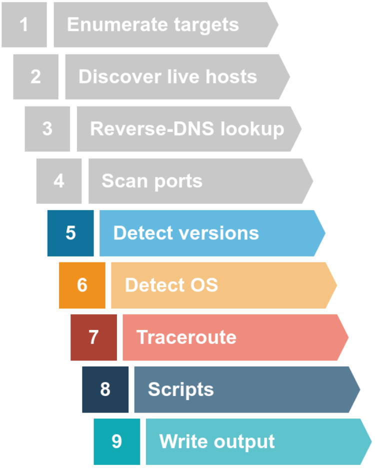
Nmap Post Port Scans

Introduction

In the first room of this series, we have learned how Nmap can enumerate targets, discover live hosts, and use reverse-DNS to find interesting names. The second and third rooms of the series focused on the basic and advanced types of scans for network ports.

In the last room, as shown in the figure below, we focus on how Nmap can be used to:

- Detect versions of the running services (on all open ports)
- Detect the OS based on any signs revealed by the target
- Run Nmap's traceroute
- Run select Nmap scripts
- Save the scan results in various formats



This room will focus on these steps and how to execute them after the port scan.

Service Detection

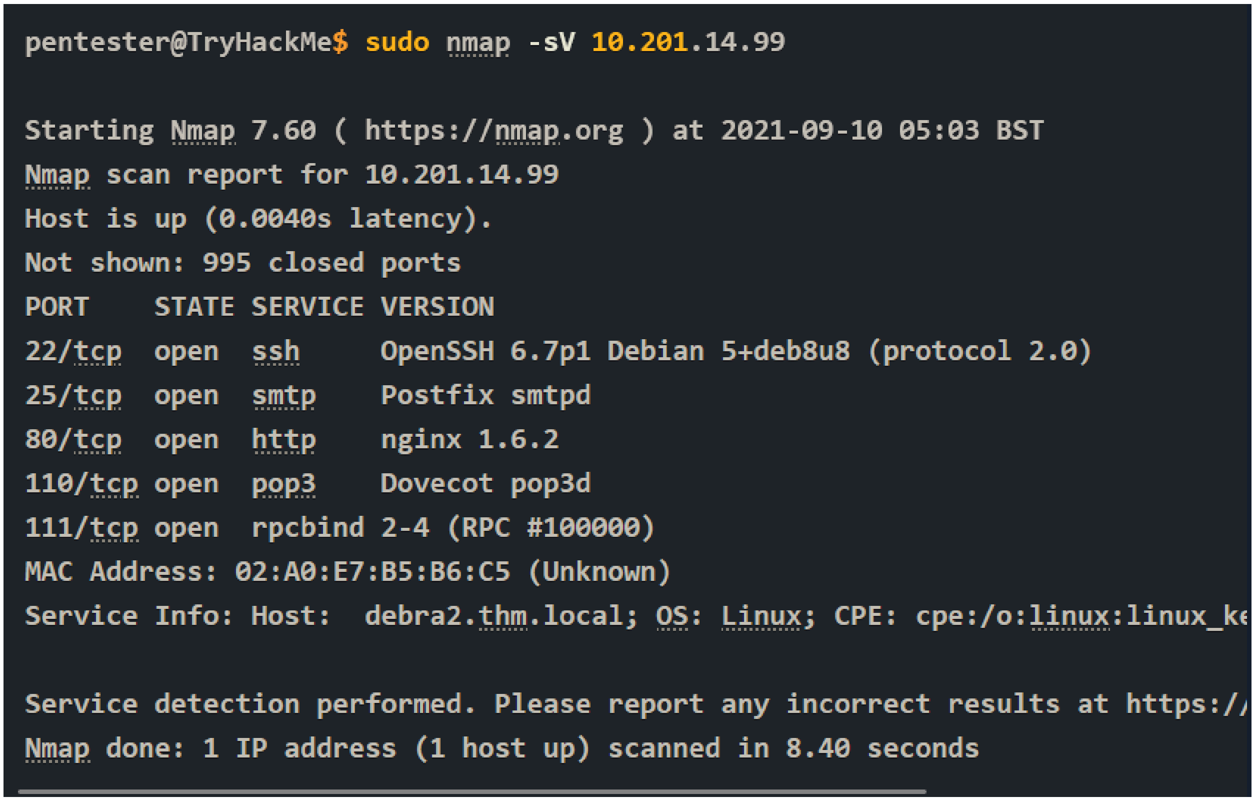
Once Nmap discovers open ports, you can probe the available port to detect the running service. Further investigation of open ports is an essential piece of information as the pentester can use it to learn if there are any known vulnerabilities of the service. Join Vulnerabilities 101 to learn more about searching for vulnerable services.

Adding -sV to your Nmap command will collect and determine service and version information for the open ports. You can control the intensity with --version-intensity

LEVEL where the level ranges between 0, the lightest, and 9, the most complete. -sV --version-light has an intensity of 2, while -sV --version-all has an intensity of 9.

It is important to note that using -sV will force Nmap to proceed with the TCP 3-way handshake and establish the connection. The connection establishment is necessary because Nmap cannot discover the version without establishing a connection fully and communicating with the listening service. In other words, stealth SYN scan -sS is not possible when -sV option is chosen.

The console output below shows a simple Nmap stealth SYN scan with the -sV option. Adding the -sV option leads to a new column in the output showing the version for each detected service. For instance, in the case of TCP port 22 being open, instead of 22/tcp open ssh, we obtain 22/tcp open ssh OpenSSH 6.7p1 Debian 5+deb8u8 (protocol 2.0). Notice that the SSH protocol is guessed as the service because TCP port 22 is open; Nmap didn't need to connect to port 22 to confirm. However, -sV required connecting to this open port to grab the service banner and any version information it can get, such as nginx 1.6.2. Hence, unlike the service column, the version column is not a guess.



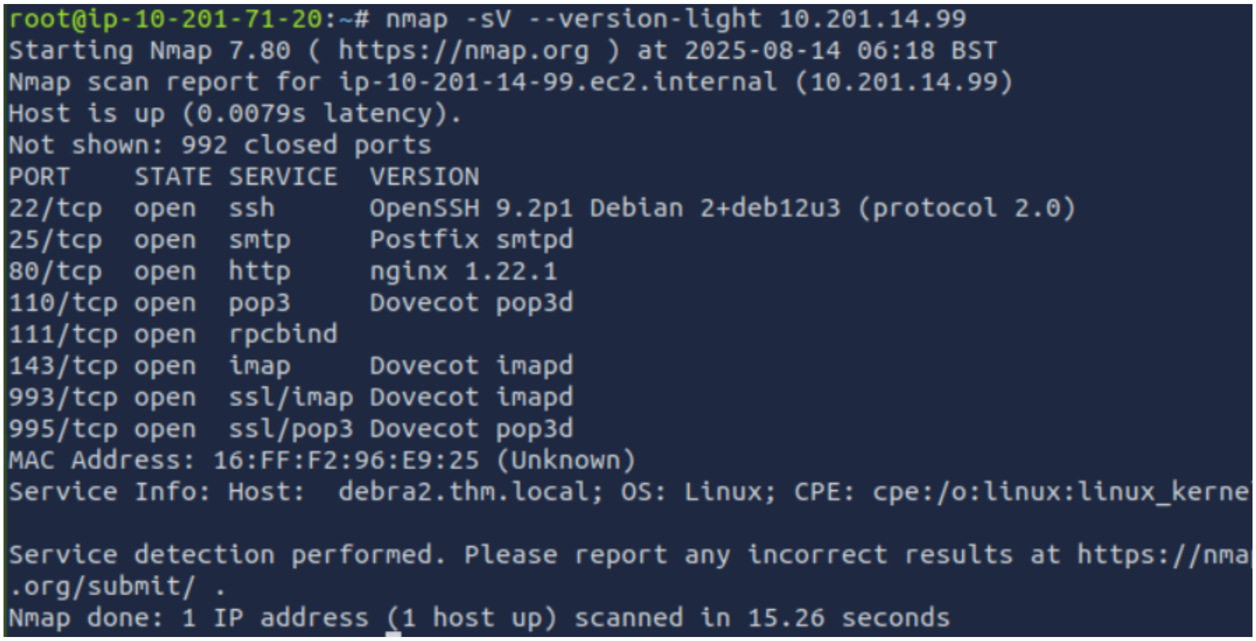
```
pentester@TryHackMe$ sudo nmap -sV 10.201.14.99

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-10 05:03 BST
Nmap scan report for 10.201.14.99
Host is up (0.0040s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.7p1 Debian 5+deb8u8 (protocol 2.0)
25/tcp  open  smtp    Postfix smtpd
80/tcp  open  http    nginx 1.6.2
110/tcp open  pop3    Dovecot pop3d
111/tcp open  rpcbind 2-4 (RPC #100000)
MAC Address: 02:A0:E7:B5:B6:C5 (Unknown)
Service Info: Host:  debra2.thm.local; OS: Linux; CPE: cpe:/o:linux:linux_ke

Service detection performed. Please report any incorrect results at https:/
Nmap done: 1 IP address (1 host up) scanned in 8.40 seconds
```

Note that many Nmap options require root privileges. Unless you are running Nmap as root, you need to use sudo as in the example above.

Start the VM. Once it is ready, open the terminal on the AttackBox to answer the following questions.
****

Answer the questions below:

Start the target machine for this task and launch the AttackBox. Run nmap -sV --version-light 10.201.14.99via the AttackBox. What is the detected version for port 143?



```
root@ip-10-201-71-20:~# nmap -sV --version-light 10.201.14.99
Starting Nmap 7.80 ( https://nmap.org ) at 2025-08-14 06:18 BST
Nmap scan report for ip-10-201-14-99.ec2.internal (10.201.14.99)
Host is up (0.0079s latency).
Not shown: 992 closed ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
25/tcp  open  smtp     Postfix smtpd
80/tcp  open  http     nginx 1.22.1
110/tcp open  pop3     Dovecot pop3d
111/tcp open  rpcbind
143/tcp open  imap     Dovecot imapd
993/tcp open  ssl/imap Dovecot imapd
995/tcp open  ssl/pop3 Dovecot pop3d
MAC Address: 16:FF:F2:96:E9:25 (Unknown)
Service Info: Host:  debra2.thm.local; OS: Linux; CPE: cpe:/o:linux:linux_kerne

Service detection performed. Please report any incorrect results at https://nma
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.26 seconds
```

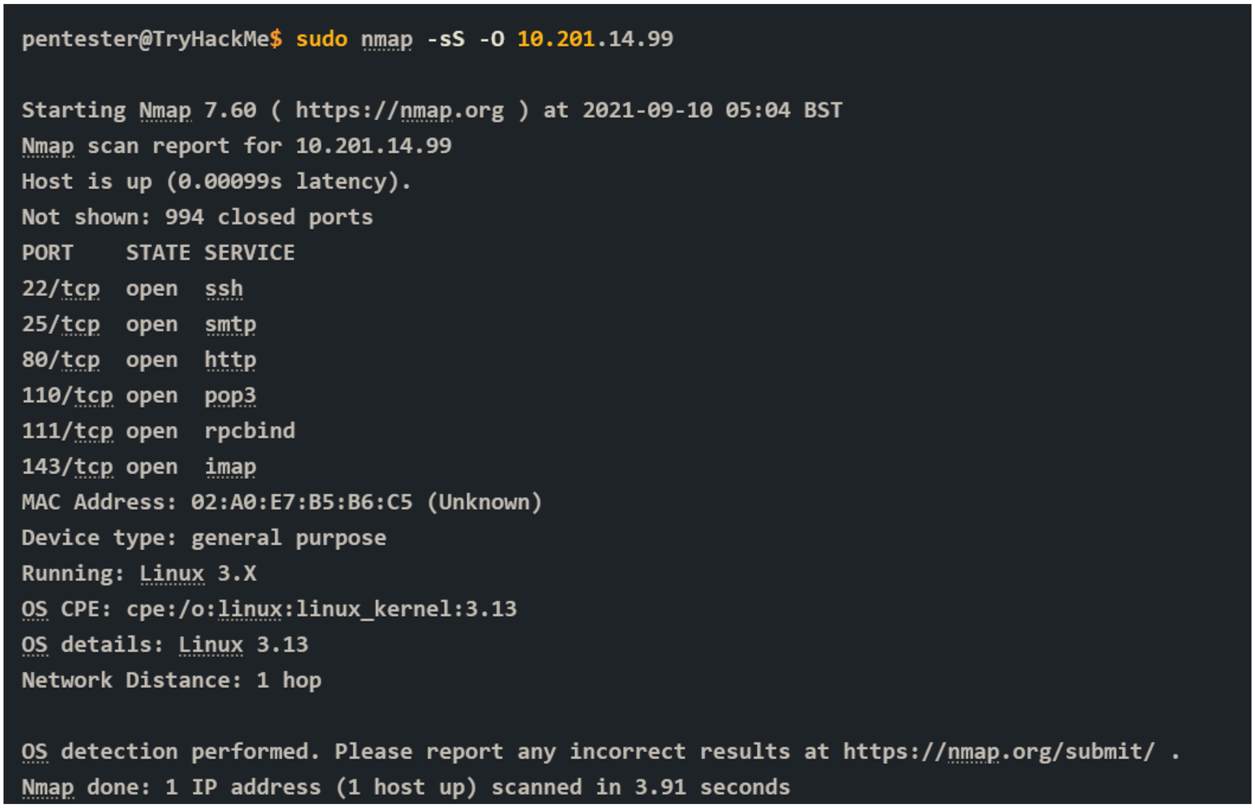
Answer: Dovecot imapd

Which service did not have a version detected with --version-light?
Answer: rpcbind
****

OS Detection and Traceroute

OS Detection

Nmap can detect the Operating System (OS) based on its behaviour and any telltale signs in its responses. OS detection can be enabled using -O; this is an uppercase O as in OS. In this example, we ran nmap -sS -O 10.201.14.99 on the AttackBox. Nmap detected the OS to be Linux 3.X, and then it guessed further that it was running kernel 3.13.



```
pentester@TryHackMe$ sudo nmap -sS -O 10.201.14.99

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-10 05:04 BST
Nmap scan report for 10.201.14.99
Host is up (0.00099s latency).
Not shown: 994 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
80/tcp  open  http
110/tcp open  pop3
111/tcp open  rpcbind
143/tcp open  imap
MAC Address: 02:A0:E7:B5:B6:C5 (Unknown)
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3.13
OS details: Linux 3.13
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.91 seconds
```

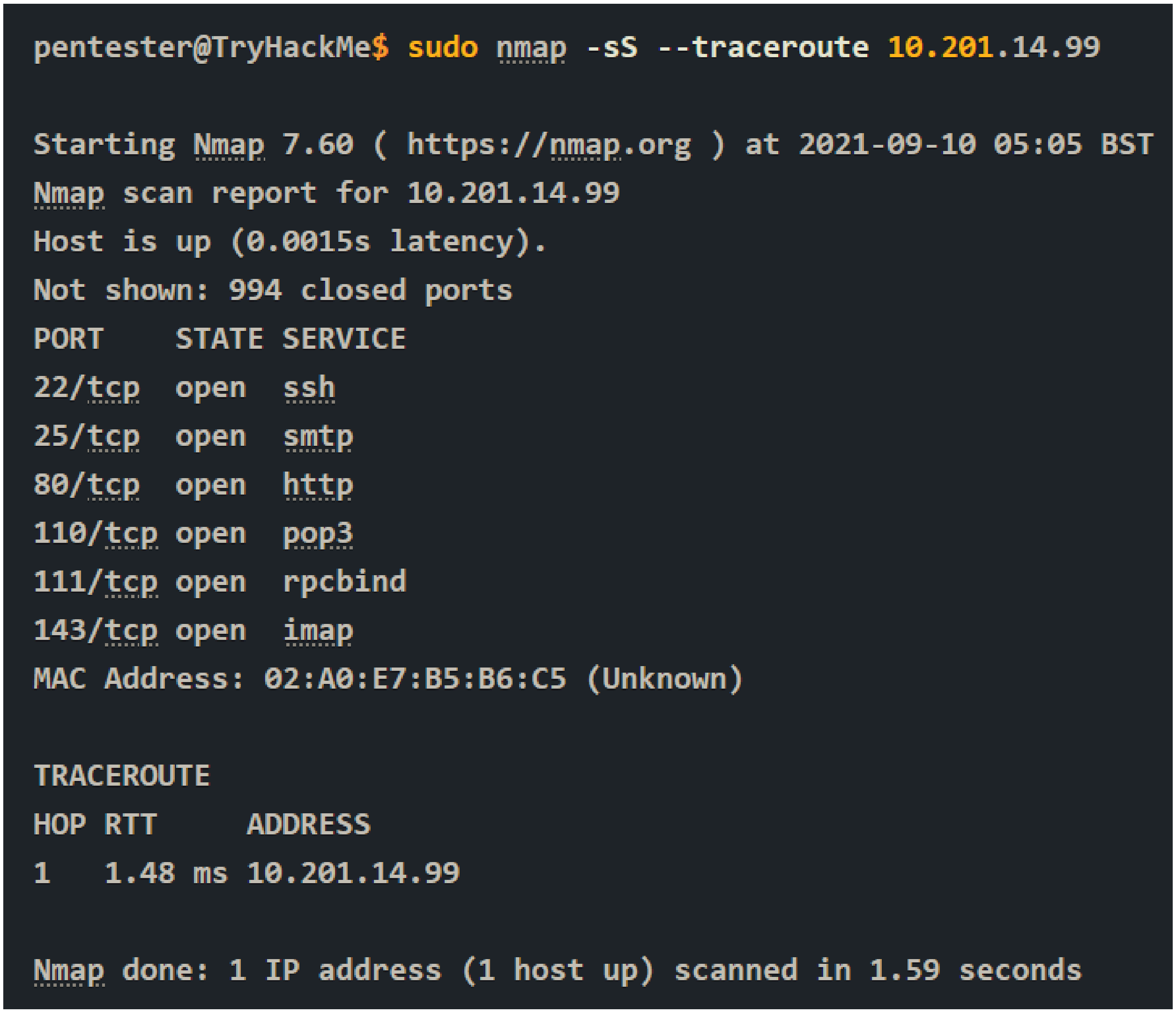
The system that we scanned and attempted to detect its OS version is running kernel version 3.16. Nmap was able to make a close guess in this case. In another case, we scanned a Fedora Linux system with kernel 5.13.14; however, Nmap detected it as Linux 2.6.X. The good news is that Nmap detected the OS correctly; the not-so-good news is that the kernel version was wrong.

The OS detection is very convenient, but many factors might affect its accuracy. First and foremost, Nmap needs to find at least one open and one closed port on the target to make a reliable guess. Furthermore, the guest OS fingerprints might get distorted due to the rising use of virtualization and similar technologies. Therefore, always take the OS version with a grain of salt.

Traceroute

If you want Nmap to find the routers between you and the target, just add --traceroute. In the following example, Nmap appended a traceroute to its scan results. Note that Nmap's traceroute works slightly different than the traceroute command found on Linux and macOS or tracert found on MS Windows. Standard traceroute starts with a packet of low TTL (Time to Live) and keeps increasing until it reaches the target. Nmap's traceroute starts with a packet of high TTL and keeps decreasing it.

In the following example, we executed nmap -sS --traceroute 10.201.14.99 on the AttackBox. We can see that there are no routers/hops between the two as they are connected directly.



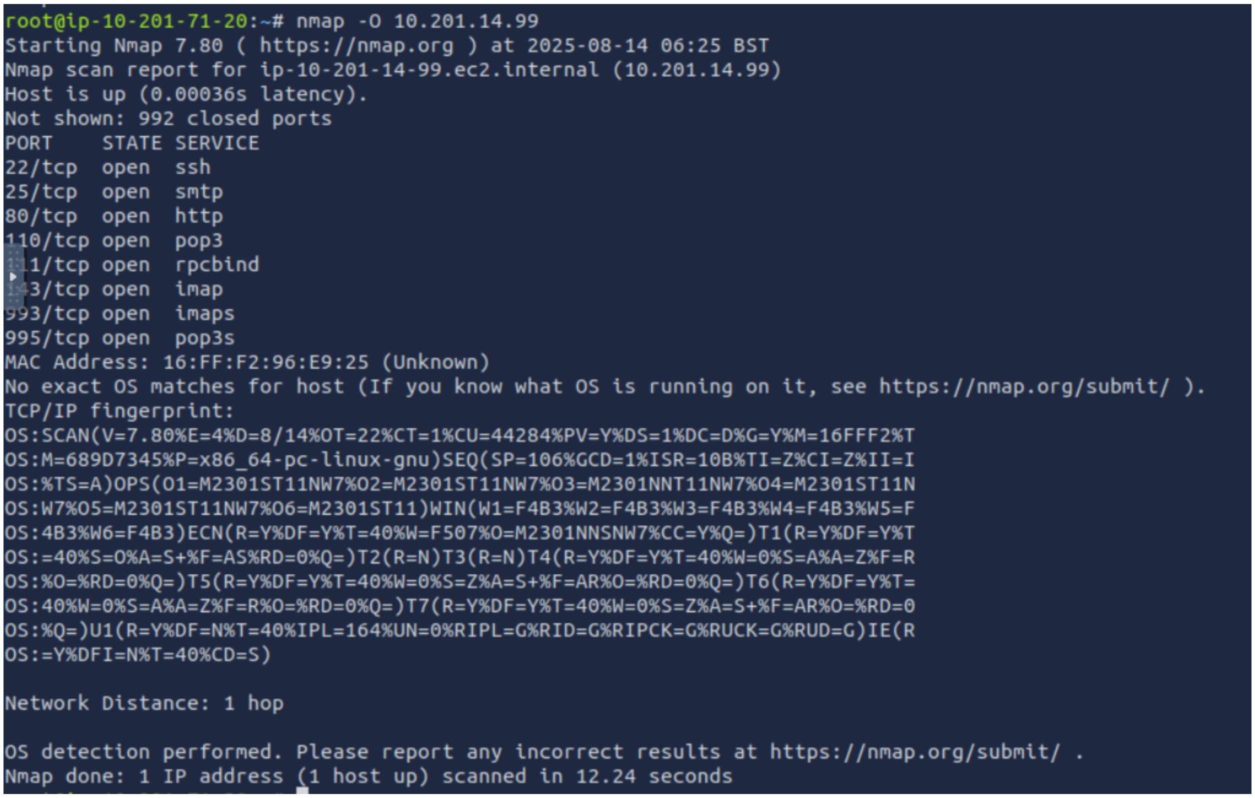
```
pentester@TryHackMe$ sudo nmap -sS --traceroute 10.201.14.99

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-10 05:05 BST
Nmap scan report for 10.201.14.99
Host is up (0.0015s latency).
Not shown: 994 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
80/tcp  open  http
110/tcp open  pop3
111/tcp open  rpcbind
143/tcp open  imap
MAC Address: 02:A0:E7:B5:B6:C5 (Unknown)

TRACEROUTE
HOP RTT     ADDRESS
1   1.48 ms 10.201.14.99

Nmap done: 1 IP address (1 host up) scanned in 1.59 seconds
```

It is worth mentioning that many routers are configured not to send ICMP Time-to-Live exceeded, which would prevent us from discovering their IP addresses. For more information, visit the Active Reconnaissance room

****

Answer the questions below:

Run nmap with -O option against 10.201.14.99. What OS did Nmap detect?



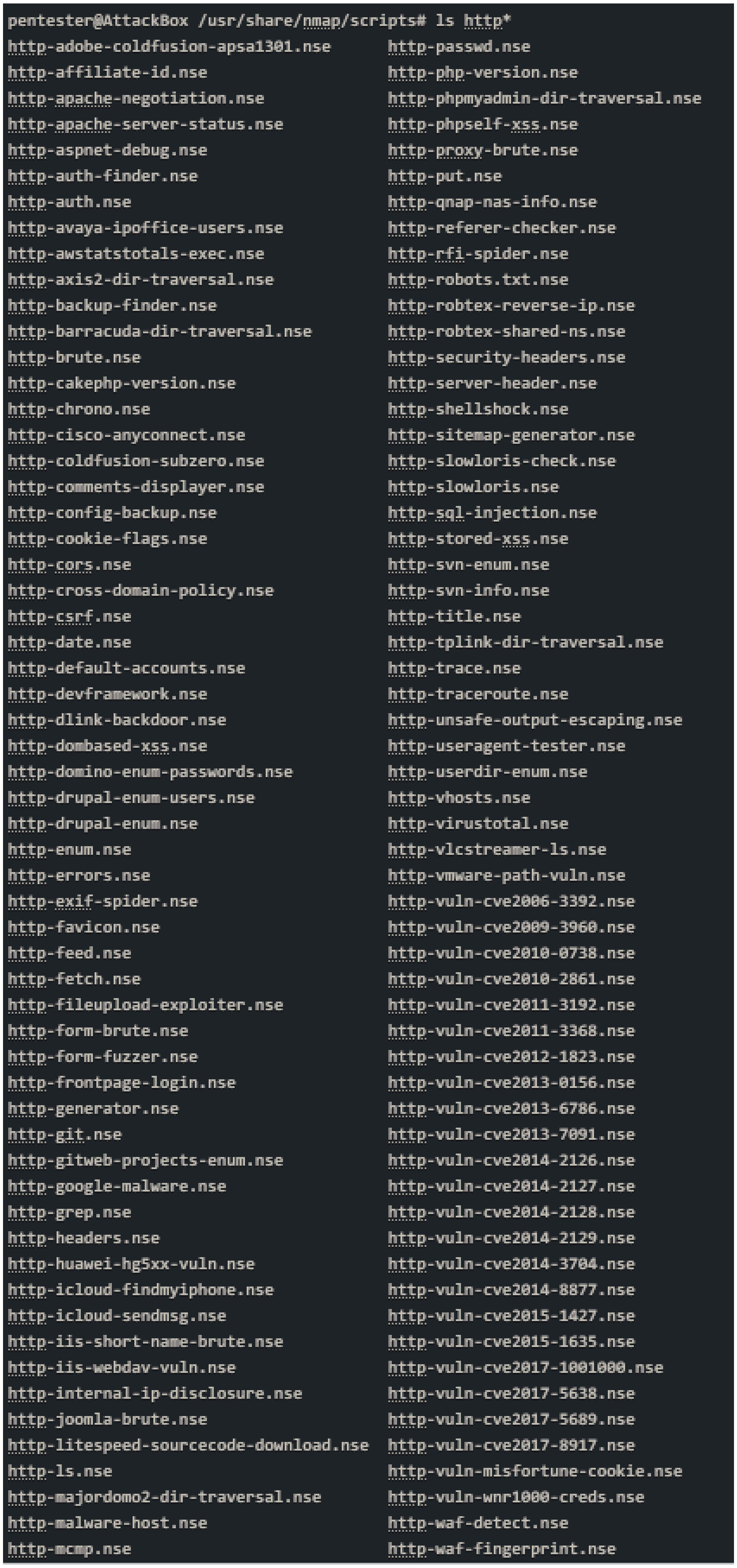
```
root@ip-10-201-71-20:~# nmap -O 10.201.14.99
Starting Nmap 7.80 ( https://nmap.org ) at 2025-08-14 06:25 BST
Nmap scan report for ip-10-201-14-99.ec2.internal (10.201.14.99)
Host is up (0.00036s latency).
Not shown: 992 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
80/tcp  open  http
110/tcp open  pop3
111/tcp open  rpcbind
143/tcp open  imap
993/tcp open  imaps
995/tcp open  pop3s
MAC Address: 16:FF:F2:96:E9:25 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=8/14%OT=22%CT=1%CU=44284%PV=Y%DS=1%DC=D%G=Y%M=16FFF2%T
OS:M=689D7345%P=x86_64-pc-linux-gnu)SEQ(SP=106%GCD=1%ISR=10B%TI=Z%CI=Z%II=I
OS:%TS=A)OPS(O1=M2301ST11NW7%O2=M2301ST11NW7%O3=M2301NNT11NW7%O4=M2301ST11N
OS:W7%O5=M2301ST11NW7%O6=M2301ST11)WIN(W1=F4B3%W2=F4B3%W3=F4B3%W4=F4B3%W5=F
OS:4B3%W6=F4B3)ECN(R=Y%DF=Y%T=40%W=F507%O=M2301NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T
OS:=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R
OS:%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=
OS:40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0
OS:%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R
OS:=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.24 seconds
```

Running the scan multiple times didn’t give a definitive OS match so I looked up other walkthroughs to see if there was something I was missing but they either got the same results as me or just listed the answer with no screenshot of the scan.
Answer: Linux
****

Nmap Scripting Engine(NSE)

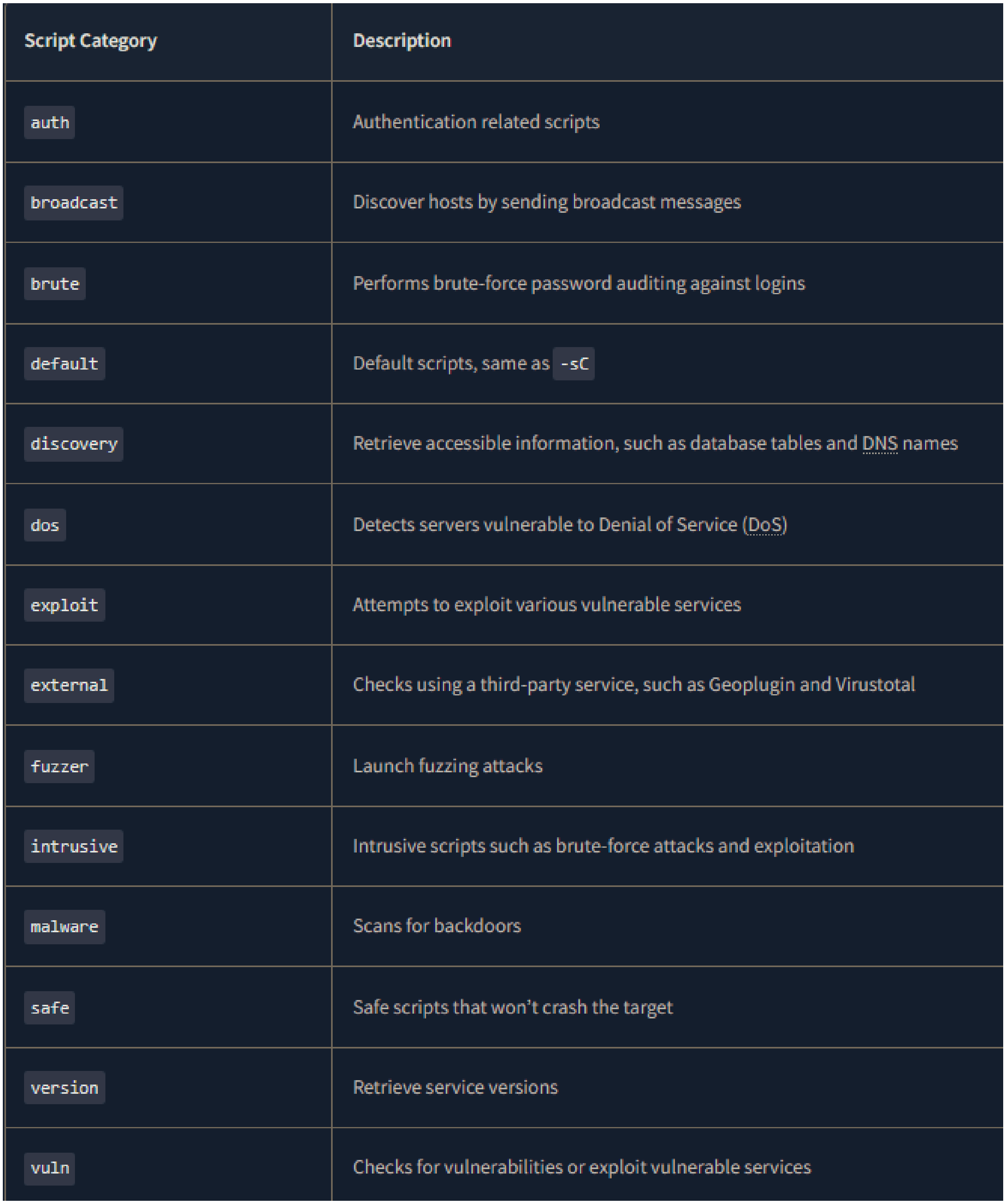
A script is a piece of code that does not need to be compiled. In other words, it remains in its original human-readable form and does not need to be converted to machine language. Many programs provide additional functionality via scripts; moreover, scripts make it possible to add custom functionality that did not exist via the built-in commands. Similarly, Nmap provides support for scripts using the Lua language. A part of Nmap, Nmap Scripting Engine (NSE) is a Lua interpreter that allows Nmap to execute Nmap scripts written in Lua language. However, we don’t need to learn Lua to make use of Nmap scripts.

Your Nmap default installation can easily contain close to 600 scripts. Take a look at your Nmap installation folder. On the AttackBox, check the files at /usr/share/nmap/scripts, and you will notice that there are hundreds of scripts conveniently named starting with the protocol they target. We listed all the scripts starting with the HTTP on the AttackBox in the console output below; we found around 130 scripts starting with http. With future updates, you can only expect the number of installed scripts to increase.



```
pentester@AttackBox /usr/share/nmap/scripts# ls http*
http-adobe-coldfusion-apsa1301.nse      http-passwd.nse
http-affiliate-id.nse                   http-php-version.nse
http-apache-negotiation.nse             http-phpmyadmin-dir-traversal.nse
http-apache-server-status.nse           http-phpself-xss.nse
http-aspnet-debug.nse                   http-proxy-brute.nse
http-auth-finder.nse                    http-put.nse
http-auth.nse                           http-qnap-nas-info.nse
http-avaya-ipoffice-users.nse           http-referer-checker.nse
http-awstatstotals-exec.nse             http-rfi-spider.nse
http-axis2-dir-traversal.nse            http-robots.txt.nse
http-backup-finder.nse                  http-robtex-reverse-ip.nse
http-barracuda-dir-traversal.nse        http-robtex-shared-ns.nse
http-brute.nse                          http-security-headers.nse
http-cakephp-version.nse                http-server-header.nse
http-chrono.nse                         http-shellshock.nse
http-cisco-anyconnect.nse               http-sitemap-generator.nse
http-coldfusion-subzero.nse             http-slowloris-check.nse
http-comments-displayer.nse             http-slowloris.nse
http-config-backup.nse                  http-sql-injection.nse
http-cookie-flags.nse                   http-stored-xss.nse
http-cors.nse                           http-svn-enum.nse
http-cross-domain-policy.nse            http-svn-info.nse
http-csrf.nse                           http-title.nse
http-date.nse                           http-tplink-dir-traversal.nse
http-default-accounts.nse               http-trace.nse
http-devframework.nse                   http-traceroute.nse
http-dlink-backdoor.nse                 http-unsafe-output-escaping.nse
http-dombased-xss.nse                   http-useragent-tester.nse
http-domino-enum-passwords.nse          http-userdir-enum.nse
http-drupal-enum-users.nse              http-vhosts.nse
http-drupal-enum.nse                    http-virustotal.nse
http-enum.nse                           http-vlcstreamer-ls.nse
http-errors.nse                         http-vmware-path-vuln.nse
http-exif-spider.nse                    http-vuln-cve2006-3392.nse
http-favicon.nse                        http-vuln-cve2009-3960.nse
http-feed.nse                           http-vuln-cve2010-0738.nse
http-fetch.nse                          http-vuln-cve2010-2861.nse
http-fileupload-exploiter.nse           http-vuln-cve2011-3192.nse
http-form-brute.nse                     http-vuln-cve2011-3368.nse
http-form-fuzzer.nse                    http-vuln-cve2012-1823.nse
http-frontpage-login.nse                http-vuln-cve2013-0156.nse
http-generator.nse                      http-vuln-cve2013-6786.nse
http-git.nse                            http-vuln-cve2013-7091.nse
http-gitweb-projects-enum.nse           http-vuln-cve2014-2126.nse
http-google-malware.nse                 http-vuln-cve2014-2127.nse
http-grep.nse                           http-vuln-cve2014-2128.nse
http-headers.nse                        http-vuln-cve2014-2129.nse
http-huawei-hg5xx-vuln.nse              http-vuln-cve2014-3704.nse
http-icloud-findmyiphone.nse            http-vuln-cve2014-8877.nse
http-icloud-sendmsg.nse                 http-vuln-cve2015-1427.nse
http-iis-short-name-brute.nse           http-vuln-cve2015-1635.nse
http-iis-webdav-vuln.nse                http-vuln-cve2017-1001000.nse
http-internal-ip-disclosure.nse         http-vuln-cve2017-5638.nse
http-joomla-brute.nse                   http-vuln-cve2017-5689.nse
http-litespeed-sourcecode-download.nse  http-vuln-cve2017-8917.nse
http-ls.nse                             http-vuln-misfortune-cookie.nse
http-majordomo2-dir-traversal.nse       http-vuln-wnr1000-creds.nse
http-malware-host.nse                   http-waf-detect.nse
http-mcmp.nse                           http-waf-fingerprint.nse
```

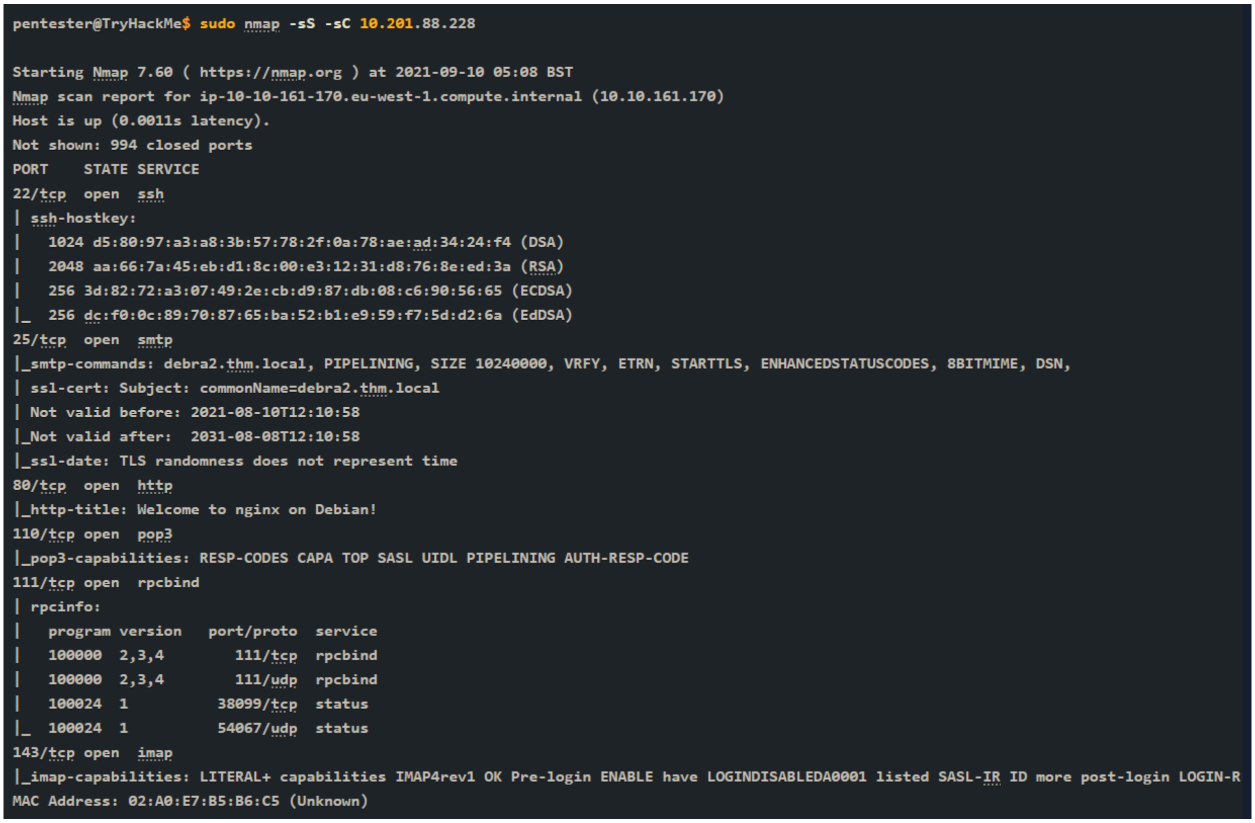
You can specify to use any or a group of these installed scripts; moreover, you can install other user's scripts and use them for your scans. Let's begin with the default scripts. You can choose to run the scripts in the default category using --script=default or simply adding -sC. In addition to default, categories include auth, broadcast, brute, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version, and vuln. A brief description is shown in the following table.



Script Category	Description
`auth`	Authentication related scripts
`broadcast`	Discover hosts by sending broadcast messages
`brute`	Performs brute-force password auditing against logins
`default`	Default scripts, same as `-sC`
`discovery`	Retrieve accessible information, such as database tables and DNS names
`dos`	Detects servers vulnerable to Denial of Service (DoS)
`exploit`	Attempts to exploit various vulnerable services
`external`	Checks using a third-party service, such as Geoplugin and Virustotal
`fuzzer`	Launch fuzzing attacks
`intrusive`	Intrusive scripts such as brute-force attacks and exploitation
`malware`	Scans for backdoors
`safe`	Safe scripts that won't crash the target
`version`	Retrieve service versions
`vuln`	Checks for vulnerabilities or exploit vulnerable services

Some scripts belong to more than one category. Moreover, some scripts launch brute-force attacks against services, while others launch DoS attacks and exploit systems. Hence, it is crucial to be careful when selecting scripts to run if you don't want to crash services or exploit them.

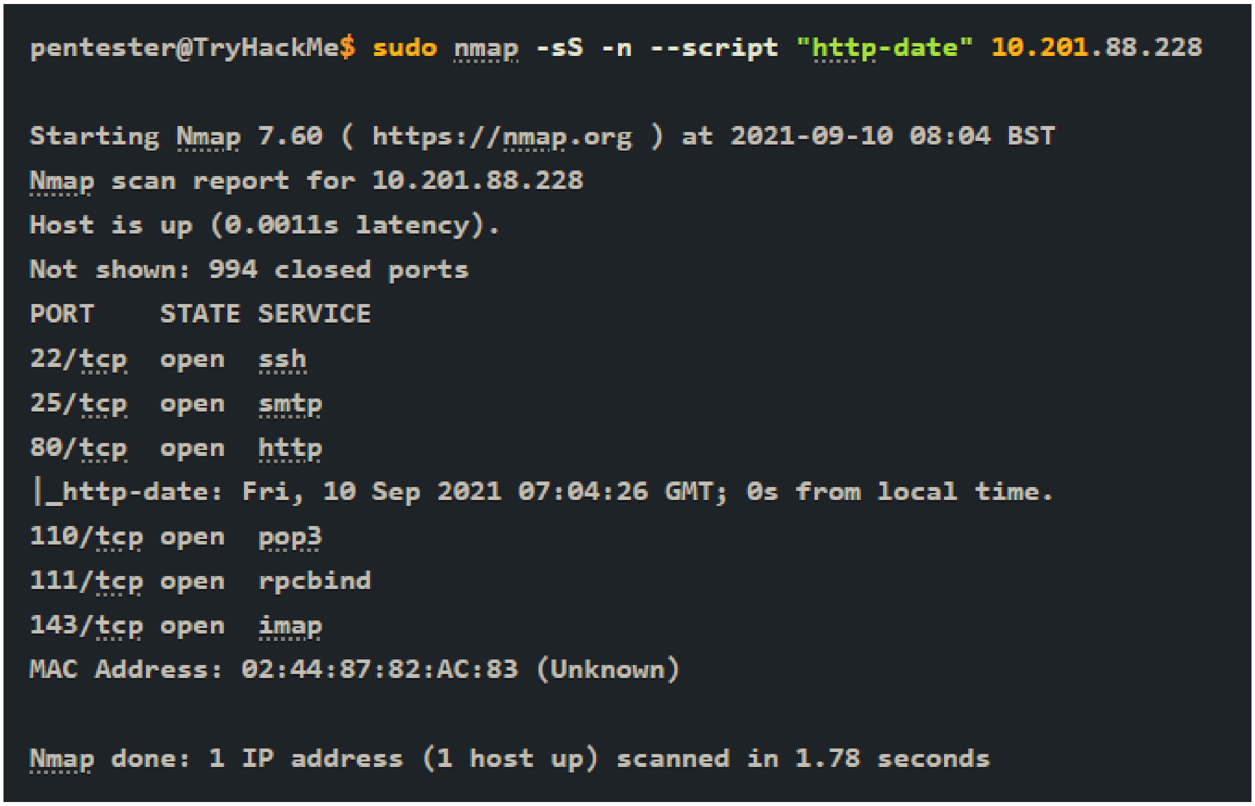
We use Nmap to run a SYN scan against 10.201.88.228 and execute the default scripts in the console shown below. The command is sudo nmap -sS -sC 10.201.88.228, where -sC will ensure that Nmap will execute the default scripts following the SYN scan. There are new details that appear below. Take a look at the SSH service at port 22; Nmap recovered all four public keys related to the running server. Consider another example, the HTTP service at port 80; Nmap retrieved the default page title. We can see that the page has been left as default.



```
pentester@TryHackMe$ sudo nmap -sS -sC 10.201.88.228

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-10 05:08 BST
Nmap scan report for ip-10-10-161-170.eu-west-1.compute.internal (10.10.161.170)
Host is up (0.0011s latency).
Not shown: 994 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
| ssh-hostkey:
|   1024 d5:80:97:a3:a8:3b:57:78:2f:0a:78:ae:ad:34:24:f4 (DSA)
|   2048 aa:66:7a:45:eb:d1:8c:00:e3:12:31:d8:76:8e:ed:3a (RSA)
|   256 3d:82:72:a3:07:49:2e:cb:d9:87:db:08:c6:90:56:65 (ECDSA)
|_  256 dc:f0:0c:89:70:87:65:ba:52:b1:e9:59:f7:5d:d2:6a (EdDSA)
25/tcp  open  smtp
|_smtp-commands: debra2.thm.local, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
| ssl-cert: Subject: commonName=debra2.thm.local
| Not valid before: 2021-08-10T12:10:58
|_Not valid after:  2031-08-08T12:10:58
|_ssl-date: TLS randomness does not represent time
80/tcp  open  http
|_http-title: Welcome to nginx on Debian!
110/tcp open  pop3
|_pop3-capabilities: RESP-CODES CAPA TOP SASL UIDL PIPELINING AUTH-RESP-CODE
111/tcp open  rpcbind
| rpcinfo:
|   program version   port/proto  service
|   100000  2,3,4        111/tcp  rpcbind
|   100000  2,3,4        111/udp  rpcbind
|   100024  1          38099/tcp  status
|_  100024  1          54067/udp  status
143/tcp open  imap
|_imap-capabilities: LITERAL+ capabilities IMAP4rev1 OK Pre-login ENABLE have LOGINDISABLEDA0001 listed SASL-IR ID more post-login LOGIN-R
MAC Address: 02:A0:E7:B5:B6:C5 (Unknown)
```

You can also specify the script by name using --script "SCRIPT-NAME" or a pattern such as --script "ftp*", which would include ftp-brute. If you are unsure what a script does, you can open the script file with a text reader, such as less, or a text editor. In the case of ftp-brute, it states: “Performs brute force password auditing against FTP servers.” You have to be careful as some scripts are pretty intrusive. Moreover, some scripts might be for a specific server and, if chosen at random, will waste your time with no benefit. As usual, make sure that you are authorized to launch such tests on the target server.

Let’s consider a benign script, http-date, which we guess would retrieve the http server date and time, and this is indeed confirmed in its description: “Gets the date from HTTP-like services. Also, it prints how much the date differs from local time…” On the AttackBox, we execute sudo nmap -sS -n --script "http-date" 10.201.88.228 as shown in the console below.



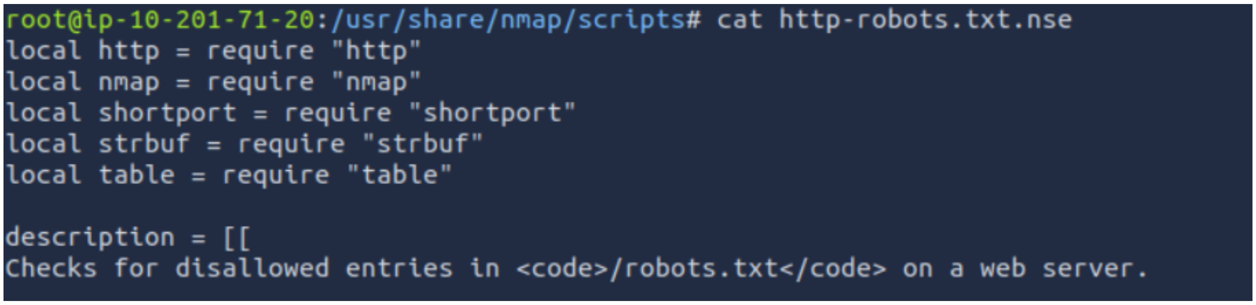
```
pentester@TryHackMe$ sudo nmap -sS -n --script "http-date" 10.201.88.228

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-10 08:04 BST
Nmap scan report for 10.201.88.228
Host is up (0.0011s latency).
Not shown: 994 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
80/tcp  open  http
|_http-date: Fri, 10 Sep 2021 07:04:26 GMT; 0s from local time.
110/tcp open  pop3
111/tcp open  rpcbind
143/tcp open  imap
MAC Address: 02:44:87:82:AC:83 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.78 seconds
```

Finally, you might expand the functionality of Nmap beyond the official Nmap scripts; you can write your script or download Nmap scripts from the Internet. Downloading and using a Nmap script from the Internet holds a certain level of risk. So it is a good idea not to run a script from an author you don't trust.

****

Answer the questions below:

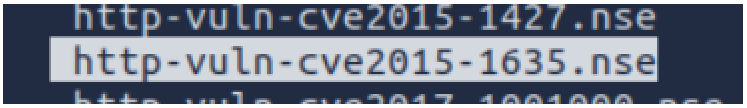
Knowing that Nmap scripts are saved in /usr/share/nmap/scripts on the AttackBox. What does the script http-robots.txt.nse check for?



```
root@ip-10-201-71-20:/usr/share/nmap/scripts# cat http-robots.txt.nse
local http = require "http"
local nmap = require "nmap"
local shortport = require "shortport"
local strbuf = require "strbuf"
local table = require "table"

description = [[
Checks for disallowed entries in <code>/robots.txt</code> on a web server.
```

Answer: disallowed entries

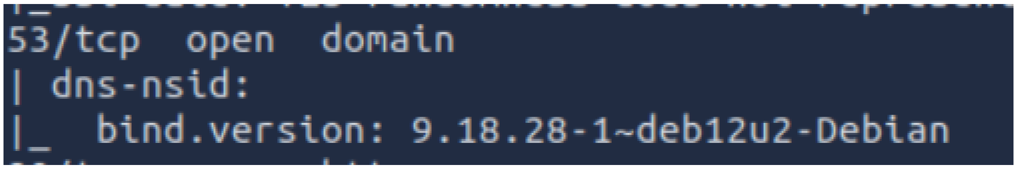
Can you figure out the name for the script that checks for the remote code execution vulnerability MS15-034 (CVE2015-1635)?



```
http-vuln-cve2015-1427.nse
http-vuln-cve2015-1635.nse
```

Answer: http-vuln-cve2015-1635

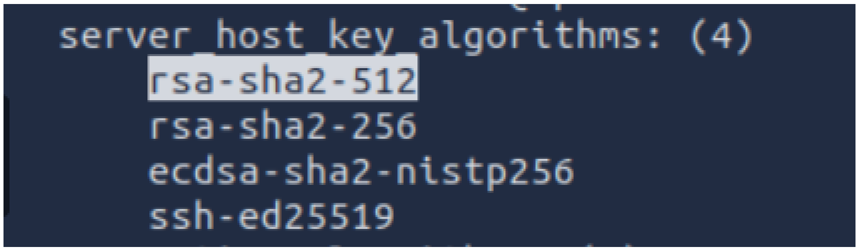
Launch the AttackBox if you haven't already. After you ensure you have terminated the VM from Task 2, start the target machine for this task. On the AttackBox, run Nmap with the default scripts -sC against 10.201.88.228. You will notice that there is a service listening on port 53. What is its full version value?



```
53/tcp  open  domain
| dns-nsid:
|_  bind.version: 9.18.28-1~deb12u2-Debian
```

Answer: 9.18.28-1~deb12u2-Debian

Based on its description, the script ssh2-enum-algos "reports the number of algorithms (for encryption, compression, etc.) that the target SSH2 server offers." What is the name of the server host key algorithm that relies on SHA2-512 and is supported by 10.201.88.228?



```
  server_host_key_algorithms: (4)
      rsa-sha2-512
      rsa-sha2-256
      ecdsa-sha2-nistp256
      ssh-ed25519
```

Answer: rsa-sha2-512

****

Saving the Output

Whenever you run a Nmap scan, it is only reasonable to save the results in a file. Selecting and adopting a good naming convention for your filenames is also crucial. The number of files can quickly grow and hinder your ability to find a previous scan result. The three main formats are:

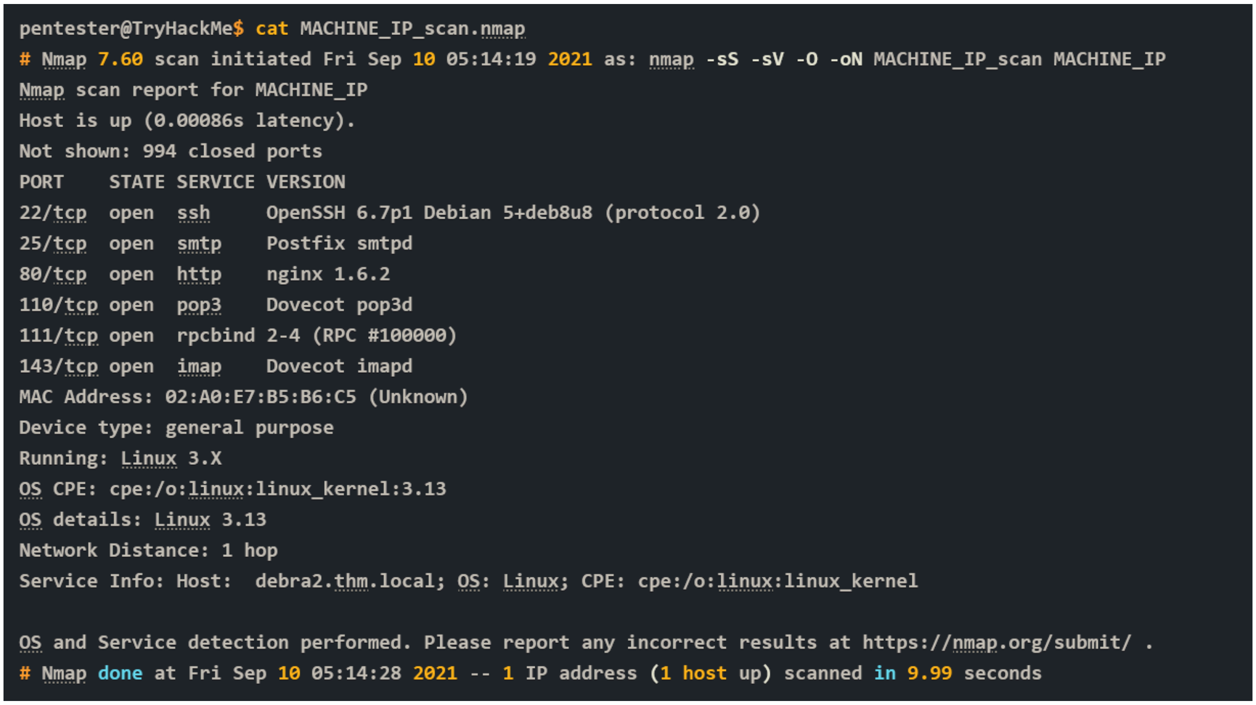
1. Normal
2. Grepable (grepable)
3. XML

There is a fourth one that we cannot recommend:

4. Script Kiddie

Normal

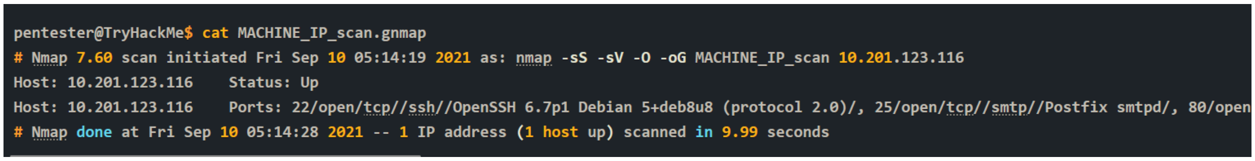
As the name implies, the normal format is similar to the output you get on the screen when scanning a target. You can save your scan in normal format by using -oN FILENAME; N stands for normal. Here is an example of the result.



```
pentester@TryHackMe$ cat MACHINE_IP_scan.nmap
# Nmap 7.60 scan initiated Fri Sep 10 05:14:19 2021 as: nmap -sS -sV -O -oN MACHINE_IP_scan MACHINE_IP
Nmap scan report for MACHINE_IP
Host is up (0.00086s latency).
Not shown: 994 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.7p1 Debian 5+deb8u8 (protocol 2.0)
25/tcp  open  smtp    Postfix smtpd
80/tcp  open  http    nginx 1.6.2
110/tcp open  pop3    Dovecot pop3d
111/tcp open  rpcbind 2-4 (RPC #100000)
143/tcp open  imap    Dovecot imapd
MAC Address: 02:A0:E7:B5:B6:C5 (Unknown)
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3.13
OS details: Linux 3.13
Network Distance: 1 hop
Service Info: Host:  debra2.thm.local; OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Sep 10 05:14:28 2021 -- 1 IP address (1 host up) scanned in 9.99 seconds
```

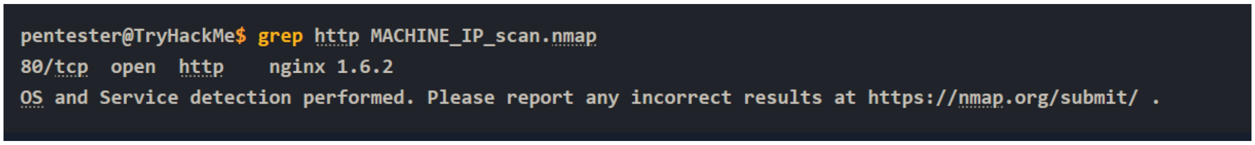
Grepable

The grepable format has its name from the command grep; grep stands for Global Regular Expression Printer. In simple terms, it makes filtering the scan output for specific keywords or terms efficient. You can save the scan result in grepable format using -oG FILENAME. The scan output, displayed above in normal format, is shown in the console below using grepable format. The normal output is 21 lines; however, the grepable output is only 4 lines. The main reason is that Nmap wants to make each line meaningful and complete when the user applies grep. As a result, in grepable output, the lines are so long and are not convenient to read compared to normal output.

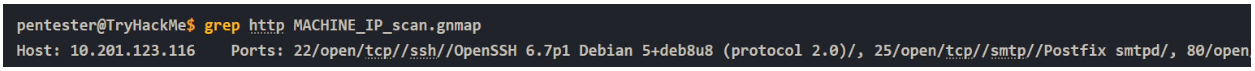


```
pentester@TryHackMe$ cat MACHINE_IP_scan.gnmap
# Nmap 7.60 scan initiated Fri Sep 10 05:14:19 2021 as: nmap -sS -sV -O -oG MACHINE_IP_scan 10.201.123.116
Host: 10.201.123.116    Status: Up
Host: 10.201.123.116    Ports: 22/open/tcp//ssh//OpenSSH 6.7p1 Debian 5+deb8u8 (protocol 2.0)/, 25/open/tcp//smtp//Postfix smtpd/, 80/open
# Nmap done at Fri Sep 10 05:14:28 2021 -- 1 IP address (1 host up) scanned in 9.99 seconds
```

An example use of grep is grep KEYWORD TEXT_FILE; this command will display all the lines containing the provided keyword. Let's compare the output of using grep on normal output and grepable output. You will notice that the former does not provide the IP address of the host. Instead, it returned 80/tcp open http nginx 1.6.2, making it very inconvenient if you are sifting through the scan results of multiple systems. However, the latter provides enough information, such as the host's IP address, in each line to make it complete.



```
pentester@TryHackMe$ grep http MACHINE_IP_scan.nmap
80/tcp   open  http    nginx 1.6.2
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```



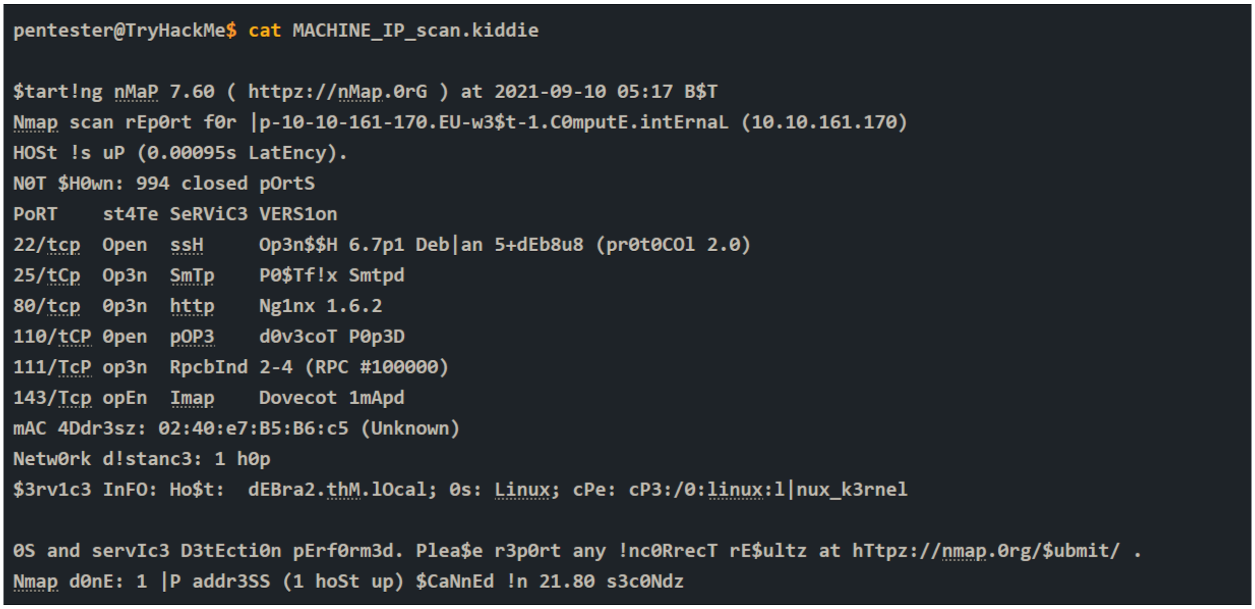
```
pentester@TryHackMe$ grep http MACHINE_IP_scan.gnmap
Host: 10.201.123.116    Ports: 22/open/tcp//ssh//OpenSSH 6.7p1 Debian 5+deb8u8 (protocol 2.0)/, 25/open/tcp//smtp//Postfix smtpd/, 80/open
```

XML

The third format is XML. You can save the scan results in XML format using -oX FILENAME. The XML format would be most convenient to process the output in other programs. Conveniently enough, you can save the scan output in all three formats using -oA FILENAME to combine -oN, -oG, and -oX for normal, grepable, and XML.

Script Kiddie

A fourth format is script kiddie. You can see that this format is useless if you want to search the output for any interesting keywords or keep the results for future reference. However, you can use it to save the output of the scan nmap -sS 127.0.0.1 -oS FILENAME, display the output filename, and look 31337 in front of friends who are not tech-savvy.



```
pentester@TryHackMe$ cat MACHINE_IP_scan.kiddie

$tart!ng nMaP 7.60 ( httpz://nMap.0rG ) at 2021-09-10 05:17 B$T
Nmap scan rEp0rt f0r |p-10-10-161-170.EU-w3$t-1.C0mputE.intErnaL (10.10.161.170)
HOSt !s uP (0.00095s LatEncy).
N0T $H0wn: 994 closed p0rtS
PoRT     st4Te SeRViC3 VERS1on
22/tcp   Open  ssH     Op3n$$H 6.7p1 Deb|an 5+dEb8u8 (pr0t0COl 2.0)
25/tCp   Op3n  SmTp    P0$Tf!x Smtpd
80/tcp   0p3n  http    Ng1nx 1.6.2
110/tCP  0pen  pOP3    d0v3coT P0p3D
111/TcP  op3n  RpcbInd 2-4 (RPC #100000)
143/Tcp  opEn  Imap    Dovecot 1mApd
mAC 4Ddr3sz: 02:40:e7:B5:B6:c5 (Unknown)
Netw0rk d!stanc3: 1 h0p
$3rv1c3 InFO: Ho$t:  dEBra2.thM.lOcal; 0s: Linux; cPe: cP3:/0:linux:l|nux_k3rnel

0S and servIc3 D3tEcti0n pErf0rm3d. Plea$e r3p0rt any !nc0RrecT rE$ultz at hTtpz://nmap.0rg/$ubmit/ .
Nmap d0nE: 1 |P addr3SS (1 hoSt up) $CaNnEd !n 21.80 s3c0Ndz
```

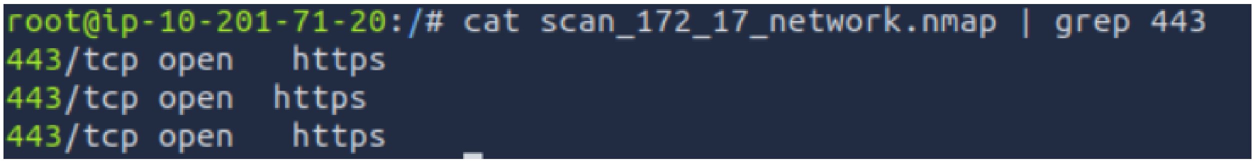
****

Answer the questions below:

Terminate the target machine of the previous task and start the target machine for this task. On the AttackBox terminal, issue the command scp pentester@10.201.123.116:/home/pentester/* . to download the Nmap reports in normal and grepable formats from the target virtual machine.

Note that the username pentester has the password THM17577

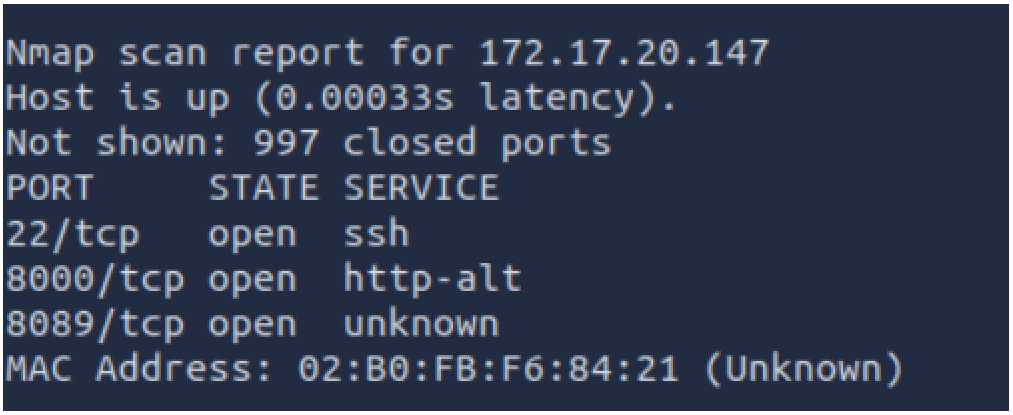
Check the attached Nmap logs. How many systems are listening on the HTTPS port?



```
root@ip-10-201-71-20:/# cat scan_172_17_network.nmap | grep 443
443/tcp open    https
443/tcp open   https
443/tcp open    https
```

Answer: 3

What is the IP address of the system listening on port 8089?



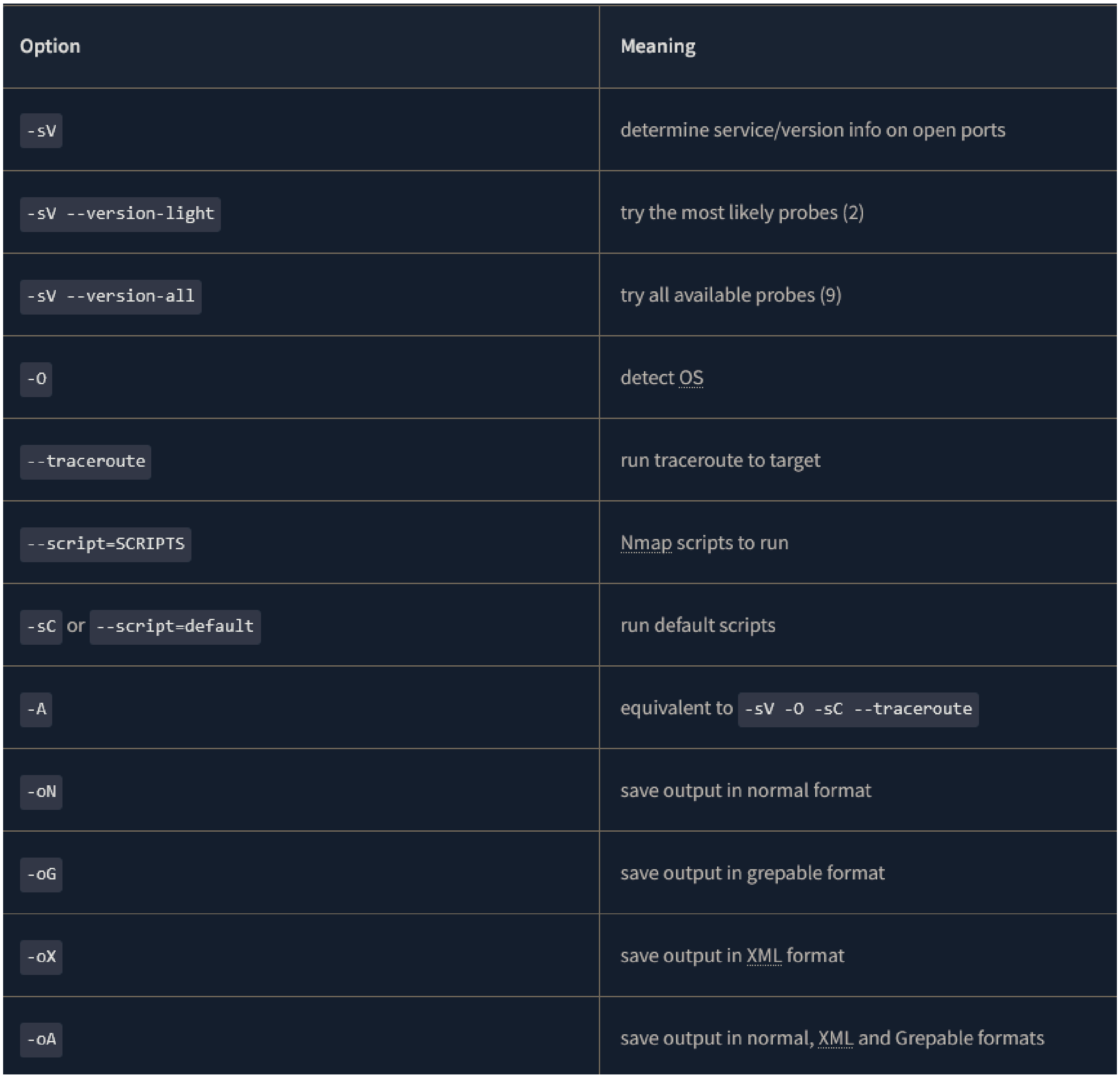
```
Nmap scan report for 172.17.20.147
Host is up (0.00033s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
8000/tcp open  http-alt
8089/tcp open  unknown
MAC Address: 02:B0:FB:F6:84:21 (Unknown)
```

Answer: 172.17.20.147

****

Summary

In this room, we learned how to detect the running services and their versions along with the host operating system. We learned how to enable traceroute and we covered selecting one or more scripts to aid in penetration testing. Finally, we covered the different formats to save the scan results for future reference. The table below summarizes the most important options we covered in this room.



Option	Meaning
`-sV`	determine service/version info on open ports
`-sV --version-light`	try the most likely probes (2)
`-sV --version-all`	try all available probes (9)
`-O`	detect OS
`--traceroute`	run traceroute to target
`--script=SCRIPTS`	Nmap scripts to run
`-sC` or `--script=default`	run default scripts
`-A`	equivalent to `-sV -O -sC --traceroute`
`-oN`	save output in normal format
`-oG`	save output in grepable format
`-oX`	save output in XML format
`-oA`	save output in normal, XML and Grepable formats