**Scanning with Nmap**

Nmap stands for Network Mapper, which is a free and open source utility for network discovery and security auditing. It's used by all users, from network administrators and security professionals, to black-hat hackers trying to probe for hosts and services on a computer network. What really makes Nmap powerful is the details it gives about devices on a network and services they are running. It sends packets to a target system and analyzes the responses. Depending on how the target system is responding, Nmap is able to deduce details such as the OS, running services, and vulnerabilities within them.

Host Discovery:

This initial critical phase of the network scanning process is widely referred to as "ping scanning." Host discovery focuses on determining those hosts in a network that are active and reachable.

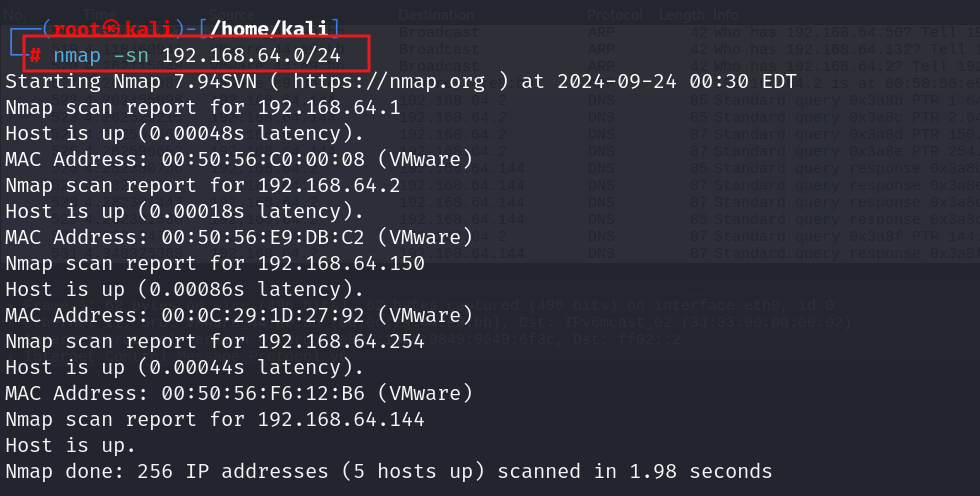
**ICMP Ping Scan**: One can perform an ICMP ping scan using the option -sn in Nmap. This is used for "Ping Scan --disable port scan." The alternative tells Nmap to conduct only host discovery and not to scan any ports.

By default, the option `-sn` of host discovery includes an ICMP echo request, a TCP SYN to port 443, a TCP ACK to port 80, and an ICMP timestamp request. When run by a non privileged user, the executed options are limited to sending only SYN packets to ports 80 and 443 on the target—a connect call. If a privileged user scans on a local Ethernet network, ARP requests are used unless the --send-ip option is specified. The -sn option can be combined with other discovery probe types as a convenience. All of the discovery probe options (-P\*) and port scan Incremental (-I) options are mutually exclusive. If any of these options are specified, they will override the default probes. These advanced techniques should be used in environments where tight firewalls exist between the Nmap scanner and the target network; at other times, it may miss some of the hosts due to dropped probes or responses.

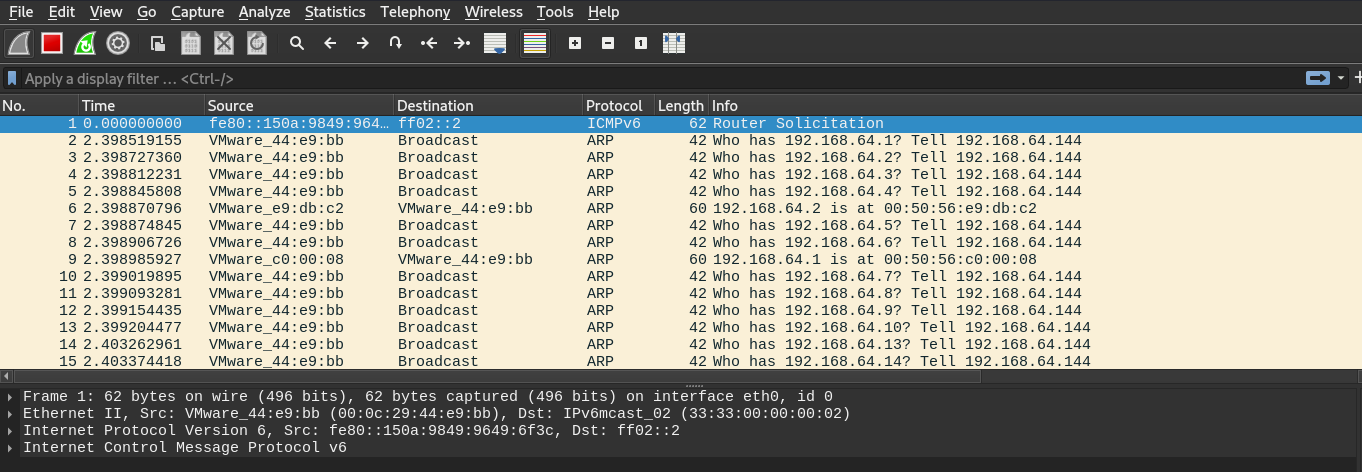
**Note:** In Previous versions of Nmap, `-sn` were referred to as `-sP`.

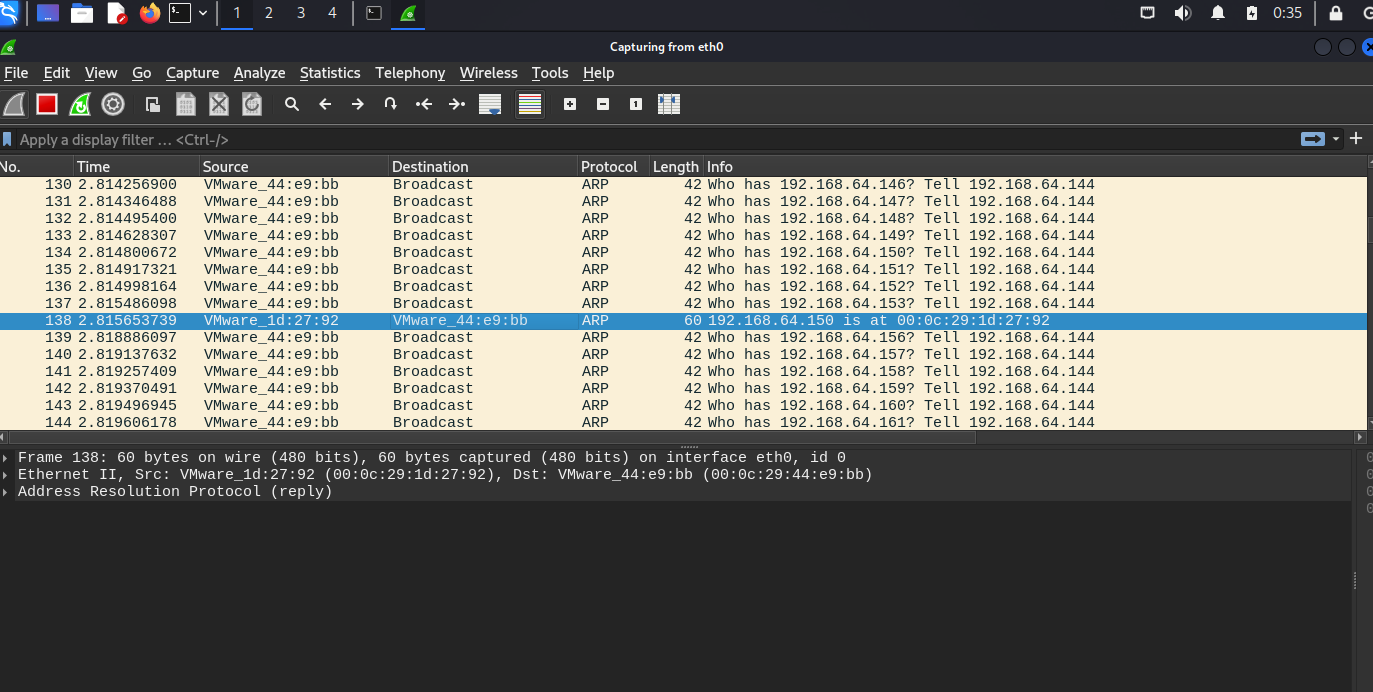
**$ nmap -sn <Target IP>**

**$ nmap -sn 192.168.64.0/24**



**This occurs in the Backend at Wireshark:**





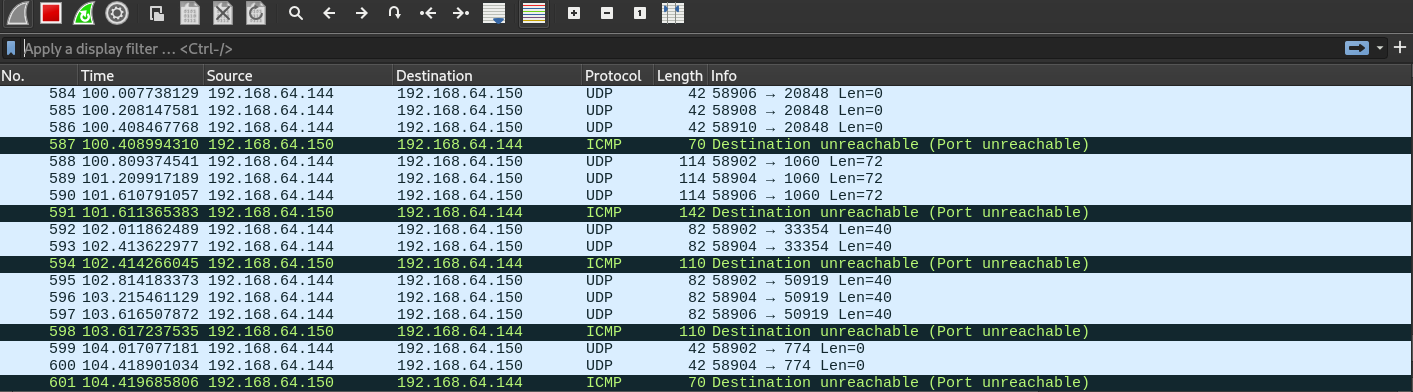
**UDP Scan**: An Nmap UDP scan is an excellent tool for discovering open UDP ports on the target machine. Unlike in TCP, there is no mechanism for connecting in UDP; thus, it is not very reliable, and scanning is extremely slow because of its high rate of retransmissions and false positives. The scanner sends UDP packets to particular ports on the target device at the very start of a UDP scan. It then awaits responses: an open port, with a running service, might respond with some sort of UDP response, or may even just acknowledge the probing. If the port is closed, it is possible that the device responds with an ICMP "Port Unreachable" message. In case there is no response, the port may be either closed or filtered by some firewall.

**$ nmap -sU <Target IP>**

**$ nmap -sU 192.168.64.150**



**This occurs in the Backend at Wireshark:**



**TCP SYN ping Scan**: In the world of cybersecurity, TCP SYN ping scanning is the process of using nmap to find active hosts in a network. The technique in the scanning process is the generation of TCP SYN packets. This kind of scan helps in determining whether a host is up by trying to probe the starting of a TCP handshake.

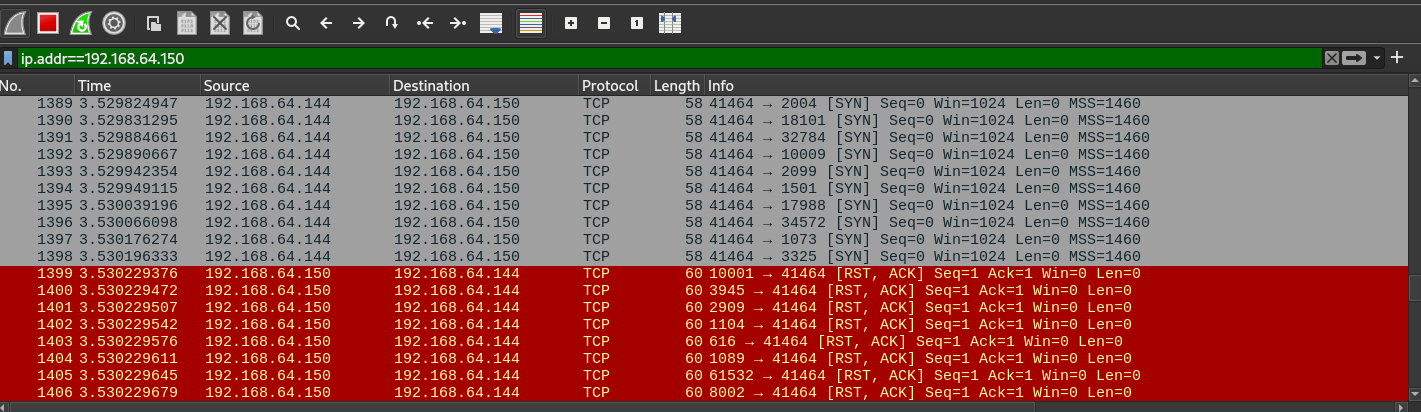
A syn flag is an indication sent to the remote system that a connection attempt is underway.

For the most part, if the distant port is closed, the remote system would return a RST (reset) packet. If the distant port is open, the target will return a SYN/ACK TCP packet as the second step of the TCP three-way handshake. Instead of following through with the handshake by sending an ACK packet, which would establish a full connection, the Nmap machine sends a RST packet to break the connection. This RST is a response packet generated by the operating system of the Nmap machine and is not sent from the Nmap application. In any case, on an open or closed port, Nmap uses the presence of either an RST or SYN/ACK to indicate that it could reach the host and that it responded to it.

**$ nmap -PS <Target IP>**

**$ nmap -PS 192.168.64.150**



**This occurs in the Backend at Wireshark:**

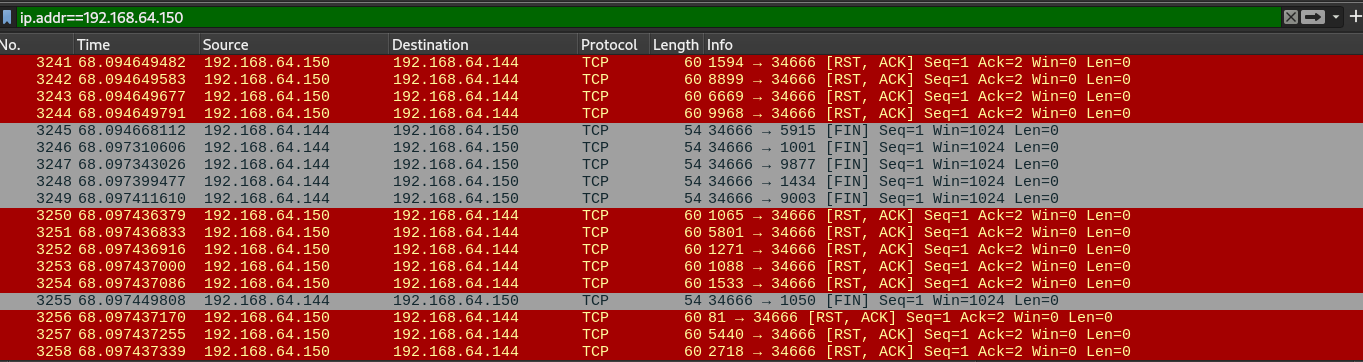
**FIN scan**: An FIN scan within Nmap focuses on the idea of port state detection with the target system by sending TCP packets with the FIN flag set. The FIN flag is used to close a connection already established for TCP. The target system does not usually respond to the FIN packet because there is no open port. Conversely, if a port is closed, then either a Rest packet is sent by the system to reply back that a connection can't be either established or terminated. Therefore, port is open= No Response and Port is closed= Response with Reset packet.

**$ nmap -sF <Target IP>**

**$ nmap -sF 192.168.64.150**



**This occurs in the Backend at Wireshark:**

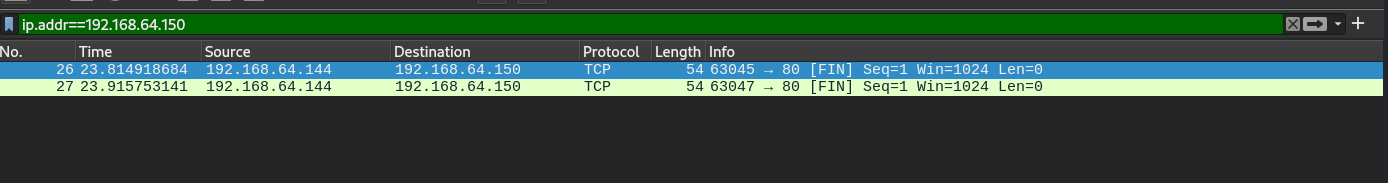


**$ nmap -sF –p80 <Target IP>**

**$ nmap -sF -p80 192.168.64.150**



**This occurs in the Backend at Wireshark:**



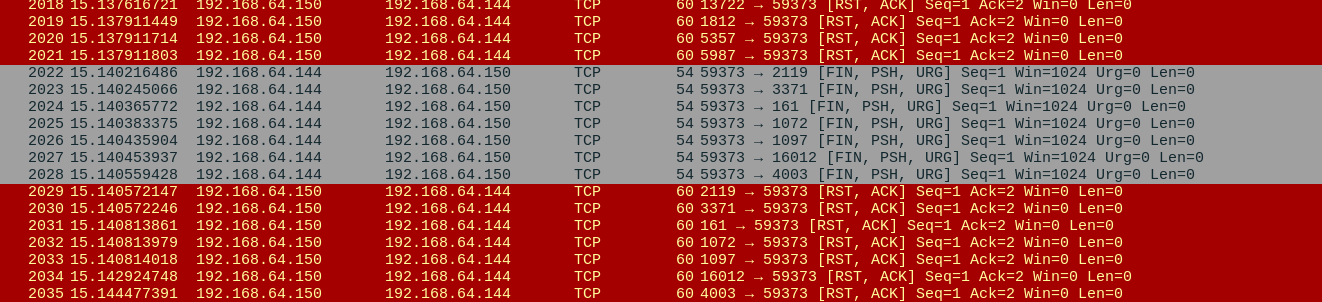
**XMAS scan** : This is yet another method of scanning for TCP ports that can be implemented using Nmap. This scan sets different flags within the TCP header into a target port probe. More precisely, an Xmas scan sets the FIN, URG, and PSH flags in the TCP header packet. As a result, this flag combination is typical within legitimate communication. The target system does not usually respond to the Xmas packet because, when a given port is open, there is no connection that can be closed. A target system sends it back in the case of a closed port, signaling that the connection request is denied, so it is responded to with a reset. Therefore, by default, no response indicates that the port is generally open, and the presence of an RST packet indicates that the port is closed.

**$ nmap -sX <Target IP>**

**$ nmap -sX 192.168.64.150**



**This occurs in the Backend at Wireshark:**

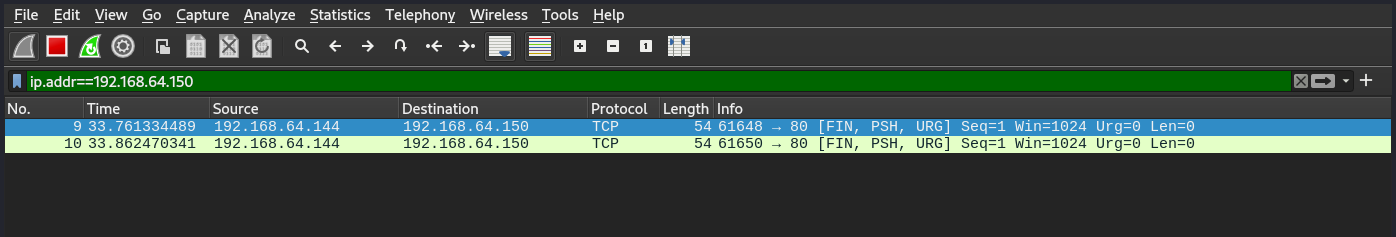


**$ nmap -sX -p80 <Target IP>**

**$ nmap -sX -p80 192.168.64.150**



**This occurs in the Backend at Wireshark:**



**ACK scan**: An ACK scan standing for Acknowledgement Scan is one of the techniques Nmap uses to map out firewall rulesets. It achieves this by sending TCP ACK packets to a target, and then analyzing the respective responses to determine whether ports are filtered, unfiltered, or open. Generally, these packets acknowledge the receipt of data with well-established connections. In an unfiltered port, it sends an RST packet, meaning that the port is not somehow filtered by the firewall or otherwise. If a port is filtered—let's say by a firewall—then the transmission of the request results in a packet not being replied to, apparently lost. It therefore follows that a packet with an RST in a response is usually announcing that the port is unfiltered and accessible, while the lack of a response elucidates the fact that the port is filtered or tailed by a firewall or other security device.

**$ nmap -sA <Target IP>**

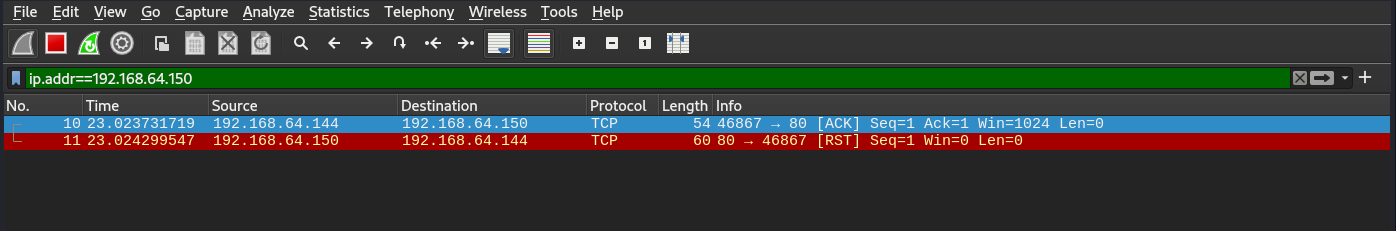
**$ nmap -sA 192.168.64.150**

**$ nmap -sA -p80 <Target IP>**

**$ nmap -sA -p80 192.168.64.150**



**This occurs in the Backend at Wireshark:**



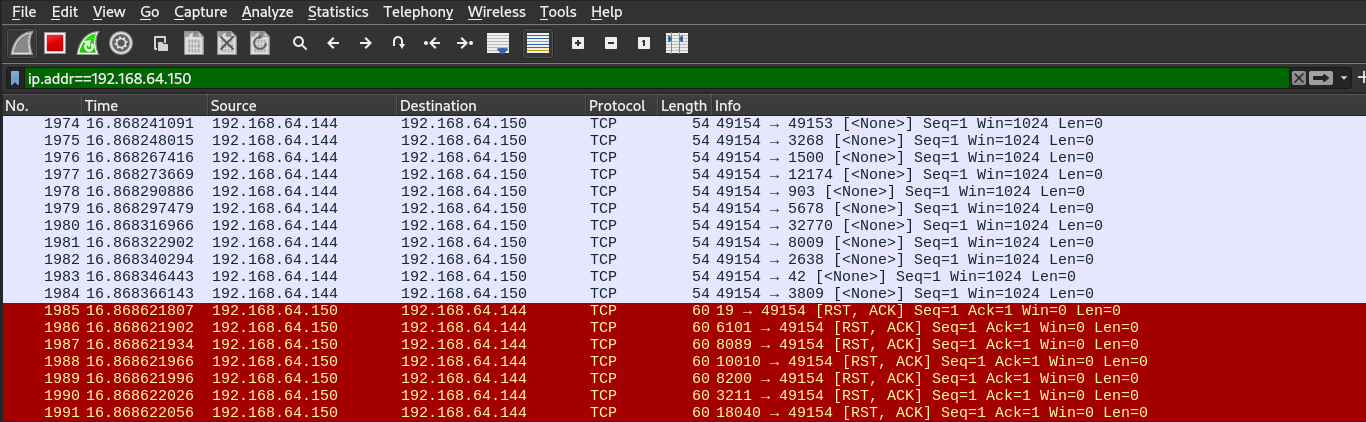
**Null Scan**: This is another way of scanning TCP ports under the Nmap utility. A null scan means the creation of a crafted TCP header that results in no flags being placed, also known as "null". This scan can be used to elicit replies from certain types of systems and firewalls that might react differently to strange or invalid TCP packets. This is normally because, if a port is open, the Null packet will not receive a response back from the target system; there will be nothing to close or announce. If the port is closed, the target system should respond with a RST (Reset) packet, thus indicating that a connection cannot be made or torn down. Which means, if you get no reply, the port is likely open. But if a RST packet is sent, a closed port is likely at the other end.

**$ nmap -sN <Target IP>**

**$ nmap -sN 192.168.64.150**



**This occurs in the Backend at Wireshark:**



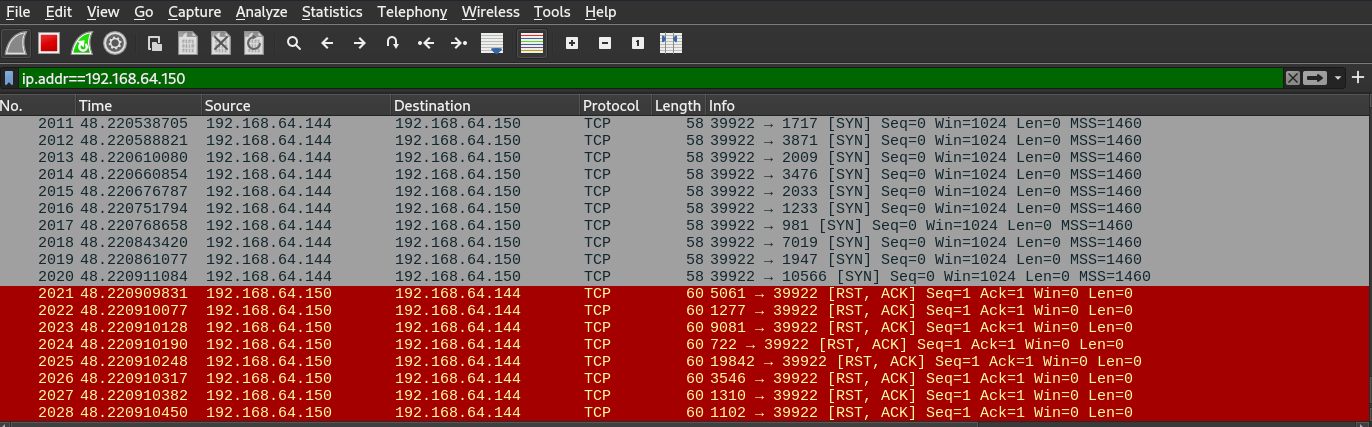
**Stealth Scan**: This is a kind of network scan with an intent to identify the available open ports on a target system by thinking out ways of reducing the chances of being busted by intrusion detection and firewalls. This is the default scan type in Nmap (nmap -sS). It sends out SYN packets to the target ports. It doesn't complete the TCP handshake (SYN, SYN-ACK, ACK) as a full TCP connection. In contrast, it closes the connection after receiving a SYN/ACK; it does so by sending a RST flag. This way, it won't make a full TCP handshake, with lowered chances that the target system will keep the connection recorded.

**$ nmap -sS <Target IP>**

**$ nmap -sS 192.168.64.150**



**This occurs in the Backend at Wireshark:**

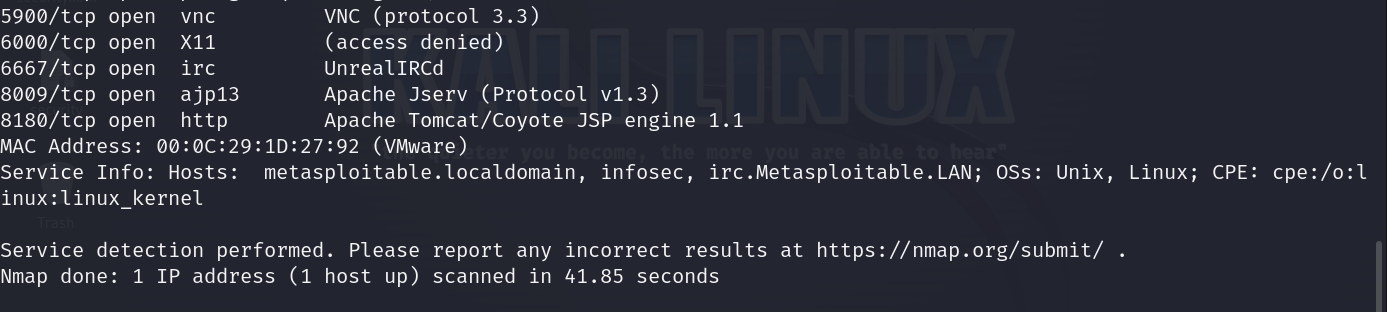


**Service version detection**: The flag `-sV` to Nmap will enable the version detection. With `-sV`, Nmap discovers all open ports and then tries to determine the versions of the services accessible on those ports. It helps to find the version of particular software along with their known vulnerabilities.

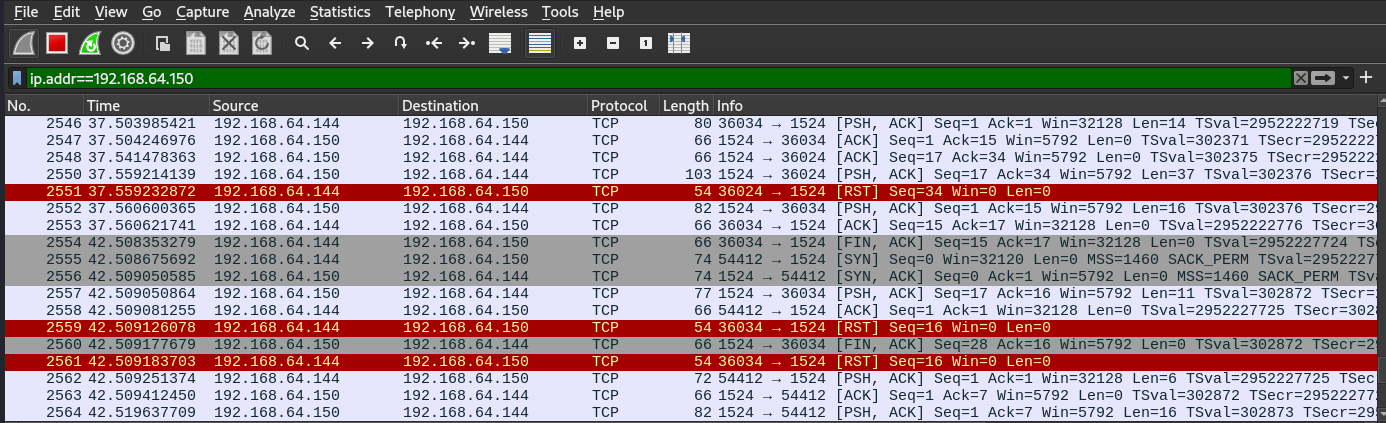
**$ nmap -sV <Target IP>**

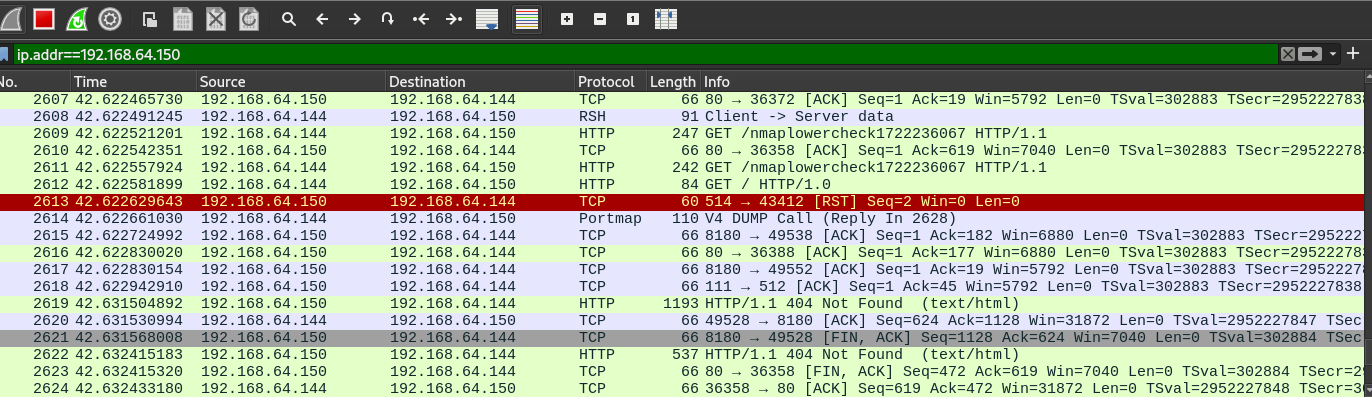
**$ nmap -sV 192.168.64.150**





**This occurs in the Backend at Wireshark:**



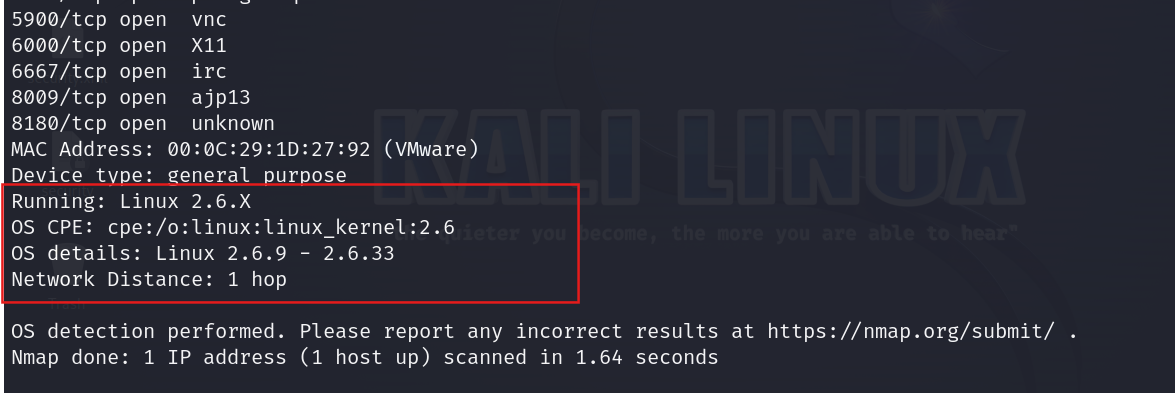


**Operating system Detection**: Nmap has an option that allows detecting the OS installed on a target device. This is done through the examination of responses on a network and fingerprinting techniques. Nmap sends a series of especially designed packets to the target system and then analyzes the responses. It looks at things such as how the TCP/IP stack behaves, response times, among other minute details. These responses are then matched against a database of known operating system fingerprints stored in Nmap to identify the OS.

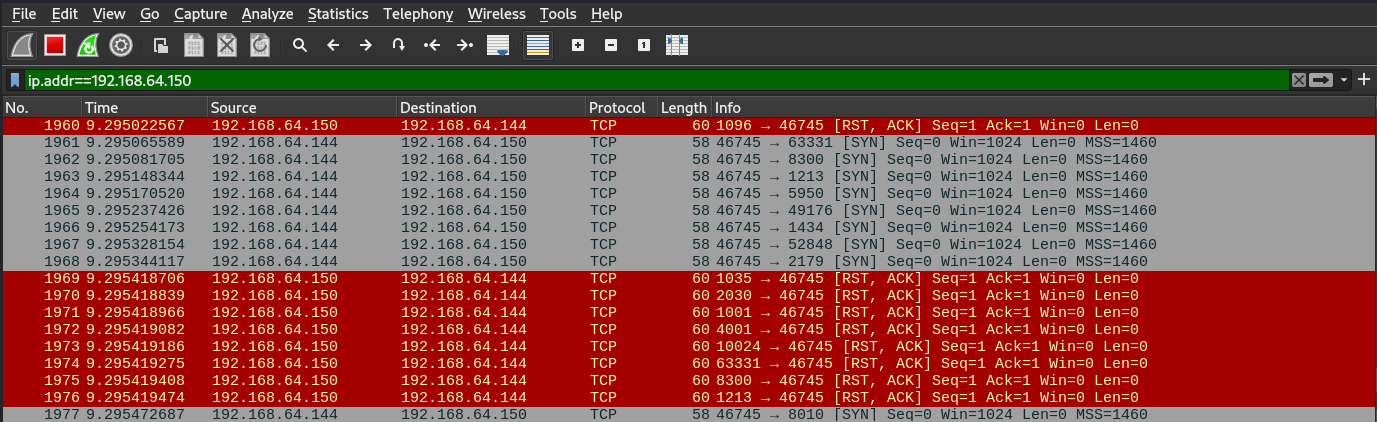
**$ nmap -O <Target IP>**

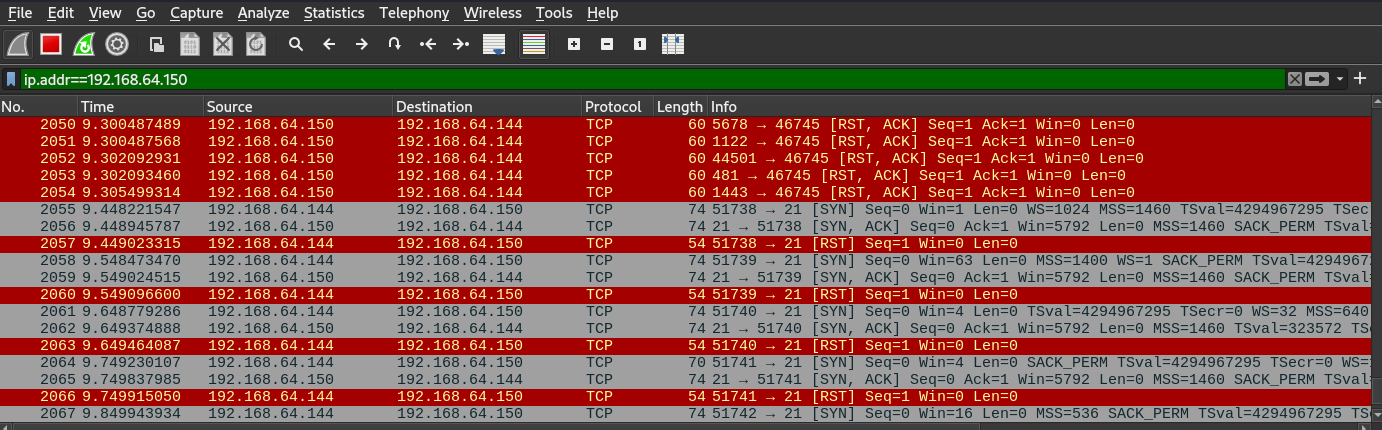
**$ nmap -O 192.168.64.150**





**This occurs in the Backend at Wireshark:**





**Evasion Techniques** : Evasion techniques are methods in Nmap that help one elude a firewall, intrusion detection systems, and other network security means while carrying out a scan. Several of the common techniques in Nmap evasion are as follows:

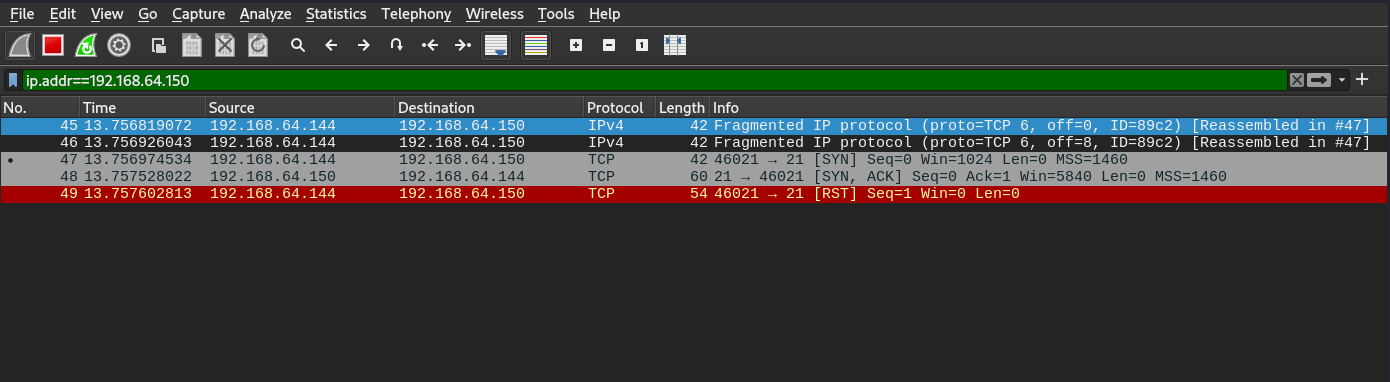
**Fragment packets**: By fragmenting the packets into small pieces, Nmap is able to bypass the packet filters and firewalls that are otherwise supposed to prevent the packet of a bigger size. Fragmentation is done through the option of `-f`.

**$ nmap -f -p21 <Target IP>**

**$ nmap -f -p21 192.168.64.150**



**This occurs in the Backend at Wireshark:**



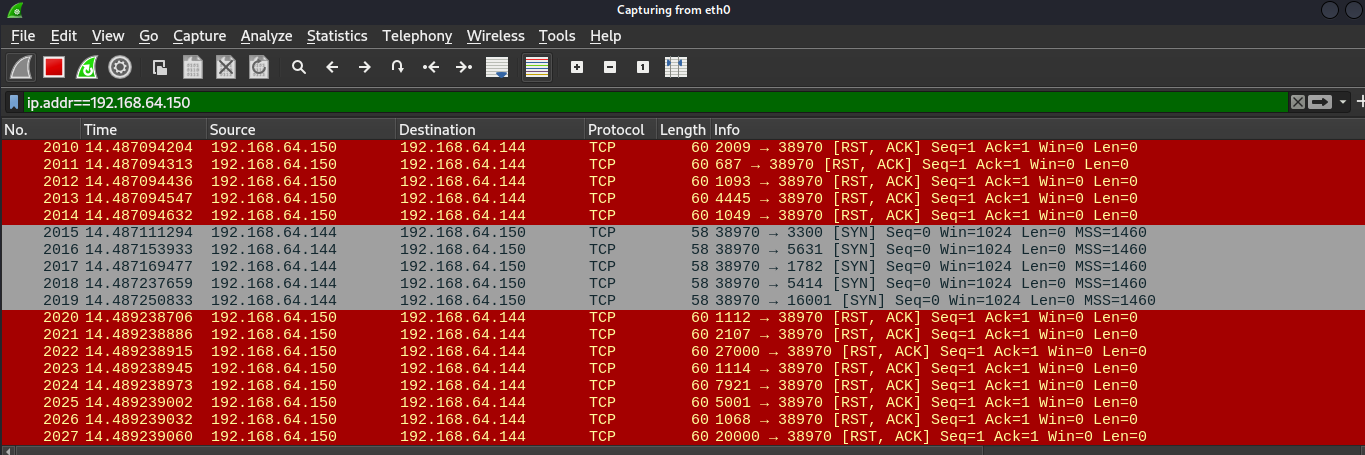
**Specifying MTU**: This is sending in the packets in an MTU size to avoid firewall/IDS detection. Configuring the packet size to be less than the MTU of the network allows Nmap to avoid most security mechanisms that block or closely examine large packets. In order to utilize this MTU evasion technique in Nmap, an end-user could assign a value to MTU by using the --mtu option. It customizes a way of bypassing filtering mechanisms, which might be triggered with normal packet sizes, hence avoiding detection.

**$ nmap -mtu 24 <Target IP>**

**$ nmap -mtu 24 192.168.64.150**



**This occurs in the Backend at Wireshark:**





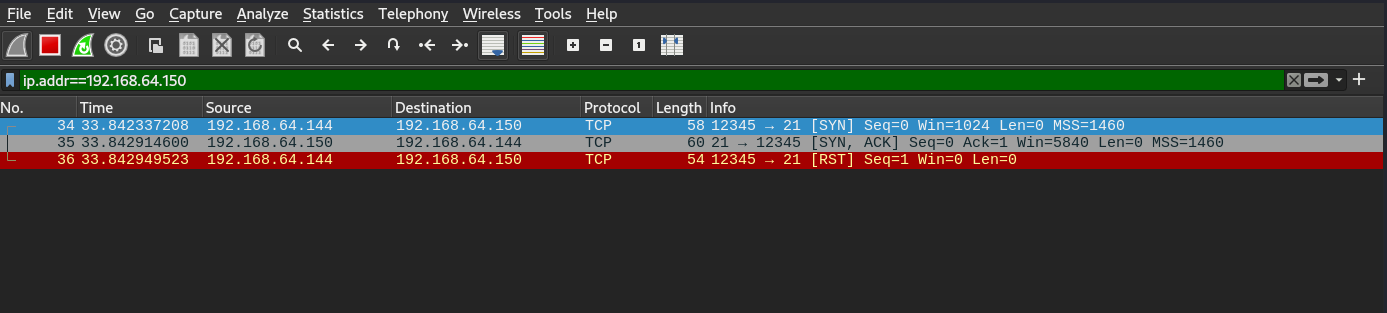
**Source port manipulation**: Setting the source port of the scan packets to a common port number that could be disregarded by some firewall or intrusion-detection system in handling the packets is accomplished by the option "--source-port".

**$ nmap -g 12345 -p21 <Target IP>**

**$ nmap -g 12345 -p21 192.169.64.150**



**This occurs in the Backend at Wireshark:**



**Conclusion:** Nmap host discovery is very critical in three ways: identifying hosts with open ports, possibly active hosts, in a network. Nmap enables efficient and panoramic visualization of the network environment with the use of different scanning techniques, such as ICMP, TCP, and UDP probes, for any network system administrator or security expert. For every scanning technique, there are some clear advantages and disadvantages that work to increase stealth, accuracy, or evasion techniques against firewalls and intrusion detection systems. State-of-the-art techniques in nmap, such as packet fragmentation, decoy scanning, and MTU adjustments, result in increased efficiency of host discovery functions and increased evasion techniques against security controls. Mastering these techniques allows for the complete assessment of the security and weaknesses of a network; thus, Nmap is a critical tool in network management and security evaluation.