

Отчет по лабораторной работе №4 :
Утилита для исследования сети и сканер
портов nmap

Бусаров Владислав

2015

Содержание

1 Цель работы

Определить набор и версии сервисов запущенных на компьютере в диапазоне адресов. Данная работа выполняется на ОС Kali linux, используется утилита nmap.

2 Ход работы

2.1 Провести поиск активных хостов

Настройки сети: в нашей сети имеется всего 3 хоста.

- Windows 10 (192.168.0.104), основная ОС
- kali linux (192.168.0.102)
- metasploitable2 (192.168.0.105)

Выведем список хостов в подсети 192.168.0.0/24

Для этого воспользуемся командой `nmap -sn 192.168.0.0/24`. (См. рисунок 1)

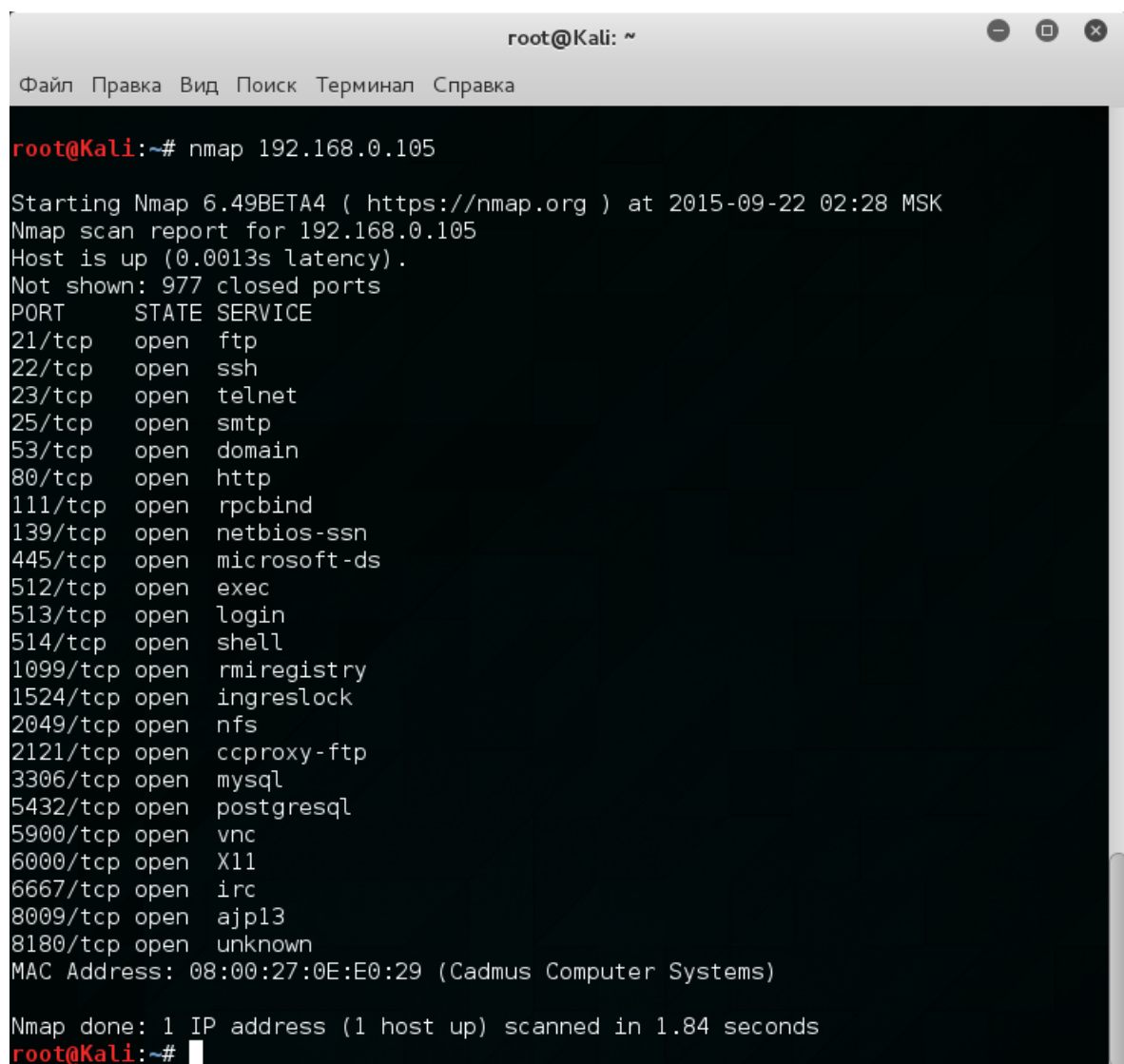
```
root@Kali: ~  
Файл Правка Вид Поиск Терминал Справка  
root@Kali:~# nmap -sn 192.168.0.0/24  
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-22 02:26 MSK  
Nmap scan report for 192.168.0.1  
Host is up (0.0010s latency).  
MAC Address: C4:6E:1F:FE:27:6A (Tp-link Technologies Co.)  
Nmap scan report for 192.168.0.104  
Host is up (0.00026s latency).  
MAC Address: 84:4B:F5:39:D6:C6 (Hon Hai Precision Ind. Co.)  
Nmap scan report for 192.168.0.105  
Host is up (-0.10s latency).  
MAC Address: 08:00:27:0E:E0:29 (Cadmus Computer Systems)  
Nmap scan report for 192.168.0.106  
Host is up (0.025s latency).  
MAC Address: 74:E5:0B:CE:05:10 (Intel Corporate)  
Nmap scan report for 192.168.0.102  
Host is up.  
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.11 seconds  
root@Kali:~#
```

Рис. 1: Поиск хостов

2.2 Определить открытые порты

Просканируем порты metasploitable2.

Для определения открытых портов достаточно просто ввести `nmap 192.168.0.105` (сканируются порты до 1024). Или же воспользоваться опцией `-p`, например `nmap -p "*" 192.168.0.105`. Данной командой просканируются все порты, если необходимо задать диапазон достаточно указать его вместо `"*"`. Результат на рисунке 2.



```
root@Kali: ~
Файл Правка Вид Поиск Терминал Справка

root@Kali:~# nmap 192.168.0.105

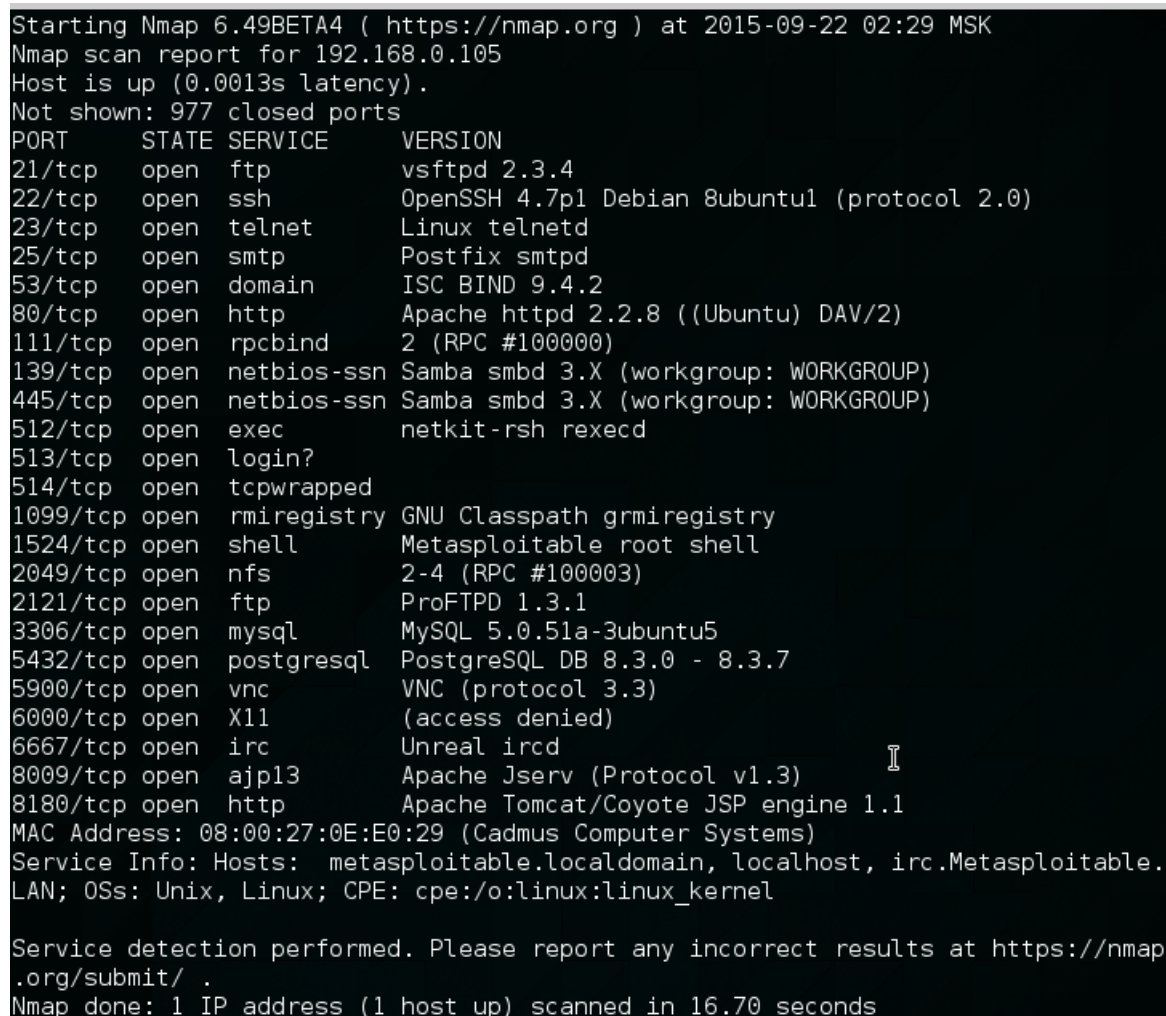
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-22 02:28 MSK
Nmap scan report for 192.168.0.105
Host is up (0.0013s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    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
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:0E:E0:29 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 1.84 seconds
root@Kali:~#
```

Рис. 2: Поиск портов

2.3 Определить версии сервисов

Чтобы определить версии сервисов необходимо воспользоваться командой `nmap` с ключем `sV` следующим образом: `nmap -sV 192.168.0.105`. Результат на рисунке 3.



```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-22 02:29 MSK
Nmap scan report for 192.168.0.105
Host is up (0.0013s latency).
Not shown: 977 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
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:0E:E0:29 (Cadmus Computer Systems)
Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploitable.
LAN; OSs: Unix, 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 16.70 seconds
```

Рис. 3: Определение версий сервисов

2.4 Изучить файлы `nmap-services`, `nmap-os-db`, `nmapservice-probes`

Рассмотрим файл `nmap-services`. Для этого введем команду `vim /usr/share/nmap/nmap-services`.

Файл служит для быстрого поиска, например с ключем `-F`. В файле в каждой строке задаются сервисное название или сокращение, число

порта и протокол, определенный разделом, частота порта мера того, как часто порт был найден открытым во время сканирования. Пример файла можно увидеть на рисунке 4.

```
# THIS FILE IS GENERATED AUTOMATICALLY FROM A MASTER - DO NOT EDIT.
# EDIT /nmap-private-dev/nmap-services-all IN SVN INSTEAD.
# Well known service port numbers -*- mode: fundamental; -*-
# From the Nmap Security Scanner ( http://nmap.org )
#
# $Id: nmap-services 34097 2015-03-21 01:08:41Z fyodor $
#
# Derived from IANA data and our own research
#
# This collection of service data is (C) 1996-2011 by Insecure.Com
# LLC. It is distributed under the Nmap Open Source license as
# provided in the COPYING file of the source distribution or at
# http://nmap.org/data/COPYING . Note that this license
# requires you to license your own work under a compatable open source
# license. If you wish to embed Nmap technology into proprietary
# software, we sell alternative licenses (contact sales@insecure.com).
# Dozens of software vendors already license Nmap technology such as
# host discovery, port scanning, OS detection, and version detection.
# For more details, see http://nmap.org/book/man-legal.html
#
# Fields in this file are: Service name, portnum/protocol, open-frequency, optional comments
#
tcpmux 1/tcp 0.001995 # TCP Port Service Multiplexer [rfc-1078]
tcpmux 1/udp 0.001236 # TCP Port Service Multiplexer
compressnet 2/tcp 0.000013 # Management Utility
compressnet 2/udp 0.001845 # Management Utility
compressnet 3/tcp 0.001242 # Compression Process
compressnet 3/udp 0.001532 # Compression Process
unknown 4/tcp 0.000477
rje 5/udp 0.000593 # Remote Job Entry
unknown 6/tcp 0.000502
echo 7/sctp 0.000000
echo 7/tcp 0.004855
```

1.1 Наверх

Рис. 4: Файл nmap-services

Файл nmap-os-db содержит сотни примеров реакций ОС на nmap. Таким образом nmap определяет какая операционная система установлена на удаленной машине. Для того чтобы узнать какая ОС установлена нужно запустить nmap с ключем -O. Содержимое файла представлено на рисунке 5.

nmap-service-probes — это простой текстовый файл состоящий из строк, в котором хранятся тесты и сигнатуры подсистем определений версий. Строки, начинающиеся с символа "решетки"(#) воспринимаются как комментарии и игнорируются обработчиком. Пустые строки также не обрабатываются.

Синтаксис:

```

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)
T1(R=100%DF=20%T=15%TG=15%S=20%A=20%F=30%RD=20%Q=20)
T2(R=80%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T3(R=80%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T4(R=100%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T5(R=100%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T6(R=100%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T7(R=80%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
U1(R=50%DF=20%T=15%TG=15%IPL=100%UN=100%RIPL=100%RID=100%RIPCK=100%RUCK=100%RUD=100)
IE(R=50%DFI=40%T=15%TG=15%CD=100)

# 4-port GSM-SIP gateway PORTech MV-374
# 2FXS VoIP gateway K-3288W
Fingerprint 2FXS K-3288W or PORTech MV-374 GSM-SIP VoIP adapter
Class 2FXS | embedded || VoIP adapter
Class PORTech | embedded || VoIP adapter
CPE cpe:/h:portech:mv-374
SEQ(SP=0-5%GCD=61A8|C350|124F8|186A0|1E848%ISR=8A-94%TI=I%II=RI%SS=0%TS=U)
OPS(O1=M5B4|WANM5B4T10S%O2=M578|M578W0ST10L%O3=M280|T10NNW5NM280%O4=M5B4|ST10WAL%O5=M218|M218ST10WAL%O6=M109|M109ST10)
WIN(W1=0|3180%W2=0|3180%W3=0|3180%W4=0|3180%W5=0|3180%W6=0|3180)
ECN(R=Y%DF=N%T=3B-45%TG=40%W=0%O=WANM5B4SNN%CC=N%Q=R)
T1(R=Y%DF=N%T=3B-45%TG=40%W=0%O=WANM5B4SNN%CC=N%Q=R)
T2(R=Y%DF=N%T=3B-45%TG=40%W=0%O=WANM109T10S%RD=0%Q=)
T3(R=Y%DF=N%T=3B-45%TG=40%W=0%O=WANM109T10S%RD=0%Q=)
T4(R=Y%DF=N%T=3B-45%TG=40%W=0%O=WANM109T10S%RD=0%Q=)
T5(R=Y%DF=N%T=3B-45%TG=40%W=0%O=WANM109T10S%RD=0%Q=)
T6(R=Y%DF=N%T=3B-45%TG=40%W=0%O=WANM109T10S%RD=0%Q=)
T7(R=Y%DF=N%T=3B-45%TG=40%W=0%O=WANM109T10S%RD=0%Q=)

```

Рис. 5: Файл nmap-os-db

- Probe <protocol> <probenamе> <probesendstring> - директива probe (тест) - указывает nmap, какие данные отправлять в процессе определения служб
- match <service> <pattern> <productname> <version> <device> <h?????> <info> <OS> - указывает nmap на то, как точно определить службу, используя полученный ответ на запрос, отправленный предыдущей директивой probe. Эта директива используется в случае, когда полученный ответ полностью совпадает с шаблоном. При этом тестирование порта считается законченным, а при помощи дополнительных спецификаторов nmap строит отчет о названии приложения, номере версии и дополнительной информации, полученной в ходе проверки
- softmatch <service> <pattern> <productname> <version> <device> <h?????> <info> <OS> - имеет аналогичный формат директиве

match. Основное отличие заключается в том, что после совпадения принятого ответа с одним из шаблонов softmatch, тестирование будет продолжено с использованием только тех тестов, которые относятся к определенной шаблону службе. Тестирование порта будет идти до тех пор, пока не будет найдено строгое соответствие (match) или не закончатся все тесты для данной службы

- ports <portlist> - группирует порты, которые обычно закрепляются за идентифицируемой данным тестом службой
- sslports <sslportlist> - аналогична директиве ports, только эта директива указывает порты, обычно используемые совместно с SSL
- totalwaitms <milliseconds> - редко используемая, т.к. указывает сколько времени (в миллисекундах) необходимо ждать ответ, прежде чем прекратить тест службы

2.5 Добавить новую сигнатуру службы в файл nmap-service-probes (для этого создать минимальный tcp server, добиться, чтобы при сканировании nmap указывал для него название и версию)

Напишем простой tcp-сервер, который просто ждет подключения клиента и отправляет ему сообщение. В файл nmap-service-probes добавим следующую строку:

```
match tcp-server m|^111| v/1.0.X/ p/Busarov V.A./ i/LoL It's works heh/
```

Код сервера:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Net;
using System.Net.Sockets;
using System.Threading;

namespace ExampleTcpListener_Console
{
    class ExampleTcpListener
    {
        static void Main(string[] args)
        {
            TcpListener server = null;
            try
```

```

{
    int MaxThreadsCount = Environment.ProcessorCount * 4;
    Console.WriteLine(MaxThreadsCount.ToString());
    ThreadPool.SetMaxThreads(MaxThreadsCount, MaxThreadsCount);
    ThreadPool.SetMinThreads(2, 2);

    Int32 port = 9596;
    IPAddress localAddr = IPAddress.Parse("192.168.0.104");
    int counter = 0;
    server = new TcpListener(localAddr, port);

    server.Start();

    while (true)
    {

        Console.WriteLine("\nWaiting for a connection... ");

        ThreadPool.QueueUserWorkItem(ObrabotkaZaprosa, server.AcceptTcpCL
        counter++;
        Console.WriteLine("\nConnection №" + counter.ToString() + "!");

    }
}
catch (SocketException e)
{
    Console.WriteLine("SocketException: {0}", e);
}
finally
{
    server.Stop();
}

Console.WriteLine("\nHit enter to continue...");
Console.Read();
}

static void ObrabotkaZaprosa(object client_obj)
{
    Byte[] bytes = new Byte[256];
    String data = null;

    TcpClient client = client_obj as TcpClient;

```



```

        data = null;

        NetworkStream stream = client.GetStream();

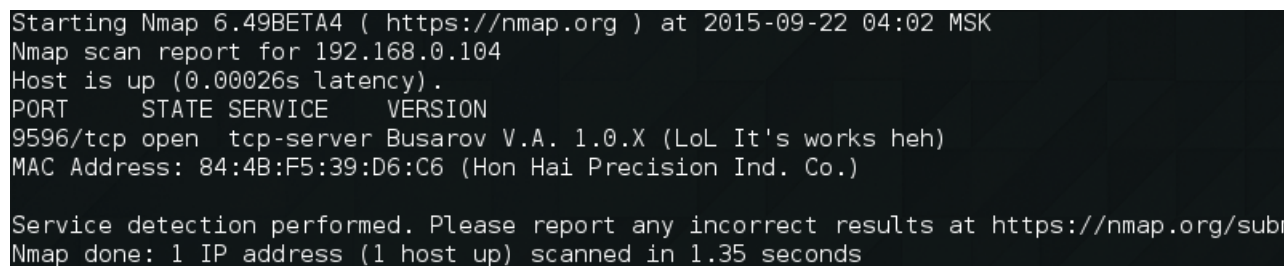
        int i;

        data = "111";
        byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);
        stream.Write(msg, 0, msg.Length);

        client.Close();
    }
}

```

Таким образом теперь nmap знает, что если при пустом запросе с сервера прихоят строка 111, значит нужно выводить информацию которая указана на рисунке 6.



```

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-22 04:02 MSK
Nmap scan report for 192.168.0.104
Host is up (0.00026s latency).
PORT      STATE SERVICE      VERSION
9596/tcp  open  tcp-server  Busarov V.A. 1.0.X (LoL It's works heh)
MAC Address: 84:4B:F5:39:D6:C6 (Hon Hai Precision Ind. Co.)

Service detection performed. Please report any incorrect results at https://nmap.org/sub
Nmap done: 1 IP address (1 host up) scanned in 1.35 seconds

```

Рис. 6: Вывод информации о сервисе

2.6 Сохранить выводы утилиты в формате xml

Для того, чтобы вывести данные в xml файл достаточно вызвать команду nmap с ключем -oX и указать имя файла. Например:

```
nmap -sn -oX output.xml 192.168.0.104
```

Результат можно увидеть на рисунке 7:

2.7 Исследовать различные этапы и режимы работы nmap с использованием утилиты Wireshark

Wireshark — это достаточно известный инструмент для захвата и анализа сетевого трафика. Wireshark работает с подавляющим большинством известных протоколов, имеет понятный и логичный графический интерфейс на основе GTK+ и мощнейшую систему фильтров. Кроссплатформенный, работает в таких ОС как Linux, Solaris, FreeBSD, NetBSD,

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/bin/./share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 6.49BETA4 scan initiated Tue Sep 22 04:11:49 2015 as: nmap -sn -oX output.xml 1
<nmaprun scanner="nmap" args="nmap -sn -oX output.xml 192.168.0.104" start="1442884309" s
11:49 2015" version="6.49BETA4" xmloutputversion="1.04">
<verbose level="0"/>
<debugging level="0"/>
<host><status state="up" reason="arp-response" reason_ttl="0"/>
<address addr="192.168.0.104" addrtype="ipv4"/>
<address addr="84:4B:F5:39:D6:C6" addrtype="mac" vendor="Hon Hai Precision Ind. Co."/>
<hostnames>
</hostnames>
<times srtt="980" rttvar="5000" to="100000"/>
</host>
<runstats><finished time="1442884309" timestr="Tue Sep 22 04:11:49 2015" elapsed="0.27" s
e Sep 22 04:11:49 2015; 1 IP address (1 host up) scanned in 0.27 seconds" exit="success"/
total="1"/>
</runstats>
</nmaprun>

```

Рис. 7: output.xml

OpenBSD, Mac OS X, и, естественно, Windows. Распространяется под лицензией GNU GPL v2. Доступен бесплатно на сайте wireshark.org.

Далее продемонстрируем простую работу с wireshark. Будем сканировать порты используя ключ -xS (xmas tree). В данном режиме отправляются tcp пакеты с установленными флагами FIN, URG, PASS. В случае если порт закрыт придет пакет с флагом RST (прерывание соединения), иначе нисего не должно прийти. Приступим. При запуске wireshark предложит выбрать интерфейс и начать сканировать его трафик(см. рисунок 8). Выберем интерфейс eth0

Wireshark по-умолчанию выводит все пакеты, которые проходят через интерфейс eth0. Все пакеты просматривать неудобно, поэтому мы будем пользоваться фильтрами. Поставим фильтр по протоколу tcp.

Введем команду сканирования порта 21 на ОС metasploitable2:

```
nmap 192.168.0.105 -p21 -sX
```

Результат на рисунке 9. Увидим два пакета, ответа на которые не пришло, значит порт открыт.

Введем команду сканирования порта 43 на ОС metasploitable2:

```
nmap 192.168.0.105 -p43 -sX
```

Результат на рисунке 10. Увидим два пакета, первый - наш запрос, второй - ответ с флагом RST. Значит порт закрыт.

Так же было проведено сканирование по всем портам, результаты совпадают с SYN сканированием и connect(). В отличие от сканирования основной ОС (Windows 10). В которой на закрытые порты не отправлялся ответ с флагом RST. Результаты на рисунке 11. Windows не

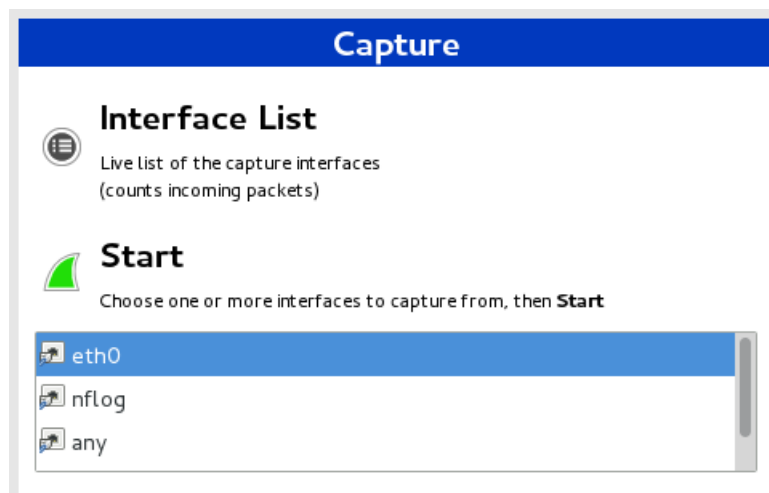


Рис. 8: wireshark выбор интерфейса и старт

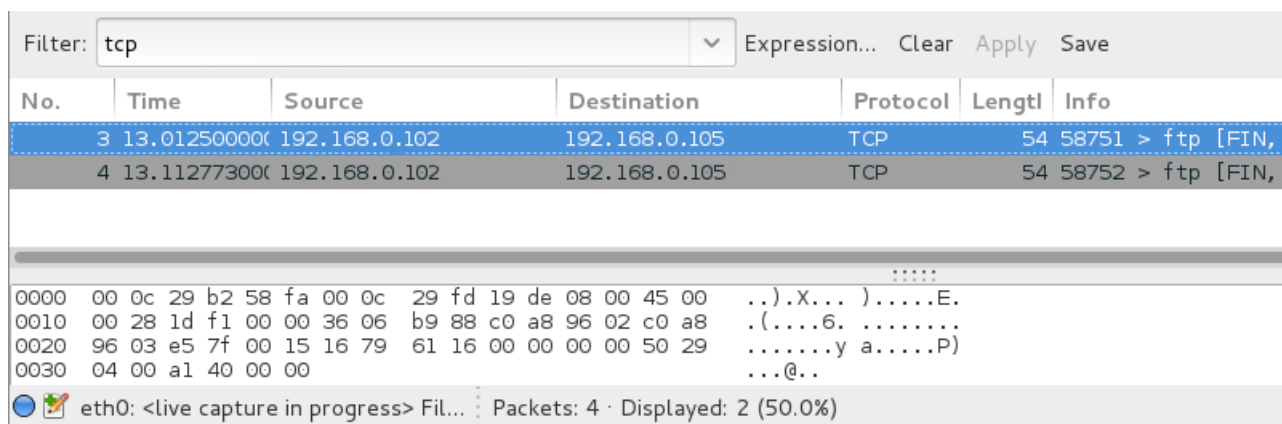


Рис. 9: xmas scan порт 21

придерживаются стандарта RFC 793. Так же не по стандарту работают системы Cisco, BSDI, HP/UX, MVS, IRIX. Они передают пакет RST при сканировании открытых портов.

2.8 Просканировать виртуальную машину Metasploitable2 используя nmap_db из состава metasploit-framework

Для работы с metasploit потребуется СУБД postgresql. Для того, чтобы ее запустить воспользуемся командой

```
service postgresql start
```

Затем запустим msf консоль командой `msfconsole`.

В данной консоли осуществляется работа с metasploit. Проверим соединение с БД:

Filter: tcp

Expression...

Clear

Apply

Save

No.	Time	Source	Destination	Protocol	Length	Info
3	13.02154600	192.168.0.102	192.168.0.105	TCP	54	38089 > nickname [R
4	13.02216400	192.168.0.105	192.168.0.102	TCP	60	nickname > 38089 [R

0000

00 0c 29 b2 58 fa 00 0c 29 fd 19 de 08 00 45 00

..).X...).....E.

0010

00 28 44 5e 00 00 3a 06 8f 1b c0 a8 96 02 c0 a8

.(D^....

0020

96 03 94 c9 00 2b 40 96 19 ab 00 00 00 00 50 29

.....+@.P)

0030

04 00 0f 2f 00 00

.../..

eth0: <live capture in progress> Fil...

Packets: 6 · Displayed: 2 (33.3%)

Рис. 10: xmas scan порт 43

Filter: tcp

▼

Expression...

Clear

Apply

Save

No.	Time	Source	Destination	Protocol	Length	Info
5	13.012707000	192.168.0.102	192.168.0.104	TCP	54	37463 > telnet [FI
6	13.113893000	192.168.0.102	192.168.0.104	TCP	54	37464 > telnet [FI

0000

00 50 56 c0 00 03 00 0c 29 fd 19 de 08 00 45 00

.PV.....).....E.

0010

00 28 51 fa 00 00 2b 06 90 81 c0 a8 96 02 c0 a8

.(Q...+.

0020

96 01 92 57 00 17 b1 80 4e 4f 00 00 00 00 50 29

...W.... NO....P)

0030

04 00 6c 28 00 00

..l(..

eth0: <live capture in progress> Fil...

Packets: 7 · Displayed: 2 (28.6%)

Рис. 11: xmas scan порт 23 (Windows 10)

db_status

Результат на рисунке 13.

```
msf > db_status
[*] postgresql connected to msf
msf >
```

Рис. 12: db_status

Таблицы БД: hosts, services, vulns, loot и notes. В каждой храниться соответствующая информация. Для заполнения этих таблиц автоматизированно, можно использовать db_nmap. Так же можно использовать какую-то любую утилиту для сканирования, экспортировать результаты её работы в XML-файл, а потом - импортировать его в метасплит. Это можно сделать, использовав db_import внутри меню метасплит.

Выполним команду: `db_nmap -sV 192.168.0.0/24`

Вывод будет такой же как и при использовании команды `nmap`. Теперь после того как была просканирована наша сеть можно посмотреть, что было записано в БД: `hosts`, `services`. Результаты на рисунках 13 и 14.

```
msf > hosts
```

Hosts						
=====						
address	mac	name	os_name	os_flavor	os_sp	purp
ose info	comments					
-----	---	----	-----	-----	-----	----
192.168.0.1	c4:6e:1f:fe:27:6a		Unknown			devi
192.168.0.102			Unknown			devi
192.168.0.104	84:4b:f5:39:d6:c6	HAPPYFANIK	Windows 10	Education		devi
192.168.0.105	08:00:27:0e:e0:29		Unknown			devi

```
msf >
```

Рис. 13: Таблица хостов

2.9 Выбрать пять записей из файла `nmap-service-probes` и описать их работу. Выбрать один скрипт из состава `Nmap` и описать его работу

- Первая запись

Возьмем самую первую запись `probe` - эта запись теста с отправкой `null`-запроса. В данной записи будет отправляться пустой запрос по протоколу `TCP`. С ожиданием ответа в 6 секунд (директива `totalwaitms`).

- Вторая запись

Второй записью рассмотрим `match` после `probe null`-запроса. Если пользователь укажет ключ `-sV` при использовании `nmap` и после отправки нулевого теста с сервера придет выражение подходящее под `mSxf5xc6x1a` тогда в колонке `SERVICE` при выводе информации он увидит наименование сервиса `1c-server`, а в колонке `VERSION` `1C:Enterprise business management server`.

- Третья запись

```
msf > services
```

Services					
=====					
host	port	proto	name	state	info
----	----	-----	----	-----	-----
192.168.0.1	22	tcp	ssh	open	Dropbear sshd 2012.55 protocol 2.0
192.168.0.1	1900	tcp	upnp	open	ip0S upnpd TP-LINK TL-WR841N WAP 9.0; UPnP 1.0
192.168.0.1	49152	tcp	http	open	Huawei HG8245T modem http config
192.168.0.1	80	tcp	http	open	TP-LINK WR841N WAP http config
192.168.0.102	6070	tcp		closed	
192.168.0.102	23	tcp		closed	
192.168.0.102	57772	tcp		closed	
192.168.0.102	25	tcp		closed	
192.168.0.102	50013	tcp		closed	
192.168.0.102	42	tcp		closed	
192.168.0.102	49	tcp		closed	
192.168.0.102	50000	tcp		closed	

Рис. 14: Таблица сервисов

```
# This is the NULL probe that just compares any banners given to us
#####NEXT PROBE#####
Probe TCP NULL q||
# Wait for at least 6 seconds for data. It used to be 5, but some
# smtp services have lately been instituting an artificial pause (see
# FEATURE('greet_pause') in Sendmail, for example)
totalwaitms 6000
```

Рис. 15: Запись 1

```
match lc-server m|^S\x5\x6\x1a{| p/1C:Enterprise business management server/
```

Рис. 16: Запись 2

В данной записи на сервер отправляется запрос по протоколу TCP в котором передается информация. Так же указан список портов и ssl-портов, по которым нужно осуществлять сканирование.

- Четвертая записи

Здесь отправляется запрос по протоколу UDP для проверки RPC.

- Пятая записи

Здесь отправляется запрос по протоколу UDP для проверки sql, через порт 1434.

Далее следует описание скрипта в переменной `description`:

```
description = [[
Detects the Skype version 2 service.
]]
```

Далее описание в комментарии в формате NSEDoc.

```
---
-- @output
-- PORT      STATE SERVICE VERSION
-- 80/tcp open  skype2  Skype
```

Далее имя автора, лицензия, категория:

```
author = "Brandon Enright"
license = "Same as Nmap--See http://nmap.org/book/man-legal.html"
categories = {"version"}
```

Далее правило порта(вместо него можно задавать правило хоста)
- функция возвращающая true или false. Если возвращает true, то выполняется функция заданная в переменной `action`. В данном случае из кода понятно, в каких случаях выполняется это условие.

```
portrule = function(host, port)
  return (port.number == 80 or port.number == 443 or
    port.service == nil or port.service == "" or
    port.service == "unknown") -- условия по портам
  and port.protocol == "tcp" and port.state == "open" -- условия по протоколу
  and port.version.name_confidence < 10 -- доверие
  and not(shortport.port_is_excluded(port.number,port.protocol)) -- порт не исключен
  and nmap.version_intensity() >= 7 -- версия интенсивности
end
```

Далее приведен код функции `action`:

```
action = function(host, port)
  local result, rand
  -- Did the service engine already do the hard work?
  if port.version and port.version.service_fp then
    -- Probes sent, replies received, but no match.
    result = U.get_response(port.version.service_fp, "GetRequest")
```



```

-- Loop through the ASCII probes most likely to receive random response
-- from Skype. Others will also receive this response, but are harder to
-- distinguish from an echo service.
for _, p in ipairs({"HTTPOptions", "RTSPRequest"}) do
    rand = U.get_response(port.version.service_fp, p)
    if rand then
        break
    end
end
end
local status
if not result then
    -- Have to send the probe ourselves.
    status, result = comm.exchange(host, port,
        "GET / HTTP/1.0\r\n\r\n", {bytes=26})

    if (not status) then
        return nil
    end
end

if (result ~= "HTTP/1.0 404 Not Found\r\n\r\n") then
    return
end

-- So far so good, now see if we get random data for another request
if not rand then
    status, rand = comm.exchange(host, port,
        "random data\r\n\r\n", {bytes=15})

    if (not status) then
        return
    end
end

if string.match(rand, "[^%s!-~].*[^%s!-~].*[^%s!-~]") then
    -- Detected
    port.version.name = "skype2"
    port.version.product = "Skype"
    nmap.set_port_version(host, port)
    return
end
return
end

```

После прохождения всех проверок, обозначается версия и продукт (после комментария Detected).

Для использования скрипта нужно ввести либо ключ `-sC`, либо `script=<имя_скрипта>`.

Для примера была просканирована основная ОС. Двумя способами. Результаты можно увидеть на рисунках 20 и 21.

```
root@kali:/usr/share/nmap/scripts# nmap 192.168.150.1 -p-
Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-31 01:35 EDT
Nmap scan report for 192.168.150.1
Host is up (0.00052s latency).
Not shown: 65515 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
902/tcp    open  iss-realsecure
912/tcp    open  apex-mesh
1025/tcp   open  NFS-or-IIS
1026/tcp   open  LSA-or-nterm
1027/tcp   open  IIS
1028/tcp   open  unknown
1031/tcp   open  iad2
1074/tcp   open  warmspotMgmt
1203/tcp   open  unknown
1433/tcp   open  ms-sql-s
2383/tcp   open  ms-olap4
2869/tcp   open  icslap
3050/tcp   open  gds_db
5357/tcp   open  wsdapi
27017/tcp  open  unknown
41535/tcp  open  unknown
```

Рис. 20: Результаты сканирования без использования скрипта

3 Вывод

В ходе проделанной работы были изучены основы работы с nmap, немного изучен сниффер wireshark, а так же запись в БД metasploit посредством команды `db_nmap`. Ранее я не был знаком с подобными программами, поэтому не с чем сравнивать. В общем интересно было посканировать хосты и порты. Понравилось сканировать трафик через wireshark. Так же понравилось писать свой сервер для порта.

```
root@kali:/usr/share/nmap/scripts# nmap 192.168.150.1 -p- --script=skypev2-version.nse

Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-31 02:03 EDT
Nmap scan report for 192.168.150.1
Host is up (0.00075s latency).
Not shown: 65515 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
902/tcp    open  iss-realsecure
912/tcp    open  apex-mesh
1025/tcp   open  NFS-or-IIS
1026/tcp   open  LSA-or-nterm
1027/tcp   open  IIS
1028/tcp   open  unknown
1031/tcp   open  iad2
1074/tcp   open  warmspotMgmt
1203/tcp   open  unknown
1433/tcp   open  ms-sql-s
2383/tcp   open  ms-olap4
2869/tcp   open  icslap
3050/tcp   open  gds_db
5357/tcp   open  wsdapi
27017/tcp  open  unknown
41535/tcp  open  skype2
```

Рис. 21: Результаты сканирования с использованием скрипта