Hello Everyone! In this video, you will learn about some more Nmap uses for a security professional as well as an attacker.

Another special type of scanning using Nmap is idle or zombie scanning. This scanning is used by attackers or security professionals to hide their identity by imitating another machine. It uses the concept of IPID, the fragmentation identification number, which is incremented by many operating systems for each packet they send. Probing for the IPID can tell an attacker how many packets have been sent since the last probe to get the status of the ports.

Here we see a scenario, in which an open port is identified using the idle scanning method. Let's understand this. Scanner (the one who is scanning, either attacker or cyber security official) first probes the zombie host by sending TCP SYN/ACK segment. Zombie host, not expecting a SYN/ACK packet, sends and RST segment, denying the connection, but discloses it's IPID, which is recorded by the scanner. In the second step, the scanner sends a SYN segment to the target, forcing it to be coming from the zombie host. The target being unaware of all this process, sends its SYN/ACK segment to zombie host (mentioned as sender in the SYN packet received). Now as the zombie host is unaware about all this, sends a RST segment, denying the connection. This step increases the IPID by 1. In the third and last step, the scanner again probes the zombie host and this time receives the same response but with the IP ID increased by 2(347 this time, which was initially 345 only).

The scanner, which is aware about the complete process, concludes that the scanned port is opened, as it sends a response to the zombie host which results in the increase of IPID.

Here is a scenario, where closed port is identified using the idle scanning method. Scanner first probes the zombie host by sending TCP SYN/ACK segment. Zombie host, not expecting a SYN/ACK packet, sends and RST segment, denying the connection, but discloses it's IPID, which is recorded by the scanner. In the second step, as before, the scanner sends a SYN segment to the target, forcing it to be coming from the zombie host. The target being unaware of all this process, sends its RST segment to zombie host (mentioned as sender in the SYN packet received), as the port is closed. The zombie host does not respond to the RST segment, which results in no increases in the IPID value.
In the third and last step, the scanner again probes the zombie host and this time receives the same response but with the IP ID increased by 1(346 this time, which was initially 345 only). The scanner, concludes that the scanned port is closed, resulting the increase in iPID value by one only.

Let's see the result of Idle scanning when the target port is filtered. Scanner first probes the zombie host by sending TCP SYN/ACK segment, as it was done in the Open Ports and Closed ports scenario. The zombie host also behaves as usual. In the second step, as before, the scanner sends a SYN segment to the target, forcing it to be coming from the zombie host. The target being unaware of all this process,

sends no segment to zombie host (mentioned as sender in the SYN packet received), as the port is filtered. Therefore, there is no response from the zombie host resulting no increases in the IPID value. In the third and last step, the scanner again probes the zombie host and this time receives the same response but with the IP ID increased by 1(346 this time, which was initially 345 only). The scanner, concludes that the scanned port is closed, resulting the increase in iPID value by one only. You can clearly observe, that it is not possible for idle scanning to differentiate between closed port and filtered port.

Now let's see, how Nmap command is used for idle scanning. Each host in a network does not support incremental IPID. Therefore, we have to first search for a host with incremental IPID. This is done with an Nmap script " ipidseq". This gives the details of all the hosts in the network stating whether, they support incremental IPID or not. Select any one host with incremental ipid and use it as a zombie host.

Idle scan is the ultimate stealth scan. It can be used to frame some other party for a scan. A unique advantage of idle scan is that it can be used to defeat certain packet filtering firewalls and routers. IP source address filtering is a common (though weak) security mechanism for limiting machines that may connect to a sensitive host or network. Simply specify the zombie hostname to the -sI option and Nmap does the rest. Here ip address 172.31.101.89 has been used as zombie host, while 172.31.101.206 is the target host.

Nmap has a streamlined firewall filtering identification function that can be used to identify firewall filtering on ports based on ACK probe responses. This function can be used to test a single port or multiple ports in sequence to determine filtering status that is to understand, whether the machine is behind a firewall or not. A parameter is used for this purpose, as shown here.

Once, you have identified that the target machine is behind a firewall, it is very difficult to use the traditional Nmap command to know the exact status of the open and closed ports. Fortunately, there are some ways to bypass the firewalls. Let's understand one by one. Using a smaller MTU (Maximum Transmission Unit) is one of the method for doing this. This is similar to the packet fragmentation technique. During the scan, Nmap will create packets with a size based on the number that we will give. IN this example, we gave the number 8, so the Nmap will create 24-byte packets, causing confusion to the firewall and will allow them to allow the packets. Keep in mind that the MTU number must be a multiple of 8 (8, 16, 24, 32, etc.).

Nmap has another option that simplifies and streamlines the process of performing TCP scans, while evading the firewalls. You can easily use the -sS command to perform TCP stealth scans with Nmap. In stealthy scan half open connections are created by the scanner, which are not recorded by the firewalls.

Nmap is equipped with a Nmap scripting engine, which is capable of complementing the Nmap output by executing the scripts for special purposes. Vuln script is one of the such very important script for finding out the vulnerabilities in the target machine, which can be used along with the –script command line option.

In order to use vuln script, uopu will need to clone its github repo. The following commands will install the vulscan script along with all the databases mentioned:
git clone https://github.com/scipag/vulscan scipag_vulscan to clone it from github and
ln -s `pwd`/scipag_vulscan /usr/share/nmap/scripts/vulscan to link it to the nmap scripting engine. After downloading use this script to check for the vulnerabilities in the target machine with nmap command as shown here.

The list of vulnerabilities will be shown in the command window. The drawback of this vulnerability scan is that it is not very organized like other specific vulnerability scanners.

Thank You!