Лабораторная работа №2
Дисциплина «Методы и средства защиты информации»
Nmap

Евгений Хандыго, гр. 53501/3

31 марта 2016 г.

Содержание

1 Постановка задачи

В рамках данной работы необходимо овладеть основными аспектами работы с утилитой Nmap. Ход работы соответствует следующим пунктам:

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

Nmap (*Network Mapper*) — это утилита предназначенная для изучения состояния и проверки безопасности компьютерных сетей. Общий синтаксис вызова даннной утилиты приведен в листинге 1.1.

```
nmap [Scan Type...] [Options] {target specification}
```

Listing 1.1: Синтаксис вызова утилиты Nmap

2 Поиск активных хостов

При вызове утилиты Nmap без дополнительных параметров поиск активных хостов в заданной сети будет произведен автоматически. Недостаток такого подхода заключается в том, что вместе с этим будут также просканированы порты обнаруженных машин, что может привлечь излишнее внимание к хосту, инициализироваашему сканирование. К тому же, при наличии достаточно большого количества активных машин в сети, такая операция может занять ощутимо большое количество времени. Таким образом при поиске активных хостов целесообразно избежать этапа сканирования портов. Для этого можно воспользоваться флагом `-sn` как показано в листинге 2.1.

```
nmap -n -sn 10.0.0.0/24

Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-30 12:47 EDT
Nmap scan report for 10.0.0.100
Host is up (0.00057s latency).
MAC Address: 08:00:27:9A:4B:EE (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.0.101
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 6.95 seconds
```

Listing 2.1: Поиск активных хостов с использованием Nmap

Здесь пр вызове утилиты Nmap также была использована опция `-n`, которая указывает Nmap не пытаться делать обратное разрешение адресов (reserve DNS resolution) обнаруженных активных машин. В данном случае использование этой опции можно считать полностью оправданным поскольку в сканируемой сети отсутствует DNS сервер. В реальной жизни DNS-имя хоста в сети потенциально может о многом сказать, однако, стоит помнить, что процесс обратного разрешения ip адресов может существенно замедлить проесс сканирования [1]. Таким образом, вызов Nmap, приведенный в листинге 2.1 соответствует операции поиска активных машин в сети с адресом `10.0.0.0` с маской `255.255.255.0` без совершения обратного разрешения ip адресов обнаруженных хостов.

Nmap предоставляет несколько опций для обнаружения активных хостов. Наиболее быстрой считается опция `-PR`. Данный флаг считается значением по умолчанию и, более того, будет использован в случаях, когда явно указан какой-либо другой флаг. Опция `-PR` соответствует широкофещательному запросу в протоколе ARP. Конечно, использование данного протокола не всегда возможно. Для принудительной отмены использования ARP можно использовать флаг `--disable-arp-ping`. В качестве примера рассмотрим опцию `-PS`, которая соответствует отправке TCP запроса (по умолчанию на порт `80`) с флагом `SYN`. Вне зависимости от ответа сканируемого хоста Nmap прервет процесс *тройного рукопожатия*. Суть данного метода заключается в том, что ответ порта в сущности не имеет значения, важен факт наличия ответа. Пример использования опции `--PS` представлен в листинге 2.2.

```
nmap -n -sn --disable-arp-ping -PS 10.0.0.0/24

Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-30 13:36 EDT
Nmap scan report for 10.0.0.100
Host is up (0.0014s latency).
MAC Address: 08:00:27:9A:4B:EE (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.0.101
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 17.35 seconds
```

Listing 2.2: Поиск активных хостов с использованием Nmap и опцией `-PS`

[1] https://nmap.org/book/man-host-discovery.html (see `-n` section)

3 Поиск открытых портов

При обнаружении открытых портов Nmap также предоставляет несколько опций. По умолчанию используется флаг `-sS`, который соответствует посылке TCP запроса с флагом `SYN`. Преимущество такого подхода заключается в том, что в ходе сканирования не устанавливается никаких соединений. Пример сканирования портов утилитой Nmap представлен в листинге 3.1.

```
nmap -n -sS 10.0.0.100

Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-30 14:00 EDT
Nmap scan report for 10.0.0.100
Host is up (0.00064s latency).
Not shown: 976 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
6789/tcp open  ibm-db2-admin
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:9A:4B:EE (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
```

Listing 3.1: Поиск активных портов с использованием Nmap и опции `-sS`

Конечно, использование опции `-sS` оправдано только для сканирования портов, на ко-

торых запущены службы, способные работать с протоколом TCP. Для обнаружения портов, которые, например, настроены на прием UDP пакетов необходимо воспользоваться опцией `-sU` как показано в листинге 3.2.

```
nmap -n -sU 10.0.0.100

Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-30 14:02 EDT
Nmap scan report for 10.0.0.100
Host is up (0.00068s latency).
Not shown: 951 closed ports, 45 open|filtered ports
PORT     STATE SERVICE
53/udp   open  domain
111/udp  open  rpcbind
137/udp  open  netbios-ns
2049/udp open  nfs
MAC Address: 08:00:27:9A:4B:EE (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1012.26 seconds
```

Listing 3.2: Поиск активных портов с использованием Nmap и опции `-sU`

Еще одной интересной опцией, предоставляемой Nmap для сканирования портов, является флаг `-sA`. Функционирование Nmap при использовании данной опции заключается в отправке TCP запроса с флагом `ACK`. В таком случае утилита не способна определит открыт или закрыт порт, однако способна определить какие из них отфильтрованы (то есть закрыты брандмауэром (firewall)). Такая функция может оказаться полезной, например, при проведении аудита безопасни. Пример использования флага `-sA` представлен в листинге 3.3.

```
nmap -n -sA 10.0.0.100

Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-30 14:28 EDT
Nmap scan report for 10.0.0.100
Host is up (0.00071s latency).
All 1000 scanned ports on 10.0.0.100 are unfiltered
MAC Address: 08:00:27:9A:4B:EE (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
```

Listing 3.3: Поиск активных портов с использованием Nmap и опции `-sA`

4 Определение версии сервисов

Nmap производит определение типа сервиса, слушающего некоторый порт на основании информации в файле `nmap-services`. По сути этот файл просто описывает наиболее ве-

роятный порт для всякого известного ему сервиса. Для более точного определения того, какая служба запущена на конкретном сервисе используется опция `-sV`. Здесь уже работают более сложные правила, которые предполагают посылку запросов определнного вида и сравнение полученных ответов с некоторыми шаблонами. Пример использования опции `-sV` представлен в листинге 4.1.

```
nmap -n -sV 10.0.0.100

Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-30 15:32 EDT
Nmap scan report for 10.0.0.100
Host is up (0.00048s latency).
Not shown: 976 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  shell         Netkit rshd
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: 08:00:27:9A:4B:EE (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, ...

Service detection performed. Please report any incorrect ...
Nmap done: 1 IP address (1 host up) scanned in 16.06 seconds
```

Listing 4.1: Подробное изучение сервисов с использованием опции `-sV`

Правила, по которым определяются слуюбы и их версии расположены в файле `nmap-service-probes`. Для того, чтобы продемонстрировать его работу создадим собствен-

ный минимальный TCP сервер и попробуем добавить соответствующее правило для его распознавания. Исходный код сервера представлен в листинге 4.2.

```
1  import java.io.*;
2  import java.net.*;
3
4  class TCPServer {
5    public static void main(final String[] args)
6    throws Exception {
7      ServerSocket s = new ServerSocket(6789);
8      while (true) {
9        Socket c = s.accept();
10       InputStreamReader isr = new InputStreamReader(c.getInputStream());
11       BufferedReader in = new BufferedReader(isr);
12       DataOutputStream out = new DataOutputStream(c.getOutputStream());
13       in.readLine();
14       out.writeBytes("Hello, nice to meet you.");
15     }
16   }
17 }
```

Listing 4.2: Исходный код минимального TCP сервера

Механизм работы данной службы тривиален:

- В строке 7 создает сокет для прослушки TCP запросов на порт с номером 6789.
- Внутри бесконечного цикла в строке 9 происходит ожидание сообщений на прослушиваемый порт.
- В строке 14 в ответ помещается сообщение `"Hello, nice to meet you."`.

Для запуска данного сервера необходимо поместить данный код в файл `TCPServer.java` и разместить его на сканируемой машине. Затем данный файл необходимо скомпилировать. Для этого используем команду `javac TCPServer.java`. Для запуска полученного таким образом `.class` файла достаточно выполнить команду `java TCPServer`.

Теперь для того, чтобы «научить» Nmap распознавать созданную службу необходимо определить какого типа запрос должен быть направлен в даннуб службу для того, чтобы получить ожидаемый и, по возможности уникальный в пределах остальных известных Nmap сервисов, ответ. Понятно, что, поскольку ответ, выдаваемый созданным сервисом не зависит от направленного ему запроса, можно использовать любой TCP запрос, например, запрос с пустым телом. Ответ сервера также не зависит ни от каких условий и будет содержать всегда одну и ту же строку. Таким, образом, сегмент файла `nmap-service-probes`, который отвечает за распознавание написанного TCP сервера, может выглядеть как показано в листинге 4.3.

```
Probe TCP NULL q||
...
match my-tcp m|^Hello, nice to meet you.$| p/my-tcp/ v/1.0-SNAPSHOT/
```

Listing 4.3: Сегмент файла `nmap-service-probes`, определяющий `my-tcp`

Для того чтобы проверить корректность определения созданного сервера, воспользуемся уже известной опцией сканирования сервисов с указанием порта как показано в листинге 4.4.

```
nmap -n -sV -p6789 10.0.0.100

Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-30 16:13 EDT
Nmap scan report for 10.0.0.100
Host is up (0.00032s latency).
PORT     STATE SERVICE VERSION
6789/tcp open  my-tcp  my-tcp 1.0-SNAPSHOT
MAC Address: 08:00:27:9A:4B:EE (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect ... .
Nmap done: 1 IP address (1 host up) scanned in 7.28 seconds
```

Listing 4.4: Пример сканирования порта с номером `6789`

5 Формат вывода

При формировании отчета Nmap позволяет сохранить его в нескольких форматах, например, XML. Для того, чтобы воспользоваться этой опцией необходимо добавить флаг `-oX filename` к строке вызова утилиты. Пример использования приведен ниже в листингах 5.1 и 5.2

```
nmap -n -sV -p6789 -oX h100p6789 10.0.0.100

Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-31 02:35 EDT
Nmap scan report for 10.0.0.100
Host is up (0.00036s latency).
PORT     STATE SERVICE VERSION
6789/tcp open  my-tcp  my-tcp 1.0-SNAPSHOT
MAC Address: 08:00:27:9A:4B:EE (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect ... .
Nmap done: 1 IP address (1 host up) scanned in 7.22 seconds
```

Listing 5.1: Пример использования опции `-oX`

```
<nmaprun scanner="nmap"
         args="nmap -n -sV -p6789 -oX h100p6789 10.0.0.100"
         start="1459406137" startstr="Thu Mar 31 02:35:37 2016"
         version="7.01" xmloutputversion="1.04">
  <scaninfo type="syn" protocol="tcp" numservices="1" services="6789"/>
  <verbose level="0"/>
  <debugging level="0"/>
  <host starttime="1459406138" endtime="1459406144">
    <status state="up" reason="arp-response" reason_ttl="0"/>
    <address addr="10.0.0.100" addrtype="ipv4"/>
    <address addr="08:00:27:9A:4B:EE" addrtype="mac"
             vendor="Oracle VirtualBox virtual NIC"/>
    <hostnames>
    </hostnames>
    <ports>
      <port protocol="tcp" portid="6789">
        <state state="open" reason="syn-ack" reason_ttl="64"/>
        <service name="my-tcp" product="my-tcp" version="1.0-SNAPSHOT"
                 method="probed" conf="10"/>
      </port>
    </ports>
    <times srtt="357" rttvar="3758" to="100000"/>
  </host>
  <runstats>
    <finished time="1459406144" timestr="Thu Mar 31 02:35:44 2016"
              elapsed="7.22"
              summary="Nmap done at Thu Mar 31 02:35:44 2016;
                1 IP address (1 host up) scanned in 7.22 seconds"
              exit="success"/>
    <hosts up="1" down="0" total="1"/>
  </runstats>
</nmaprun>
```

Listing 5.2: Содержимое файла `h100p6789`

6 Анализ работы Nmap

Для того, чтобы проанализировать работу Nmap воспользуемся утилитой Wireshark.

Рассмотрим задачу поиска активных хостов в сети. Для удобства предположим, что заранее известен не очень широкий диапазон потенциально уязвимых адресов и возможно применение опции `-PR`. Таким образом, для решения поставленной задачи можно воспользоваться командой

`nmap -n -sn 10.0.0.98-102`.

Результаты анализа трафика данной команды с помощью инструмента Wireshark представлены в литиснге 6.1.

```
1  Source             Destination        Protocol Info
2  CadmusCo_fa:25:8e  Broadcast          ARP      Who has 10.0.0.99?
3                                                  Tell 10.0.0.101
4  CadmusCo_fa:25:8e  Broadcast          ARP      Who has 10.0.0.100?
5                                                  Tell 10.0.0.101
6  CadmusCo_fa:25:8e  Broadcast          ARP      Who has 10.0.0.102?
7                                                  Tell 10.0.0.101
8  CadmusCo_fa:25:8e  Broadcast          ARP      Who has 10.0.0.98?
9                                                  Tell 10.0.0.101
10 CadmusCo_9a:4b:ee  CadmusCo_fa:25:8e  ARP      10.0.0.100 is at
11                                                 08:00:27:9a:4b:ee
12 CadmusCo_fa:25:8e  Broadcast          ARP      Who has 10.0.0.98?
13                                                 Tell 10.0.0.101
14 CadmusCo_fa:25:8e  Broadcast          ARP      Who has 10.0.0.99?
15                                                 Tell 10.0.0.101
16 CadmusCo_fa:25:8e  Broadcast          ARP      Who has 10.0.0.102?
17                                                 Tell 10.0.0.101
```

Listing 6.1: Анализ трафика для `nmap -n -sn 10.0.0.98-102`

Как видно из отчета, процесс поиска активных хостов по протоколу ARP достаточно прост: Nmap просто рассылает широковещательный запрос для того, чтобы узнат, есть ли в сети машина с определенным ip адресом. В случае получения ответа (строка 10) хост считается активным.

Проанализируем теперь сканирование известного сервиса на порту `6789`. Для этого воспользуемся командой

`nmap -n -sV -p6789 10.0.0.100`.

Результаты анализа приведены в листинге 6.2. Провдем подробный разбор:

- В строках $2 - -9$ отражена проверка активности хоста по сценарию, аналогичному 6.1.
- В строках $10 - -12$ происходит проверка того, что порт открыт.
- В строках $13 - -15$ происходит процесс тройного рукопожатия и установление соеденения.
- В строках $15 - -23$ происходит обмен сообщениями и завершение соеденения.

```
Source             Destination        Protocol Info
CadmusCo_fa:25:8e  CadmusCo_9a:4b:ee  ARP      Who has 10.0.0.100?
                                               Tell 10.0.0.101
CadmusCo_9a:4b:ee  CadmusCo_fa:25:8e  ARP      10.0.0.100 is at
                                               08:00:27:9a:4b:ee
CadmusCo_fa:25:8e  Broadcast          ARP      Who has 10.0.0.100?
                                               Tell 10.0.0.101
CadmusCo_9a:4b:ee  CadmusCo_fa:25:8e  ARP      10.0.0.100 is at
                                               08:00:27:9a:4b:ee
10.0.0.101         10.0.0.100         TCP      35170  6789 [SYN]
10.0.0.100         10.0.0.101         TCP      6789  35170 [SYN, ACK]
10.0.0.101         10.0.0.100         TCP      35170  6789 [RST]
10.0.0.101         10.0.0.100         TCP      36538  6789 [SYN]
10.0.0.100         10.0.0.101         TCP      6789  36538 [SYN, ACK]
10.0.0.101         10.0.0.100         TCP      36538  6789 [ACK]
10.0.0.101         10.0.0.100         TCP      36538  6789 [PSH, ACK]
10.0.0.100         10.0.0.101         TCP      6789  36538 [ACK]
10.0.0.100         10.0.0.101         TCP      6789  36538 [PSH, ACK]
10.0.0.101         10.0.0.100         TCP      36538  6789 [ACK]
10.0.0.100         10.0.0.101         TCP      6789  36538 [PSH, ACK]
10.0.0.101         10.0.0.100         TCP      36538  6789 [ACK]
10.0.0.101         10.0.0.100         TCP      36538  6789 [FIN, ACK]
10.0.0.100         10.0.0.101         TCP      6789  36538 [ACK]
```

Listing 6.2: Анализ трафика для `nmap -n -sV -p6789 10.0.0.100`

7 Использование db_nmap

Утилита db_nmap входит в состав фреймворка metasploit. Данный инструмент расширяет возможности Nmap тем, что позволяет сохранять результаты сканирования в базу данных. Для того, чтобы начать работу с db_nmap необходимо сначала запустить сервис базы данных. В случае дистрибутива kali linux это будет PostgreSQL. Для запуска сервиса базы данных воспользуемся командой `service postgresql start`. Для работы непосредственно с db_nmap теперь необходимо открыть консольный интерфейс для metasploit. Для удобной работы с базой данных заведем новый workspace, который будет хранить результаты для сканирования хоста по адресу `10.0.0.100` как показано в листинге 7.1.

```
msf > workspace -a host100
[*] Added workspace: host100
msf > workspace
  default
* host100
```

Listing 7.1: Создание нового workspace для работы с db_nmap

Синтаксис утилиты db_nmap повторяет синтаксис вызова Nmap. Для демонстрации работы с db_nmap проведем сканирование TCP портов хоста с адресом `10.0.0.100` как показано в листинге 7.2.

```
msf > db_nmap -n -v -sV 10.0.0.100
[*] Nmap: Nmap scan report for 10.0.0.100
[*] Nmap: Host is up (0.0016s latency).
[*] Nmap: Not shown: 976 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
[*] 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
[*] Nmap: 445/tcp  open  netbios-ssn   Samba smbd 3.X
[*] Nmap: 512/tcp  open  exec          netkit-rsh rexecd
[*] Nmap: 513/tcp  open  login?
[*] Nmap: 514/tcp  open  shell         Netkit rshd
[*] 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: 6789/tcp open  my-tcp        my-tcp 1.0-SNAPSHOT
[*] 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: 08:00:27:9A:4B:EE (Oracle VirtualBox virtual NIC)
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 18.12 seconds
[*] Nmap: Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.124KB)
```

Listing 7.2: Пример использования утилиты db_nmap

Теперь для проверки результатов работы db_nmap воспользуемся командами `hosts` и `services` как показано в листингах 7.3 и 7.4.

```
msf > hosts -c address,mac,os_name

Hosts
=====
address     mac                os_name
-------     ---                -------
10.0.0.100  08:00:27:9a:4b:ee  Linux
```

Listing 7.3: Результаты сканирования db_nmap

```
msf > services

Services
========
host        port  proto  name           state  info
----        ----  -----  ----           -----  ----
10.0.0.100  21    tcp    ftp            open   vsftpd 2.3.4
10.0.0.100  22    tcp    ssh            open   OpenSSH 4.7p1 Debian
10.0.0.100  23    tcp    telnet         open   Linux telnetd
10.0.0.100  25    tcp    smtp           open   Postfix smtpd
10.0.0.100  53    tcp    domain         open   ISC BIND 9.4.2
10.0.0.100  80    tcp    http           open   Apache httpd 2.2.8
10.0.0.100  111   tcp    rpcbind        open   2 RPC #100000
10.0.0.100  139   tcp    netbios-ssn    open   Samba smbd 3.X
10.0.0.100  445   tcp    netbios-ssn    open   Samba smbd 3.X
10.0.0.100  512   tcp    exec           open   netkit-rsh rexecd
10.0.0.100  513   tcp    login          open
10.0.0.100  514   tcp    shell          open   Netkit rshd
10.0.0.100  1099  tcp    rmiregistry    open   GNU Classpath grmiregistry
10.0.0.100  1524  tcp    shell          open   Metasploitable root shell
10.0.0.100  2049  tcp    nfs            open   2-4 RPC #100003
10.0.0.100  2121  tcp    ftp            open   ProFTPD 1.3.1
10.0.0.100  3306  tcp    mysql          open   MySQL 5.0.51a-3ubuntu5
10.0.0.100  5432  tcp    postgresql     open   PostgreSQL DB 8.3.0 - 8.3.7
10.0.0.100  5900  tcp    vnc            open   VNC protocol 3.3
10.0.0.100  6000  tcp    x11            open   access denied
10.0.0.100  6667  tcp    irc            open   Unreal ircd
10.0.0.100  6789  tcp    my-tcp-server  open   my-tcp-server 1.0-SNAPSHOT
10.0.0.100  8009  tcp    ajp13          open   Apache Jserv Protocol v1.3
10.0.0.100  8180  tcp    http           open   Apache Tomcat
```

Listing 7.4: Результаты сканирования db_nmap

8 Разбора файлов в составе Nmap

Проведем теперь разбор некоторых файлов в составе утилиты Nmap. Сначала обратимся к файлу nmap-service-probes и рассмотрим несколько записей из него.

`Probe TCP NULL q||`
Указывает использовать в качестве пробы TCP запрос с пустым телом.

`totalwaitms 6000`
Указывает ожидать ответа в течении 6ти секунд.

`match acmp m|^ACMP Server Version ([\w._-]+)\r\n| p/Aagon ACMP Inventory/ v/$1/`

Данная запись задает строку сравнения для определения ACMP сервиса. В соответствии с регулярным выражением ACMP сервис распознается если строка ответа начинается с подстроки `ACMP Server Version` за которой следует последовательность символов, которая будет распознана как версия.

`match AndroMouse m|^AMServer$|s p/AndroMouse Android remote mouse server/` Данная запись задает строку сравнения для определения сервиса AndroMouse. В соответствии с регулярным выражением данный сервис распознается если строка ответа в точности соответствует `AMServer`.

Проведем теперь разбор какого-либо скрипта из состава Nmap. В качестве примера был выбран скрипт `vnc-info.nse`. Его исходный код прелставлен в листинге 8.1. Разберем работу этого скрипта:

- Строки $1 - -3$ содержат подключение необходимых библиотек.
- В строках $5 - -11$ содержится некоторая метаинформация о скрипте.
- В строках 13 и 14 содержится описание необходимых локальных переменных и функций.
- В строке 19 создается объект класса типа `VNC`.
- В строке 23 происходит установка соеденения с выбранным портом по указанному адресу.
- Затем в строке 26 роисходит попытка «авторизации» клиента.
- Строки с 26ой по 42ую отвечают за формирование результата.

```
local shortport = require "shortport"
local stdnse = require "stdnse"
local vnc = require "vnc"

description = [[
Queries a VNC server for its protocol version and supported security types.
]]

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

portrule = shortport.port_or_service( {5900, 5901, 5902} , "vnc", "tcp", "open")

local function fail(err) return stdnse.format_output(false, err) end

action = function(host, port)

  local vnc = vnc.VNC:new( host, port )
  local status, data
  local result = stdnse.output_table()

  status, data = vnc:connect()
  if ( not(status) ) then return fail(data) end

  status, data = vnc:handshake()
  if ( not(status) ) then return fail(data) end

  status, data = vnc:getSecTypesAsTable()
  if ( not(status) ) then return fail(data) end

  result["Protocol version"] = vnc:getProtocolVersion()

  if ( data and #data ~= 0 ) then
    result["Security types"] = data
  end

  if ( vnc:supportsSecType(vnc.sectypes.NONE) ) then
    result["WARNING"] = "Server does not require authentication"
  end

  return result
end
```

Listing 8.1: Текст скрипта sec $vnc - info.nse$

9 Заключение

Утилита Nmap предоставляет широкий функционал по сканированию сети. В рамках данной работы были подробно разобраны некоторые аспекты работы с данной утилитой.