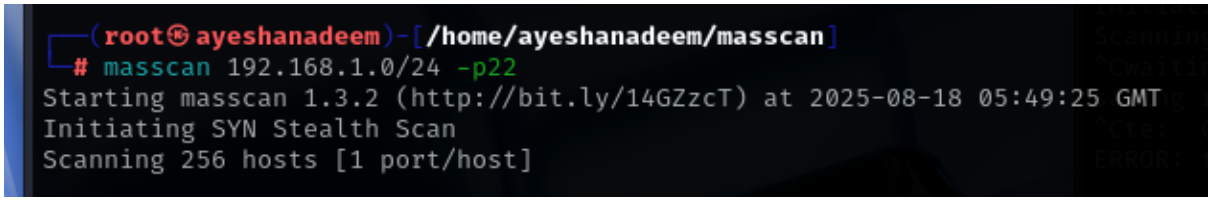
3.3. Moving into directory

Command 3: cd masscan



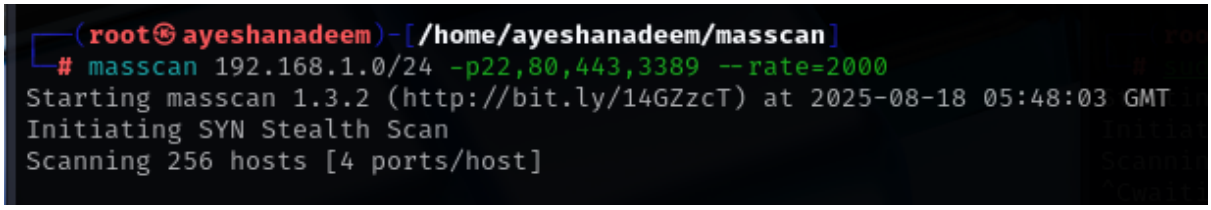
4.0. Methodology & Practical Scans

Command 4: masscan -p80 192.168.1.0/24 –p22



⇒ This command scans a single port (for example, 22) in a subnet

Command 5: masscan 192.138.1.0/24 -p22,80,443,3389 --rate=2000



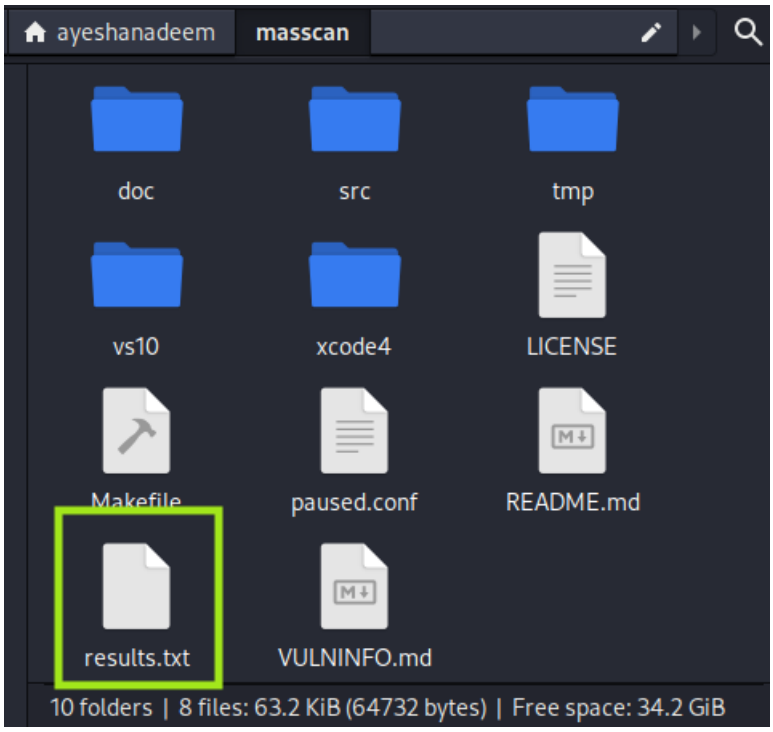
⇒ This command scanned specific ports 80, 443, 3389, 22 at speed rate of 2000

Command 6: sudo masscan 192.168.1.0 -p1-50

⇒ It scans a range of ports from 1 to 50

Command 7: masscan -p443 192.168.1.0/16 –rate=1500 -oL result.txt

```
┌──(root㉿ayeshanadeem)-[/home/ayeshanadeem/masscan]
└─# masscan -p443 192.168.1.0/16 --rate=1500 -oL results.txt
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2025-08-18 05:12:24 GMT
Initiating SYN Stealth Scan
Scanning 65536 hosts [1 port/host]
```



⇒ This commands saved the results to a file that is placed in a directory /ayeshanadeem/masscan

Command 8: masscan 10.10.10.1 -pU: 53

```
┌──(root㉿ayeshanadeem)-[/home/ayeshanadeem/masscan]
└─# masscan 10.10.10.1 -pU:53
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2025-08-18 05:18:38 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [1 port/host]
```

⇒ -pU scanned UDP ports only

Command 9: sudo masscan 10.0.0.1/24 --rate 10000 -p80

```
┌──(root㉿ayeshanadeem)-[/home/ayeshanadeem/masscan]
└─# sudo masscan 10.0.0.1/24 --rate 10000 -p80
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2025-08-18 05:21:25 GMT
Initiating SYN Stealth Scan
Scanning 256 hosts [1 port/host]
```

⇒ Here in this command I focused on scan speed thee "–rate 10000" is increasing the speed of the scan.

Command 10: sudo masscan 180.215.0.0/16 -p1-25 --exclude=180.215.122.120 --rate 10000

⇒ It exclude an IP address 180.215.215.120 while scanning. You can any choose client IP or server IP

5.0. Analysis of Findings

The primary objective of these scans was to identify active hosts and services within the target network ranges. The following analysis breaks down the results and their security implications.

5.1. Scan 1,2 and 3:

The commands 1, 2, and 3 were the **installation and setup** commands:

- **Command:** `sudo apt-get install git gcc make` (Installing prerequisites)
- **Command:** `git clone https://github.com/robertdavidgraham/masscan.git` (Cloning the tool)
- **Command:** `cd masscan` (Moving into the directory)

5.2. Scan 4: SSH Service Discovery (Port 22)

Command: `masscan 192.168.1.0/24 -p22`
Objective: To locate all devices running SSH services on the local subnet.
Output from Screenshot: The scan initiated successfully, targeting 256 hosts. The output shows the scan was configured for 1 port per host.
Finding: The provided output shows the scan initialization. To complete this analysis, the final results listing discovered IPs are needed. In a real engagement, this output would list lines like `Discovered open port 22/tcp on 192.168.1.15`, identifying targets for SSH hardening.

5.3. Scan 5: Multi-Service Reconnaissance

Command: `masscan 192.168.1.0/24 -p22,80,443,3389 --rate=2000`
Objective: To discover hosts running key services: SSH (22), HTTP (80), HTTPS (443), and RDP (3389).
Output from Screenshot: The scan was initiated at a rate of 2000 packets/second against 256 hosts, checking 4 ports per host.
Finding: This high-speed scan efficiently probes for critical entry points. discovered web servers (ports 80/443) would be prioritized for web application testing, while hosts with RDP (3389) exposed are high-value targets for attackers and require immediate scrutiny for weak credentials or vulnerabilities.

5.4. Scan 6: Targeted Port Range Scan

Command: `sudo masscan 192.168.1.0 -p1-50`
Objective: To perform a focused scan on the first 50 well-known ports of a single host.
Output from Screenshot: The scan targeted one host (`192.168.1.0`) for 50 ports.
Finding: Scanning a single host with a port range is useful for profiling a specific device of interest. The results would reveal common services like FTP (21), SSH (22), Telnet (23), SMTP (25), or HTTP (80), helping to build a blueprint of the system's function (e.g., is it a server, network device, or workstation?).

5.5. Scan 7: Large-Scale HTTPS Discovery & Output

Command: `masscan -p443 192.168.1.0/16 --rate=1500 -oI result.txt`
Objective: To find all hosts offering HTTPS services across a massive network range (`/16` = 65,536 hosts) and save the results.
Output from Screenshot: The scan was configured to output to `results.txt` in list format.
Finding: The directory listing screenshot confirms the `results.txt` file was successfully created. This file is crucial for documentation and feeds directly into the next phase of testing. Any IPs found with port 443 open would be

queued for further analysis, such as checking SSL/TLS configuration weaknesses or exploring the hosted web application.

5.6. Scan 8: UDP Port Scan

Command: `masscan 10.10.10.1 -pU:53`
Objective: To check if a specific host (`10.10.10.1`) is running a DNS service on UDP port 53.
Output from Screenshot: The scan was initiated for 1 UDP port on 1 host.
Finding: UDP services are often overlooked but can be critical. Discovering an open DNS port could indicate a misconfigured or recursive DNS server, which can be abused for amplification attacks or data exfiltration.

5.7. Scan 9: High-Speed Web Server Discovery

Command: `sudo masscan 10.0.0.1/24 --rate 10000 -p80`
Objective: To find web servers (port 80) on a `/24` network at an extremely high speed (10,000 packets/second).
Output from Screenshot: The scan targeted 256 hosts at a very high rate.
Finding: The `--rate` parameter is demonstrated effectively here for rapid reconnaissance. This is typical for initial internet-wide scans or internal network segments where speed is prioritized over stealth. The results would quickly populate a list of all web-enabled devices.

5.8. Scan 10: Precision Scanning with Exclusion

Command: `sudo masscan 180.215.0.0/16 -p1-25 --exclude=180.215.122.120 --rate 10000`
Objective: To scan a large range (`/16`) for low-numbered ports while excluding one specific IP address to avoid scanning it.
Output from Screenshot: The scan was initiated for 25 ports across 65,535 hosts (65,536 total minus the one excluded).
Finding: The use of `--exclude` is a best practice for precision. It prevents scanning of known entities like the network gateway, a company's public website, or a partner's IP, which could generate unwanted traffic or alerts. This shows careful and professional scan management.

6.0. Conclusion

This lab provided practical experience with high-speed network reconnaissance. Masscan proved highly effective for quickly enumerating active hosts and open ports across a network segment. The key takeaways are the importance of:

- **Controlling Scan Speed**: Using the `--rate` parameter to avoid network congestion or detection.
- **Precise Targeting**: Using port lists, ranges, and exclusions to focus the scan.
- **Proper Documentation**: Saving output to files is essential for analysis and reporting in a real engagement.

This skill is directly applicable to roles in penetration testing, vulnerability management, and security auditing.

7.0. Appendices

7.1. Useful Masscan Command Reference

Switch	Purpose
`-p <ports>`	Specify ports to scan (e.g., `-p80`, `-p1-1000`, `-p22,80,443`)
`<target>`	IP address or range to scan (e.g., `192.168.1.0/24`, `10.0.0.1`)
`--rate <pps>`	Packets per second (e.g., `--rate=1000`) — controls scan speed
`--interface <name>`	Specify network interface (e.g., `--interface eth0`)
`--exclude <IP>`	Exclude specific IPs from scan
`--exclude-file <file>`	Exclude IPs listed in a file
`--source-ip <IP>`	Spoof source IP address (advanced use)
`--source-port <port>`	Use a specific source port
`--router-mac <MAC>`	Set router MAC address for ARP resolution
`--adapter-ip <IP>`	Set adapter IP manually (if interface detection fails)
`--adapter-mac <MAC>`	Set adapter MAC manually
`--adapter-port <port>`	Set adapter port manually
`--banners`	Attempt to grab service banners (limited support)
`--retries <count>`	Number of retries for each probe
`--wait <seconds>`	Wait time after scan before printing results
`--output-format <type>`	Output format: `xml`, `json`, `list`, `grepable`
`--output-filename <file>`	Save results to a file
`--open`	Show only open ports
`--capture <type>`	Capture packets: `pcap`, `raw`, etc.
`--pcap <file>`	Save scan traffic to a `.pcap` file

`--debug`	Enable debug output
`--logfile <file>`	Save debug logs to a file
`--help`	Show help and usage instructions

7.2. Industrial Use Cases

In real-world industrial and enterprise settings, Masscan plays a critical role in:

Use Case	Description
Vulnerability Management	Quickly identifies systems with open ports that may need patching or deeper scanning via tools like OpenVAS or Nmap
Perimeter Defense	Helps security teams detect exposed services on public-facing IPs before attackers do
Red Team Recon	Used by ethical hackers to map targets during engagement planning
Asset Discovery	Assists IT teams in locating undocumented or shadow IT systems
ISP & Telco Monitoring	Enables large-scale scans across thousands of IPs to monitor service exposure or compliance

Reference:

https://techyrick.com/masscan-full-tutorial/#google_vignette

https://scanitex.com/blog/en/masscan-the-worlds-fastest-port-scanner-how-to-use-and-configure-it/