Федеральное государственное автономное образовательное учреждение Высшего профессионального образования

Санкт-Петербургский политехнический университет

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Лабораторная работа №3

Утилита для исследования сети и сканер портов Nmap

Выполнил студент
группы 53501/3

_____
П. П. Жук
«____» _____ 2016 г.

Проверил преподаватель

_____
К. Д. Вылегжанина
«____» _____ 2016 г.

Санкт-Петербург
2016 г.

Содержание

1 Цель работы

Изучить основные аспекты работы с утилитой Nmap на различных примерах.

2 Задание

1. Провести поиск активных хостов.
2. Определить открытые порты.
3. Определить версии сервисов.
4. Изучить файлы nmap-services, nmap-os-db, nmap-service-probes
5. Добавить новую сигнатуру службы в файл nmap-service-probes.
6. Сохранить вывод утилиты в формате xml.
7. Исследвать различные этапы и режимы работы Nmap с использованием утилиты Wireshark.
8. Просканировать виртуальную машину Metasploitable2 используя утилиту db_nmap из состава metasploit-framework.
9. Выбрать пять записей из файла nmap-service-probes и описать их работу.
10. Выбрать один скрипт из состава Nmap и описать его работу.

3 Ход работы

Для проведения лабораторной работы было создано две виртуальные машины, объединенные в общую сеть. Metasploitable2 - виртуальная машина, намеренно содержащая ряд уязвимостей. Kali Linux - виртуальная машина, необходима для сканирования и поиска уязвимостей на первой виртуальной машине. Определим IP-адреса виртуальных машин:

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.226.135  netmask 255.255.255.0  broadcast 192.168.226.255
        inet6 fe80::20c:29ff:fe48:9c2a  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:48:9c:2a  txqueuelen 1000  (Ethernet)
        RX packets 81  bytes 11060 (10.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 74  bytes 6535 (6.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Виртуальная машина с ОС Metasploitable2 имеет адрес 192.168.226.134, вторая (с Kali Linux) имеет адрес 192.168.226.135.

Для проверки работоспособности сети "пропингуем"виртуальные машины:

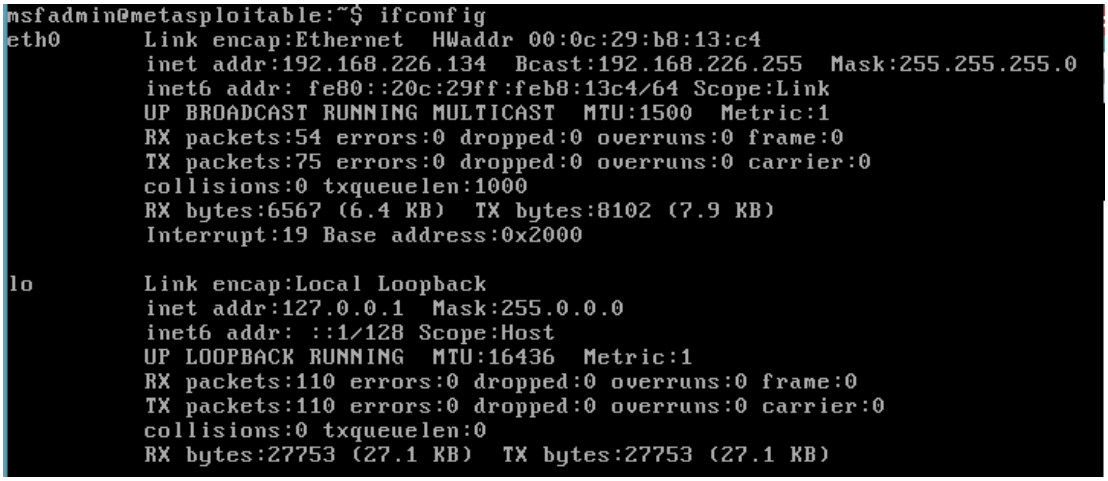

Рис. 1: IP-адрес Metasploitable2

```
root@kali:~# ping 192.168.226.134
PING 192.168.226.134 (192.168.226.134) 56(84) bytes of data.
64 bytes from 192.168.226.134: icmp_seq=1 ttl=64 time=0.230 ms
64 bytes from 192.168.226.134: icmp_seq=2 ttl=64 time=0.384 ms
64 bytes from 192.168.226.134: icmp_seq=3 ttl=64 time=0.376 ms
64 bytes from 192.168.226.134: icmp_seq=4 ttl=64 time=0.410 ms
^C
--- 192.168.226.134 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.230/0.350/0.410/0.070 ms
```

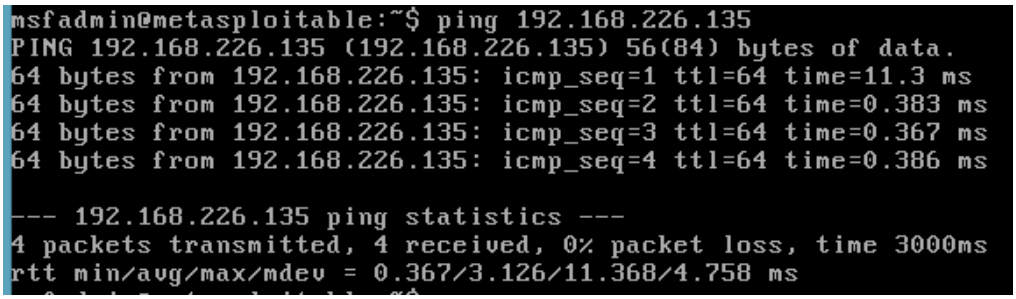

Рис. 2: Пингуем Kali Linux

Пакеты проходят, значит сеть работоспособна.

3.1 Провести поиск активных хостов

Для поиска активных хостов используется флаг -sP. При этом также нужно указать адрес подсети, который в данном случае будет 192.168.226.* .

```
root@kali:~# nmap -sP 192.168.226.*
```

```
Starting Nmap 7.01 ( https://nmap.org ) at 2016-06-14 09:54 EDT
Nmap scan report for 192.168.226.1
Host is up (0.00018s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.226.2
Host is up (0.000098s latency).
MAC Address: 00:50:56:E5:ED:96 (VMware)
Nmap scan report for 192.168.226.134
Host is up (0.00035s latency).
MAC Address: 00:0C:29:B8:13:C4 (VMware)
Nmap scan report for 192.168.226.254
Host is up (0.00020s latency).
MAC Address: 00:50:56:F2:AB:24 (VMware)
Nmap scan report for 192.168.226.135
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.01 seconds
```

В результате было найдено 5 активных хостов, среди которых присутсвует и Metasploitable2.

3.2 Определить открытые порты

Для определения открытых портов утилите необходимо передать адрес хоста. Простой командой nmap <цель сканирования> будет произведено сканирование более чем 1660 TCP портов на <целевой машине>.

```
root@kali:~# nmap 192.168.226.134

Starting Nmap 7.01 ( https://nmap.org ) at 2016-06-14 10:27 EDT
Nmap scan report for 192.168.226.134
Host is up (0.00013s latency).
Not shown: 978 closed ports
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
```

```
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8180/tcp open  unknown
MAC Address: 00:0C:29:B8:13:C4 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

В результате мы получили адреса открытых портов и названия соответсвующих им сервисов.

3.3 Определить версии сервисов

Для определения вресии сервиса используется флаг -sV.

```
root@kali:~# nmap -sV 192.168.226.134

Starting Nmap 7.01 ( https://nmap.org ) at 2016-06-14 10:40 EDT
Nmap scan report for 192.168.226.134
Host is up (0.00015s latency).
Not shown: 978 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         Unreal ircd
8180/tcp open  unknown
MAC Address: 00:0C:29:B8:13:C4 (VMware)
Service Info: Hosts:  metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OSs:

Service detection performed. Please report any incorrect results at https://nmap.org/submi
Nmap done: 1 IP address (1 host up) scanned in 147.51 seconds
```

3.4 Изучить файлы nmap-services, nmap-os-db, nmap-service-probes

Служебные файлы для nmap можно найти в директории "/usr/share/nmap".

Файл **nmap-services** описывает назначение стандартных портов. Каждая запись имеет следующий вид: имя_сервиса, номер_порта/протокол, вероятность_открытости_порта, опциональный_комментарий.

Для известных, зарезервированных номеров портов, файл содержит подробное описание. Для незакрепленных портов также есть записи, но с "unknown"в качестве имени:

```
fln-spx 221/udp 0.000577 # Berkeley rlogind with SPX auth
rsh-spx 222/tcp 0.000941 # Berkeley rshd with SPX auth
rsh-spx 222/udp 0.000774 # Berkeley rshd with SPX auth
cdc 223/tcp 0.000125 # Certificate Distribution Center
cdc 223/udp 0.000346 # Certificate Distribution Center
masqdialer 224/tcp 0.000025
unknown 225/tcp 0.000100
unknown 225/udp 0.000330
unknown 226/tcp 0.000013
unknown 228/tcp 0.000013
```

Файл **nmap-os-db** содержит примеры ответов различных операционных систем при сканировании. Это необходимо для того, что бы узнать какая операционная система находится на данном хосте. Пример:

```
MatchPoints
SEQ(SP=25%GCD=75%ISR=25%TI=100%CI=50%II=100%SS=80%TS=100)
OPS(O1=20%O2=20%O3=20%O4=20%O5=20%O6=20)
WIN(W1=15%W2=15%W3=15%W4=15%W5=15%W6=15)
ECN(R=100%DF=20%T=15%TG=15%W=15%O=15%CC=100%Q=20)
```

Для вывода информации об ОС на хосте используется ключ -O:

```
root@kali:~# nmap -O 192.168.226.134

Starting Nmap 7.01 ( https://nmap.org ) at 2016-06-14 11:04 EDT
Nmap scan report for 192.168.226.134
Host is up (0.00036s latency).
Not shown: 978 closed ports
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
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8180/tcp open  unknown
MAC Address: 00:0C:29:B8:13:C4 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
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 1.76 seconds
```

Видно, что на Metasploitable2 используется Linux с версией ядра 2.6.

Файл **nmap-service-probes** содержит сигнатуры для определения сервисов, прослушивающих тот или иной порт. Понимание устройства этого файла позволит добавить сигнатуру для, например, собственного сервиса. Каждая запись в данном файле определяется одной сторокой. Строки, начинающиеся с хэша (#) рассматриваются как комментарии и игнорируются. Пустые строки игнорируются также. Другие строки должны содержать одну из директив, описанных ниже.

- Probe <protocol> <probename> <probestring>
  Директива Probe содержит строку, необходимую для отправки при распознавании сервиса.
- match <service> <pattern> [<versioninfo>]
  Директива match необходима при распознавании сервиса на основе ответов на строку, отправленную предыдущей директивой Probe.
- ports <portlist>
  Директива ports содержит порты сервиса.
- rarity <value between 1 and 9>
  Директива rarity указывает частоту, с которой от сервиса можно ожидать возвращения корректных результатов.
- некоторые другие

Пример из файла nmap-service-probes

```
# Microsoft ActiveSync Version 3.7 Build 3083 (It's used for syncing
# my ipaq it disappears when you remove the ipaq.)
match activesync m|^.\0\x01\0[^\0]\0[^\0]\0[^\0]\0[^\0]\0[^\0]\0.*\0\0\0$|s p/Microsoft Ac
match activesync m|^\(\0\0\0\x02\0\0\0\x03\0\0\0\+\0\0\x003\0\0\0\0\0\0\0\x04\0\0‘\x01\0\0
```

3.5 Добавить новую сигнатуру службы в файл nmap-service-probes

Создадим простой сервер, который мы будем идентифицировать с помощью утилиты nmap.

```
#include <sys/socket.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define PORT 8133

int main(int argc, char** argv)
{
char str[100];
char *msg="Hello";
struct sockaddr_in server_addr;

server_addr.sin_family = AF_INET;
server_addr.sin_port = htons(PORT);
server_addr.sin_addr.s_addr = htonl(INADDR_ANY);

int listener = socket(AF_INET,SOCK_STREAM,0);

if(listener < 0) {
    perror("Can't create socket.");
    exit(1);
}

if(bind(listener,(struct sockaddr *) &server_addr,sizeof(server_addr)) < 0) {
    perror("Can't bind socket.");
    exit(1);
}

if(listen(listener,5)) {
    perror("Erro while listening:");
    exit(1);
}

int client = accept(listener,NULL,NULL);
```

```
while(1)
{
    bzero( str, 100);
    recv(client,str, 100, 0);
    printf("Message from client - %s",str);
    send(client, msg, (int)strlen(msg), 0);
}

return 0;
}
```

Сервер слушает порт 8133 и на входящее сообщение отправляет "Hello".

Для определения данного сервиса, добавим в файл nmap-service-probes следующие строки:

```
Probe TCP testServer q|Hi|
rarity 1
ports 8133
match testServer m|Hello|
```

В данных строках мы описываем, что серсиву с именем testServer мы посылаем сообщение Hi на порт 8133 и ожидаем в ответ сообщение Hello.

Запустим на испытуемой виртуальной машине данный сервер и попробуем определить его при помощи утилиты nmap:

```
root@kali:~# nmap 192.168.226.134

Starting Nmap 7.01 ( https://nmap.org ) at 2016-06-14 12:06 EDT
Nmap scan report for 192.168.226.134
Host is up (0.00014s latency).
Not shown: 977 closed ports
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
```

```
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8133/tcp open  testServer
8180/tcp open  unknown
MAC Address: 00:0C:29:B8:13:C4 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

Видно, что утилита корректно определила наш сервер.

3.6 Сохранить вывод утилиты в формате xml

Для сохранения вывода nmap в файле в формате xml необходимо указать флаг -oX и имя файла.

```
root@kali:~# nmap 192.168.226.134 -oX xml_output.xml
```

Содержимое файла **xml_output.xml** приведено ниже.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/bin/../share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 7.01 scan initiated Tue Jun 14 12:27:40 2016 as: nmap -oX xml_output.xml 192.168
<nmaprun scanner="nmap" args="nmap -oX xml_output.xml 192.168.226.134" start="1465921660"
<scaninfo type="syn" protocol="tcp" numservices="1000" services="1,3-4,6-7,9,13,17,19-26,3
<verbose level="0"/>
<debugging level="0"/>
<host starttime="1465921660" endtime="1465921660"><status state="up" reason="arp-response"
<address addr="192.168.226.134" addrtype="ipv4"/>
<address addr="00:0C:29:B8:13:C4" addrtype="mac" vendor="VMware"/>
<hostnames>
</hostnames>
<ports><extraports state="closed" count="978">
<extrareasons reason="resets" count="978"/>
</extraports>
<port protocol="tcp" portid="21"><state state="open" reason="syn-ack" reason_ttl="64"/><se
<port protocol="tcp" portid="22"><state state="open" reason="syn-ack" reason_ttl="64"/><se
<port protocol="tcp" portid="23"><state state="open" reason="syn-ack" reason_ttl="64"/><se
<port protocol="tcp" portid="25"><state state="open" reason="syn-ack" reason_ttl="64"/><se
<port protocol="tcp" portid="53"><state state="open" reason="syn-ack" reason_ttl="64"/><se
<port protocol="tcp" portid="80"><state state="open" reason="syn-ack" reason_ttl="64"/><se
<port protocol="tcp" portid="111"><state state="open" reason="syn-ack" reason_ttl="64"/><s
<port protocol="tcp" portid="139"><state state="open" reason="syn-ack" reason_ttl="64"/><s
<port protocol="tcp" portid="445"><state state="open" reason="syn-ack" reason_ttl="64"/><s
<port protocol="tcp" portid="512"><state state="open" reason="syn-ack" reason_ttl="64"/><s
<port protocol="tcp" portid="513"><state state="open" reason="syn-ack" reason_ttl="64"/><s
<port protocol="tcp" portid="514"><state state="open" reason="syn-ack" reason_ttl="64"/><s
<port protocol="tcp" portid="1099"><state state="open" reason="syn-ack" reason_ttl="64"/><
<port protocol="tcp" portid="1524"><state state="open" reason="syn-ack" reason_ttl="64"/><
```

```
<port protocol="tcp" portid="2049"><state state="open" reason="syn-ack" reason_ttl="64"/><
<port protocol="tcp" portid="2121"><state state="open" reason="syn-ack" reason_ttl="64"/><
<port protocol="tcp" portid="3306"><state state="open" reason="syn-ack" reason_ttl="64"/><
<port protocol="tcp" portid="5432"><state state="open" reason="syn-ack" reason_ttl="64"/><
<port protocol="tcp" portid="5900"><state state="open" reason="syn-ack" reason_ttl="64"/><
<port protocol="tcp" portid="6000"><state state="open" reason="syn-ack" reason_ttl="64"/><
<port protocol="tcp" portid="6667"><state state="open" reason="syn-ack" reason_ttl="64"/><
<port protocol="tcp" portid="8180"><state state="open" reason="syn-ack" reason_ttl="64"/><
</ports>
<times srtt="140" rttvar="21" to="100000"/>
</host>
<runstats><finished time="1465921660" timestr="Tue Jun 14 12:27:40 2016" elapsed="0.24" su
</runstats>
</nmaprun>
```

Также поддерживаются некоторые другие форматы, для которых определены соответсвующие флаги: -oS, -oG, -oN , -oA.

3.7 Исследвать различные этапы и режимы работы Nmap с использованием утилиты Wireshark

Wireshark - программа-анализатор трафика для компьютерных сетей Ethernet. Если, после запуска Wireshark, запусть сканирование nmap, то мы сможем проанализировать пакеты, посылаемые nmap.

При сканировании сети на наличие активных хостов утилита nmap обращается к DNS серверу для получения доступных узлов в данной подсети:

```
626 5387.413380494 192.168.226.135 192.168.226.2 DNS 88 Standard query 0x3a05 PTR 1.226.16
627 5387.413527354 192.168.226.135 192.168.226.2 DNS 88 Standard query 0x3a06 PTR 2.226.16
628 5387.413590884 192.168.226.135 192.168.226.2 DNS 90 Standard query 0x3a07 PTR 134.226.
629 5387.413648402 192.168.226.135 192.168.226.2 DNS 90 Standard query 0x3a08 PTR 254.226.
630 5387.420341514 192.168.226.2 192.168.226.135 DNS 88 Standard query response 0x3a06 No
631 5387.420373160 192.168.226.2 192.168.226.135 DNS 90 Standard query response 0x3a08 No
```

При анализе открытых портов, утилита nmap пытается установить соединение с доступными портами. Порт считается открытым, если на запрос установления соединения приходит пакет с флагами [SYN, ACK]. После этого nmap обращается к файлу nmap-services. Пример сканирования открытого порта 80 приведен ниже.

Запуск nmap.

```
root@kali:~# nmap 192.168.226.134 -p 80

Starting Nmap 7.01 ( https://nmap.org ) at 2016-06-14 14:27 EDT
Nmap scan report for 192.168.226.134
Host is up (0.00025s latency).
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 00:0C:29:B8:13:C4 (VMware)
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
```

Вывод wireshark.

```
499 192.168.226.135 192.168.226.2 DNS 88 Standard query 0x52ac PTR 134.226.168.192.in-addr
500 192.168.226.2 192.168.226.135 DNS 88 Standard query response 0x52ac No such name PTR 1
501 192.168.226.135 192.168.226.134 TCP 58 43956 -> 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
502 192.168.226.134 192.168.226.135 TCP 60 80 -> 43956 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len
503 192.168.226.135 192.168.226.134 TCP 54 43956 -> 80 [RST] Seq=1 Win=0 Len=0
```

При сканировании порта, nmap посылает пакет с флагом запроса на установление соединения(SYN). Если в ответ приходит пакет с флагами [SYN, ACK], это означает, что порт открыт, и утилита nmap разрывает соединение, посылая пакет с флагом сброса соединения(RST).

Результат сканирования закрытого порта 112 приведен ниже:

Запуск nmap.

```
root@kali:~# nmap 192.168.226.134 -p 112

Starting Nmap 7.01 ( https://nmap.org ) at 2016-06-14 14:34 EDT
Nmap scan report for 192.168.226.134
Host is up (0.00029s latency).
PORT    STATE  SERVICE
112/tcp closed mcidas
MAC Address: 00:0C:29:B8:13:C4 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
```

Вывод утилиты wireshark:

```
549 192.168.226.135 192.168.226.2 DNS 88 Standard query 0x0b13 PTR 134.226.168.192.in-addr
550 192.168.226.2 192.168.226.135 DNS 88 Standard query response 0x0b13 No such name PTR 1
551 192.168.226.135 192.168.226.134 TCP 58 59528 -> 112 [SYN] Seq=0 Win=1024 Len=0 MSS=146
552 192.168.226.134 192.168.226.135 TCP 60 112 -> 59528 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

Видно, что nmap так же посылает пакет с флагом SYN и пытается установить соединение по протоколу TCP. Так как порт закрыт, в ответ приходит пакет с флагами [RST, ACK]. После этого nmap считает, что порт закрыт, о чем свидетельсвует статус closed. Также nmap находит и выводит информацию об этом порте в файле nmap-services.

3.8 Просканировать виртуальную машину Metasploitable2 используя утилиту db_nmap

Для того, чтобы использовать db_nmap, сначала необходимо выполнить три шага:

1. запустить posgresql сервер
2. инициализировать базу данных командой msfdb init

3. запустить консоль msfconsole

```
root@kali:~# service postgresql start
root@kali:~# msfdb init
Creating database user 'msf'
Enter password for new role:
Enter it again:
Creating databases 'msf' and 'msf_test'
Creating configuration file in /usr/share/metasploit-framework/config/database.yml
Creating initial database schema
root@kali:~# msfconsole



        .                                                .
 .

      dBBBBBBb  dBBBP dBBBBBBP dBBBBBb  .                       o
        '   dB'                       BBP
    dB'dB'dB' dBBP     dBP     dBP BB
   dB'dB'dB' dBP      dBP     dBP  BB
  dB'dB'dB' dBBBBP   dBP     dBBBBBBB

                                  dBBBBBP  dBBBBBb  dBP    dBBBBP dBP dBBBBBBP
          .                  .                   dB' dBP    dB'.BP
                               |       dBP    dBBBB' dBP    dB'.BP dBP    dBP
                            --o--    dBP    dBP    dBP    dB'.BP dBP    dBP
                               |     dBBBBP dBP    dBBBBP dBBBBP dBP    dBP


                                                                      .
                .
        o                  To boldly shell were no
                            shell has gone before


Frustrated with proxy pivoting? Upgrade to layer-2 VPN pivoting with
Metasploit Pro -- learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.8-                           ]
+ -- --=[ 1519 exploits - 880 auxiliary - 259 post       ]
+ -- --=[ 437 payloads - 38 encoders - 8 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

После этого вместо nmap можно использовать db_nmap, которая имеет аналагичную функциональность за тем исключением, что все результаты сохраняются в базу данных, что облегчает их дальнейшую обработку.

```
msf > db_nmap 192.168.226.134
[*] Nmap: Starting Nmap 7.01 ( https://nmap.org ) at 2016-06-14 15:13 EDT
```

```
[*] Nmap: Nmap scan report for 192.168.226.134
[*] Nmap: Host is up (0.00014s latency).
[*] Nmap: Not shown: 978 closed ports
[*] Nmap: PORT     STATE SERVICE
[*] Nmap: 21/tcp   open  ftp
[*] Nmap: 22/tcp   open  ssh
[*] Nmap: 23/tcp   open  telnet
[*] Nmap: 25/tcp   open  smtp
[*] Nmap: 53/tcp   open  domain
[*] Nmap: 80/tcp   open  http
[*] Nmap: 111/tcp  open  rpcbind
[*] Nmap: 139/tcp  open  netbios-ssn
[*] Nmap: 445/tcp  open  microsoft-ds
[*] Nmap: 512/tcp  open  exec
[*] Nmap: 513/tcp  open  login
[*] Nmap: 514/tcp  open  shell
[*] Nmap: 1099/tcp open  rmiregistry
[*] Nmap: 1524/tcp open  ingreslock
[*] Nmap: 2049/tcp open  nfs
[*] Nmap: 2121/tcp open  ccproxy-ftp
[*] Nmap: 3306/tcp open  mysql
[*] Nmap: 5432/tcp open  postgresql
[*] Nmap: 5900/tcp open  vnc
[*] Nmap: 6000/tcp open  X11
[*] Nmap: 6667/tcp open  irc
[*] Nmap: 8180/tcp open  unknown
[*] Nmap: MAC Address: 00:0C:29:B8:13:C4 (VMware)
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

3.9 Выбрать пять записей из файла nmap-service-probes и описать их работу

Для рассмотрения выберем следующие строки.

```
Probe TCP GetRequest q|GET / HTTP/1.0\r\n\r\n|
rarity 1
ports 1,70,79,80-85,88,113,139,143,280,497,505,514,515,540,554,591,620,631,783,888,898,900
sslports 443,993,995,1311,1443,3443,4443,5061,7443,8443,9443,10443,14443,44443,60443

match ajp13 m|^AB\0\x13\x04\x01\x90\0\x0bBad Request\0\0\0AB\0\x02\x05\x01$| p/Apache Jser
```

Директива Probe задает какому сервису, по какому протоколу, какое сообщение нужно отправить. В данном случае протокол - TCP, сервис - GetRequest, а отправляемое сообщение:

```
GET / HTTP/1.0\r\n\r\n
```

Строка с директивой rarity указывает частоту, с которой от сервиса можно ожидать ответ. Чем больше число - тем реже. Максимальное значение - 9. Минимальное значение - 1. В рассматриваемом примере выбрано минимальное значение.

Директивы ports и sslports указывают на порты, которые использует данный сервис.

Директива match используется для сопоставления ответа сервиса на отправленный ему ранее запрос. Формат - m/[regex]/[opts] Где m говорит nmap о начале сопоставляемой строки, regex - регулярное выражение для сопоставления, а opts - опции. Из опций сейчас поддерживается только две: i - делает сопоставление нечувствительным к регистру и s - который позваляет парсить переход на новую строку.

```
match ajp13 m|^AB\0\x13\x04\x01\x90\0\x0bBad Request\0\0\0AB\0\x02\x05\x01$| p/Apache Jser
```

В строке выше параметры не используются, зато есть сексия с опциональными параметрами. В данном случае это p - указывает авторство и часто название продукта. В данном случае Apache Jserv.

Рассмотрим еще одну строку, например:

```
fallback GetRequest
```

Дерриктива fallback говорит о том, какие Probe нужно использовать,если с текущей секцией Probe совпадений не будет. В строке выше в этом случае рекомендуется перейти к Probe GetRequest.

3.10 Выбрать один скрипт из состава Nmap и описать его работу

Nmap имеет возможность запусткать скрипты. Скрипты пишутся на языке Lua. Помимо того, что скрипт можно написать самомстоятельно, имеются уже готовые скрипты, информацию о которых можно почитать на сайте nmap.org.
Был выбран скрипт **mysql-users**. Данный скрипт обращается к серверу MySQL и возвращает список пользователей.

Пример использования:

```
nmap -sV --script=mysql-users <target>
```

Пример результата работы:

```
3306/tcp open  mysql
| mysql-users:
|   test
|   root
|   test2
|   album
|   debian-sys-maint
|   horde
|   mediatomb
|_  squeezecenter
```

Листинг кода скрипта:

```
local mysql = require "mysql"
local nmap = require "nmap"
local shortport = require "shortport"
local stdnse = require "stdnse"

local openssl = stdnse.silent_require "openssl"

description = [[
Attempts to list all users on a MySQL server.
]]

---
-- @args mysqluser The username to use for authentication. If unset it
-- attempts to use credentials found by <code>mysql-brute</code> or
-- <code>mysql-empty-password</code>.
-- @args mysqlpass The password to use for authentication. If unset it
-- attempts to use credentials found by <code>mysql-brute</code> or
-- <code>mysql-empty-password</code>.
--
-- @output
-- 3306/tcp open  mysql
-- | mysql-users:
-- |   test
-- |   root
-- |   test2
-- |   album
-- |   debian-sys-maint
-- |   horde
-- |   mediatomb
-- |_  squeezecenter


author = "Patrik Karlsson"
license = "Same as Nmap--See https://nmap.org/book/man-legal.html"
categories = {"auth", "intrusive"}


dependencies = {"mysql-brute", "mysql-empty-password"}

-- Version 0.1
-- Created 01/23/2010 - v0.1 - created by Patrik Karlsson <patrik@cqure.net>

portrule = shortport.port_or_service(3306, "mysql")

action = function( host, port )

  local socket = nmap.new_socket()
  local catch = function() socket:close() end
  local try = nmap.new_try(catch)
  local result, response = {}, nil
```

```
  local users = {}
  local nmap_args = nmap.registry.args
  local status, rows

  -- set a reasonable timeout value
  socket:set_timeout(5000)

  -- first, let's see if the script has any credentials as arguments?
  if nmap_args.mysqluser then
    users[nmap_args.mysqluser] = nmap_args.mysqlpass or ""
  -- next, let's see if mysql-brute or mysql-empty-password brought us anything
  elseif nmap.registry.mysqlusers then
    -- do we have root credentials?
    if nmap.registry.mysqlusers['root'] then
      users['root'] = nmap.registry.mysqlusers['root']
    else
      -- we didn't have root, so let's make sure we loop over them all
      users = nmap.registry.mysqlusers
    end
  -- last, no dice, we don't have any credentials at all
  else
    stdnse.debug1("No credentials supplied, aborting ...")
    return
  end

  --
  -- Iterates over credentials, breaks once it successfully receives results
  --
  for username, password in pairs(users) do

    try( socket:connect(host, port) )

    response = try( mysql.receiveGreeting( socket ) )
    status, response = mysql.loginRequest( socket, { authversion = "post41", charset = res

    if status and response.errorcode == 0 then
      status, rows = mysql.sqlQuery( socket, "SELECT DISTINCT user FROM mysql.user" )
      if status then
        result = mysql.formatResultset(rows, { noheaders = true })
      end
    end
    socket:close()
  end

  return stdnse.format_output(true, result)

end
```

Скрипт проверяет были ли ему переданы логин и пароль в качестве параметров. Если нет, то выполняется попытка зайти от имени root поль-

зователя с пустым паролем. Затем происходит првоерка прав у текущего пользователя, если с правами все в порядке, то формируется запрос на выборку пользователей и затем возвращается отформатированный ответ.

4 Выводы

В ходе лабораторной работы были изучены основные возможности утилиты nmap: сканирование открытых портов и доступных хостов, определение версий сервисов, сохранение вывода в формате xml. Так же были рассмотрены основные служебные файлы, которые используются для работы утилиты nmap, рассмотрены скрипты. Так же было рассмотрено расширение db_map, которое позволяет сохранять результаты сканирования в базу данных. Файл nmap-service-probe был изменен, путем добавления записей о собственном сервисе. Исследована работа утилиты nmap с помощью Wireshark.