Phòng thí nghiệm An toàn Thông tin InsecLab
Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO THỰC HÀNH

Môn học: An Toàn Mạng

Tên chủ đề: Vulnerability Scanning

GVHD: Tô Trọng Nghĩa

Nhóm: 3

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT140.P11.ANTT.1

STT	Họ và tên	MSSV	Email
1	Hà Minh Quân	22521177	22521177@gm.uit.edu.vn
2	Từ Chí Kiên	22520713	22520713@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Nội dung	Tình trạng	Trang
1	Quét lỗ hổng sử dụng công cụ Nessus	100%	Xem mục lục
2	Bài tập nhóm	100%	Xem mục lục
Điểm tự đánh giá			**10/10**

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

A. TỔNG QUAN

B. THỰC HÀNH

1. Quét lỗ hổng sử dụng công cụ Nessus

a) Cài đặt Nessus

```
┌──(root㉿kali)-[/home/kali/Documents/NT140/Lab03]
└─# apt update && apt upgrade
Get:1 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Ign:2 https://download.docker.com/linux/debian kali-rolling InRelease
Err:4 https://download.docker.com/linux/debian kali-rolling Release
  404  Not Found [IP: 13.224.163.28 443]
Get:3 http://mirror.kku.ac.th/kali kali-rolling InRelease [41.5 kB]
Get:5 http://mirror.kku.ac.th/kali kali-rolling/main amd64 Packages [20.2 MB]
Get:6 http://mirror.kku.ac.th/kali kali-rolling/main amd64 Contents (deb) [47.9 MB]
Get:7 http://mirror.kku.ac.th/kali kali-rolling/contrib amd64 Packages [111 kB]
Get:8 http://mirror.kku.ac.th/kali kali-rolling/contrib amd64 Contents (deb) [270 kB]
Get:9 http://mirror.kku.ac.th/kali kali-rolling/non-free amd64 Packages [197 kB]
Get:10 http://mirror.kku.ac.th/kali kali-rolling/non-free amd64 Contents (deb) [876 kB]
Get:11 http://mirror.kku.ac.th/kali kali-rolling/non-free-firmware amd64 Packages [10.8 kB]
Warning: https://download.docker.com/linux/ubuntu/dists/bionic/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), se
e the DEPRECATION section in apt-key(8) for details.
Error: The repository 'https://download.docker.com/linux/debian kali-rolling Release' does not have a Release file.
Notice: Updating from such a repository can't be done securely, and is therefore disabled by default.
Notice: See apt-secure(8) manpage for repository creation and user configuration details.
```

Kiểm tra tệp tải xuống:

```
┌──(kali㉿kali)-[~/Documents/NT140/Lab03]
└─$ sha256sum Nessus-10.8.3-debian10_amd64.deb
2426a5c11d45fd373c7ca6c7716f9505bfbf0c0a623a82ab709fc7f6af9b5d94  Nessus-10.8.3-debian10_amd64.deb
```

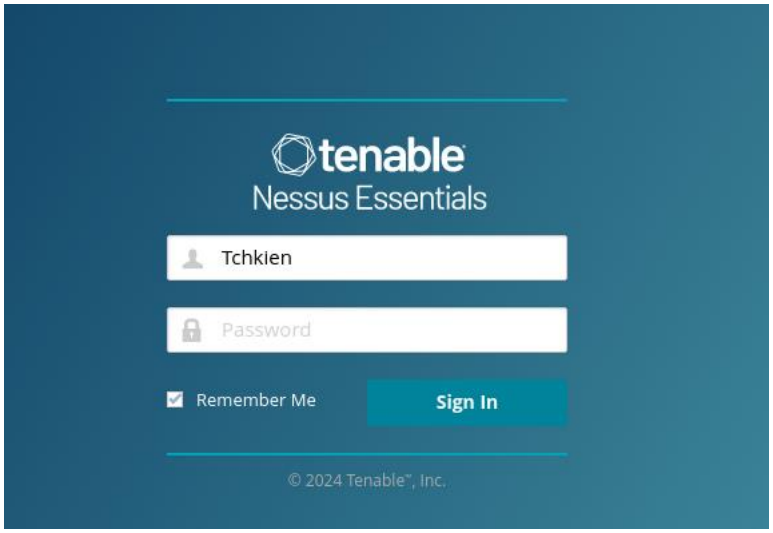
Cài nessus:

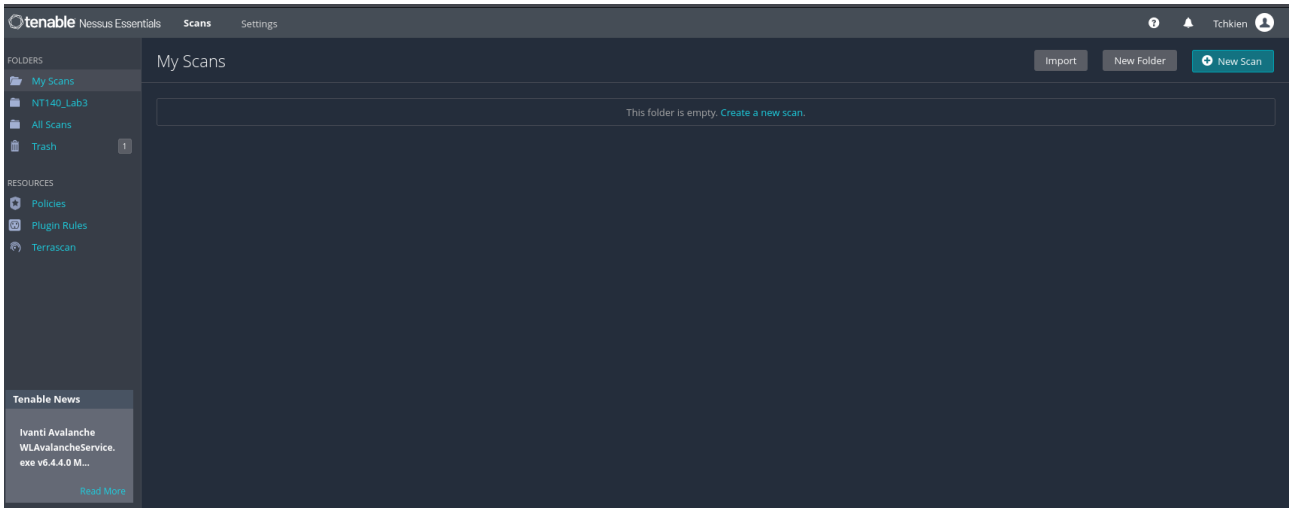
```
┌──(root㉿kali)-[/home/kali/Documents/NT140/Lab03]
└─# apt install ./Nessus-10.8.3-debian10_amd64.deb
Note, selecting 'nessus' instead of './Nessus-10.8.3-debian10_amd64.deb'
The following packages were automatically installed and are no longer required:
  libintl-perl     libmodule-find-perl      libproc-processtable-perl  needrestart     tini
  libintl-xs-perl  libmodule-scandeps-perl  libsort-naturally-perl     openjdk-17-jre
Use 'sudo apt autoremove' to remove them.

Installing:
  nessus

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 995
  Download size: 0 B / 68.8 MB
  Space needed: 0 B / 46.2 GB available

Get:1 /home/kali/Documents/NT140/Lab03/Nessus-10.8.3-debian10_amd64.deb nessus amd64 10.8.3 [68.8 MB]
Selecting previously unselected package nessus.
(Reading database ... 425624 files and directories currently installed.)
Preparing to unpack .../Nessus-10.8.3-debian10_amd64.deb ...
```

Chạy và kiểm tra nessus:

```
┌──(root㉿kali)-[/home/kali/Documents/NT140/Lab03]
└─# /bin/systemctl start nessusd.service

┌──(root㉿kali)-[/home/kali/Documents/NT140/Lab03]
└─# systemctl status nessusd
● nessusd.service - The Nessus Vulnerability Scanner
     Loaded: loaded (/usr/lib/systemd/system/nessusd.service; disabled; preset: disabled)
     Active: active (running) since Fri 2024-10-18 08:06:15 +07; 23s ago
  Invocation: cebe27df1f7d426290642acac50f1ed7
    Main PID: 7180 (nessus-service)
       Tasks: 14 (limit: 2258)
      Memory: 164.5M (peak: 164.8M)
         CPU: 23.621s
      CGroup: /system.slice/nessusd.service
              ├─7180 /opt/nessus/sbin/nessus-service -q
              └─7182 nessusd -q

Oct 18 08:06:15 kali systemd[1]: Started nessusd.service - The Nessus Vulnerability Scanner.
Oct 18 08:06:16 kali nessus-service[7182]: Cached 0 plugin libs in 0msec
Oct 18 08:06:16 kali nessus-service[7182]: Cached 0 plugin libs in 0msec
```

Sau khi đã thực hiện đăng ký, đăng nhập vào tài khoản





Địa chỉ của metasploitable2: 192.168.108.128

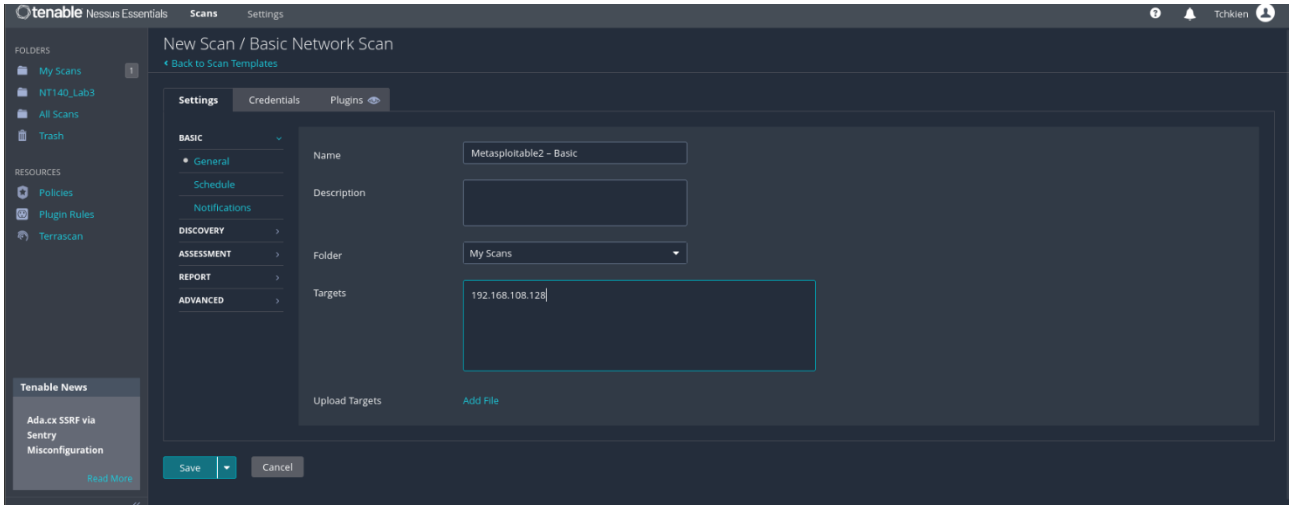
```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:8b:d0:cd
          inet addr:192.168.108.128  Bcast:192.168.108.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe8b:d0cd/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:36 errors:0 dropped:0 overruns:0 frame:0
          TX packets:63 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3805 (3.7 KB)  TX bytes:6722 (6.5 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$ _
```

b) Khai báo đối tượng

® Bài tập về nhà (yêu cầu làm)

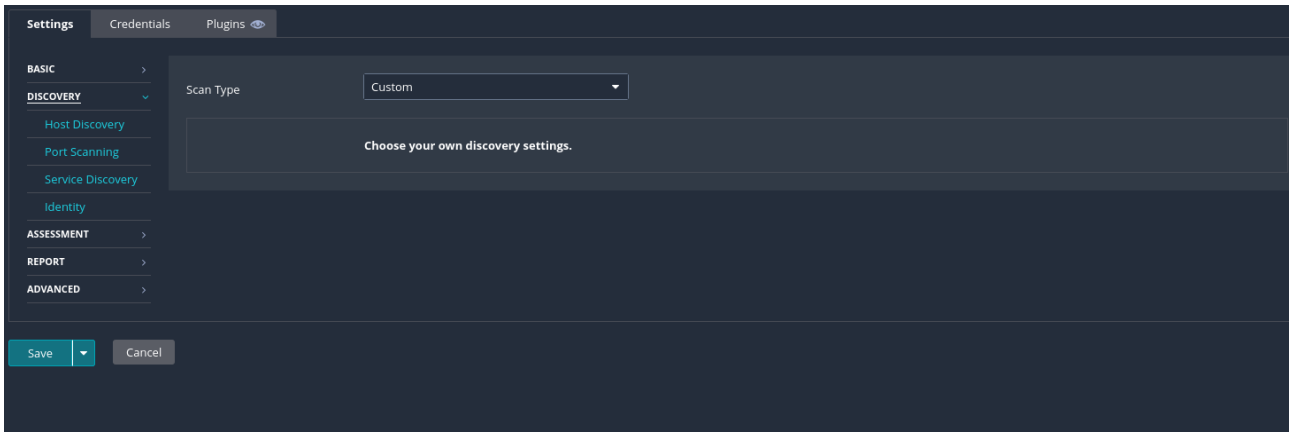
1. *Thực hiện lại các bước trên để quét máy Metasploitable 2 không sử dụng tài khoản chứng thực.*

Tạo một new scan với tên và địa chỉ taget là địa chỉ của máy metasploitable 2

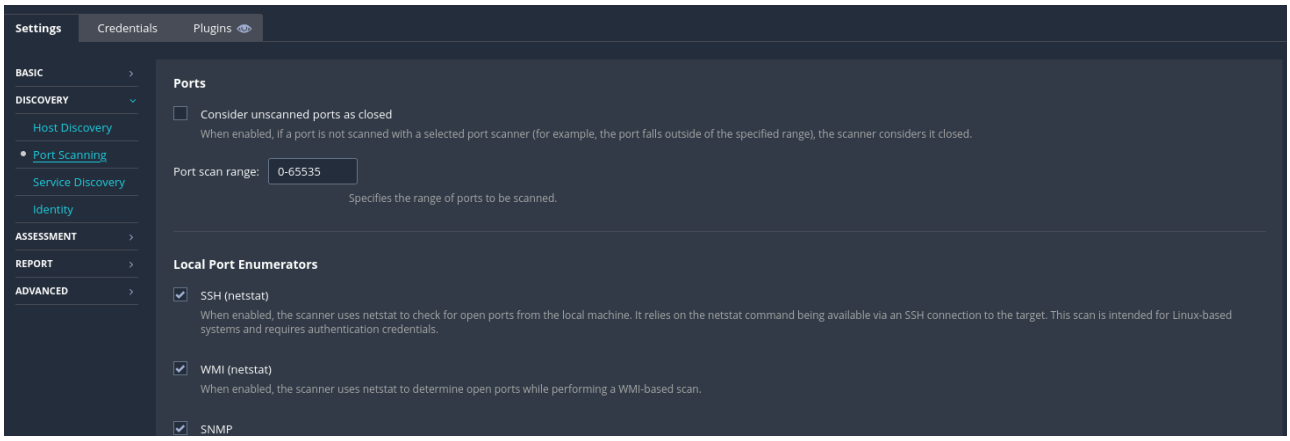


c) Cấu hình các định nghĩa quét (Scan Definitions)

Phần scan type chọn custom

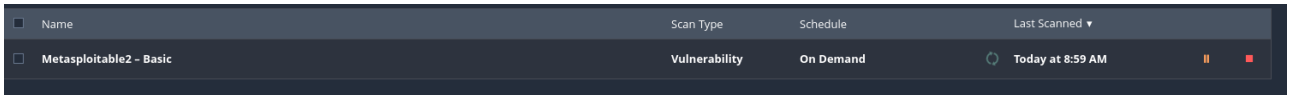


Phần port sẽ cho khoảng từ 0 đến 65535

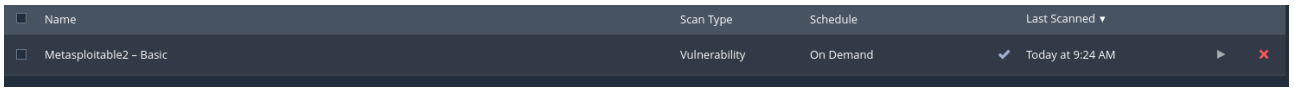


d) Quét lỗ hổng không sử dụng tài khoản chứng thực

Sau khi lưu bắt đầu chạy để quét

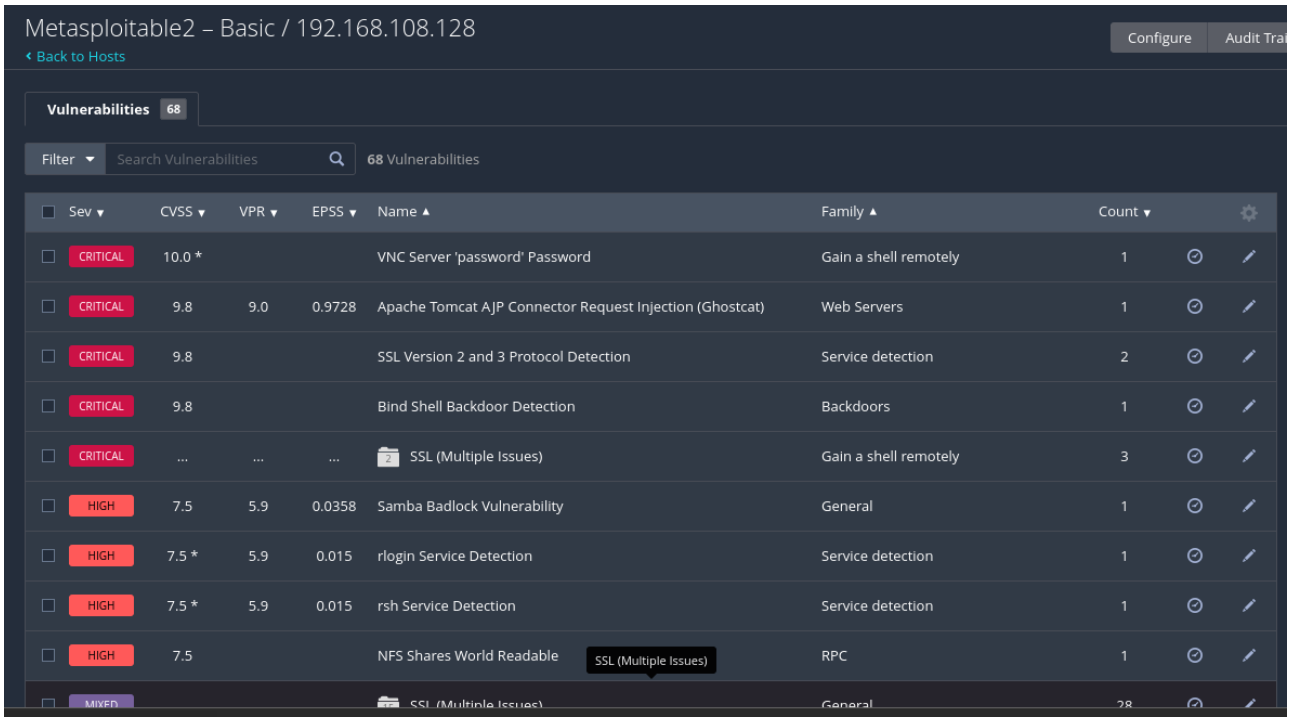


Sau một khoảng thời gian thì quét thành công

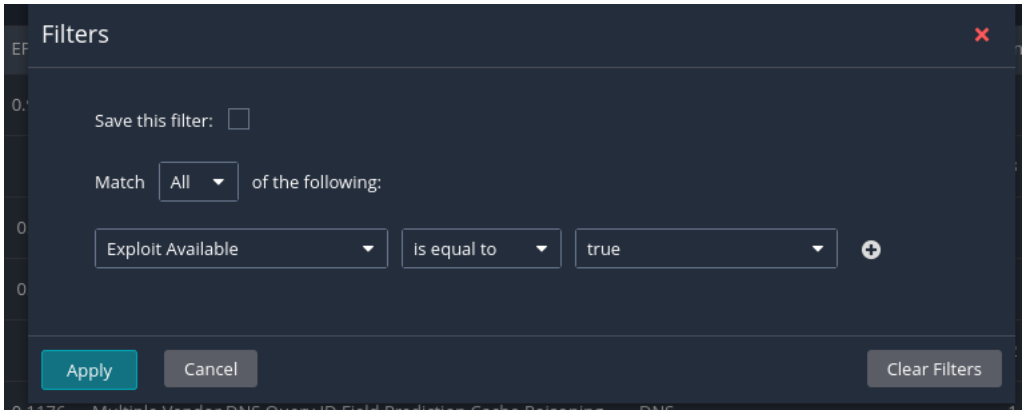


Mở lên để xem kết quả

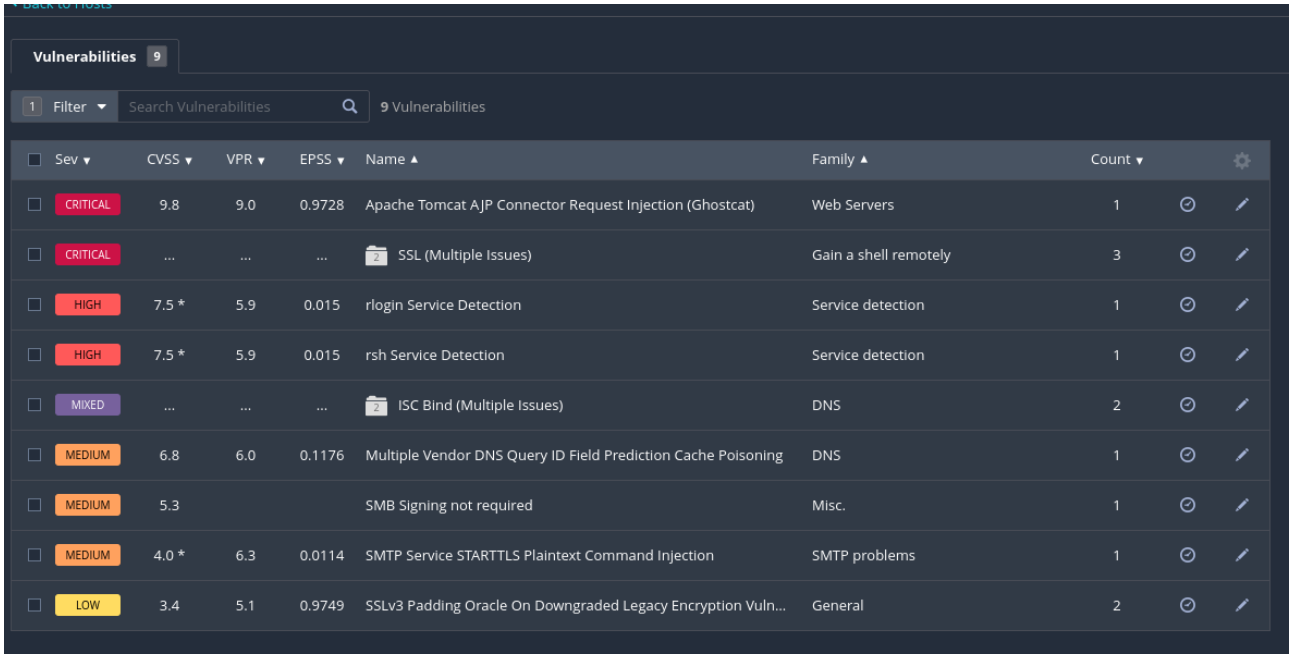
Truy cập vào vulnerabilities để xem chi tiết



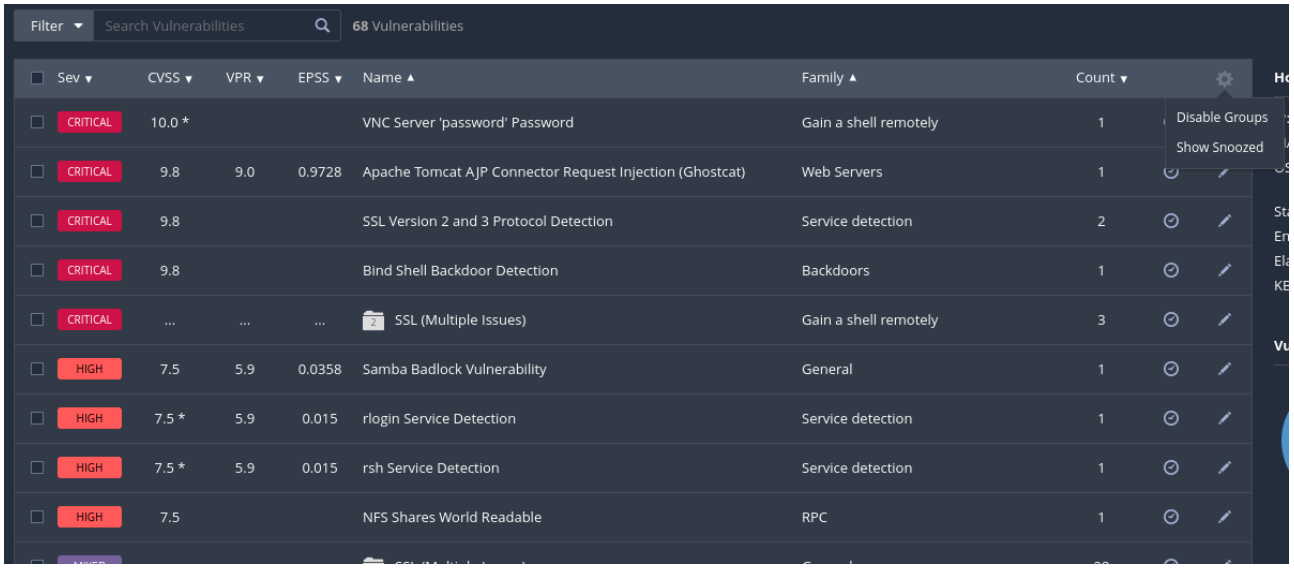
Sử dụng bộ lọc với cấu hình exploit available is equal to true

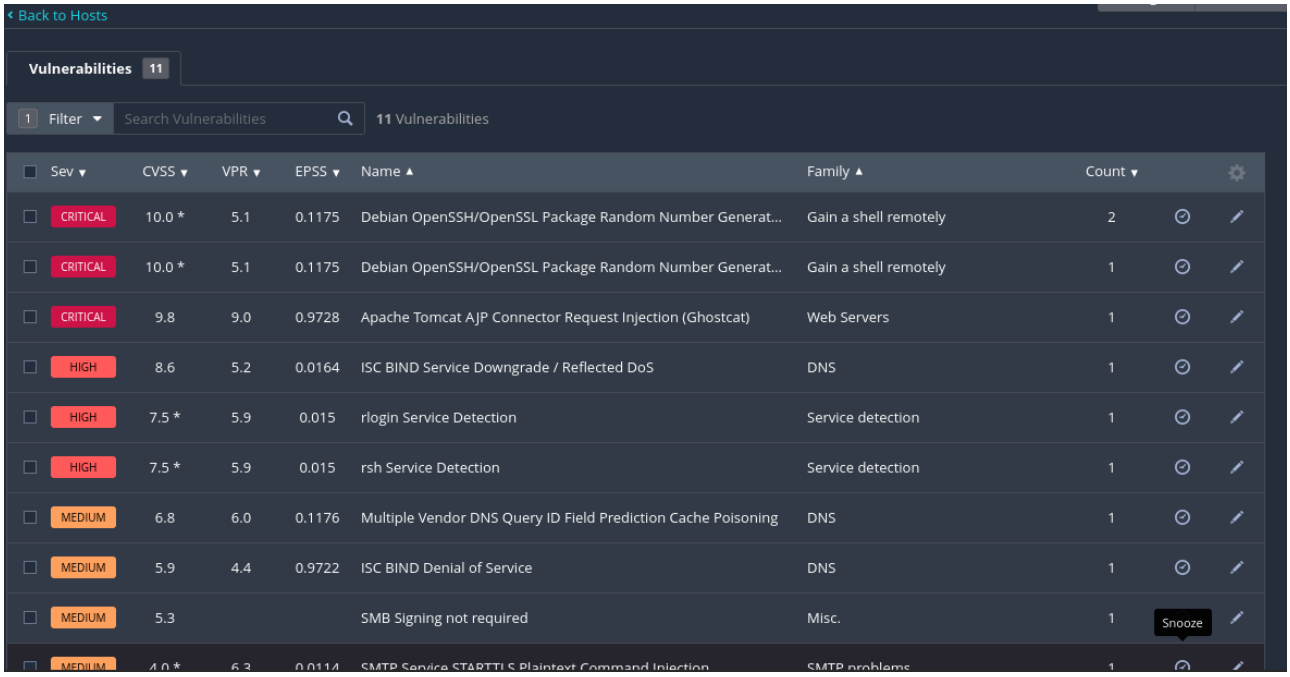


Kết quả sau khi lọc:



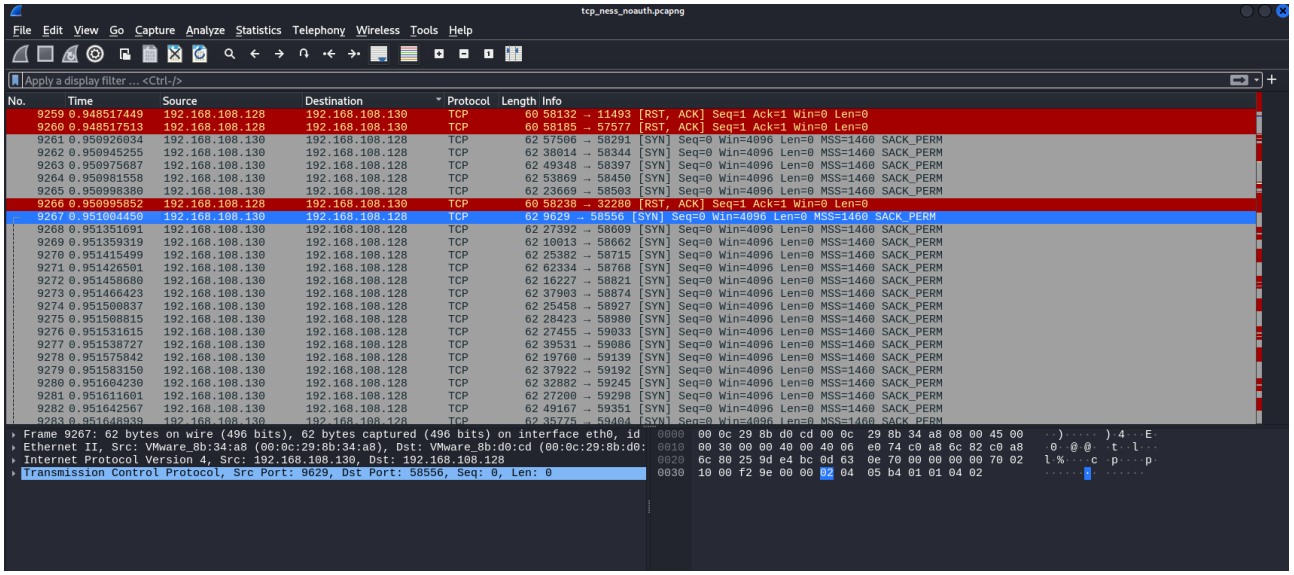
Để bỏ nhóm exploit sử dụng disable group:





2. *Bật Wireshark sau đó tiến hành quét và xác định các bước mà Nessus đã thực hiện để hoàn tất quá trình quét.*

Chạy wireshark và thực hiện scan lại trong nessus



Thì sau khi phân tích thì chỉ có 2 trường hợp:

Ví dụ: gói tin 9267 khi máy khách nessus yêu cầu kết đến cổng 58556 metasploitable 2 qua cổng 9629, sử dụng gói tin SYN

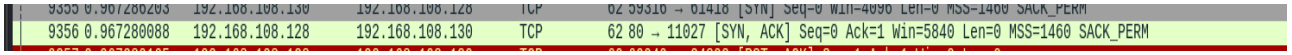
Sau đó máy chủ đáp lại cũng qua 2 cổng nói trên ở gói tin thứ 9297, sử dụng gói tin RST,ACK:

Thì trường hợp này là máy khách gửi SYN để yêu cầu kết nối nhưng máy chủ từ chối kết nối đến cổng đó, RST(không chấp nhận hoặc nhận dữ liệu), ACK(cho biết đã nhận được gói tin SYN của client)

Một trường hợp khác sẽ là:

Ở gói tin 9327 thì máy khách gửi gói tin SYN yêu cầu kết nối đến cổng 80

Sau đó máy chủ gửi gói tin SYN, ACK cho phép máy khách kết nối

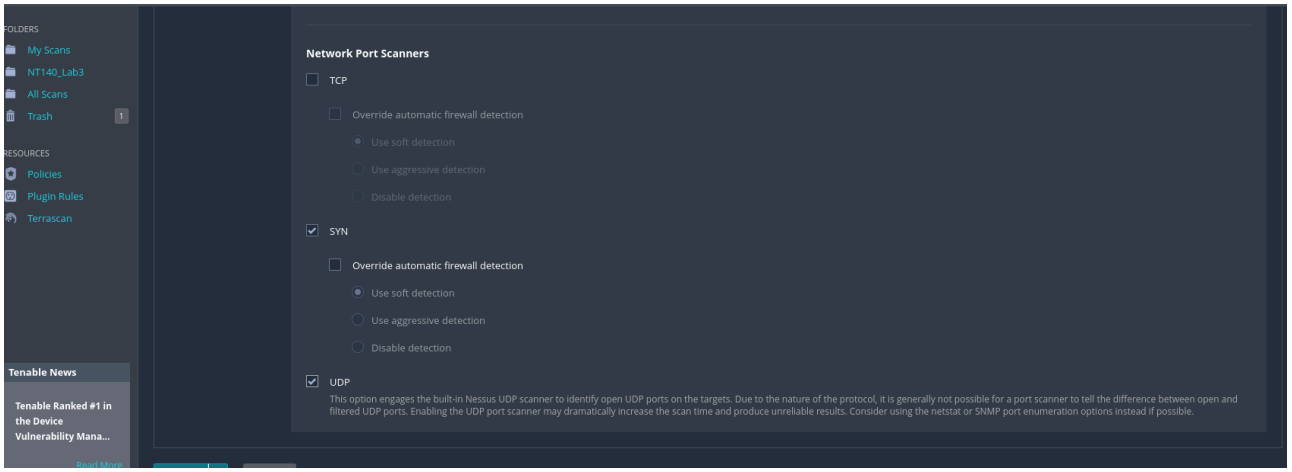


Máy khách gửi gói RST để đóng kết nối

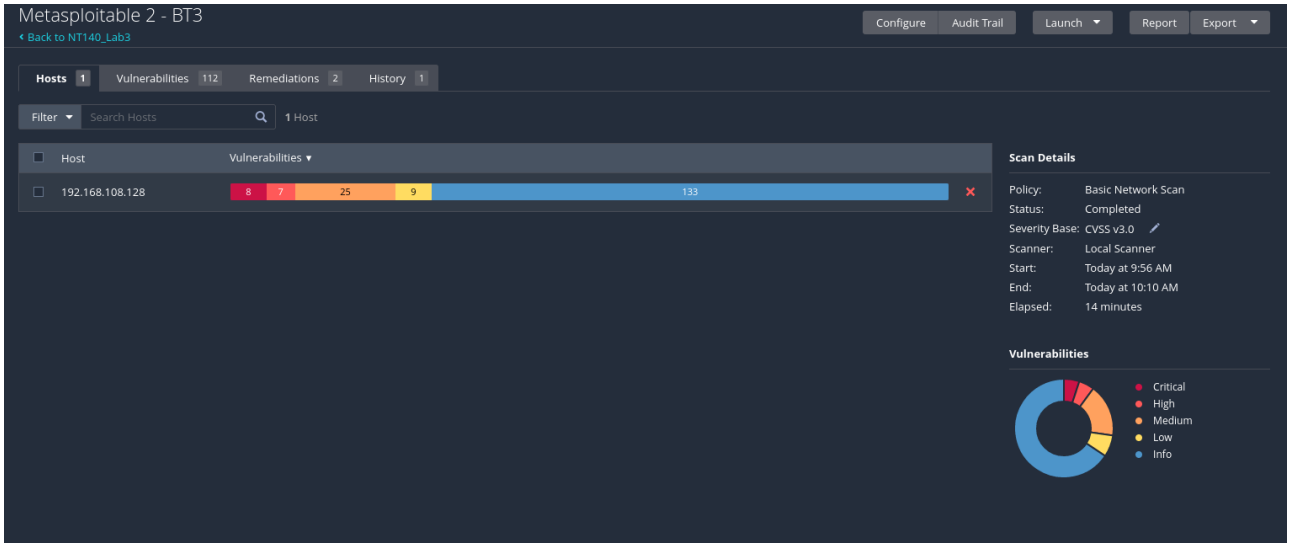
Thì trong trường hợp này máy khách kết nối thành công đến cổng 80

3. *Quét lại nhưng quét thêm port UDP.*

Để có thể quét UDP, trong phần cấu hình bật check này:



Sau đó chạy như trên, kết quả sau:

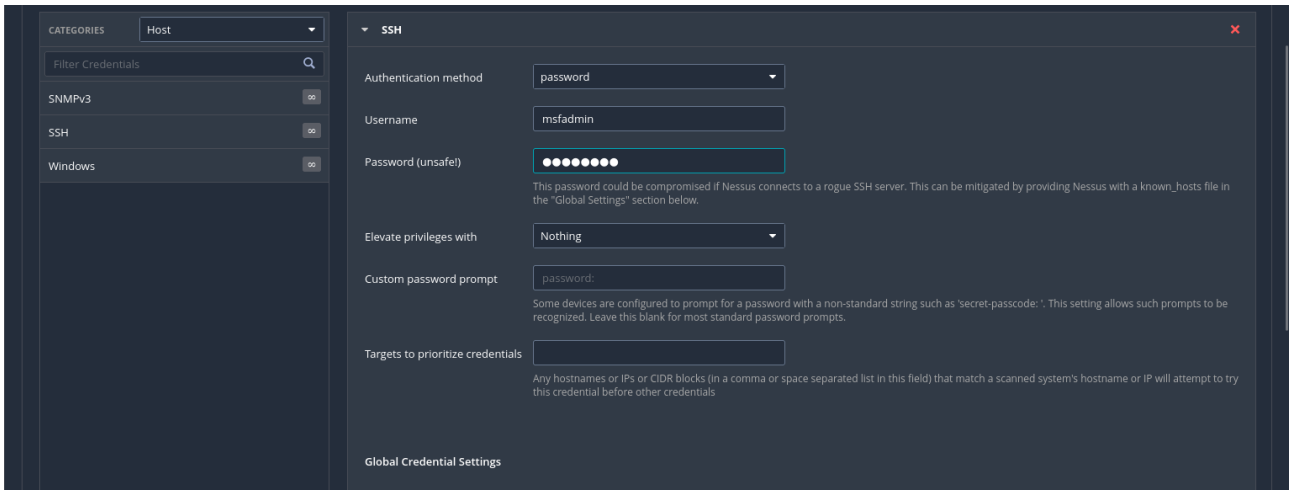


So với tcp thì udp ít hơn tcp 2 info

e) Quét lỗ hổng sử dụng tài khoản chứng thực

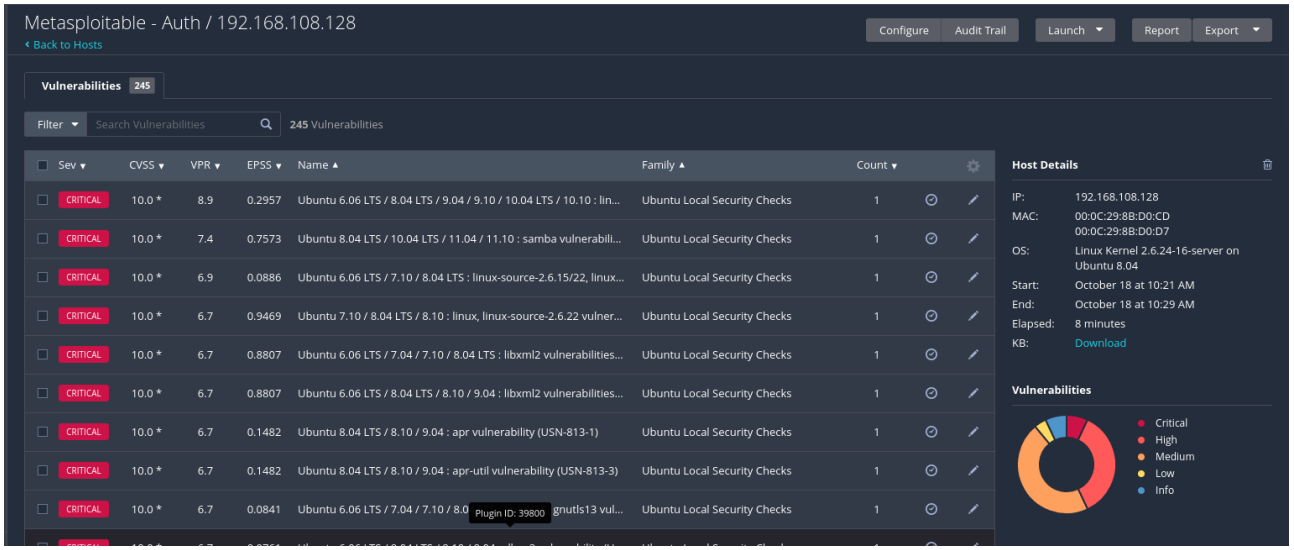
> **® Bài tập về nhà (yêu cầu làm)**
>
> 4. *Thực hiện lại các bước trên để quét máy Metasploitable 2 có sử dụng tài khoản chứng thực.*

Cũng làm như trên với phần đặt tên và địa chỉ đích nhưng với template là *Credentialed Patch Audit* , ở phần credentials sẽ chọn method password, sử dụng username và password mặc định của metasploitable 2 msfadmin:msfadmin



Sau đó lưu và chạy chương trình

Kết quả như sau



5. *Kiểm tra kết quả quét và so sánh với việc quét không sử dụng tài khoản chứng thực.*

Không chứng thực

Có chứng thực:

	Không chứng thực	Có chứng thực
Số lỗ hổng	68	245
Số info	135	60
Số Critical	8	17
Số High	7	88
Số Medium	25	115
Số Low	9	9

Nhận xét:

-Khi quét dùng tài khoản không chứng thực sẽ lấy được nhiều thông tin hơn

-Khi nói về tìm lỗ hổng thì tài khoản chứng thực sẽ tìm thấy lỗ hổng gấp 4 so với tài khoản không chứng thực, nhất là những lỗ hổng có mức độ cao và vừa

6. Hãy liệt kê các ưu, nhược điểm khi quét có tài khoản chứng thực và không có tài khoản chứng thực.

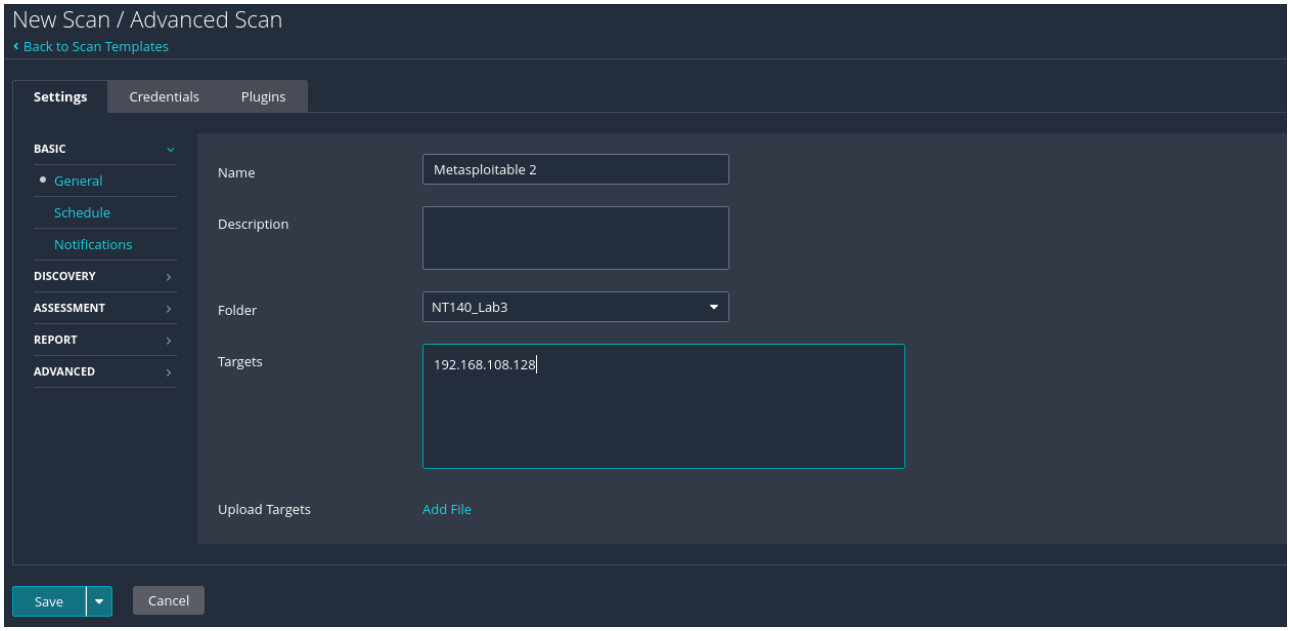
	Không chứng thực	Có chứng thực
Ưu điểm	-Không cần thông tin chứng thực -Tìm thấy những plugin cục bộ	-Có thể tìm thấy và kiểm tra plugin bên ngoài -Tìm thấy nhiều lỗ hổng hơn
Nhược điểm	-Không tìm thấy plugin bên ngoài -Tìm thấy ít lỗ hổng hơn	-Cần biết thông tin chứng thực

f) Quét với Plugin được chỉ định

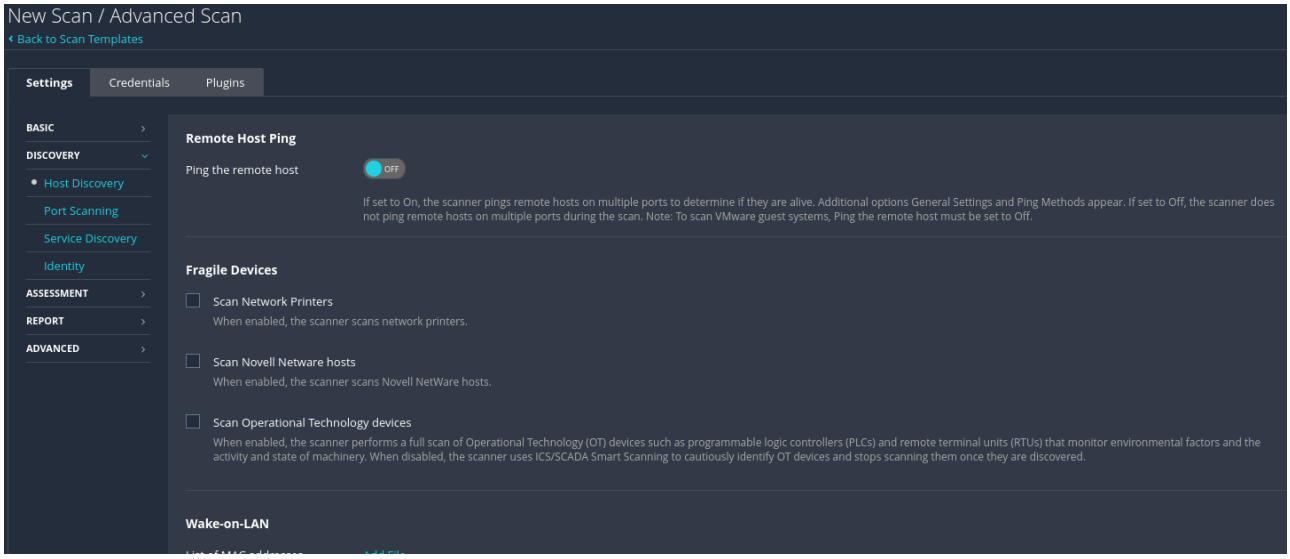
Bài tập về nhà (yêu cầu làm)

7. *Thực hiện lại các bước trên để quét máy Metasploitable 2 sử dụng plugin NFS Exported Share Information Disclosure*

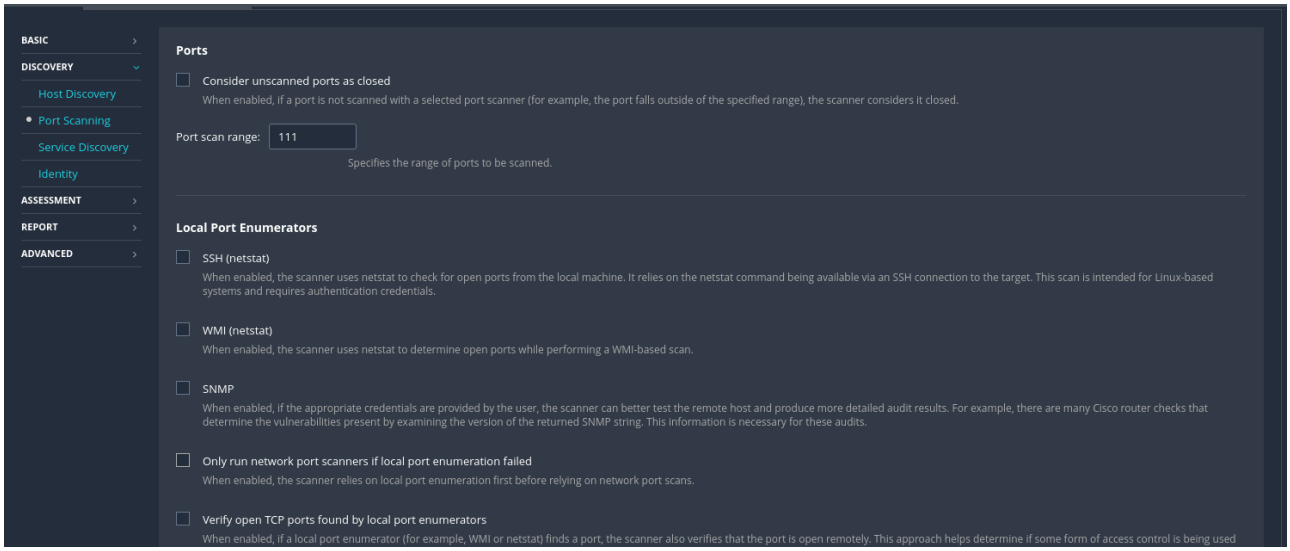
Sử dụng Advanced Scan template, đặt tên và địa chỉ đích



Tắt ping remote host để quét nhanh hơn

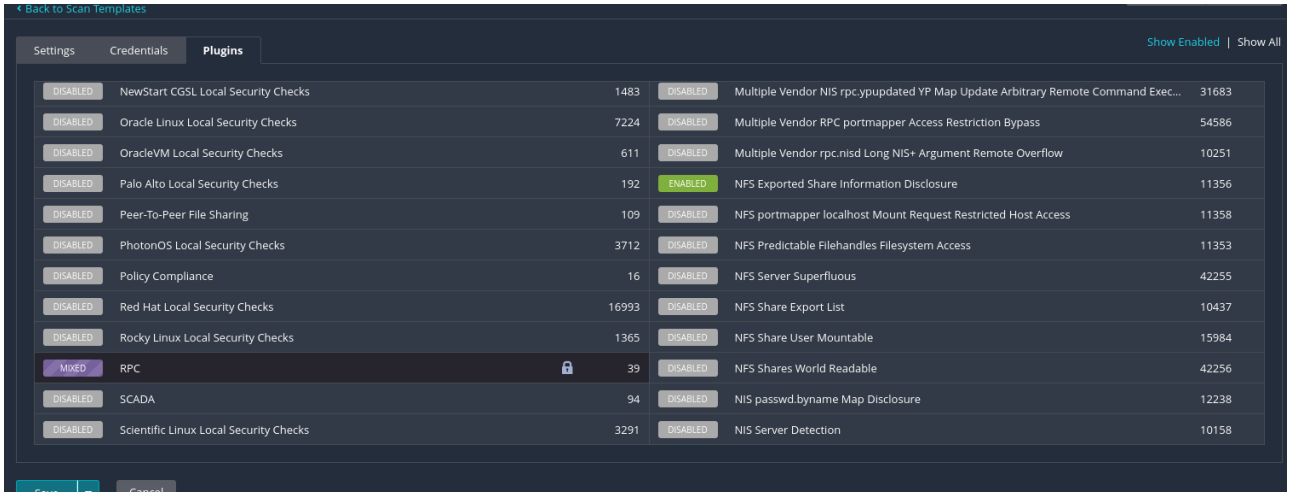


Đặt cổng là 111 và tắt các tính năng local port enumerators

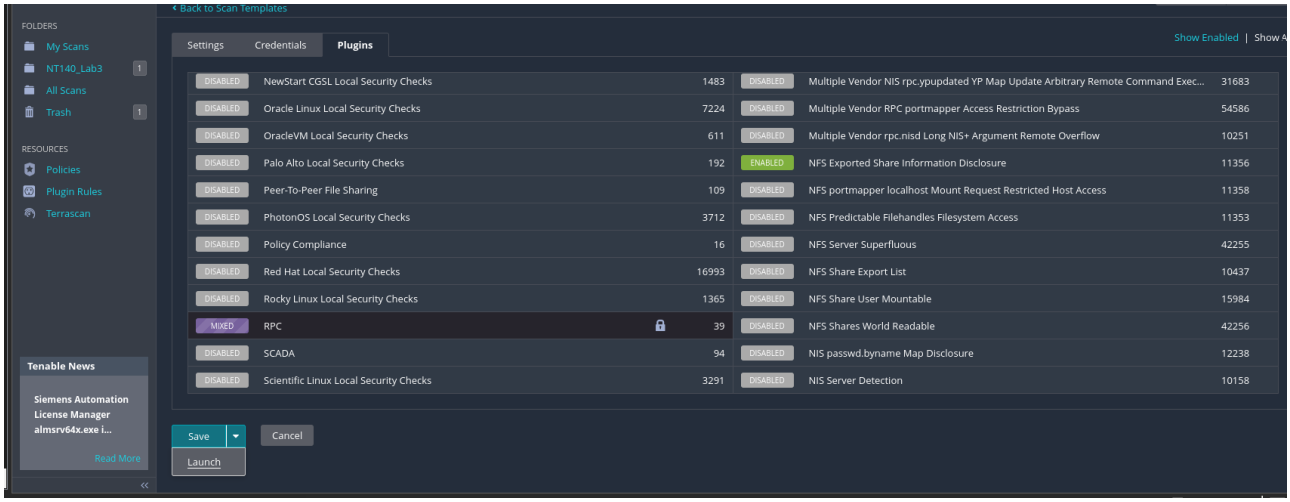


Tắt các plugin ở phần plugin

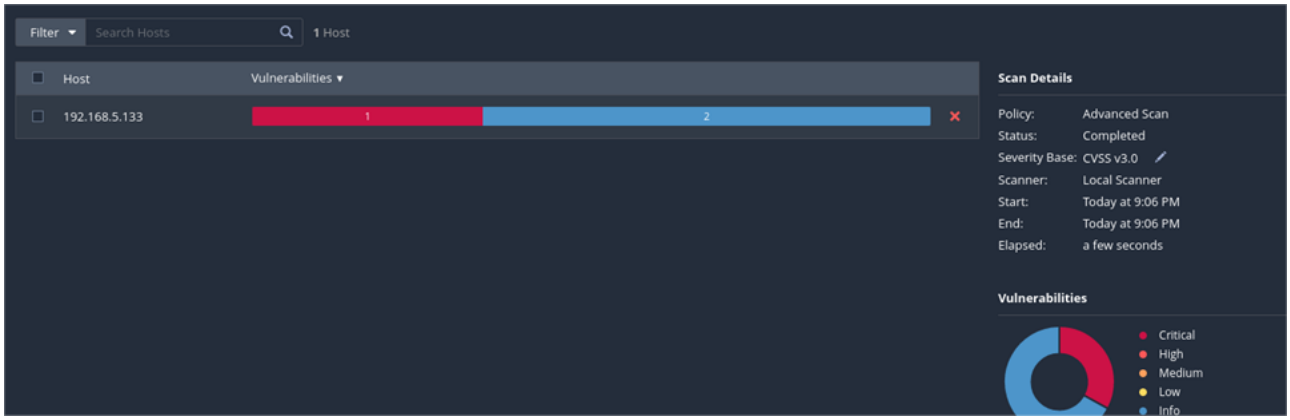
Chỉ bật đúng plugin NFS Exported Share Information Discosure của RPC

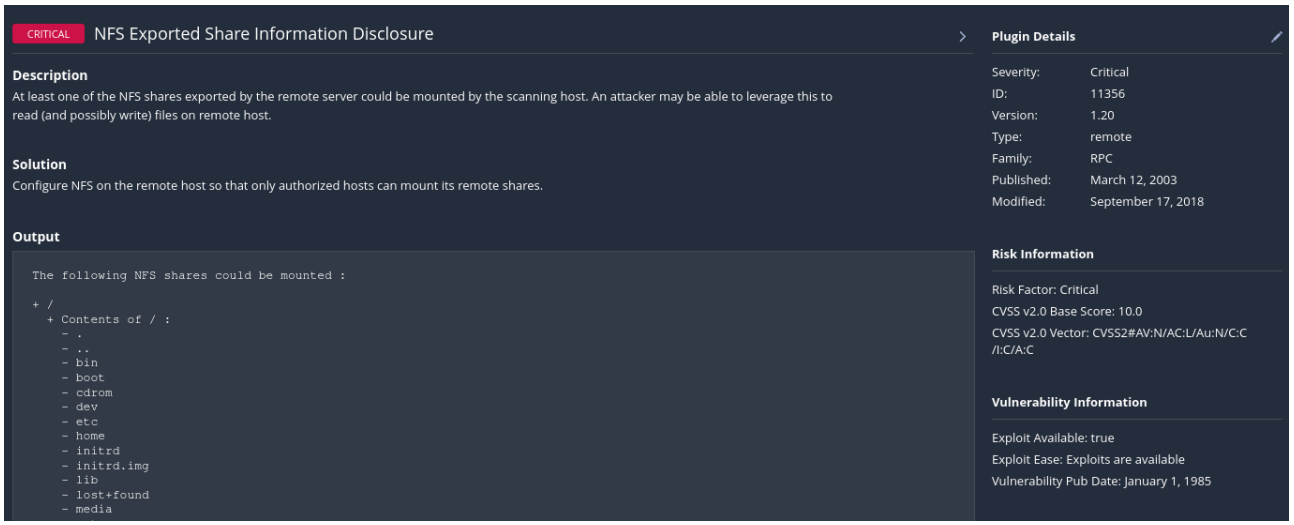


Chọn Launch để quét



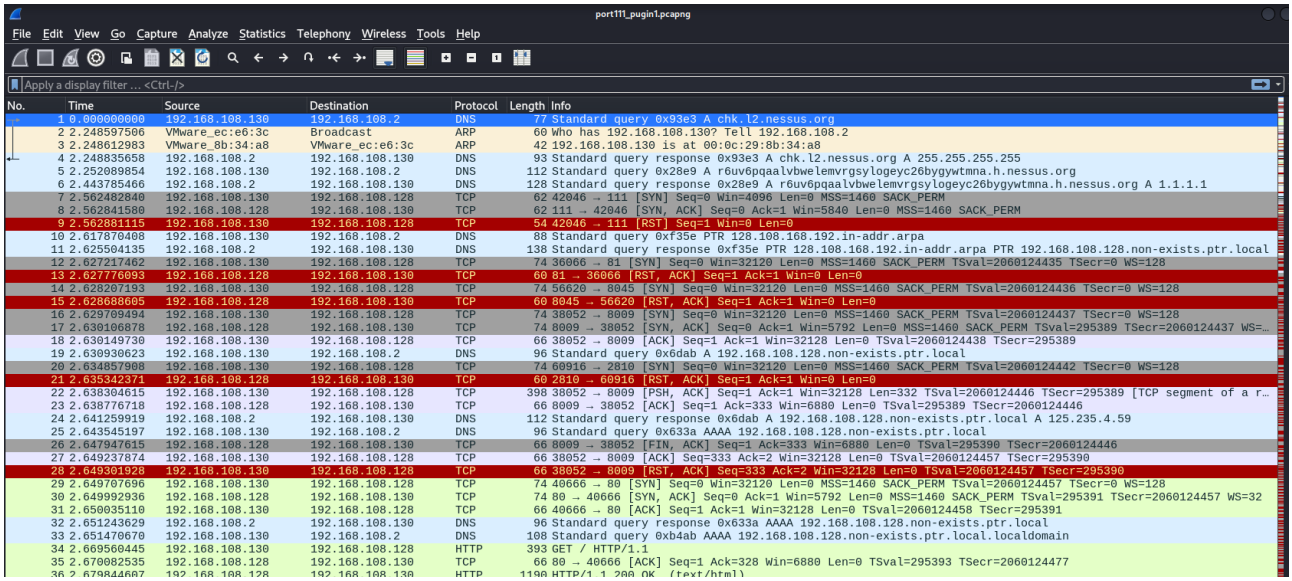
Kết quả sau khi quét sẽ có đúng một lỗ hổng





8. *Chạy Wireshark hoặc tcpdump trong suốt quá trình scan sử dụng 1 plugin duy nhất. Liệt kê các port khác mà Nessus thực hiện scan, mà không phải port 111? Tại sao Nessus lại scan các port khác, trong khi chúng ta đã chỉ định chỉ scan duy nhất 1 port là 111?*

Mở wireshark và quét lại



Kết quả trên cho thấy ngoài cổng 111 thì có cổng: 81, 8045, 8009, 2810, ...

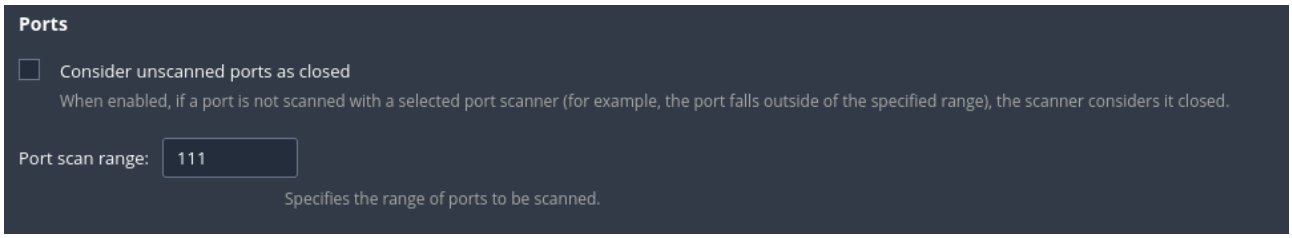
Theo thông tin trên https://community.tenable.com/s/article/Why-Is-Nessus-Scanning-Ports-Outside-Of-The-Port-Range?language=en_US

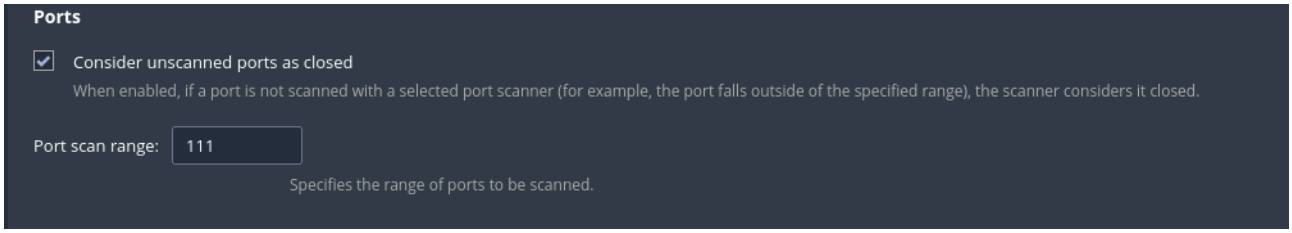
Nessus quét các cổng khác là do một số plugin sẽ kiểm tra trạng thái các cổng được hardcode mặc định, Những cổng ngoài khoảng yêu cầu quét sẽ mặc định có trạng thái không xác định.

Cũng theo mặc định, hàm get_port_state() sẽ được đặt trạng thái những cổng không xác định làm true, dẫn đến việc kết nối thử nghiệm đến cổng

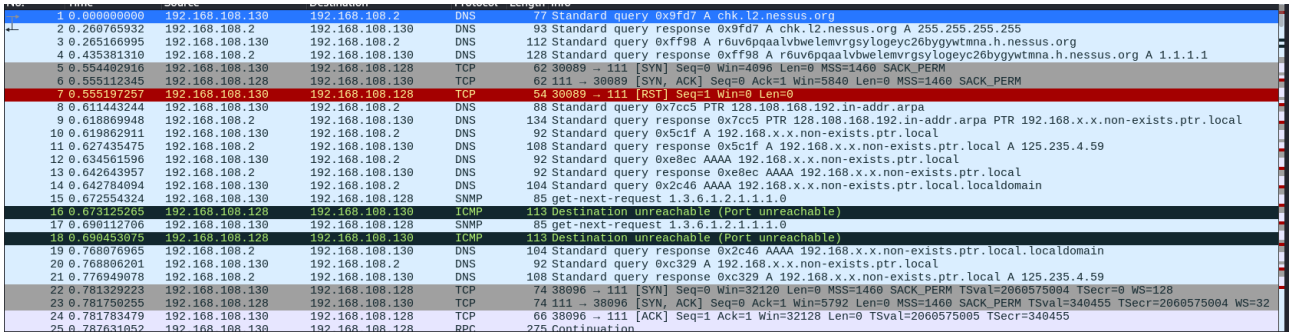
9. Mô tả cách làm để ngăn chặn việc Nessus scan port khác không phải là port được chỉ định?

Cũng trong trang web trên thì để tránh việc này, thì phải chọn vào check sau đây vì điều này sẽ làm cho hàm nói trên đặt các cổng không xác định là false

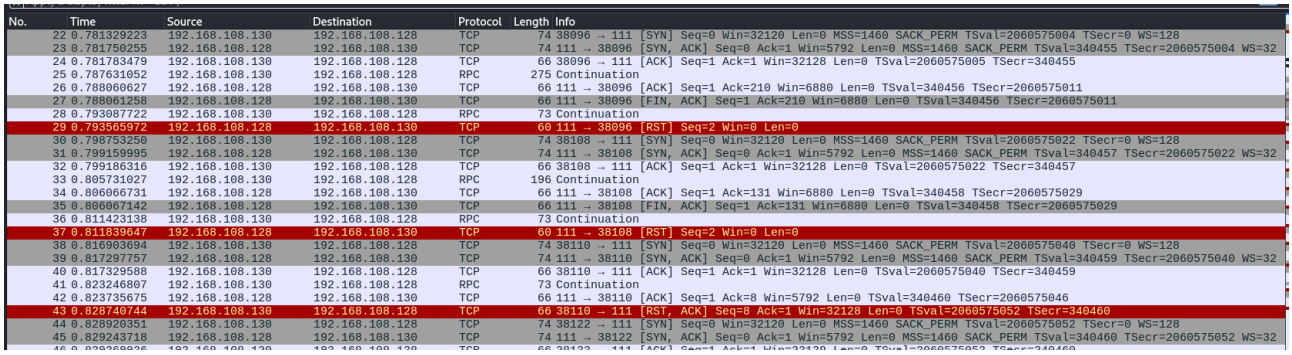




Kết quả wireshark sau khi quét lại:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.108.130	192.168.108.2	DNS	77	Standard query 0x9fd7 A chk.l2.nessus.org
2	0.260765932	192.168.108.2	192.168.108.130	DNS	93	Standard query response 0x9fd7 A chk.l2.nessus.org A 255.255.255.255
3	0.265166995	192.168.108.130	192.168.108.2	DNS	112	Standard query 0xff98 A r6uv6pqaalvbwelemvrgsylogeyc26bygywtmna.h.nessus.org
4	0.435381310	192.168.108.2	192.168.108.130	DNS	128	Standard query response 0xff98 A r6uv6pqaalvbwelemvrgsylogeyc26bygywtmna.h.nessus.org A 1.1.1.1
5	0.554402916	192.168.108.130	192.168.108.128	TCP	62	30089 → 111 [SYN] Seq=0 Win=4096 Len=0 MSS=1460 SACK_PERM
6	0.555112345	192.168.108.128	192.168.108.130	TCP	62	111 → 30089 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
7	0.555197257	192.168.108.130	192.168.108.128	TCP	54	30089 → 111 [RST] Seq=1 Win=0 Len=0
8	0.611443244	192.168.108.130	192.168.108.2	DNS	88	Standard query 0x7cc5 PTR 128.108.168.192.in-addr.arpa
9	0.618869948	192.168.108.2	192.168.108.130	DNS	134	Standard query response 0x7cc5 PTR 128.108.168.192.in-addr.arpa PTR 192.168.x.x.non-exists.ptr.local
10	0.619862911	192.168.108.130	192.168.108.2	DNS	92	Standard query 0x5c1f A 192.168.x.x.non-exists.ptr.local
11	0.627435475	192.168.108.2	192.168.108.130	DNS	108	Standard query response 0x5c1f A 192.168.x.x.non-exists.ptr.local A 125.235.4.59
12	0.634561596	192.168.108.130	192.168.108.2	DNS	92	Standard query 0xe8ec AAAA 192.168.x.x.non-exists.ptr.local
13	0.642643957	192.168.108.2	192.168.108.130	DNS	92	Standard query response 0xe8ec AAAA 192.168.x.x.non-exists.ptr.local
14	0.642784094	192.168.108.130	192.168.108.2	DNS	104	Standard query 0x2c46 AAAA 192.168.x.x.non-exists.ptr.local.localdomain
15	0.672554324	192.168.108.130	192.168.108.128	SNMP	85	get-next-request 1.3.6.1.2.1.1.1.0
16	0.673125265	192.168.108.128	192.168.108.130	ICMP	113	Destination unreachable (Port unreachable)
17	0.690112706	192.168.108.130	192.168.108.128	SNMP	85	get-next-request 1.3.6.1.2.1.1.1.0
18	0.690453075	192.168.108.128	192.168.108.130	ICMP	113	Destination unreachable (Port unreachable)
19	0.768076965	192.168.108.2	192.168.108.130	DNS	104	Standard query response 0x2c46 AAAA 192.168.x.x.non-exists.ptr.local.localdomain
20	0.768806201	192.168.108.130	192.168.108.2	DNS	92	Standard query 0xc329 A 192.168.x.x.non-exists.ptr.local
21	0.776949078	192.168.108.2	192.168.108.130	DNS	108	Standard query response 0xc329 A 192.168.x.x.non-exists.ptr.local A 125.235.4.59
22	0.781329223	192.168.108.130	192.168.108.128	TCP	74	38096 → 111 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=2060575004 TSecr=0 WS=128
23	0.781750255	192.168.108.128	192.168.108.130	TCP	74	111 → 38096 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=340455 TSecr=2060575004 WS=32
24	0.781783479	192.168.108.130	192.168.108.128	TCP	66	38096 → 111 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=2060575005 TSecr=340455
25	0.787631052	192.168.108.130	192.168.108.128	RPC	275	Continuation

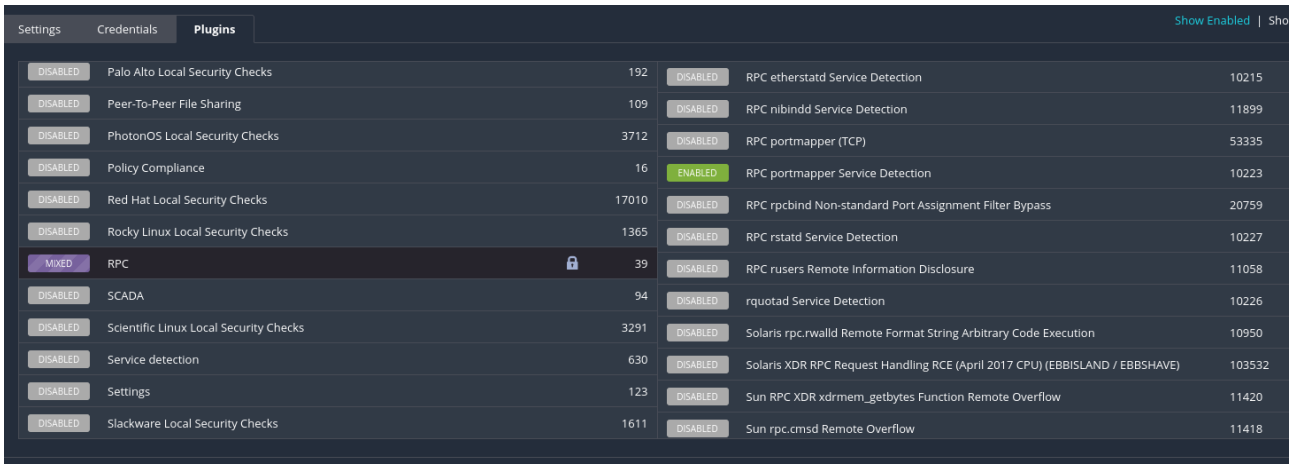


No.	Time	Source	Destination	Protocol	Length	Info
22	0.781329223	192.168.108.130	192.168.108.128	TCP	74	38096 → 111 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=2060575004 TSecr=0 WS=128
23	0.781750255	192.168.108.128	192.168.108.130	TCP	74	111 → 38096 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=340455 TSecr=2060575004 WS=32
24	0.781783479	192.168.108.130	192.168.108.128	TCP	66	38096 → 111 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=2060575005 TSecr=340455
25	0.787631052	192.168.108.130	192.168.108.128	RPC	275	Continuation
26	0.788060627	192.168.108.128	192.168.108.130	TCP	66	111 → 38096 [ACK] Seq=1 Ack=210 Win=6880 Len=0 TSval=340456 TSecr=2060575011
27	0.788061258	192.168.108.128	192.168.108.130	TCP	66	111 → 38096 [FIN, ACK] Seq=1 Ack=210 Win=6880 Len=0 TSval=340456 TSecr=2060575011
28	0.793087722	192.168.108.130	192.168.108.128	RPC	73	Continuation
29	0.793565972	192.168.108.128	192.168.108.130	TCP	60	111 → 38096 [RST] Seq=2 Win=0 Len=0
30	0.798753250	192.168.108.130	192.168.108.128	TCP	74	38108 → 111 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=2060575022 TSecr=0 WS=128
31	0.799159995	192.168.108.128	192.168.108.130	TCP	74	111 → 38108 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=340457 TSecr=2060575022 WS=32
32	0.799186316	192.168.108.130	192.168.108.128	TCP	66	38108 → 111 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=2060575022 TSecr=340457
33	0.805731027	192.168.108.130	192.168.108.128	RPC	196	Continuation
34	0.806066731	192.168.108.128	192.168.108.130	TCP	66	111 → 38108 [ACK] Seq=1 Ack=131 Win=6880 Len=0 TSval=340458 TSecr=2060575029
35	0.806067142	192.168.108.128	192.168.108.130	TCP	66	111 → 38108 [FIN, ACK] Seq=1 Ack=131 Win=6880 Len=0 TSval=340458 TSecr=2060575029
36	0.811423138	192.168.108.130	192.168.108.128	RPC	73	Continuation
37	0.811839647	192.168.108.128	192.168.108.130	TCP	60	111 → 38108 [RST] Seq=2 Win=0 Len=0
38	0.816903694	192.168.108.130	192.168.108.128	TCP	74	38110 → 111 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=2060575040 TSecr=0 WS=128
39	0.817297757	192.168.108.128	192.168.108.130	TCP	74	111 → 38110 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=340459 TSecr=2060575040 WS=32
40	0.817329588	192.168.108.130	192.168.108.128	TCP	66	38110 → 111 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=2060575040 TSecr=340459
41	0.823246807	192.168.108.130	192.168.108.128	RPC	73	Continuation
42	0.823735675	192.168.108.128	192.168.108.130	TCP	66	111 → 38110 [ACK] Seq=1 Ack=8 Win=5792 Len=0 TSval=340460 TSecr=2060575046
43	0.828740744	192.168.108.130	192.168.108.128	TCP	66	38110 → 111 [RST, ACK] Seq=8 Ack=1 Win=32128 Len=0 TSval=2060575052 TSecr=340460
44	0.828920351	192.168.108.130	192.168.108.128	TCP	74	38122 → 111 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=2060575052 TSecr=0 WS=128
45	0.829243718	192.168.108.128	192.168.108.130	TCP	74	111 → 38122 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=340460 TSecr=2060575052 WS=32

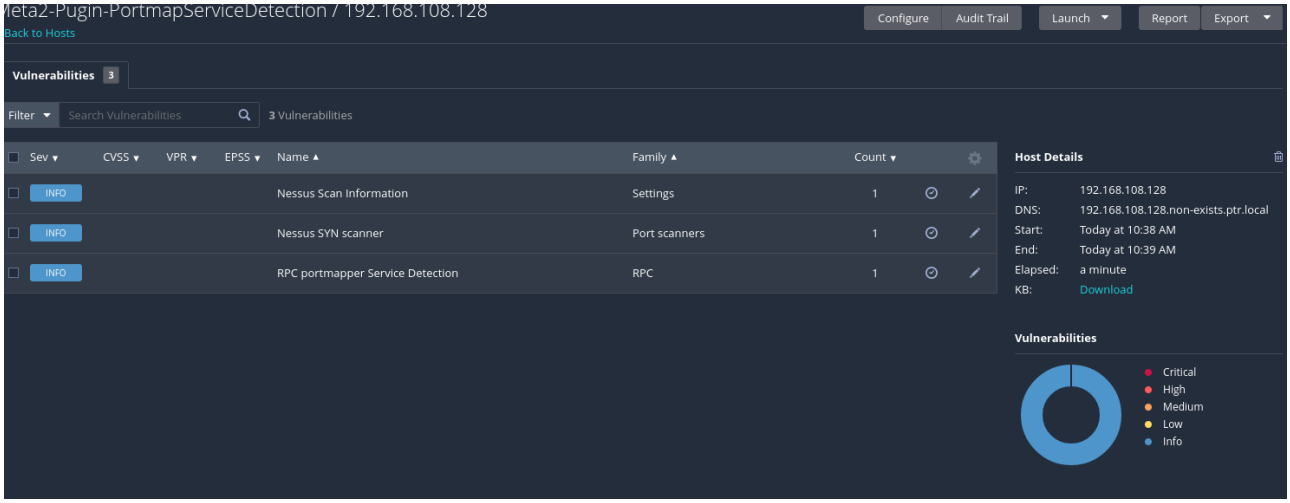
Chỉ có đúng cổng 111

10. Thực hiện quét lại sử dụng 2 plugin khác.

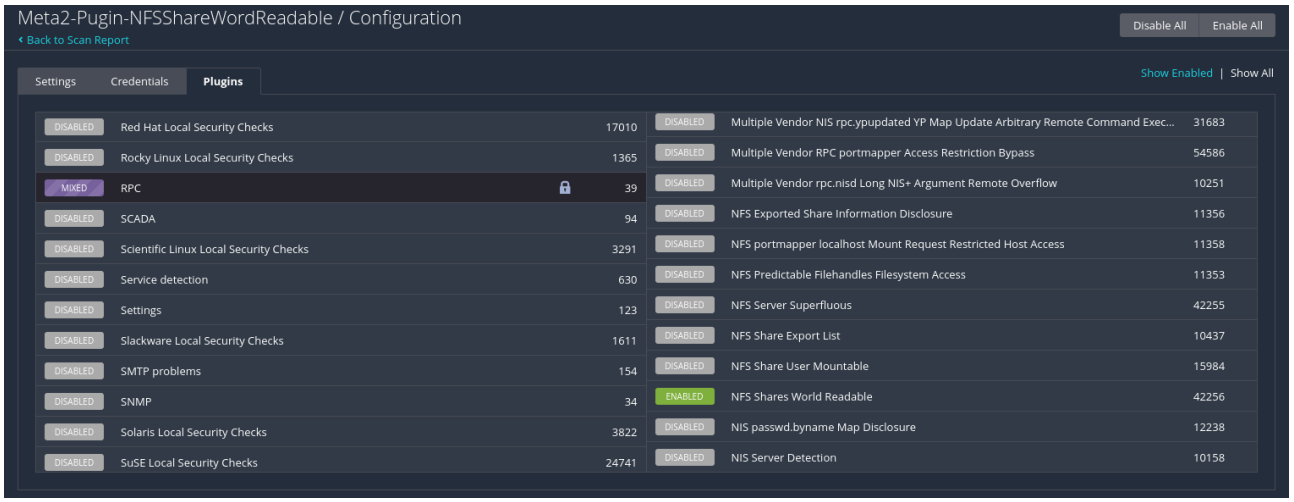
Ví dụ 1: sử dụng plugin RPC portmapper Service Detection



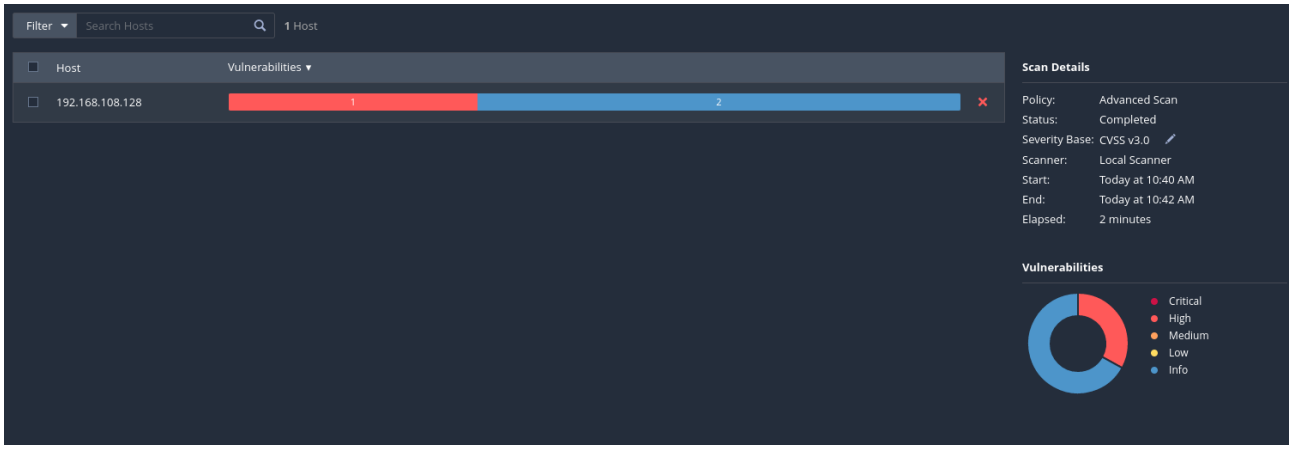
Kết quả:



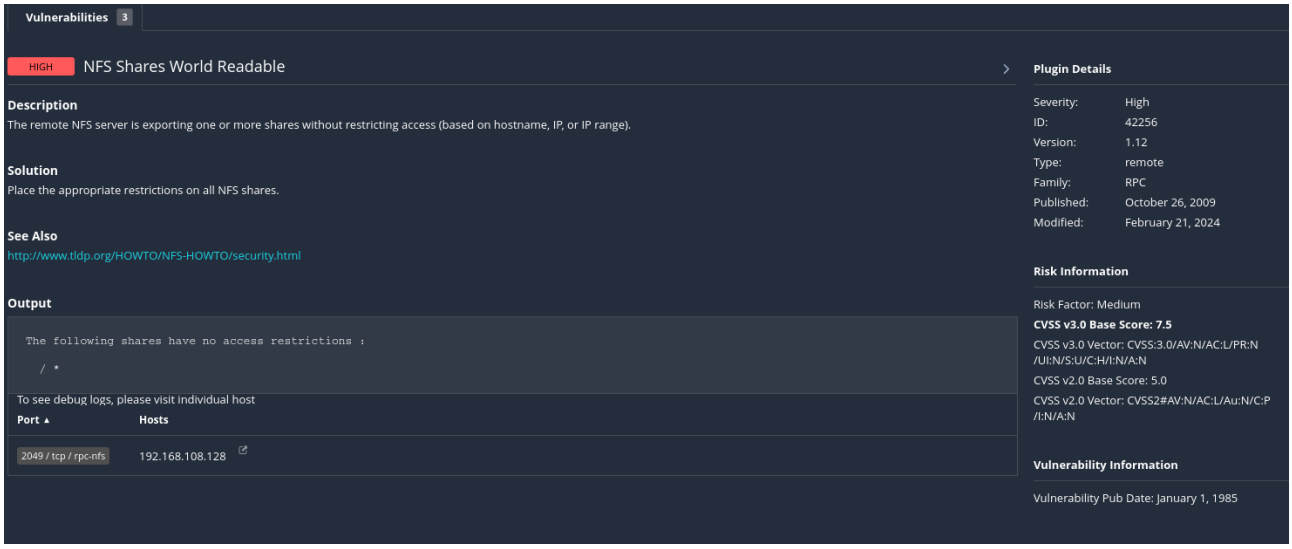
Ví dụ 2: sử dụng plugin NFS Share World Readable



Kết quả:



Chi tiết lỗ hổng:

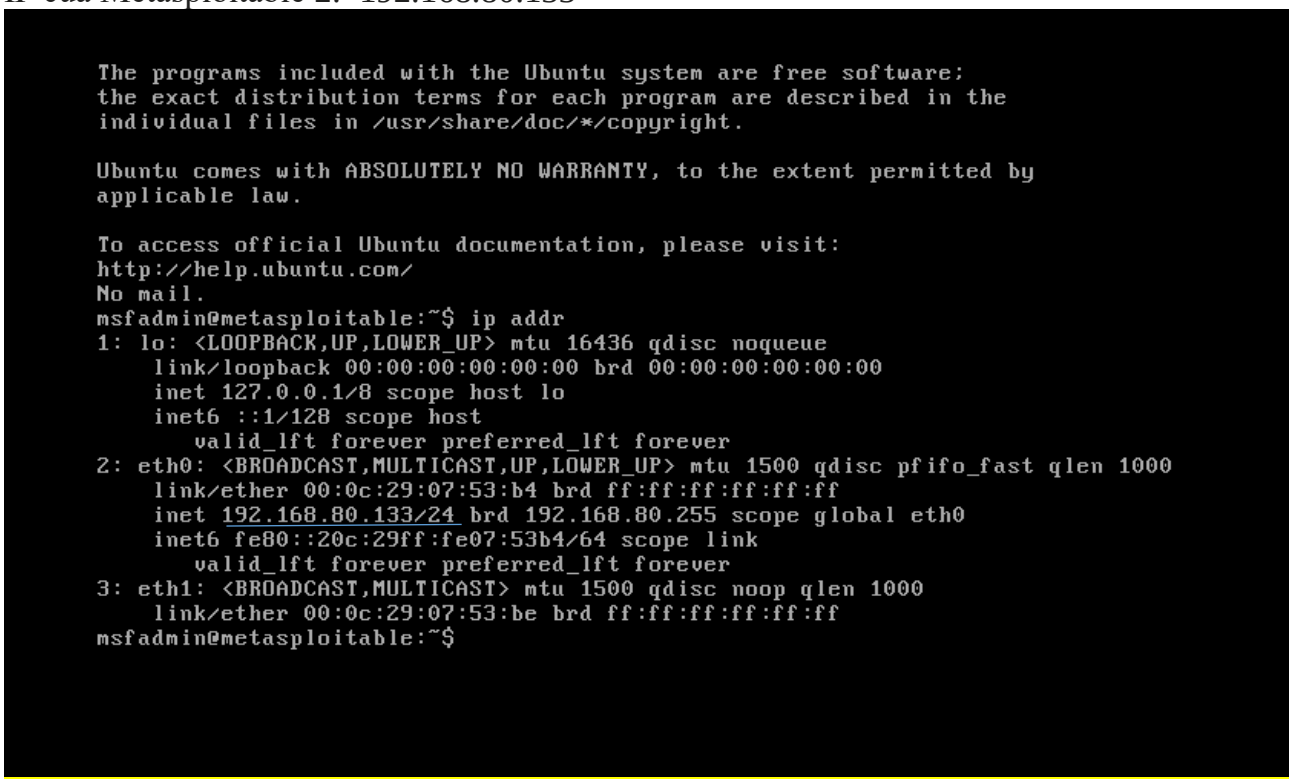


2. Bài tập nhóm

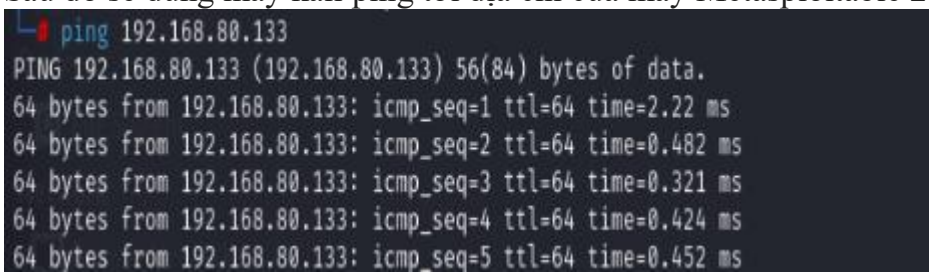
> **Bài tập về nhà (yêu cầu làm)**
>
> *11. Sinh viên/nhóm sinh viên tìm hiểu 1 trong các công cụ quét lỗ hổng tự động sau đây, và viết báo cáo kết quả theo như các phần đã chia ở bài tập 1:*

Ở đây nhóm em sẽ chọn công cụ là sniper

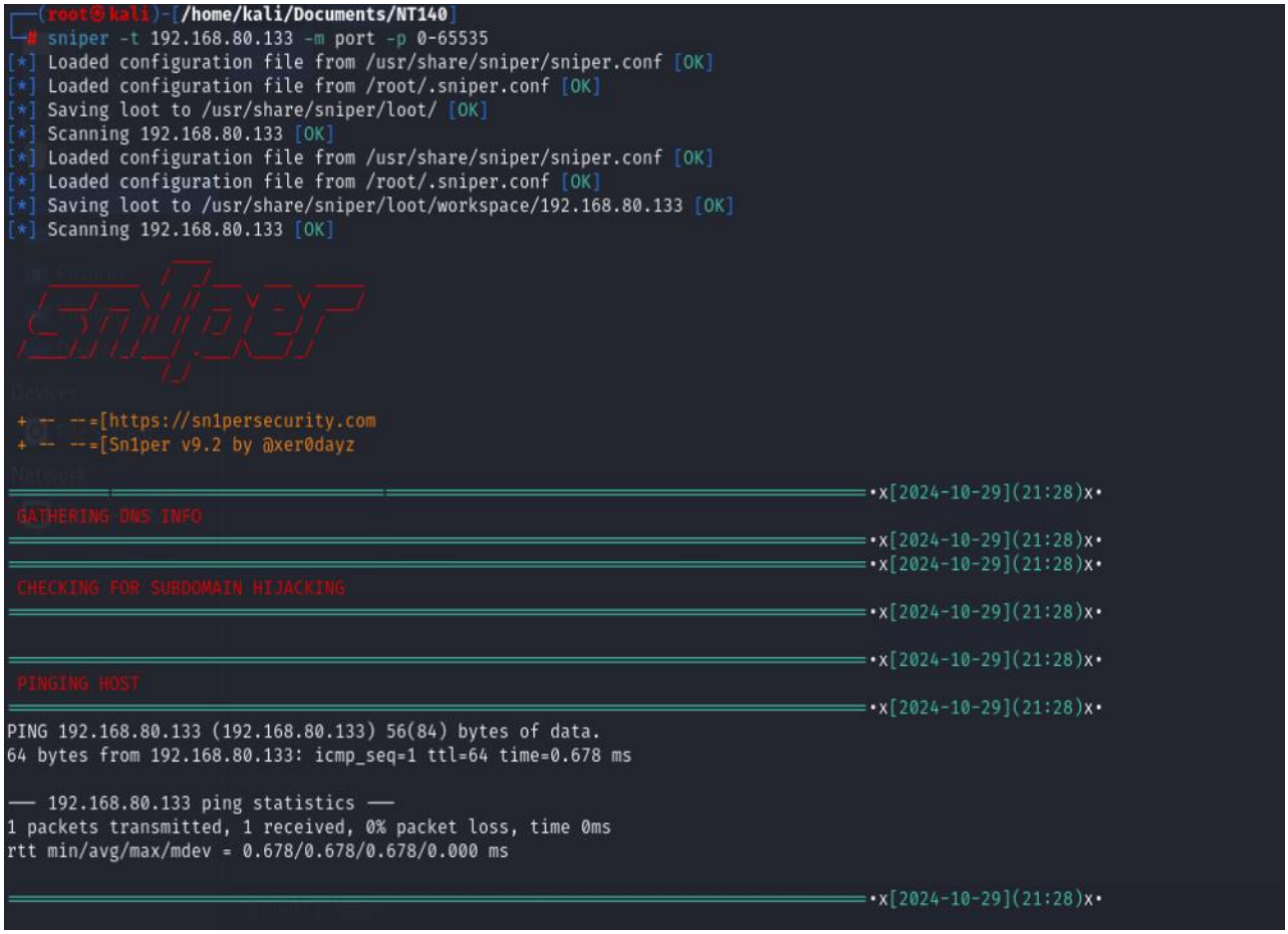
IP của Metasploitable 2: 192.168.80.133



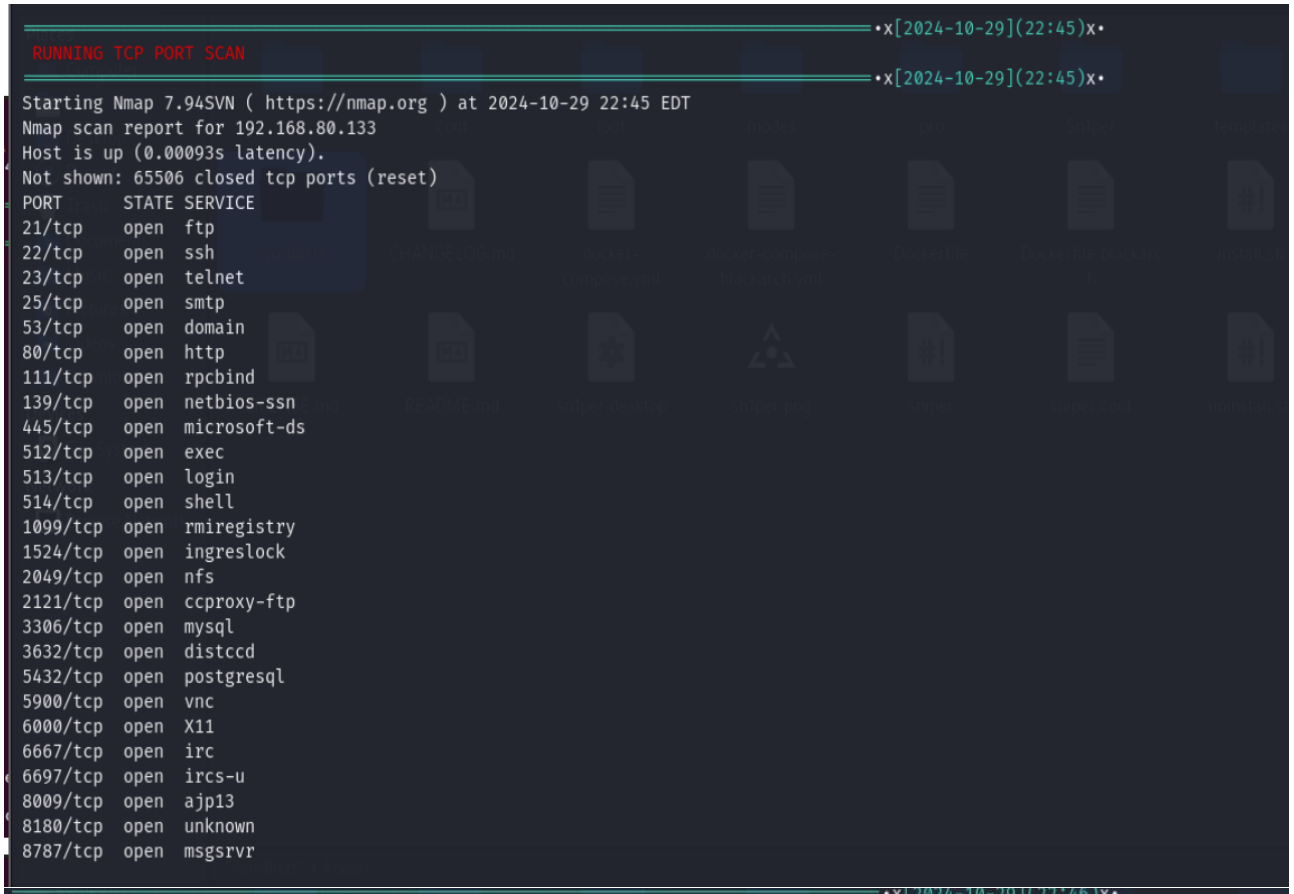
Sau đó sẽ dùng máy kali ping tới địa chỉ của máy Metasploitable 2

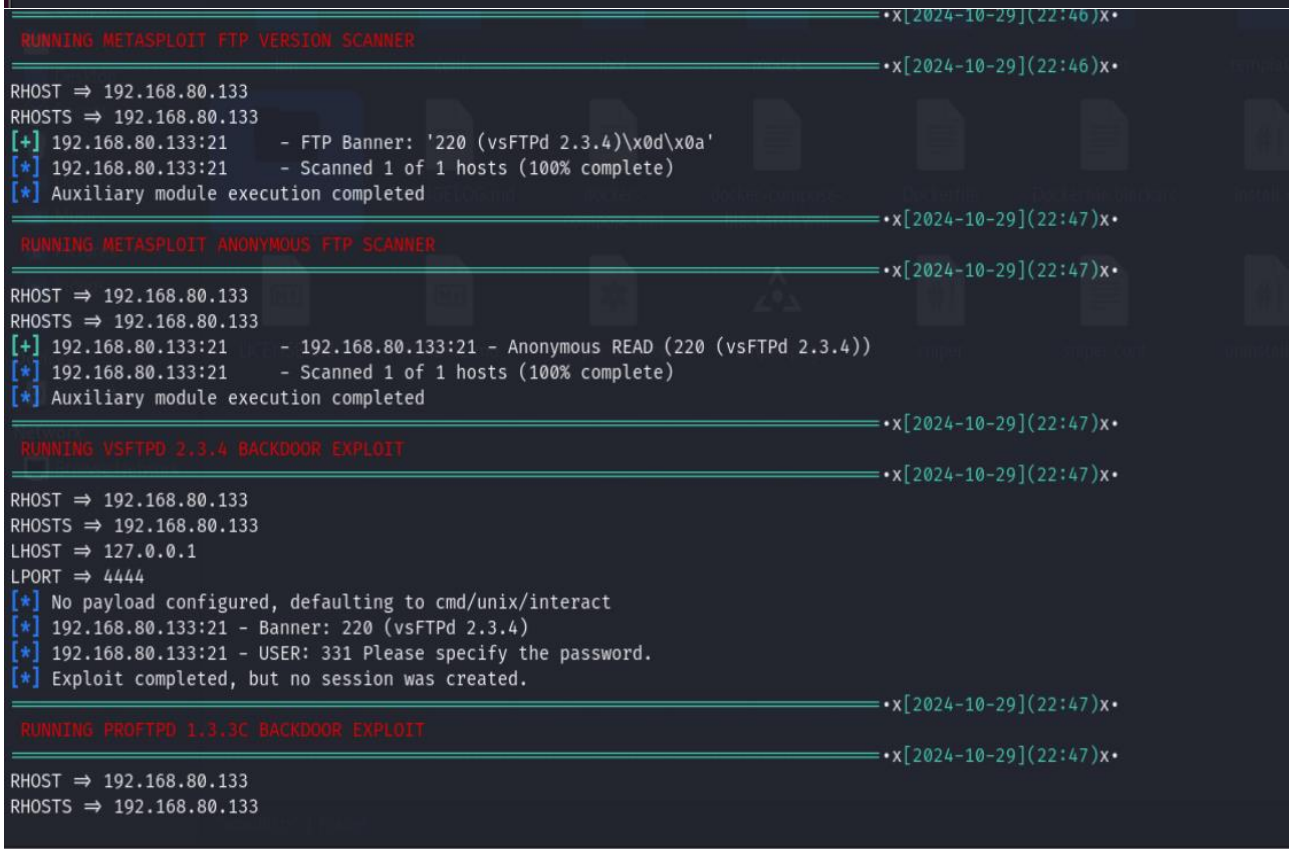


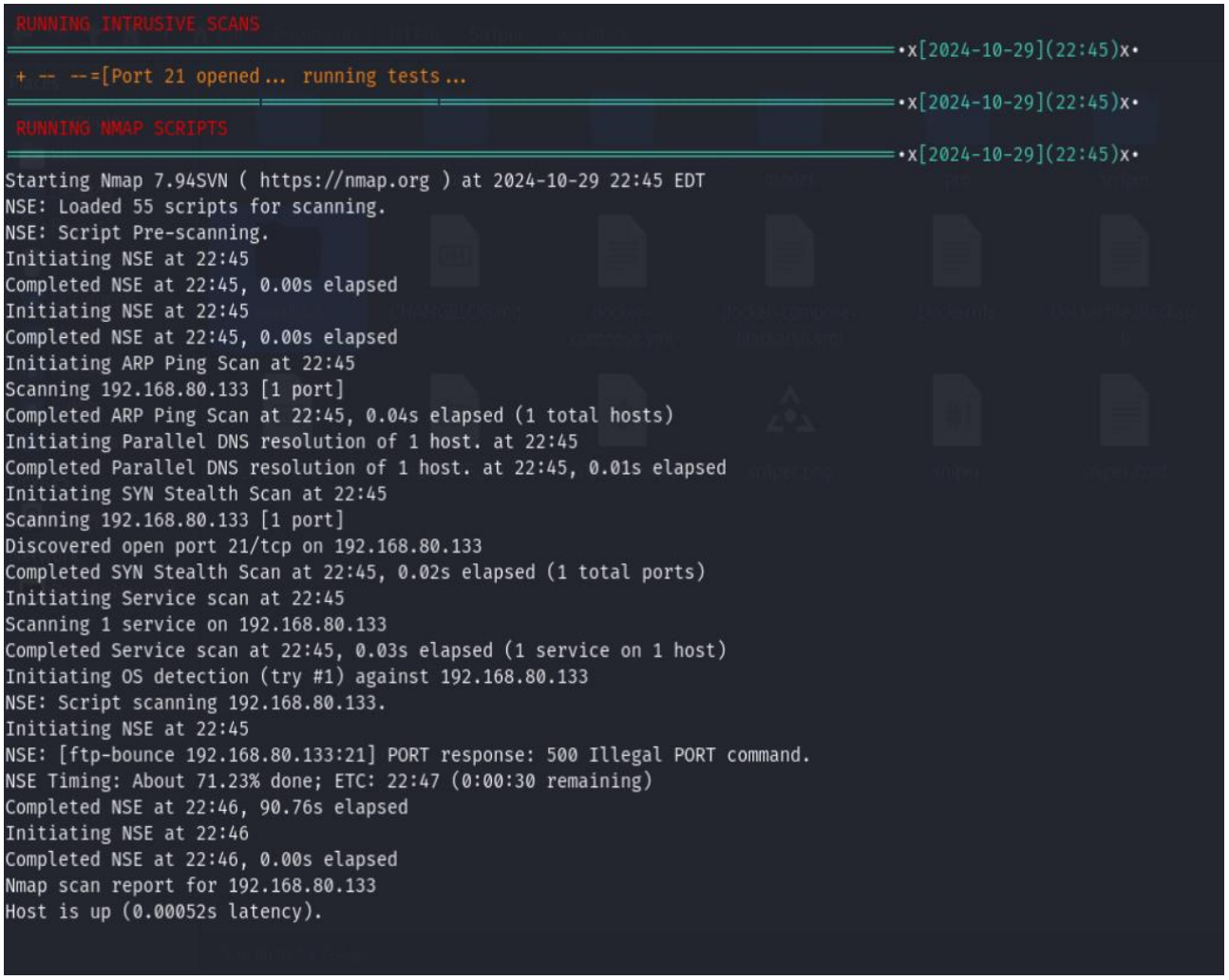
Sau khi tải được sniper sẽ có giao diện và chạy với lệnh "sniper – t 192.168.80.133 -m port -p 0-65535" với IP của Metasploitable 2 là 192.168.80.133, target là máy ảo Metasploitable 2, sử dụng port từ 0 đến 65535:



Sau khi sử dụng lệnh có thể thấy sniper dùng rất nhiều cách để khai thác lỗ hổng.


•x[2024-10-29](22:45)x•
RUNNING TCP PORT SCAN
•x[2024-10-29](22:45)x•
Starting Nmap 7.94SVN (https://nmap.org) at 2024-10-29 22:45 EDT
Nmap scan report for 192.168.80.133
Host is up (0.00093s latency).
Not shown: 65506 closed tcp ports (reset)
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
23/tcp open telnet
25/tcp open smtp
53/tcp open domain
80/tcp open http
111/tcp open rpcbind
139/tcp open netbios-ssn
445/tcp open microsoft-ds
512/tcp open exec
513/tcp open login
514/tcp open shell
1099/tcp open rmiregistry
1524/tcp open ingreslock
2049/tcp open nfs
2121/tcp open ccproxy-ftp
3306/tcp open mysql
3632/tcp open distccd
5432/tcp open postgresql
5900/tcp open vnc
6000/tcp open X11
6667/tcp open irc
6697/tcp open ircs-u
8009/tcp open ajp13
8180/tcp open unknown
8787/tcp open msgsrvr


RUNNING METASPLOIT FTP VERSION SCANNER
•x[2024-10-29](22:46)x•
RHOST ⇒ 192.168.80.133
RHOSTS ⇒ 192.168.80.133
[+] 192.168.80.133:21 - FTP Banner: '220 (vsFTPd 2.3.4)\x0d\x0a'
[*] 192.168.80.133:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
•x[2024-10-29](22:47)x•
RUNNING METASPLOIT ANONYMOUS FTP SCANNER
•x[2024-10-29](22:47)x•
RHOST ⇒ 192.168.80.133
RHOSTS ⇒ 192.168.80.133
[+] 192.168.80.133:21 - 192.168.80.133:21 - Anonymous READ (220 (vsFTPd 2.3.4))
[*] 192.168.80.133:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
•x[2024-10-29](22:47)x•
RUNNING VSFTPD 2.3.4 BACKDOOR EXPLOIT
•x[2024-10-29](22:47)x•
RHOST ⇒ 192.168.80.133
RHOSTS ⇒ 192.168.80.133
LHOST ⇒ 127.0.0.1
LPORT ⇒ 4444
[*] No payload configured, defaulting to cmd/unix/interact
[*] 192.168.80.133:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.80.133:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
•x[2024-10-29](22:47)x•
RUNNING PROFTPD 1.3.3C BACKDOOR EXPLOIT
•x[2024-10-29](22:47)x•
RHOST ⇒ 192.168.80.133
RHOSTS ⇒ 192.168.80.133



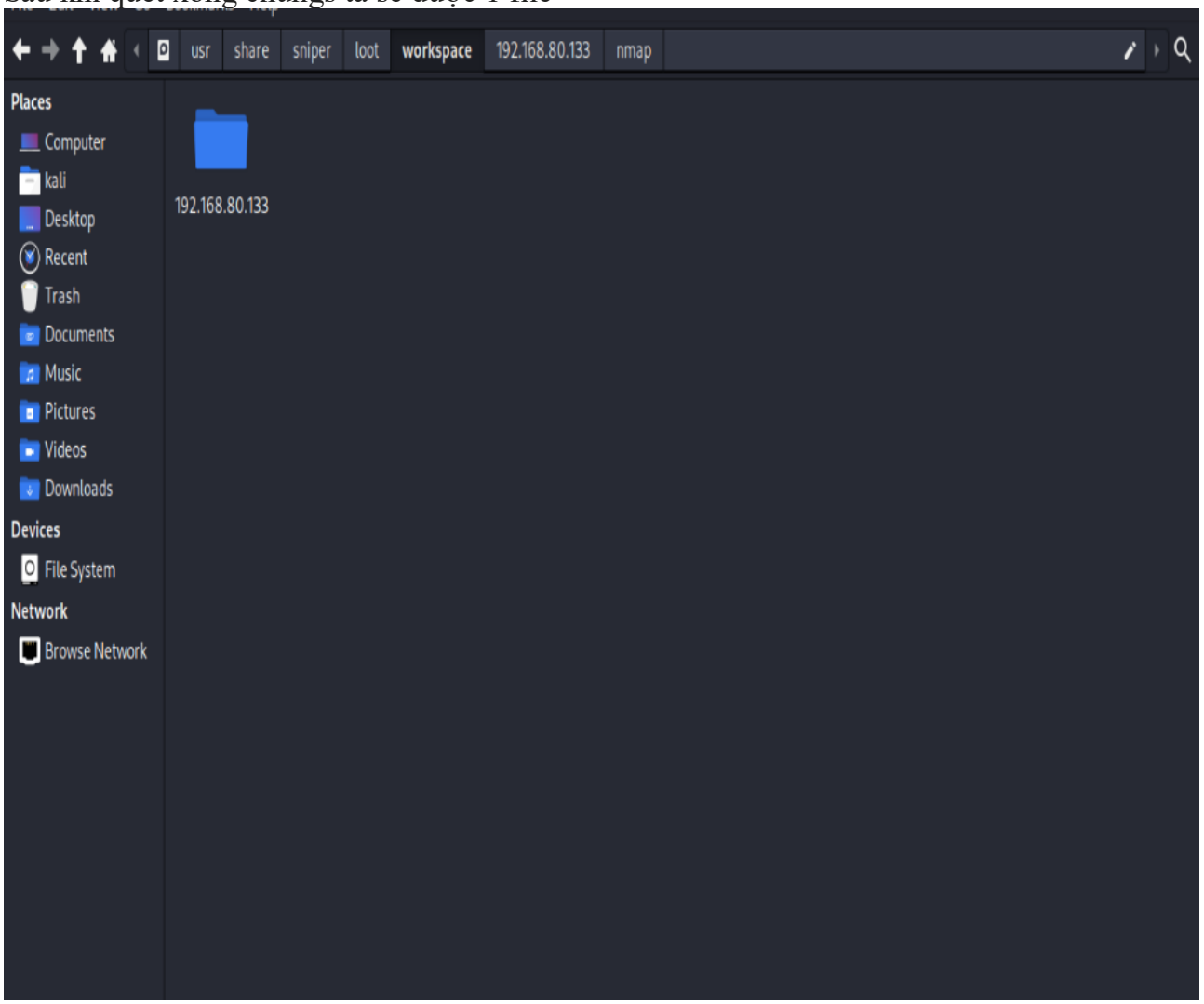
Và quét ra rất nhiều lỗ hổng và 1 số CVE

```
| vulners:
|   cpe:/a:openbsd:openssh:4.7p1:
|       95499236-C9FE-56A6-9D7D-E943A24B633A    10.0    https://vulners.com/githubexploit/95499236-C9FE-56A6-9D7D-E943A24B633A  *EX
|       2C119FFA-ECE0-5E14-A4A4-354A2C38071A    10.0    https://vulners.com/githubexploit/2C119FFA-ECE0-5E14-A4A4-354A2C38071A  *EX
|       CVE-2023-38408  9.8     https://vulners.com/cve/CVE-2023-38408
|       CVE-2016-1908   9.8     https://vulners.com/cve/CVE-2016-1908
|       B8190CDB-3EB9-5631-9828-8064A1575B23    9.8     https://vulners.com/githubexploit/B8190CDB-3EB9-5631-9828-8064A1575B23  *EX
|       8FC9C5AB-3968-5F3C-825E-E8DB5379A623    9.8     https://vulners.com/githubexploit/8FC9C5AB-3968-5F3C-825E-E8DB5379A623  *EX
|       8AD01159-548E-546E-AA87-2DE89F3927EC    9.8     https://vulners.com/githubexploit/8AD01159-548E-546E-AA87-2DE89F3927EC  *EX
|       5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A    9.8     https://vulners.com/githubexploit/5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A  *EX
|       0221525F-07F5-5790-912D-F4B9E2D1B587    9.8     https://vulners.com/githubexploit/0221525F-07F5-5790-912D-F4B9E2D1B587  *EX
|       CVE-2015-5600   8.5     https://vulners.com/cve/CVE-2015-5600
|       SSV:78173       7.8     https://vulners.com/seebug/SSV:78173    *EXPLOIT*
|       SSV:69983       7.8     https://vulners.com/seebug/SSV:69983    *EXPLOIT*
|       PACKETSTORM:98796       7.8     https://vulners.com/packetstorm/PACKETSTORM:98796       *EXPLOIT*
|       PACKETSTORM:94556       7.8     https://vulners.com/packetstorm/PACKETSTORM:94556       *EXPLOIT*
|       PACKETSTORM:140070      7.8     https://vulners.com/packetstorm/PACKETSTORM:140070      *EXPLOIT*
|       PACKETSTORM:101052      7.8     https://vulners.com/packetstorm/PACKETSTORM:101052      *EXPLOIT*
|       EXPLOITPACK:71D51B69AA2D3A74753D7A921EE79985    7.8     https://vulners.com/exploitpack/EXPLOITPACK:71D51B69AA2D3A74753D7A9
5       *EXPLOIT*
|       EXPLOITPACK:67F6569F63A082199721C069C852BBD7    7.8     https://vulners.com/exploitpack/EXPLOITPACK:67F6569F63A082199721C06
7       *EXPLOIT*
|       EXPLOITPACK:5BCA798C6BA71FAE29334297EC0B6A09    7.8     https://vulners.com/exploitpack/EXPLOITPACK:5BCA798C6BA71FAE2933429
9       *EXPLOIT*
|       EDB-ID:24450    7.8     https://vulners.com/exploitdb/EDB-ID:24450      *EXPLOIT*
|       EDB-ID:15215    7.8     https://vulners.com/exploitdb/EDB-ID:15215      *EXPLOIT*
|       CVE-2020-15778  7.8     https://vulners.com/cve/CVE-2020-15778
|       CVE-2016-10012  7.8     https://vulners.com/cve/CVE-2016-10012
|       CVE-2015-8325   7.8     https://vulners.com/cve/CVE-2015-8325
|       1337DAY-ID-26494        7.8     https://vulners.com/zdt/1337DAY-ID-26494        *EXPLOIT*
|       SSV:92579       7.5     https://vulners.com/seebug/SSV:92579    *EXPLOIT*
```

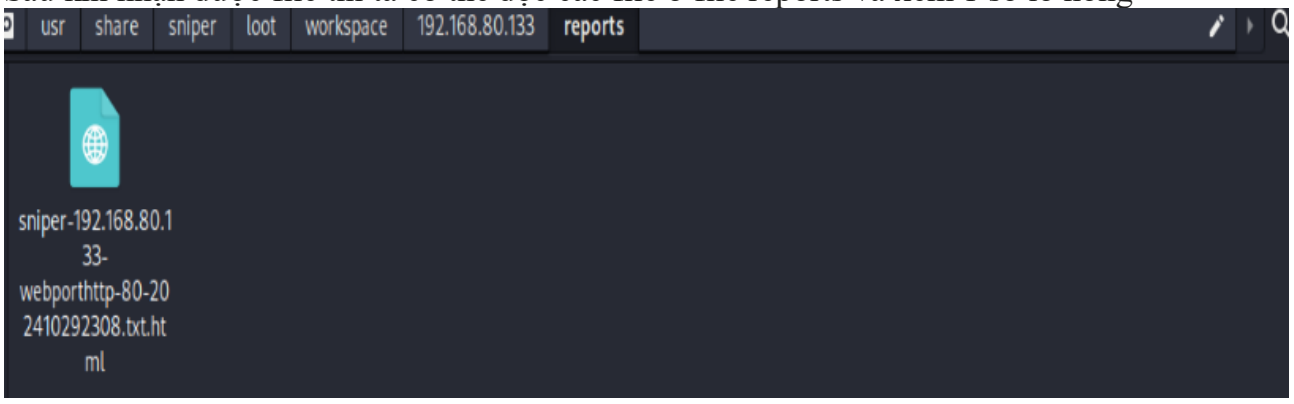
Thực hiện quét wireshark

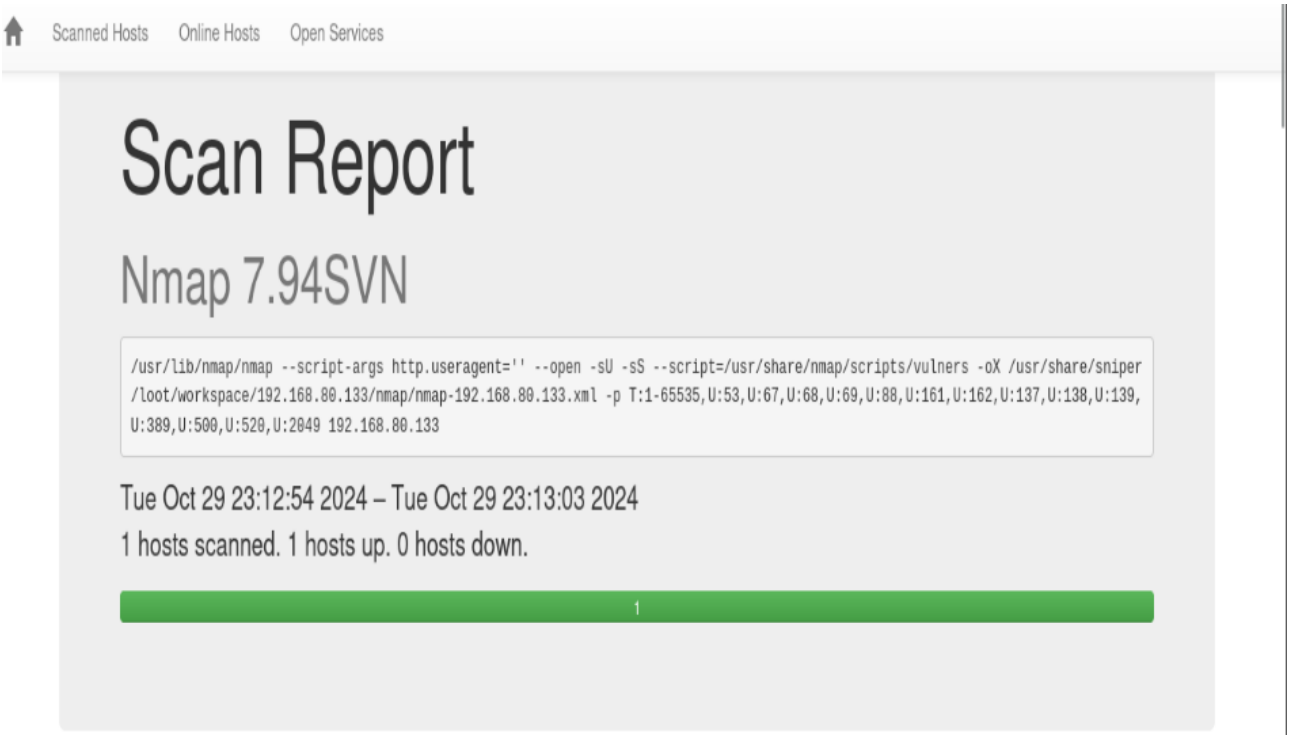
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.80.133	192.168.80.131	SMTP	80	S: 250 2.1.0 Ok
2	0.000253340	192.168.80.131	192.168.80.133	SMTP	109	C: RCPT TO: rooty@metasploitable.localdomain
3	0.000543790	192.168.80.133	192.168.80.131	TCP	66	25 → 41337 [ACK] Seq=15 Ack=44 Win=181 Len=0 TSval=5
4	1.001286261	192.168.80.133	192.168.80.131	SMTP	179	S: 550 5.1.1 <rooty@metasploitable.localdomain>: Rec
5	1.001545347	192.168.80.131	192.168.80.133	SMTP	72	C: RSET
6	1.001779945	192.168.80.133	192.168.80.131	TCP	66	25 → 41337 [ACK] Seq=128 Ack=50 Win=181 Len=0 TSval=
7	2.002449485	192.168.80.133	192.168.80.131	SMTP	80	S: 250 2.0.0 Ok
8	2.002892579	192.168.80.131	192.168.80.133	SMTP	110	C: MAIL FROM: root@metasploitable.localdomain
9	2.003164289	192.168.80.133	192.168.80.131	TCP	66	25 → 41337 [ACK] Seq=142 Ack=94 Win=181 Len=0 TSval=

Sau khi quét xong chungs ta sẽ được 1 file



Sau khi nhận được file thì ta có thể đọc các file ở file reports và xem 1 số lỗ hổng





Và sau khi quét xong sẽ được kết quả: