THM-Nmap Post Port Scans(Nmap端口扫描后期工作)-学习

本文相关的TryHackMe实验房间链接： https://tryhackme.com/room/nmap04

通过学习相关知识点：了解如何利用 Nmap 进行服务和操作系统检测，使用 Nmap 脚本引擎(NSE)并保存扫描结果。

H2 介绍

在本文中，我们将专注于Nmap端口扫描之后的步骤：主要是目标服务探测、目标操作系统探测、Nmap脚本引擎的使用以及如何保存扫描结果。

具体涉及以下几个知识点：

- 探测正在运行的服务的版本（在所有打开的端口上）
- 根据目标显示的任何迹象探测操作系统信息
- 运行Nmap的 traceroute 选项
- 选择并运行Nmap脚本
- 以各种格式保存Nmap的扫描结果

H2 服务探测

一旦 Nmap 发现了开放的目标端口，你就可以探测开放端口以检测正在运行的服务；对开放端口的进一步调查是一项重要步骤，因为渗透测试者可以通过它来了解目标机器上的服务是否存在任何已知漏洞。

在 Nmap 命令中添加 `-sV` 将会收集并确定开放的目标端口上的服务和相关版本信息。你可以使用 `--version-intensity LEVEL` 控制检测强度，其级别介于 0（最轻）和 9（最完整）之间，`-sV --version-light` 的强度为 2，而 `-sV --version-all` 的强度为 9。

需要注意的是，使用 `-sV` 选项将强制 Nmap 继续进行 TCP 3 次握手并建立TCP连接；此处建立连接是必要的，因为 Nmap 在没有完全建立连接并与所侦听的服务通信的情况下无法发现服务的版本信息；所以当选择Nmap的 `-sV` 选项时，并不能进行完全的隐蔽性SYN 扫描（`-sS`）。

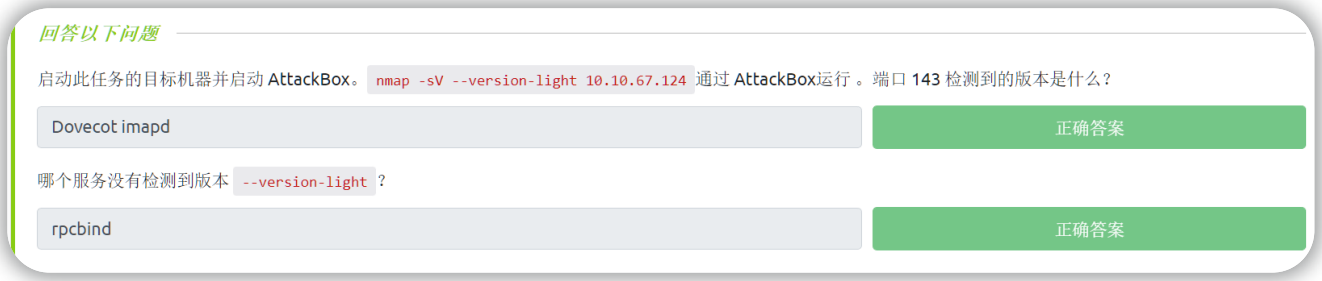
下面的控制台输出显示了使用 `-sV` 选项进行的简单Nmap扫描结果，添加 -sV 选项会导致输出结果能显示每个检测到的端口上的服务的版本。例如，在 TCP 端口 22 开放的情况下，我们能得到 `22/tcp open ssh OpenSSH 6.7p1 Debian 5+deb8u8 (protocol 2.0)`，而不是简单的 `22/tcp open ssh`。

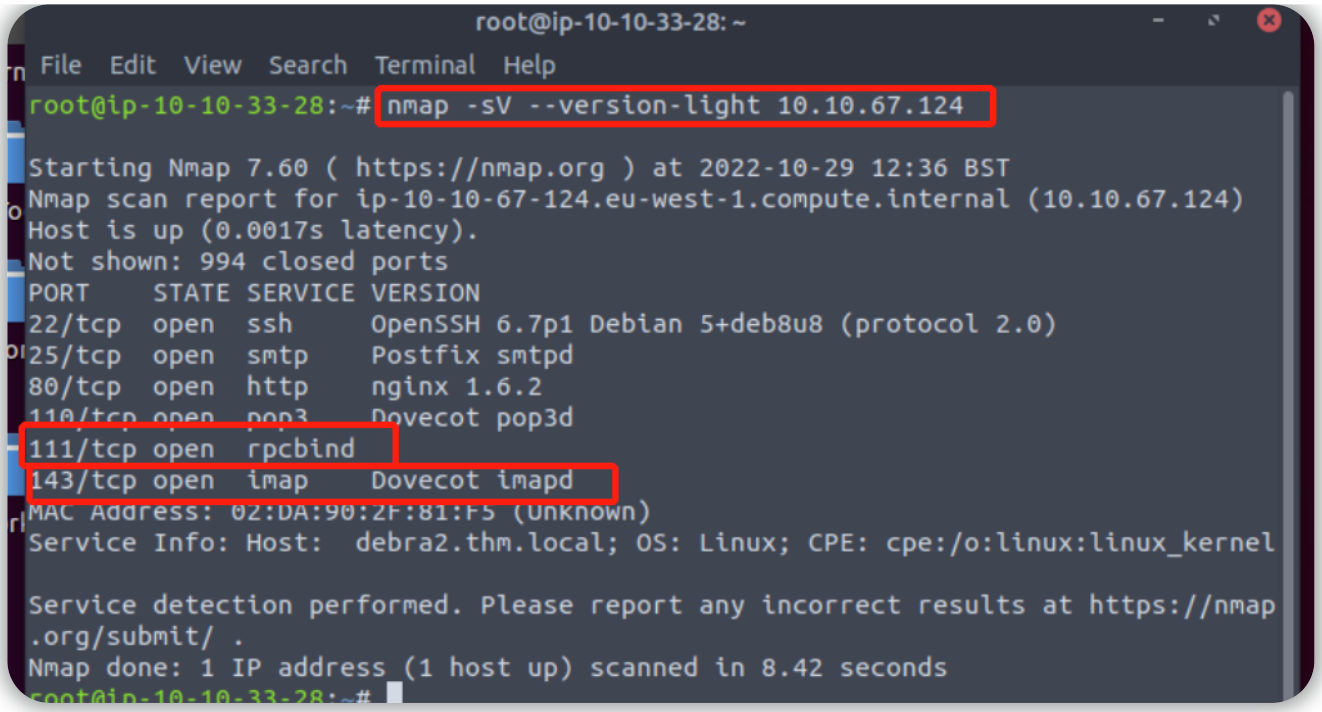
注意：SSH协议之所以被猜测为运行在端口上的服务，是因为目标的TCP端口22是开放的（22端口默认服务是SSH），Nmap 并不需要连接到 22 端口来确认该服务是否真实运行；但是使用-sV时，需要连接到这个开放端口来获取服务的banner(横幅) 以及获取任何关于服务的版本信息，例如 `nginx 1.6.2`；因此，与服务信息不同，版本信息并不是由简单的猜测得出。

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

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-10 05:03 BST
Nmap scan report for MACHINE_IP
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_kernel

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

请注意，许多Nmap选项需要root 权限，除非你以root身份运行Nmap，否则你需要像上面的示例一样使用sudo。

答题





H2 操作系统探测和跟踪路由

操作系统检测

Nmap 可以根据目标机器的行为及其响应中的任何迹象来检测操作系统 (OS)；可以使用 `-O` 启用操作系统检测，这是 OS 中的大写 O。

在下面的例子中，我们在 AttackBox 上运行了 `nmap -sS -O 10.10.67.124`，Nmap 检测到操作系统是 Linux 3.X，然后进一步猜测它运行的是 3.13 内核。

```
pentester@TryHackMe$ sudo nmap -sS -O 10.10.67.124

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-10 05:04 BST
Nmap scan report for 10.10.67.124
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
```

使用Nmap进行操作系统检测非常方便，但有许多因素可能会影响其检测结果的准确性。首先，Nmap 需要在目标机器上找到至少一个开放和一个封闭的端口才能做出比较可靠的猜测；此外，由于越来越多地使用虚拟化和类似技术，来宾操作系统的指纹可能会失真；因此，应该始终对 由Nmap检测得出的操作系统版本持保留态度。

Traceroute

如果你想让 Nmap 找到你和目标之间的路由器，只需添加--traceroute选项即可。

在以下示例中，Nmap 将跟踪路由附加到其扫描结果。请注意，Nmap 的 traceroute 与 Linux 和 macOS 上的 traceroute 命令或 MS Windows 上的 tracert 略有不同：标准的跟踪路由命令从一个低 TTL（生存时间）的数据包开始，并不断增加TTL，直到到达目标；Nmap 的 traceroute 从一个高 TTL（生存时间）的数据包开始，并不断减少TTL。

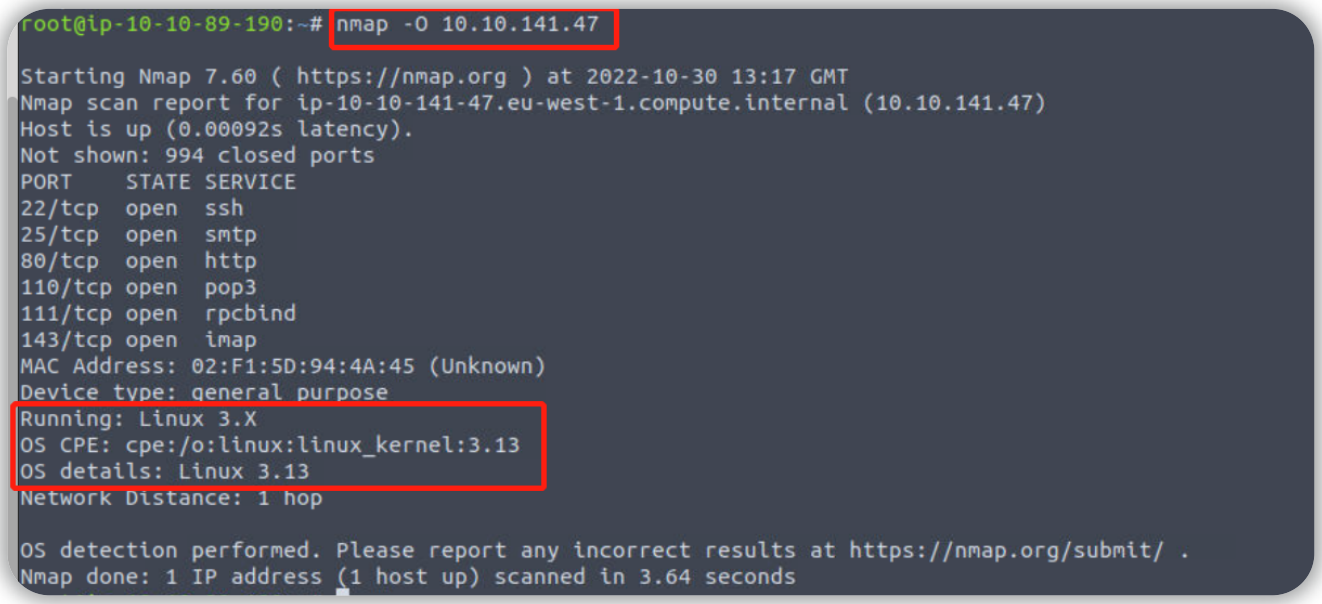
在以下示例中，我们在 AttackBox 上执行了 `nmap -sS --traceroute MACHINE_IP` 命令；我们可以看到两者之间（攻击机和目标机）没有路由器/跃点，因为它们是直接连接的。

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

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-10 05:05 BST
Nmap scan report for MACHINE_IP
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 MACHINE_IP
```

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

值得一提的是，许多路由器被配置为不发送 ICMP Time-to-Live exceeded，这将阻止我们发现它们的 IP 地址；欲了解更多信息，请查看"主动侦察"知识点博客。

答题



H2 Nmap脚本引擎（NSE）

脚本是一段不需要编译的代码，换句话说，脚本能够保持其原始的人类可读形式，并不需要转换为机器语言。许多程序通过脚本提供附加功能；此外，脚本也可以添加自定义功能（使用内置命令无法实现的功能）。

同样，在Nmap中也支持使用Lua语言编写的脚本。作为Nmap 的一部分，Nmap 脚本引擎 (NSE-Nmap Scripting Engine) 是一个Lua解释器，它能够允许Nmap 执行用Lua语言编写的Nmap脚本（我们不需要专门学习Lua语言来使用 Nmap 脚本）。

默认安装的Nmap可以轻松包含近 600 个脚本，你可以查看Nmap 安装文件夹。在 AttackBox 上，如果检查 /usr/share/nmap/scripts 中的文件，你会注意到有数百个以它们所针对的协议开头的方便命名的脚本。

我们在下面的控制台输出中列出了所有在 AttackBox 上以 HTTP 开头的脚本； 我们发现了大约 130 个以 http 开头的脚本。随着未来的更新，已安装脚本的数量还会逐渐增加。

```
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
http-method-tamper.nse                  http-webdav-scan.nse
http-methods.nse                        http-wordpress-brute.nse
http-mobileversion-checker.nse          http-wordpress-enum.nse
http-ntlm-info.nse                      http-wordpress-users.nse
http-open-proxy.nse                     http-xssed.nse
http-open-redirect.nse
```

你可以指定使用任何或一组这些已安装的脚本； 此外，你还可以安装其他用户的脚本并将它们用于你的扫描。

让我们从默认脚本开始，你可以使用 `--script=default` 或简单地添加 `-sC` 选项来选择运行默认类别的脚本。

除了默认脚本之外，其他脚本类别还包括 auth、broadcast、brute、default、discovery、dos、exploit、external、fuzzer、intrusive、malware、safe、version 和 vuln等，脚本类别的简要说明如下表所示。

脚本类别	描述
auth	身份验证相关脚本
broadcast	通过发送广播消息发现主机
brute	针对登录名执行暴力密码审计
default	默认脚本，-sC
discovery	检索可访问的信息，例如数据库表和DNS名称
dos	检测易受拒绝服务（DoS）攻击的服务器
exploit	尝试利用各种易受攻击的服务
external	使用第三方服务进行检查，例如 Geoplugin 和 Virustotal
fuzzer	发起模糊攻击
intrusive	侵入性脚本，例如暴力攻击和利用
malware	扫描后门
safe	不会使目标崩溃的安全脚本
version	检索服务版本
vuln	检查漏洞或利用易受攻击的服务

一些脚本可能会属于多个类别，此外一些脚本会对服务发起暴力（brute-force）攻击，而另一些脚本则会发起 DoS 攻击并进行系统漏洞利用。因此，如果你不想让目标服务崩溃或对目标进行漏洞利用，那么在选择要运行的Nmap脚本时要格外小心谨慎。

我们使用 Nmap 对 MACHINE_IP 运行 SYN 扫描，并在如下所示的控制台中执行默认脚本；使用的命令是 `sudo nmap -sS -sC MACHINE_IP`，其中 -sC 将确保 Nmap 在 SYN 扫描之后执行默认脚本。

下面显示了扫描结果的详细信息。看一下目标端口22的SSH服务，Nmap 恢复了与正在运行的服务器相关的四个公钥；再看一下目标端口80的 HTTP 服务，Nmap 检索到一个默认页面的标题，我们可以看到该页面已被保留为默认值。

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

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-REFERRALS IDLE
MAC Address: 02:A0:E7:B5:B6:C5 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 2.21 seconds
```

你还可以使用 `--script "SCRIPT-NAME"` 或诸如 `--script "ftp*"` （包括 ftp-brute）之类的模式按名称指定并运行脚本。

如果你不确定脚本的作用，可以使用文本阅读器（例如 less）或文本编辑器打开脚本文件进行内容查看。在查看ftp-brute脚本内容时，可以看到它的声明：“对 FTP 服务器执行暴力密码审计。”你必须尽量小心，因为某些脚本非常具有侵入性；此外，某些脚本可能只适用于特定服务器，如果随机选择脚本可能只会浪费你的时间。像往常一样，你首先需要确保你已经得到授权，然后才能在目标服务器上启动此类测试。

让我们考虑一个良性脚本 http-date，我们猜它会检索 http 服务器的日期和时间，这确实在其内容描述中得到证实：“从类似 HTTP 的服务中获取日期，此外，它会打印该日期与当地时间的差异.....”

现在我们使用AttackBox，执行命令 `sudo nmap -sS -n --script "http-date" MACHINE_IP`，输出结果如下面的控制台所示。

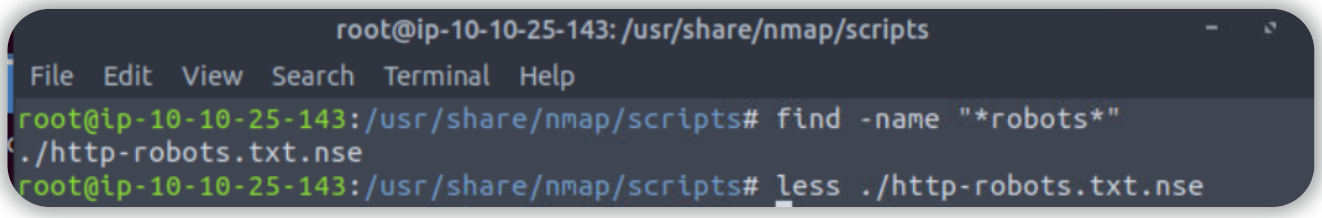
```
pentester@TryHackMe$ sudo nmap -sS -n --script "http-date" MACHINE_IP

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-10 08:04 BST
```

```
Nmap scan report for MACHINE_IP
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
```

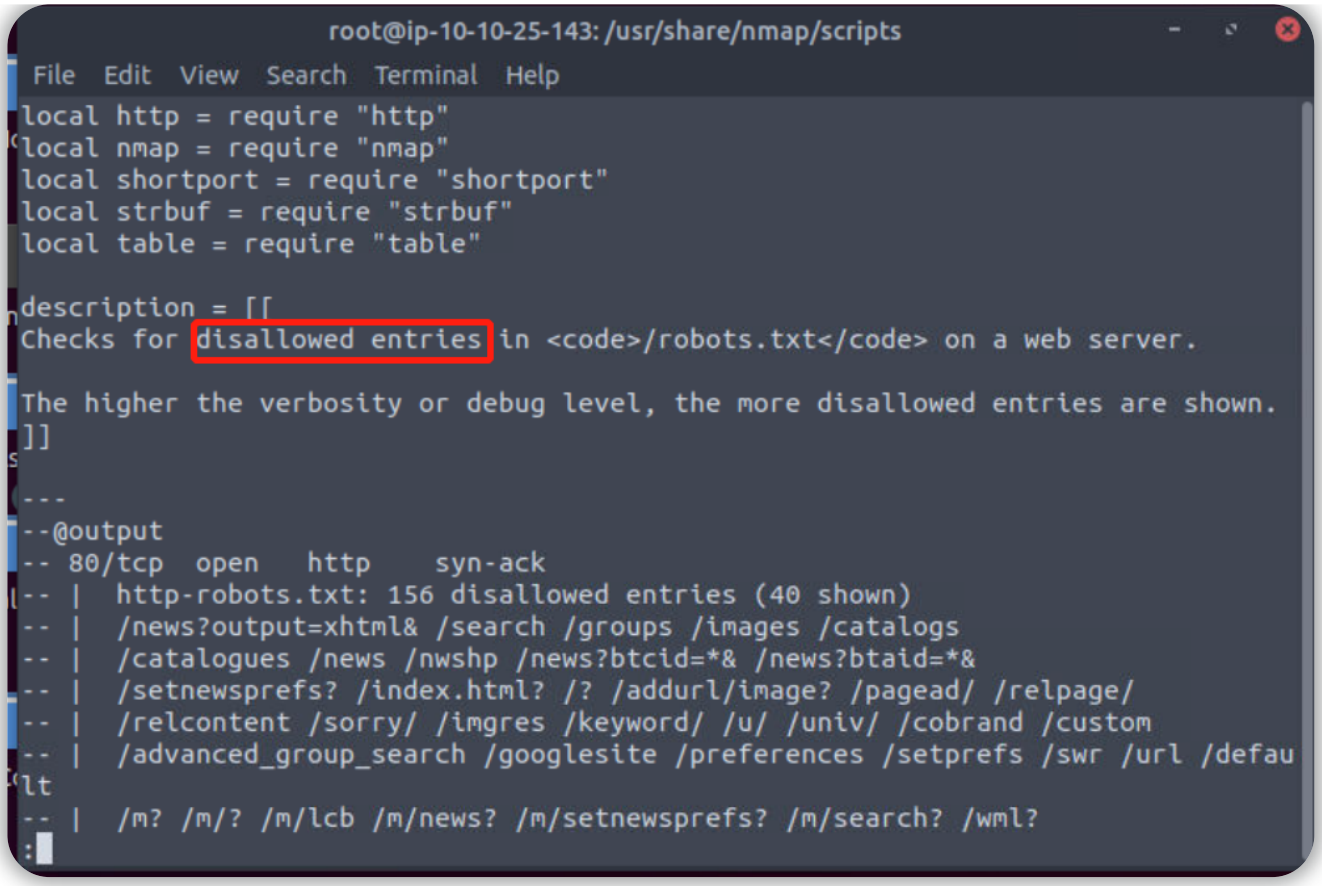
最后，你还可以将 Nmap 的脚本功能扩展到官方提供的Nmap脚本之外；你可以编写Nmap脚本或从 Internet 下载 Nmap 脚本进行使用；从 Internet 下载和使用 Nmap 脚本存在一定程度的风险，因此，最好不要运行你不信任的作者的Nmap脚本。

答题

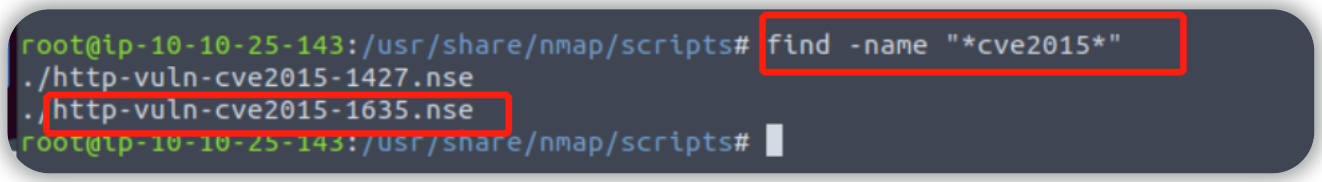
查找目标脚本：



使用less命令查看相关脚本的界面：



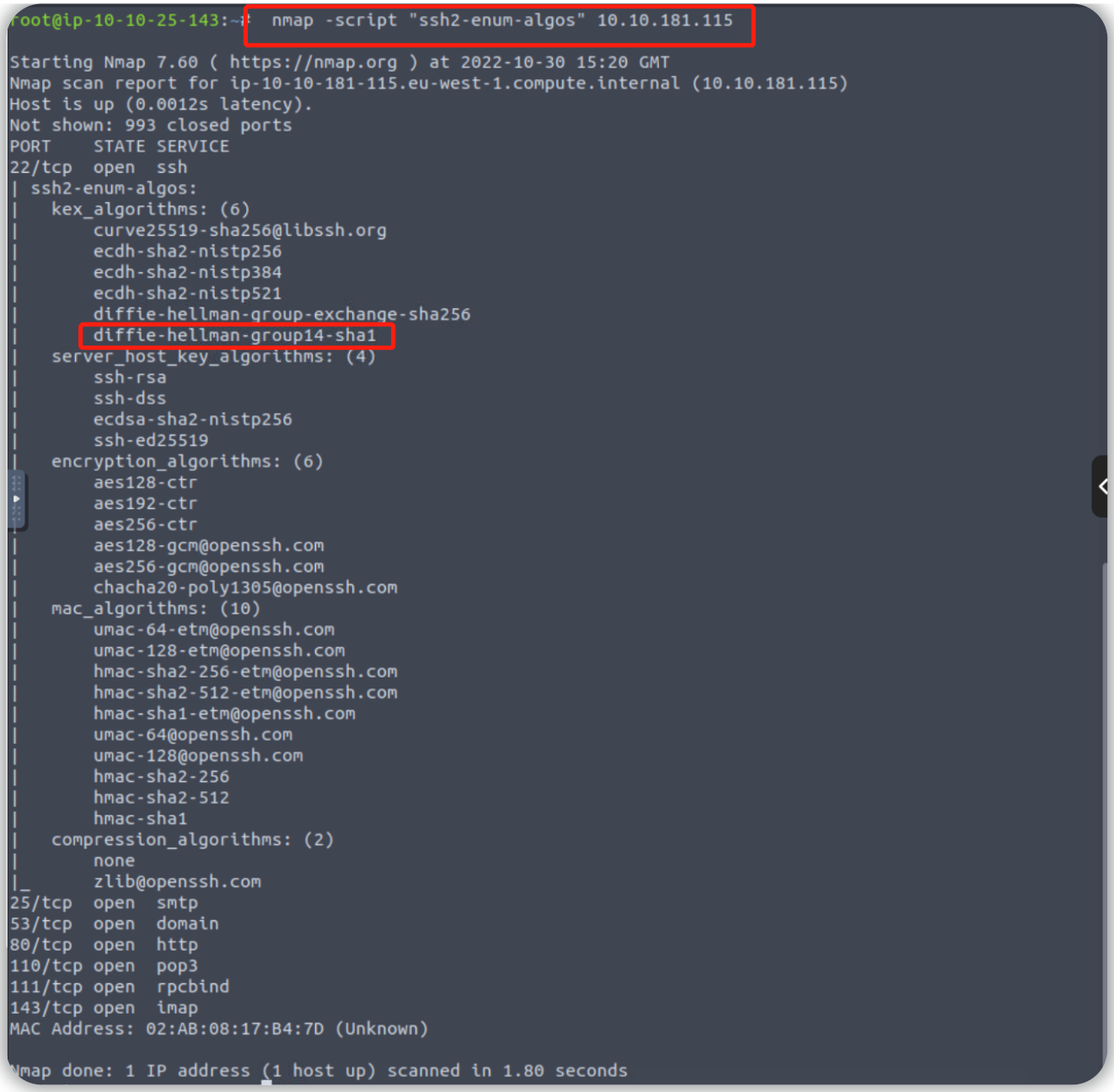
查找目标脚本：



使用默认脚本扫描目标：

```
root@ip-10-10-25-143:~# nmap -sC 10.10.181.115

Starting Nmap 7.60 ( https://nmap.org ) at 2022-10-30 15:12 GMT
Nmap scan report for ip-10-10-181-115.eu-west-1.compute.internal (10.10.181.115)
Host is up (0.0013s latency).
Not shown: 993 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
| ssh-hostkey:
|   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
53/tcp  open  domain
| dns-nsid:
|_  bind.version: 9.9.5-9+deb8u19-Debian
80/tcp  open  http
|_http-title: Welcome to nginx on Debian!
110/tcp open  pop3
|_pop3-capabilities: PIPELINING CAPA RESP-CODES AUTH-RESP-CODE SASL TOP UIDL
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          33671/udp  status
|_  100024  1          41751/tcp  status
143/tcp open  imap
|_imap-capabilities: IMAP4rev1 LOGIN-REFERRALS ID SASL-IR have ENABLE LITERAL+ more Pre-login OK post-l
ogin listed IDLE LOGINDISABLEDA0001 capabilities
MAC Address: 02:AB:08:17:B4:7D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 10.73 seconds
```

运行脚本查看结果：



H2 保存Nmap的输出结果

每当你运行 Nmap 扫描时，将扫描结果保存在文件中才是合理的，为文件名选择和采用良好的命名约定也很重要，否则随着文件数量的增加，会妨碍你查找以前的扫描结果。

保存Nmap输出结果的三种主要格式是：

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

Normal

顾名思义，正常保存格式类似于Nmap扫描目标时在屏幕上得到的输出，你可以使用 `-oN FILENAME` 以正常格式保存扫描结果，其中的N代表正常。

示例如下：

```
pentester@TryHackMe$ cat 10.10.181.115_scan.nmap
# Nmap 7.60 scan initiated Fri Sep 10 05:14:19 2021 as: nmap -sS -sV -O -oN
10.10.181.115_scan MACHINE_IP
Nmap scan report for 10.10.181.115
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

grepable 格式的名称来自命令 `grep`，而 `grep` 代表全局正则表达式打印（Global Regular Expression Printer）。简而言之，它可以得到 有效地过滤特定关键字或术语的扫描输出。

你可以使用 `-oG FILENAME` 以 grepable 格式保存扫描结果。上面以正常格式显示的扫描输出也可以使用grepable格式显示在下面的控制台中，上面正常格式的输出结果有21行关键信息，然而，grepable格式的输出结果只有 4 行关键信息。

主要原因是 Nmap在应用 grep 时想让每一行都有完整的意义，所以grepable格式的输出与正常格式的输出相比，每行都很长并且也不方便阅读。

```
pentester@TryHackMe$ cat 10.10.181.115_scan.gnmap
# Nmap 7.60 scan initiated Fri Sep 10 05:14:19 2021 as: nmap -sS -sV -O -oG
10.10.181.115_scan MACHINE_IP
Host: 10.10.181.115 Status: Up
Host: MACHINE_IP    Ports: 22/open/tcp//ssh//OpenSSH 6.7p1 Debian 5+deb8u8 (protocol
2.0)/, 25/open/tcp//smtp//Postfix smtpd/, 80/open/tcp//http//nginx 1.6.2/,
110/open/tcp//pop3//Dovecot pop3d/, 111/open/tcp//rpcbind//2-4 (RPC #100000)/,
143/open/tcp//imap//Dovecot imapd/   Ignored State: closed (994) OS: Linux 3.13  Seq
Index: 257  IP ID Seq: All zeros
# Nmap done at Fri Sep 10 05:14:28 2021 -- 1 IP address (1 host up) scanned in 9.99
seconds
```

`grep` 的一个示例用法是 `grep KEYWORD TEXT_FILE`，此命令将显示包含所提供的关键字的所有行。

让我们比较在正常格式输出上和在grepable格式输出上分别使用 `grep` 命令的结果。你会注意到前者并没有提供主机的 IP 地址，相反，它返回的是 `80/tcp open http nginx 1.6.2`，如果你正在筛选多个系统的扫描结果，这将非常不方便；而后者则在每一行中都提供了足够的信息，例如主机的 IP 地址，以使其信息较为完整。

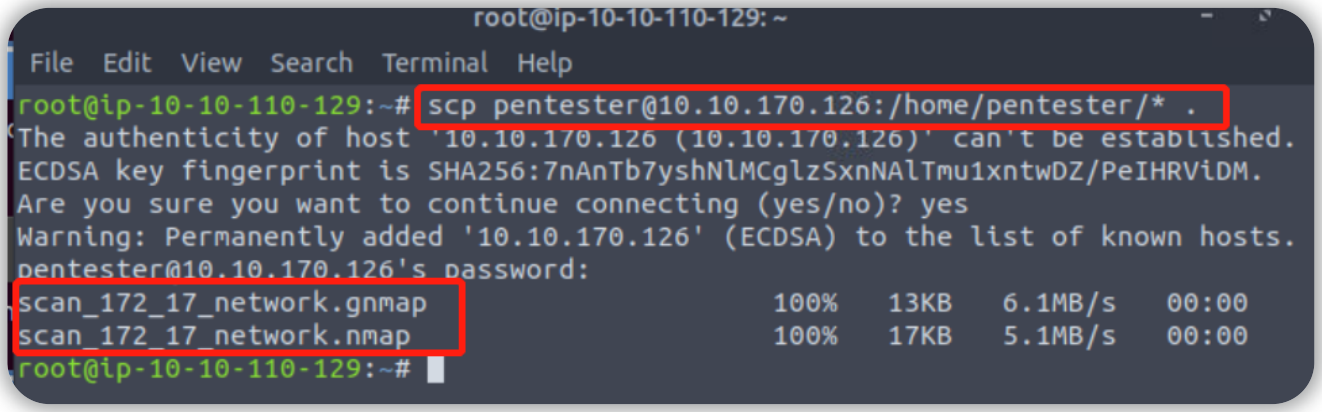
```
pentester@TryHackMe$ grep http 10.10.181.115_scan.nmap
80/tcp  open  http    nginx 1.6.2
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
```

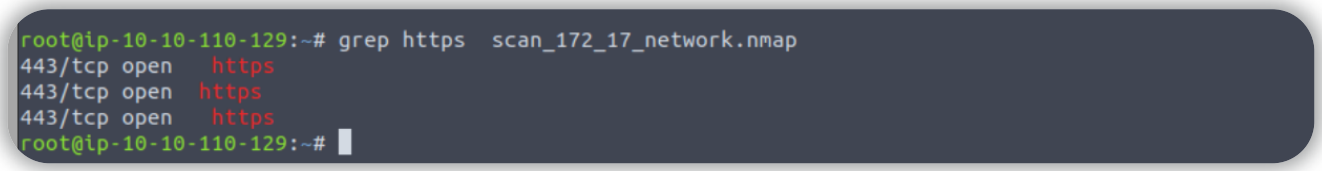
```
pentester@TryHackMe$ grep http 10.10.181.115_scan.gnmap
Host: 10.10.181.115 Ports: 22/open/tcp//ssh//OpenSSH 6.7p1 Debian 5+deb8u8 (protocol
2.0)/, 25/open/tcp//smtp//Postfix smtpd/, 80/open/tcp//http//nginx 1.6.2/,
110/open/tcp//pop3//Dovecot pop3d/, 111/open/tcp//rpcbind//2-4 (RPC #100000)/,
143/open/tcp//imap//Dovecot imapd/   Ignored State: closed (994) OS: Linux 3.13  Seq
Index: 257  IP ID Seq: All zeros
```

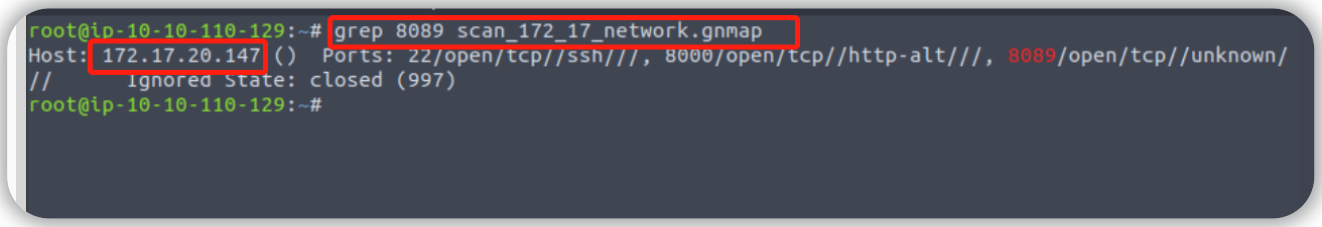
XML

第三种格式是 XML，你可以在Nmap中使用 `-oX FILENAME` 以 XML 格式保存扫描结果。 XML 格式最方便在其他程序中处理输出结果，你也可以使用 `-oA FILENAME` 将 -oN、-oG 和 -oX 组合，以Normal（正常）、grepable和 XML三种格式来保存Nmap扫描的输出结果。

答题







H2 小结

在本文中，我们学习了如何检测目标主机上正在运行的服务及其版本以及操作系统信息；我们学习了Nmap中的traceroute命令，并介绍了如何选择一个或多个Nmap脚本来帮助进行渗透测试；最后，我们介绍了使用不同的格式保存Nmap的扫描结果 以供将来参考。

下表总结了我们在本文中讨论的最重要的选项。

选项	意义
`-sV`	确定开放端口上的服务/版本信息
`-sV --version-light`	尝试最有可能的探针 (2)
`-sV --version-all`	尝试所有可用的探头 (9)
`-O`	检测操作系统
`--traceroute`	运行跟踪路由到目标
`--script=SCRIPTS`	要运行的 Nmap 脚本
`-sC` 或者 `--script=default`	运行默认脚本
`-A`	相当于 `-sV -O -sC --traceroute`
`-oN`	以正常格式保存输出
`-oG`	以 grepable 格式保存输出
`-oX`	以XML格式保存输出
`-oA`	以普通、XML和 Grepable 格式保存输出