

Утилита для исследования сети и сканер
портов Nmap

Виктор БОРИСОВ

March 27, 2016

Contents

1	Цель работы	3
2	Описание	3
3	Ход работы	3
3.1	Поиск активных хостов	3
3.2	Определение открытых портов	4
3.3	Определение версии сервисов	4
3.4	nmap-services, nmap-os-db, nmap-service-probes	5
3.5	Добавление сигнатур в nmap-service-probes	6
3.6	Сохранение вывода в xml	7
3.7	Исследование работы Nmap с помощью Wireshark	7
3.8	Metasploit Framework	8
3.9	Примеры записей из nmap-service-probes	8
3.10	Описание скрипта finger	12

1 Цель работы

Научиться работать с Nmap

2 Описание

nmap — свободная утилита, предназначенная для разнообразного настраиваемого сканирования IP-сетей с любым количеством объектов, определения состояния объектов сканируемой сети (портов и соответствующих им служб). Изначально программа была реализована для систем UNIX, но сейчас доступны версии для множества операционных систем.

Nmap использует множество различных методов сканирования, таких как UDP, TCP (connect), TCP SYN (полукоткрытое), FTP-proxy (прорыв через ftp), Reverse-ident, ICMP (ping), FIN, ACK, Xmas tree, SYN- и NULL-сканирование. Nmap также поддерживает большой набор дополнительных возможностей, а именно: определение операционной системы удалённого хоста с использованием отпечатков стека TCP/IP, «невидимое» сканирование, динамическое вычисление времени задержки и повтор передачи пакетов, параллельное сканирование, определение неактивных хостов методом параллельного ping-опроса, сканирование с использованием ложных хостов, определение наличия пакетных фильтров, прямое (без использования portmapper) RPC-сканирование, сканирование с использованием IP-фрагментации, а также произвольное указание IP-адресов и номеров портов сканируемых сетей.

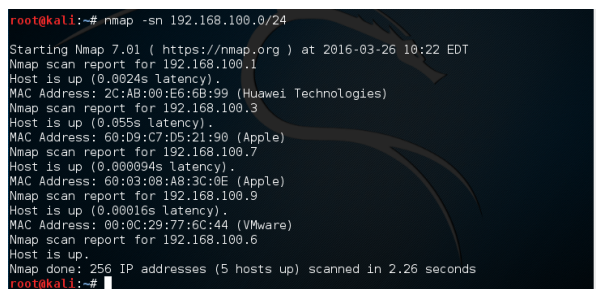
3 Ход работы

3.1 Поиск активных хостов

Сканируем локальную сеть и ищем активные хосты с помощью команды nmap с ключом -sn и диапазоном ip адресов

```
$ nmap -sn 192.168.100.0/24
```

Результат выполнения представлен на Рисунке 1



```
root@kali:~# nmap -sn 192.168.100.0/24
Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-26 10:22 EDT
Nmap scan report for 192.168.100.1
Host is up (0.0024s latency).
MAC Address: 2C:AB:00:E6:6B:99 (Huawei Technologies)
Nmap scan report for 192.168.100.3
Host is up (0.055s latency).
MAC Address: 60:D9:C7:05:21:90 (Apple)
Nmap scan report for 192.168.100.7
Host is up (0.000094s latency).
MAC Address: 60:03:08:A8:3C:0E (Apple)
Nmap scan report for 192.168.100.9
Host is up (0.00016s latency).
MAC Address: 00:0C:29:77:6C:44 (VMware)
Nmap scan report for 192.168.100.6
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.26 seconds
root@kali:~#
```

Figure 1: Поиск активных хостов.

3.2 Определение открытых портов

Для определения открытых портов необходимо воспользоваться командой

```
$ nmap --open адресХоста
```

Результат выполнения для устройства под управлением ОС Linux представлен на Рисунке ??, для iOS - Рисунок 3

```
root@kali:~# nmap --open 192.168.100.9
Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-26 11:07 EDT
Nmap scan report for 192.168.100.9
Host is up (0.00032s 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
8000/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:77:6C:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
```

Figure 2: Открытые порты ОС Linux.

```
root@kali:~# nmap --open 192.168.100.7
Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-26 11:07 EDT
Nmap scan report for 192.168.100.7
Host is up (0.00028s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
MAC Address: 60:03:00:A8:3C:0E (Apple)

Nmap done: 1 IP address (1 host up) scanned in 4.57 seconds
```

Figure 3: Открытые порты ОС iOS.

3.3 Определение версии сервисов

Для определения версии необходимо воспользоваться командой

```
$ nmap -sV адресХоста
```

Результат выполнения для устройства под управлением ОС Windows представлен на Рисунке 4

```

root@kali:~# nmap -sV 192.168.100.9
Starting Nmap 7.81 ( https://nmap.org ) at 2016-03-26 11:13 EDT
Nmap scan report for 192.168.100.9
Host is up (0.00036s 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 gmiregistry
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-ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.8 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          Unreal ircd
8080/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 98:0C:29:77:6C:44 (VMware)
Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OS
s: 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 15.54 seconds

```

Figure 4: Версии сервисов.

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

Файл **nmap-services** содержит сопоставление имен портов с их номером и протоколом. Каждая запись содержит вероятность того, что порт открыт. Пример записей:

```

msp          18/udp    0.000610 # Message Send Protocol
chargen      19/tcp    0.002559 # ttytst source Character Generator
chargen      19/udp    0.015865 # ttytst source Character Generator
ftp-data     20/tcp    0.001079 # File Transfer [Default Data]
ftp-data     20/udp    0.001878 # File Transfer [Default Data]
ftp          21/tcp    0.197667 # File Transfer [Control]
ftp          21/udp    0.004844 # File Transfer [Control]
ssh          22/tcp    0.182286 # Secure Shell Login
ssh          22/udp    0.003905 # Secure Shell Login
telnet       23/tcp    0.221265
telnet       23/udp    0.006211
priv-mail    24/tcp    0.001154 # any private mail system
priv-mail    24/udp    0.000329 # any private mail system
smtp         25/tcp    0.131314 # Simple Mail Transfer
smtp         25/udp    0.001285 # Simple Mail Transfer

```

Файл **nmap-os-db** необходим для определения ОС хоста. В ней содержится примеры ответов различных ОС на специальные запросы Nmap. Он разделен на блоки, так называемые отпечатки, содержащие название ОС, классификацию и данные ответа. Пример:

```

Fingerprint Linux 2.6.17 - 2.6.24
Class Linux | Linux | 2.6.X | general purpose
SEQ(SP=A5-D5%GCD=1-6%ISR=A7-D7%TI=Z%II=I%TS=U)
OPS(O1=M400C%O2=M400C%O3=M400C%O4=M400C%O5=M400C%O6=M400C)

```

```

WIN(W1=8018%W2=8018%W3=8018%W4=8018%W5=8018%W6=8018)
ECN(R=Y%DF=Y%T=3B-45%TG=40%W=8018%O=M400C%CC=N%Q=)
T1(R=Y%DF=Y%T=3B-45%TG=40%S=0%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=Y%DF=Y%T=3B-45%TG=40%W=8018%S=0%A=S+%F=AS%O=M400C%RD=0%Q=)
T4(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T5(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T7(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
U1(DF=N%T=3B-45%TG=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(DFI=N%T=3B-45%TG=40%CD=S)

```

Файл **nmap-service-probes** содержит "пробы", используемые для определения программ по прослушиваемому порту (ключи -sV и -A). Пример:

```

#####NEXT PROBE#####
# DNS Server status request: http://www.rfc-editor.org/rfc/rfc1035.txt
Probe UDP DNSStatusRequest q|\0\0\x10\0\0\0\0\0\0\0\0|
ports 53,135
match domain m|\0\0\x90\x04\0\0\0\0\0\0\0\0|
# This one below came from 2 tested Windows XP boxes
match msrpc m|\0\0\x06\0\0\x10\0\0\0\0\0\0\0\0|
[...]
#####NEXT PROBE#####
Probe UDP Help q|help\r\n\r\n|
ports 7,13,37
match chargen m|@ABCDEFGHJKLMNOPQRSTUVWXYZ|
match echo m|^help\r\n\r\n$|
match time m|^\xc0-\xc5]...$|

```

3.5 Добавление сигнатур в nmap-service-probes

Можно добавить собственную пробу в файл **nmap-service-probes**. Для этого необходимо знание хотя бы следующих конструкций:

a. Директива Probe

Синтаксис: Probe <protocol> <probenam> <probestring>
 <protocol> - название протокола (TCP or UDP)
 <probenam> - имя пробы
 <probestring> - отправляемое сообщение

b. Директива match (для сопоставления ответа)

Синтаксис:
 match <service> <pattern> [<versioninfo>]
 <service> - название сервиса
 <pattern> - шаблон ответа
 <versioninfo> - информация о версии

c.

Добавим в файл следующую пробу

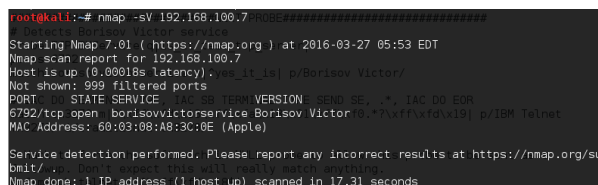
```
#####NEXT PROBE#####  
# Detects Borisov Victor service  
Probe TCP BVService q|is_it_test_tcp_server|  
ports 6792  
match borisovvictorservice m|^yes_it_is| p/Borisov Victor/
```

Создим и на исследуемом компьютере запустим TCP сервер, который слушает порт 6792 и в случае получения сообщения "is_it_test_tcp_server" отправляет обратно "yes_it_is".

Запустим проверку версии сервисов

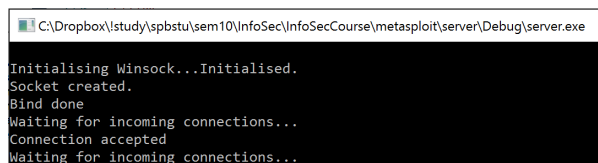
```
nmap -sV 192.168.100.7
```

На Рисунке 5 отображен результат выполнения команды, на Рисунке 6 - вывод TCP сервера



```
root@kali:~# nmap -sV 192.168.100.7 -n -oX /home/nmap_sv_1.xml  
Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-27 05:53 EDT  
Nmap scan report for 192.168.100.7  
Host is up (0.00018s latency).yes_it_is| p/Borisov Victor/  
Not shown: 999 filtered ports  
PORT      STATE SERVICE  
6792/tcp  open  borisovvictorservice Borisov Victor [0.7715716513] p/BM Telnet  
MAC Address: 60:03:08:A8:3C:0E (Apple)  
  
Service detection performed. Please report any incorrect results at https://nmap.org/support/  
bmit/ .. Don't expect this will really match anything.  
Nmap done: 1 IP address (1 host up) scanned in 17.31 seconds
```

Figure 5: Новый сервис.



```
C:\Dropbox\Istudy\spbstu\sem10\InfoSec\InfoSecCourse\metasploit\server\Debug\server.exe  
Initialising Winsock...Initialised.  
Socket created.  
Bind done  
Waiting for incoming connections...  
Connection accepted  
Waiting for incoming connections...
```

Figure 6: TCP сервер.

3.6 Сохранение вывода в xml

Для сохранения вывода утилиты в файл xml необходимо воспользоваться **-oX <file>**. Например,

```
nmap -sV 192.168.100.7 -oX /home/nmap_sv_1.xml
```

3.7 Исследование работы Nmap с помощью Wireshark

Для исследования сетевой активности nmap воспользуемся утилитой Wireshark. Для примера отследим пакеты относящиеся к определению TCP сервиса, написанного нами ранее.

На Рисунках 7, 8 отображены TCP пакеты для определения версии сервиса.

В исходящем пакете в передаваемых данных можно наблюдать текст "is_it_test_tcp_server", а в ответном пакете - "yes_it_is".

No.	Time	Source	Destination	Protocol	Length	Info
734	0.363361879	192.168.100.6	192.168.100.7	TCP	58	6792 → 6792 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
735	0.36336188	192.168.100.7	192.168.100.6	TCP	60	6792 → 6792 [SYN, ACK] Seq=8 Ack=1 Win=0 Len=0
736	0.363674135	192.168.100.6	192.168.100.7	TCP	54	6792 → 6792 [RST] Seq=1 Win=0 Len=0
1481	0.51540577	192.168.100.6	192.168.100.7	TCP	58	6792 → 6792 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1482	0.51540557	192.168.100.7	192.168.100.6	TCP	60	6792 → 6792 [SYN, ACK] Seq=8 Ack=1 Win=0 Len=0
1483	0.515392193	192.168.100.6	192.168.100.7	TCP	54	6792 → 6792 [RST] Seq=1 Win=0 Len=0
1978	10.530328021	192.168.100.6	192.168.100.7	TCP	74	42358 → 6792 [SYN] Seq=0 Win=2920 Len=0 MSS=1460
1979	10.530585558	192.168.100.7	192.168.100.6	TCP	74	6792 → 42358 [SYN, ACK] Seq=8 Ack=1 Win=0 Len=0
1980	10.530544189	192.168.100.6	192.168.100.7	TCP	66	42358 → 6792 [ACK] Seq=1 Ack=1 Win=29312 Len=0
1981	10.530522333	192.168.100.6	192.168.100.7	TCP	57	6792 → 6792 [ACK] Seq=1 Ack=1 Win=29312 Len=0
1982	10.53768461	192.168.100.7	192.168.100.6	TCP	75	6792 → 42358 [FIN, ACK] Seq=1 Ack=22 Win=65504 Len=0
1983	10.537748404	192.168.100.6	192.168.100.7	TCP	66	42358 → 6792 [ACK] Seq=22 Ack=10 Win=29312 Len=0
1984	10.537870146	192.168.100.6	192.168.100.7	TCP	66	42358 → 6792 [ACK] Seq=22 Ack=10 Win=29312 Len=0
1985	10.538011045	192.168.100.7	192.168.100.6	TCP	66	6792 → 42358 [ACK] Seq=10 Ack=23 Win=65504 Len=0

Frame 1981: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface 0
 Ethernet II, Src: VMware_a8:37:55 (00:0c:29:a5:37:55), Dst: Apple_a8:3c:0e (08:00:27:a5:37:55)
 Internet Protocol version 4, Src: 192.168.100.6, Dst: 192.168.100.7
 Transmission Control Protocol, Src Port: 42358 (42358), Dst Port: 6792 (6792), Seq: 1, Ack: 1, Len: 21
 Data (21 bytes)
 Data: 6973697465f746573745f7463705f736572766572
 [Length: 21]

```

0000  68 03 08 a8 3c 0e 00 0c 29 a5 37 55 08 00 45 08  ...8...7U...E
0010  00 49 7c c8 40 00 40 06 74 85 c0 a8 64 06 c8 a8  .I.@.t...d...
0020  04 07 a5 76 18 08 40 4f ef 18 32 51 93 ff 80 18  d...v...0...
0030  00 65 06 77 08 00 01 00 08 00 74 80 00 04 ef    .....R...t
0040  30 49 69 73 5f 69 74 5f 74 65 73 74 5f 74 63 70  _Is..t..test_tcp
0050  5f 73 65 72 76 65 72                               _server
  
```

Figure 7: Исходящий TCP пакет.

No.	Time	Source	Destination	Protocol	Length	Info
734	0.363361879	192.168.100.6	192.168.100.7	TCP	58	6792 → 6792 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
735	0.36336188	192.168.100.7	192.168.100.6	TCP	60	6792 → 6792 [SYN, ACK] Seq=8 Ack=1 Win=0 Len=0
736	0.363674135	192.168.100.6	192.168.100.7	TCP	54	6792 → 6792 [RST] Seq=1 Win=0 Len=0
1481	0.51540577	192.168.100.6	192.168.100.7	TCP	58	6792 → 6792 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1482	0.51540557	192.168.100.7	192.168.100.6	TCP	60	6792 → 6792 [SYN, ACK] Seq=8 Ack=1 Win=0 Len=0
1483	0.515392193	192.168.100.6	192.168.100.7	TCP	54	6792 → 6792 [RST] Seq=1 Win=0 Len=0
1978	10.530328021	192.168.100.6	192.168.100.7	TCP	74	42358 → 6792 [SYN] Seq=0 Win=2920 Len=0 MSS=1460
1979	10.530585558	192.168.100.7	192.168.100.6	TCP	74	6792 → 42358 [SYN, ACK] Seq=8 Ack=1 Win=0 Len=0
1980	10.530544189	192.168.100.6	192.168.100.7	TCP	66	42358 → 6792 [ACK] Seq=1 Ack=1 Win=29312 Len=0
1981	10.53768461	192.168.100.7	192.168.100.6	TCP	75	6792 → 42358 [FIN, ACK] Seq=1 Ack=22 Win=65504 Len=0
1982	10.537748404	192.168.100.6	192.168.100.7	TCP	66	42358 → 6792 [ACK] Seq=22 Ack=10 Win=29312 Len=0
1983	10.537870146	192.168.100.6	192.168.100.7	TCP	66	42358 → 6792 [ACK] Seq=22 Ack=10 Win=29312 Len=0
1984	10.538011045	192.168.100.7	192.168.100.6	TCP	66	6792 → 42358 [ACK] Seq=10 Ack=23 Win=65504 Len=0

Frame 1982: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface 0
 Ethernet II, Src: Apple_a8:3c:0e (08:00:27:a5:37:55), Dst: VMware_a8:37:55 (00:0c:29:a5:37:55)
 Internet Protocol version 4, Src: 192.168.100.7, Dst: 192.168.100.6
 Transmission Control Protocol, Src Port: 6792 (6792), Dst Port: 42358 (42358), Seq: 1, Ack: 22, Len: 9
 Data (9 bytes)
 Data: 7965735f69745f6973
 [Length: 9]

```

0000  00 0c 29 a5 37 55 08 03 08 a8 3c 0e 08 00 45 08  ...7U...8...E
0010  00 30 50 c9 40 00 06 60 93 c0 a8 64 07 c0 a8  .sp...d...
0020  64 80 1a 08 a5 73 32 51 93 ff 40 4f ef c0 80 18  d...v...0...
0030  01 04 50 7b 00 00 01 00 08 0a 04 ef 52 c0 00 74  .L.....R...t
0040  00 30 79 65 73 5f 69 74 5f 69 73                ...yes..t.._Is
  
```

Figure 8: Входящий TCP пакет.

3.8 Metasploit Framework

- инструмент для создания, тестирования и использования эксплойтов. Позволяет конструировать эксплойты с необходимой в конкретном случае «боевой 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 или nessus.

Проверим на уязвимости виртуальную машину Metasploitable2, используя db_nmap (аналог nmap, сохраняющий результаты в БД) командой

```
db_nmap -A 192.168.100.9
```

Результат сканирования отображен на Рисунках 9, 10, 11

3.9 Примеры записей из nmap-service-probes

```

#####NEXT PROBE#####
# Detects TN3270 Servers which send IAC DO TTYPE on initial connection
# instead of IAC DO TN3270E
Probe TCP tn3270 q|\xff\xfb\x18\xff\xfa\x18\x00IBM-3279-4-E\xff\x0|
  
```



```

msf > db nmap -A 192.168.100.9
[*] Nmap: Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-27 07:14 EDT
[*] Nmap: Nmap scan report for 192.168.100.9
[*] Nmap: Host is up (0.00037s latency).
[*] Nmap: Not shown: 977 closed ports
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 21/tcp    open  ftp          vsftpd 2.3.4
[*] Nmap: |_ftp-anon: Anonymous FTP login allowed (FTP code 230)
[*] Nmap: 22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
[*] Nmap: |_ssh-hostkey:
[*] Nmap: |_ 1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
[*] Nmap: |_ 2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
[*] Nmap: 23/tcp    open  telnet       Linux telnetd
[*] Nmap: 25/tcp    open  smtp          Postfix smtpd
[*] Nmap: |_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000
, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
[*] Nmap: |_ssl-cert: Subject: commonName=ubuntu004-base.localdomain/organizatio
nName=0C0SA/stateOrProvinceName=There is no such thing outside US/countryName=XX
[*] Nmap: |_Not valid before: 2010-03-17T14:07:45
[*] Nmap: |_Not valid after: 2010-04-16T14:07:45
[*] Nmap: |_ssl-date: 2016-03-26T18:57:23+00:00; -16h17m31s from scanner time.
[*] Nmap: 53/tcp    open  domain       ISC BIND 9.4.2
[*] Nmap: |_dns-nsid:
[*] Nmap: |_bind.version: 9.4.2
[*] Nmap: 80/tcp    open  http          Apache httpd 2.2.8 ((Ubuntu) DAV/2)
[*] Nmap: |_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
[*] Nmap: |_http-title: Metasploitable2 - Linux

```

Figure 9: db_nmap

rarity 8
ports 23,2323,2023,623
sslports 992

Согласно данной записи nmap для обнаружения сервиса использует протокол
tcp для отправки пакета, содержащего

\xff\xfb\x18\xff\xfa\x18\x00IBM-3279-4-E\xff\x0

Целевые порты 23, 2323, 2023, 623, ssl - 992. rarity - индикатор того,
насколько часто возвращаемые пакеты содержат полезную информацию.

```

#####NEXT PROBE#####
Probe UDP AndroMouse q|AMSNIFF|
rarity 9
ports 8888

```

match AndroMouse m|^GOTBACK\$|s p/AndroMouse Android remote mouse server/

Протокол - UDP, редкость полезных ответов - 9, порт - 8888. Данные для
отправки

AMSNIFF

Шаблон ответа

m|^GOTBACK\$|s

Дополнительная информация - "AndroMouse Android remote mouse server"

```

#####NEXT PROBE#####
Probe UDP AirHID q|from:airhid|
rarity 9
ports 13246

```

match AirHID m|^andReceiver-\d+\.\d+\.\d+\$|s p/AirHID Andrioid remote mouse server/

```

[*] Nmap: 111/tcp open rpcbind 2 (RPC #100000)
[*] Nmap: | rpcinfo:
[*] Nmap: | program version port/proto service
[*] Nmap: | 100000 2 111/tcp rpcbind
[*] Nmap: | 100000 2 111/udp rpcbind
[*] Nmap: | 100003 2,3,4 2049/tcp nfs
[*] Nmap: | 100003 2,3,4 2049/udp nfs
[*] Nmap: | 100005 1,2,3 42841/tcp mountd
[*] Nmap: | 100005 1,2,3 58287/udp mountd
[*] Nmap: | 100021 1,3,4 44677/tcp nlockmgr
[*] Nmap: | 100021 1,3,4 54520/udp nlockmgr
[*] Nmap: | 100024 1 41270/tcp status
[*] Nmap: | 100024 1 51630/udp status
[*] 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 rexecd
[*] Nmap: 513/tcp open login?
[*] Nmap: 514/tcp open tcpwrapped
[*] Nmap: 1099/tcp open rmiregistry GNU Classpath grmiregistry
[*] Nmap: |_rmi-dumpregistry: Registry listing failed (No return data received from server)
[*] 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: | mysql-info:
[*] Nmap: | Protocol: 53
[*] Nmap: | Version: .0.51a-3ubuntu5
[*] Nmap: | Thread ID: 12
[*] Nmap: | Capabilities flags: 43564
[*] Nmap: | Some Capabilities: Support41Auth, SupportsCompression, Speaks41ProtocolNew, SwitchToSSLAfterHandshake, ConnectWithDatabase, SupportsTransactions, LongColumnFlag
[*] Nmap: | Status: Autocommit
[*] Nmap: | Salt: X^QZPc~sGFhZk*/IeBWh
[*] Nmap: 5432/tcp open postgresql PostgreSQL DB 8.3.0 - 8.3.7

```

Figure 10: db_nmap

Протокол - UDP, редкость полезных ответов - 9, порт - 13246. Данные для отправки

from:airhid

Шаблон ответа

m|^andReceiver-\d+\.\d+\.\d+\$|s

Дополнительная информация - "AirHID Android remote mouse server"

#####NEXT PROBE#####

Queries z/OS Network Job Entry

Sends an NJE Probe with the following information (text is converted to EBCDIC):

TYPE = OPEN

OHOST = FAKE

RHOST = FAKE

RIP and OIP = 0.0.0.0

R = 0

Based on http://www-01.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.has
Probe TCP NJE q|\xd6\xd7\xc5\xd5@@@@\xc6\xc1\xd2\xc5@@@@\0\0\0\0\xc6\xc1\xd2\xc5@@@@\0\0\0\0

rarity 9

ports 175

sslports 2252

If the port supports NJE it will respond with either a 'NAK' or 'ACK' in EBCDIC

match nje m|^ \xd5\xc1\xd2| p/IBM Network Job Entry (JES)/

match nje m|^ \xc1\xc3\xd2| p/IBM Network Job Entry (JES)/

```

[*] Nmap: 5900/tcp open  vnc          VNC (protocol 3.3)
[*] Nmap: | vnc-info:
[*] Nmap: |   Protocol version: 3.3
[*] Nmap: |   Security types:
[*] Nmap: |     Unknown security type (33554432)
[*] 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: |_ajp-methods: Failed to get a valid response for the OPTION request
[*] Nmap: 8180/tcp open  http          Apache Tomcat/Coyote JSP engine 1.1
[*] Nmap: |_http-favicon: Apache Tomcat
[*] Nmap: |_http-server-header: Apache-Coyote/1.1
[*] Nmap: |_http-title: Apache Tomcat/5.5
[*] Nmap: MAC Address: 00:0C:29:77:6C:44 (VMware)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 2.6.X
[*] Nmap: OS CPE: cpe:/o:linux:linux kernel:2.6
[*] Nmap: OS details: Linux 2.6.9 - 2.6.33
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metas
ploitabile.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: Host script results:
[*] Nmap: |_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBI
OS MAC: <unknown> (unknown)
[*] Nmap: | smb-os-discovery:
[*] Nmap: |   OS: Unix (Samba 3.0.20-Debian)
[*] Nmap: |   NetBIOS computer name:
[*] Nmap: |   Workgroup: WORKGROUP
[*] Nmap: |   System time: 2016-03-26T14:57:21-04:00
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT      ADDRESS
[*] Nmap: 1    0.37 ms 192.168.100.9
[*] Nmap: OS and Service detection performed. Please report any incorrect result
s at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 77.56 seconds

```

Figure 11: db_nmap

Протокол - TCP, редкость полезных ответов - 9, порт - 175, ssl порт - 2252.

Данные для отправки

\xd6\xd7\xc5\xd5@@@\xc6\xc1\xd2\xc5@@@\0\0\0\0\xc6\xc1\xd2\xc5@@@\0\0\0\0\0

Шаблоны ответа

\xd5\xc1\xd2

\xc1\xc3\xd2

Дополнительная информация для обоих шаблонов - "IBM Network Job Entry (JES)"

```

#####NEXT PROBE#####
# Sends a ServerInfo PBC request to the Basho Riak distributed database
Probe TCP riak-pbc ql\0\0\0\x01\x07|
rarity 8
ports 8087
match riak-pbc m|^\....\x08..(riak@[\w._-]+)..([\w._-]+)$|s p/Basho Riak/ v/$2/ h/$1/

```

Протокол - TCP, редкость полезных ответов - 8, порт - 8087. Данные для отправки

\0\0\0\x01\x07

Шаблоны ответа

^\....\x08..(riak@[\w._-]+)..([\w._-]+)\$

Дополнительная информация - "Basho Riak", версия и имя хоста получаются из регулярного выражения.

3.10 Описание скрипта finger

В начале содержится описание скрипта

```
description = [[
Attempts to get a list of usernames via the finger service.
]]
```

```
author = "Eddie Bell"
```

```
license = "Same as Nmap--See https://nmap.org/book/man-legal.html"
```

Категории, к которым принадлежит скрипт

```
categories = {"default", "discovery", "safe"}
```

Пример вывода

```
---
-- @output
-- PORT      STATE SERVICE
-- 79/tcp    open  finger
-- | finger:
-- | Welcome to Linux version 2.6.31.12-0.2-default at linux-pb94.site !
-- | 01:14am up 18:54, 4 users, load average: 0.14, 0.08, 0.01
-- |
-- | Login      Name           Tty      Idle  Login Time  Where
-- | Gutek      Ange Gutek      *:0      -      Wed 06:19  console
-- | Gutek      Ange Gutek      pts/1    18:54  Wed 06:20
-- | Gutek      Ange Gutek      *pts/0   -      Thu 00:41
-- | _Gutek     Ange Gutek      *pts/4   3      Thu 01:06
```

Подключение библиотек

```
require "comm"
require "shortport"
```

Проверка называется ли сервис "finger" или порт равен 79.

```
portrule = shortport.port_or_service(79, "finger")
```

nmap.new_try создает обработчик исключений, comm.exchange - обрабатывает сетевые транзакции. В данном случае происходит ожидание пока не получено хотя бы 100 строк, не менее 5 секунд или пока хост не закроет подключение.

```
action = function(host, port)
local try = nmap.new_try()

return try(comm.exchange(host, port, "\r\n",
                        {lines=100, proto=port.protocol, timeout=5000}))
end
```