**Define what network scanning is as a method of penetration testing.**

Network scanning is the process of troubleshooting the active devices on your system for vulnerabilities. It identifies and examines the connected devices by deploying one or more features in the network protocol. These features pick up vulnerability signals and give you feedback on the security status of your network.

**How Network Scanning Works.**

Network scanning can offer critical insights into devices and performance, making it easier to troubleshoot issues. Scanning your network regularly is important for ensuring the network and devices on it are functioning healthily. Here’s how to go about scanning your network to understand what devices are on your network, view how they’re performing, and understand the traffic moving between them. There are two types of scanning methods: Active and Passive scanning.

**Active scanning** is when the tool sends a **ping** to each device on the network and awaits a response. The scanner then looks at the responses it gets to see if there are inconsistencies or vulnerabilities. For IP networks, this is often done by sending a ping to each possible IP address and getting a response to determine its status. It’s possible to manually ping your subnet to using an Address Resolution Protocol (ARP) scan. But to view all devices on the network across all subnets, your best bet is to use a tool that can automatically run scans and discover devices. Using the necessary Internet Control Message Protocol (ICMP) scan is more complicated, but it can be done—you’ll need to use echo, timestamp, or subnet mask requests. This method is often used to map network topology.

The purpose of network scanning is to manage, maintain, and secure the system using data found by the scanner. Network scanning is used to recognize available network services, discover, and recognize any filtering systems in place, look at what operating systems are in use, and to protect the network from attacks. It can also be used to determine the overall health of the network.

**Passive Scanning Network** scanning can also refer to packet sniffing, which captures and tracks the traffic moving over the network in the form of data packets. If you’d like to track the packet-level traffic on your network, you’ll need to implement sensors on managed devices and applications and deploy a tool for you to easily translate packet data into intelligible and relevant information. This approach looks at network information as soon as a device or system appears and starts sending messages to the network. In reality, networks release a lot of information in their normal communications, enough that passive scanners can simply look at this traffic flow, rather than pinging the devices themselves. This can help reveal traffic types, protocols, and bottlenecks within the network. It can also reveal potential security risks by catching anomalies.

**How to use Network Scanning Tools**

Using a network scanning tool is essential if you have more than a few devices on your network, or if your network is large enough to include multiple subnets. Trying to manually manage a large network is difficult and can expose your business to major security risks. There are several different scanning tools on the market, each with a slightly different approach to the task.

Nmap. The network Mapper.

**Nmap, or network mapper**, is a toolkit for functionality and penetration testing throughout a network, including port scanning and vulnerability detection and assessment.

Nmap scripting engine (NSE) Script is one of Nmap's most popular and powerful capabilities. These Nmap vulnerability scan scripts are used by hackers to examine commonly known vulnerabilities.

Common Vulnerabilities and Exposures (CVE); is a database of publicly disclosed data security issues. It serves as a reference model for detecting vulnerabilities and threats related to the security of information systems.

In this article, we’ll look at how to use Nmap for Vulnerability Scan.

Let’s get started!

**map Installation**

Nmap is pre-installed in almost every Linux distribution. In case it’s missing, you need to install it manually. It can be easily installed with the following command.

apt-get install nmap

And you can also install it by cloning the official git Repository.

git clone https://github.com/nmap/nmap.git

Next, navigate to that directory and install the requirements using the below commands.

. /configure

make

make install.

This software’s most recent version and binary installers for Windows, macOS, and Linux (RPM) are:

Vulnerability scan with Nmap

Nmap-vulners, vulscan, and vuln are the common and most popular CVE detection scripts in the Nmap search engine. These scripts allow you to discover important information about system security flaws.

Nmap-vulners

One of the most well-known vulnerability scanners is Nmap-Vulner. Let’s look at how to set up this tool as well as how to run a basic CVE scan. The Nmap script engine searches HTTP responses to identify CPEs for the given script.

Installation

To install the Nmap-vulners script, navigate to the Nmap scripts directory using the following command.

cd /usr/share/nmap/scripts/

The Next step is to clone the git repository.

git clone https://github.com/vulnersCom/nmap-vulners.git

After cloning the git repository, you won’t need to do anything else for the configuration. The tool will be automatically installed.

And if you want to see the NSE scripts present in Nmap-vulners database, use ls command. It will display all the. nse extension scripts on the terminal.

Usage

It’s easy to use NSE scripts. Simply pass the -script argument to our Nmap command to instruct what NSE script to use.

nmap -sV --script vulners [--script-args mincvss=<arg\_val>] <target>

Don’t forget to pass “-sV” argument while using NSE scripts. Nmap-vulners will be unable to access the Vulners exploit database if it does not receive any version information from Nmap. So, the -sV parameter is required all the time.

Example command

The syntax is quite straightforward. Just call the script with the “–script” option and specify the vulners engine and target to begin scanning.

nmap -sV --script nmap-vulners/ <target>

nmap -sV --script nmap-vulners/ <target>

If you wish to scan any specific ports, just add “-p” option to the end of the command and pass the port number you want to scan.

nmap -sV --script nmap-vulners/ <target> -p80,223

Nmap – vuln

NSE scripts are classified according to a set of predetermined categories to which each script belongs. Authentication, broadcast, brute force, intrusive, malware, safe, version, and vuln are some of the categories.

The scripts which come under the “vuln” category look for specific known vulnerabilities and only report back if any are identified in the target system.

nmap -sV --script vuln <target>

Nmap-vulscan

Vulscan is an NSE script that assists Nmap in detecting vulnerabilities on targets based on services and version detections. vulscan is like a module for Nmap that transforms it into a vulnerability scanner. The Nmap option -sV allows for per-service version detection, which is used to identify potential exploits for the detected vulnerabilities in the system.

Currently, the following pre-installed databases are available:

* exploitdb.csv
* osvdb.csv
* securitytracker.csv
* openvas.csv
* scipvuldb.csv
* xforce.csv
* securityfocus.csv
* cve.csv

Installation

To install the Vulscan, First, go to the Nmap scripts directory by using the following command.

cd /usr/share/nmap/scripts/

The Next step is to clone the git repository and install all the requirements.

git clone https://github.com/scipag/vulscan.git

ln -s `pwd`/scipag\_vulscan /usr/share/nmap/scripts/vulscan

Vulscan makes use of pre-configured databases saved locally on our machine. To update the database, go to the updater directory. Type the following command into a terminal to navigate to the updater directory.

cd vulscan/utilities/updater/

Next, change the permissions of the file to be run in the system.

chmod +x updateFiles.sh

And finally, update the exploit databases with the below command.

./updateFiles.sh

Usage

Let’s use vulscan to do a Nmap vulnerability scan. The vulscan NSE script can be used in the same way as nmap-vulners.

nmap -sV --script vulscan <target>

By default, Vulscan will search all of the databases simultaneously. It takes a lot of time to query information using all the databases. Using the vulscandb parameter, you can pass only one CVE database at a time.

--script-args vulscandb=database\_name

Example Command

nmap -sV --script vulscan --script-args vulscandb=exploit.csv <target> -p 80,233

Individual vulnerability Scanning

Individual vulnerability scans can also be performed utilizing scripts within each category. Here is a list of all 600+ Nse Scripts and 139 NSE Libraries

Examples

* http-csrf: Cross-Site Request Forgery (CSRF) vulnerabilities are detected by this script.

nmap -sV --script http-csrf <target>

Copy

* http-sherlock: Intends to exploit the “shellshock” vulnerability in web applications.

nmap -sV --script http-sherlock <target>

Copy

* http-slowloris-attack: Without launching a DoS attack, this script checks a web server or a target system for vulnerability to perform the Slowloris DoS attack.

nmap -sV --script http-slowloris-check <target>

* http-vmware-path-vuln: VMWare ESX, ESXi, and Server are all tested for a path-traversal vulnerability.

nmap -sV --script http-vmware-path-vuln <target>

* http-passwd: Attempts to retrieve /etc/passwd or boot.ini to see if a web server is vulnerable to directory traversal.

nmap -sV --script http-passwd <target>

* http-internal-ip-disclosure: When sending an HTTP/1.0 request without a Host header, this check determines if the web server leaks its internal Ip Address

nmap -sV --script http-internal-ip-disclosure <target>

* http-vuln-cve2013-0156: Detects Ruby on Rails servers that are vulnerable to Dos attacks and command injection.

nmap -sV --script http-vuln-cve2013-0156 <target-address>