**[Discover Open Ports Using Metasploit's Built-in Port Scanner](https://null-byte.wonderhowto.com/how-to/discover-open-ports-using-metasploits-built-port-scanner-0186829/)**

* By [**drd\_**](https://creator.wonderhowto.com/drd_/)
* 09/01/2018 4:33 am
* [Metasploit Basics](https://null-byte.wonderhowto.com/how-to/metasploit-basics/)

One of the first steps in [reconnaissance](https://null-byte.wonderhowto.com/how-to/recon/) is determining the open ports on a system. [Nmap](https://null-byte.wonderhowto.com/how-to/hack-like-pro-advanced-nmap-for-reconnaissance-0151619/) is widely considered the undisputed king of port scanning, but certain situations call for different tools. [Metasploit](https://null-byte.wonderhowto.com/how-to/metasploit-basics/) makes it easy to conduct port scanning from directly inside the framework, and we'll show you three types of port scans: TCP, SYN, and XMAS.

**What Is Port Scanning?**

[Port scanning](https://null-byte.wonderhowto.com/how-to/build-stealth-port-scanner-with-scapy-and-python-0164779/) is the process of probing a range of ports in order to determine the state of those ports — generally open or closed. There are 65,536 available ports on a host, with the first 1,024 ports being reserved for [well-known services](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers). Ports can communicate using the TCP protocol, UDP, or both.

* **Don't Miss:** [**The Ultimate List of Hacking Scripts for Metasploit's Meterpreter**](https://null-byte.wonderhowto.com/how-to/hack-like-pro-ultimate-list-hacking-scripts-for-metasploits-meterpreter-0149339/)

The first type of scan we will be covering is the TCP scan, also known as TCP connect. This type of scan utilizes a system call to establish a connection, much like web browsers or other networked applications. When a port is open, the TCP scan will initiate and complete a full three-way handshake, and then close the connection. This type of scan is effective, but noisy since the IP address can be logged.

The second type of scan is the SYN scan. This is the default Nmap scan and is considered the most popular type of port scan. In contrast to the TCP connect scan, an SYN scan uses raw packets to connect to ports rather than a system call. This is advantageous because the connection is never fully completed, making it relatively stealthy and more likely to evade firewalls. There is also more control over the requests and responses since there is access to raw networking.

The third type of scan we will be going over is the XMAS scan. This scan sets the FIN, PSH, and URG flags on the packet, which is said to light it up like a Christmas tree (hence the name). XMAS scans can be even stealthier than SYN scans, although modern intrusion detection systems can still detect them. Regardless, it is worth trying out if other scanning methods fail.

**Option 1TCP Scan**

The first thing we need to to before conducting any scans is start Metasploit by typing **msfconsole** in the terminal. A random banner will be displayed, as well as version information and the number of modules currently loaded.

msfconsole

, ,

/ \

((\_\_---,,,---\_\_))

(\_) O O (\_)\_\_\_\_\_\_\_\_\_

\ \_ / |\

o\_o \ M S F | \

\ \_\_\_\_\_ | \*

||| WW|||

||| |||

=[ metasploit v4.17.8-dev ]

+ -- --=[ 1803 exploits - 1027 auxiliary - 311 post ]

+ -- --=[ 538 payloads - 41 encoders - 10 nops ]

+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >

Scanners are a type of [auxiliary module](https://null-byte.wonderhowto.com/how-to/hack-like-pro-exploring-metasploit-auxiliary-modules-ftp-fuzzing-0155574/) in Metasploit, and to locate the port scanners, we can type **search portscan** at the prompt.

msf > search portscan

[!] Module database cache not built yet, using slow search

Matching Modules

================

Name Disclosure Date Rank Description

---- --------------- ---- -----------

auxiliary/scanner/http/wordpress\_pingback\_access normal Wordpress Pingback Locator

auxiliary/scanner/natpmp/natpmp\_portscan normal NAT-PMP External Port Scanner

auxiliary/scanner/portscan/ack normal TCP ACK Firewall Scanner

auxiliary/scanner/portscan/ftpbounce normal FTP Bounce Port Scanner

auxiliary/scanner/portscan/syn normal TCP SYN Port Scanner

auxiliary/scanner/portscan/tcp normal TCP Port Scanner

auxiliary/scanner/portscan/xmas normal TCP "XMas" Port Scanner

auxiliary/scanner/sap/sap\_router\_portscanner normal SAPRouter Port Scanner

This returns a few results, including the three types of port scans we will be looking at. Let's start with a simple TCP scan. Type **use auxiliary/scanner/portscan/tcp** to load the module. We can now take a look at the module settings by typing **options**:

msf auxiliary(scanner/portscan/tcp) > options

Module options (auxiliary/scanner/portscan/tcp):

Name Current Setting Required Description

---- --------------- -------- -----------

CONCURRENCY 10 yes The number of concurrent ports to check per host

DELAY 0 yes The delay between connections, per thread, in milliseconds

JITTER 0 yes The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.

PORTS 1-10000 yes Ports to scan (e.g. 22-25,80,110-900)

RHOSTS yes The target address range or CIDR identifier

THREADS 1 yes The number of concurrent threads

TIMEOUT 1000 yes The socket connect timeout in milliseconds

Here, we can the current settings and their descriptions. Unlike many exploit modules, this scanner can take a range of target addresses in addition to a single IP address. In this case, since we only have one target machine, a single address will do.

The number of threads can also be increased to help the scan run faster. It's recommended to keep this value under 256 for Unix systems and under 16 for native Win32 systems. To be safe, we can set this to something like 8. All the other options can be left as default for now.

msf auxiliary(scanner/portscan/tcp) > set rhosts 172.16.1.102

rhosts => 172.16.1.102

msf auxiliary(scanner/portscan/tcp) > set threads 8

threads => 8

Now we're ready to start the scan. In Metasploit, the **run** command is simply an alias for **exploit**, so it will do the exact same thing. Given we are only conducting scans, run seems more appropriate, though it really doesn't matter.

msf auxiliary(scanner/portscan/tcp) > run

[+] 172.16.1.102: - 172.16.1.102:21 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:23 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:22 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:25 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:53 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:80 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:111 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:139 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:445 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:513 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:514 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:512 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:1099 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:1524 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:2049 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:2121 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:3306 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:3632 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:5432 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:5900 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:6000 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:6667 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:6697 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:8009 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:8180 - TCP OPEN

[+] 172.16.1.102: - 172.16.1.102:8787 - TCP OPEN

[\*] Scanned 1 of 1 hosts (100% complete)

[\*] Auxiliary module execution completed

The TCP scan will run pretty quickly, and once it's complete, we can see that there are many open ports on the target.

**Option 2SYN Scan**

Next, we'll move on to a SYN scan. Type **back** to return to the main prompt, followed by **use auxiliary/scanner/portscan/syn** to load the module. Again, we can type **options** to view the current settings for this module:

msf auxiliary(scanner/portscan/syn) > options

Module options (auxiliary/scanner/portscan/syn):

Name Current Setting Required Description

---- --------------- -------- -----------

BATCHSIZE 256 yes The number of hosts to scan per set

DELAY 0 yes The delay between connections, per thread, in milliseconds

INTERFACE no The name of the interface

JITTER 0 yes The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.

PORTS 1-10000 yes Ports to scan (e.g. 22-25,80,110-900)

RHOSTS yes The target address range or CIDR identifier

SNAPLEN 65535 yes The number of bytes to capture

THREADS 1 yes The number of concurrent threads

TIMEOUT 500 yes The reply read timeout in milliseconds

There are a few different options here compared to the TCP scan, but for the most part, it's very similar, including the option to accept a range of target addresses and the number of threads to set.

When performing a number of scans or exploits on a singular target, it can get tiring setting the same options over and over again. Luckily, there is a command that will set an option globally, meaning it won't have to be re-entered when using a different module. Use **setg** to set a global option.

msf auxiliary(scanner/portscan/syn) > setg rhosts 172.16.1.102

rhosts => 172.16.1.102

msf auxiliary(scanner/portscan/syn) > setg threads 8

threads => 8

Now, type **run** to start the scan.

msf auxiliary(scanner/portscan/syn) > run

[+] TCP OPEN 172.16.1.102:21

[+] TCP OPEN 172.16.1.102:22

[+] TCP OPEN 172.16.1.102:23

[+] TCP OPEN 172.16.1.102:25

[+] TCP OPEN 172.16.1.102:53

[+] TCP OPEN 172.16.1.102:80

[+] TCP OPEN 172.16.1.102:111

[+] TCP OPEN 172.16.1.102:139

[+] TCP OPEN 172.16.1.102:445

[+] TCP OPEN 172.16.1.102:512

[+] TCP OPEN 172.16.1.102:513

[+] TCP OPEN 172.16.1.102:514

[+] TCP OPEN 172.16.1.102:1099

[+] TCP OPEN 172.16.1.102:1524

[+] TCP OPEN 172.16.1.102:2049

[+] TCP OPEN 172.16.1.102:2121

[+] TCP OPEN 172.16.1.102:3306

[+] TCP OPEN 172.16.1.102:3632

[+] TCP OPEN 172.16.1.102:5432

[+] TCP OPEN 172.16.1.102:5900

[+] TCP OPEN 172.16.1.102:6000

[+] TCP OPEN 172.16.1.102:6667

[+] TCP OPEN 172.16.1.102:6697

[+] TCP OPEN 172.16.1.102:8009

[+] TCP OPEN 172.16.1.102:8180

[+] TCP OPEN 172.16.1.102:8787

[\*] Scanned 1 of 1 hosts (100% complete)

[\*] Auxiliary module execution completed

The SYN scan will take a little longer to complete compared to the TCP scan, but once it's done, we can see that we obtained similar results compared to the previous scan.

**Option 3XMAS Scan**

The third type of scan we'll do is the XMAS scan. Again, type **back** to exit the current module, and then **use auxiliary/scanner/portscan/xmas** to load the module. Since we previously set global options for the remote host and threads, we should see these settings already populated when we view **options** now.

msf auxiliary(scanner/portscan/xmas) > options

Module options (auxiliary/scanner/portscan/xmas):

Name Current Setting Required Description

---- --------------- -------- -----------

BATCHSIZE 256 yes The number of hosts to scan per set

DELAY 0 yes The delay between connections, per thread, in milliseconds

INTERFACE no The name of the interface

JITTER 0 yes The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.

PORTS 1-10000 yes Ports to scan (e.g. 22-25,80,110-900)

RHOSTS 172.16.1.102 yes The target address range or CIDR identifier

SNAPLEN 65535 yes The number of bytes to capture

THREADS 8 yes The number of concurrent threads

TIMEOUT 500 yes The reply read timeout in milliseconds

The other options are pretty much identical to the SYN scan, so we can leave these as default. Feel free to play around with the other settings and see how it affects the timing and accuracy. Now we can **run** the scan.

msf auxiliary(scanner/portscan/xmas) > run

[\*] TCP OPEN|FILTERED 172.16.1.102:21

[\*] TCP OPEN|FILTERED 172.16.1.102:22

[\*] TCP OPEN|FILTERED 172.16.1.102:23

[\*] TCP OPEN|FILTERED 172.16.1.102:25

[\*] TCP OPEN|FILTERED 172.16.1.102:53

[\*] TCP OPEN|FILTERED 172.16.1.102:80

[\*] TCP OPEN|FILTERED 172.16.1.102:111

[\*] TCP OPEN|FILTERED 172.16.1.102:139

[\*] TCP OPEN|FILTERED 172.16.1.102:445

[\*] TCP OPEN|FILTERED 172.16.1.102:512

[\*] TCP OPEN|FILTERED 172.16.1.102:513

[\*] TCP OPEN|FILTERED 172.16.1.102:514

[\*] TCP OPEN|FILTERED 172.16.1.102:1099

[\*] TCP OPEN|FILTERED 172.16.1.102:1524

[\*] TCP OPEN|FILTERED 172.16.1.102:2049

[\*] TCP OPEN|FILTERED 172.16.1.102:2121

[\*] TCP OPEN|FILTERED 172.16.1.102:3306

[\*] TCP OPEN|FILTERED 172.16.1.102:3632

[\*] TCP OPEN|FILTERED 172.16.1.102:5432

[\*] TCP OPEN|FILTERED 172.16.1.102:5900

[\*] TCP OPEN|FILTERED 172.16.1.102:6000

[\*] TCP OPEN|FILTERED 172.16.1.102:6667

[\*] TCP OPEN|FILTERED 172.16.1.102:6697

[\*] TCP OPEN|FILTERED 172.16.1.102:8009

[\*] TCP OPEN|FILTERED 172.16.1.102:8180

[\*] TCP OPEN|FILTERED 172.16.1.102:8787

[\*] Scanned 1 of 1 hosts (100% complete)

[\*] Auxiliary module execution completed

Once again, we obtained similar results to the other scans, with additional information about whether the port is filtered or not. Depending on the target (or targets) and the type of environment in place, these scans can sometimes yield different results, so it certainly doesn't hurt to try out multiple scans.

**These Find Open Ports with Ease**

In this guide, we've covered how to do three types of port scan — TCP, SYN, and XMAS — right from Metasploit's [interactive console](https://tag.wonderhowto.com/msfconsole/). These scanners are quick and dirty, but can accomplish the objective of finding open ports with relative ease. This just goes to show that Metasploit is packed full of features that make it easier for white hat hackers to do what they best.