

Лабораторная работа №2.  
Утилита ntar

Скрипаль Борис

28 марта 2016 г.

# Оглавление

1	Цель работы . . . . .	2
2	Описание лабораторных условий . . . . .	2
3	Поиск активных хостов . . . . .	3
4	Сканирование портов хоста . . . . .	3
5	Исследование служебных файлов nmap-services, nmap-os-db, nmap-service-probes . . . . .	4
5.1	Файл nmap-services . . . . .	4
5.2	Файл nmap-os-db . . . . .	5
5.3	Файл nmap-service-probes . . . . .	6
6	Добавление собственной сигнатуры в файл nmap-service-probes	6
7	Сохранение вывода утилиты nmap в формате XML . . . . .	8
8	Исследование работы утилиты nmap при помощи wireshark .	15
9	Просканировать виртуальную машину Metasploitable2 исполь- зуя db_nmap из состава metasploit-framework . . . . .	16
10	Описать работу пяти записей из файла nmap-service-probes .	18
11	Описать работу скрипта из состава Nmap . . . . .	19
12	Выводы . . . . .	21

# 1 Цель работы

Изучение принципов работы утилиты nmap.

## 2 Описание лабораторных условий

Для проведения лабораторной работы было создано две виртуальные машины, объединенные в общую виртуальную сеть. Первая виртуальная машина (Metasploitable2) намеренно содержит ряд уязвимостей. Вторая виртуальная машина (Kali Linux) необходима для сканирования и поиска уязвимостей на первой виртуальной машине.

Виртуальная машина с ОС Metasploitable2 имеет адрес 192.168.202.2, вторая (с Kali Linux) имеет адрес 192.168.202.3.

Конфигурация первой машины:

```
msfadmin@metasploitable:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:3b:18:a4
inet addr:192.168.202.2  Bcast:0.0.0.0  Mask:255.255.255.255
inet6 addr: fe80::a00:27ff:fe3b:18a4/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:269 errors:0 dropped:0 overruns:0 frame:0
TX packets:142 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:24841 (24.2 KB)  TX bytes:22808 (22.2 KB)
Base address:0xd010 Memory:f0000000-f0020000
```

Конфигурация второй машины:

```
root@kali:~# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
inet 192.168.202.3  netmask 255.255.255.0  broadcast 192.168.202.255
inet6 fe80::a00:27ff:fe35:a17a  prefixlen 64  scopeid 0x20<link>
ether 08:00:27:35:a1:7a  txqueuelen 1000  (Ethernet)
RX packets 7  bytes 496 (496.0 B)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 51  bytes 6218 (6.0 KiB)
TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Для проверки работоспособности сети "пропингуем" виртуальные машины:

```
msfadmin@metasploitable:~$ ping 192.168.202.3
PING 192.168.202.3 (192.168.202.3) 56(84) bytes of data.
64 bytes from 192.168.202.3: icmp_seq=1 ttl=64 time=10.5 ms
64 bytes from 192.168.202.3: icmp_seq=2 ttl=64 time=0.609 ms
```

```
--- 192.168.202.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.609/5.564/10.520/4.956 ms
```

```
root@kali:~# ping 192.168.202.2
PING 192.168.202.2 (192.168.202.2) 56(84) bytes of data.
```

```
64 bytes from 192.168.202.2: icmp_seq=1 ttl=64 time=0.325 ms
64 bytes from 192.168.202.2: icmp_seq=2 ttl=64 time=0.514 ms
^C
--- 192.168.202.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.325/0.419/0.514/0.096 ms
```

Как видно из результатов, оба хоста находятся в одной сети и видят друг друга.

### 3 Поиск активных хостов

Для поиска активных хостов можно воспользоваться флагами `-sP` утилиты `nmap`. При задании данных флагов, утилита не сканирует порты, а ищет только активные хосты в сети. Ввиду того, что маска нашей подсети `/24`, то для того, что бы утилита просмотрела всю подсеть, необходимо задать ей адрес подсети. В нашем случае `192.168.202`.

Результат сканирования подсети на наличие активных хостов:

```
root@kali:~# nmap -sP 192.168.202.*
Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-20 16:39 EDT
Nmap scan report for 192.168.202.1
Host is up (0.00053s latency).
MAC Address: 0A:00:27:00:00:2C (Unknown)
Nmap scan report for 192.168.202.2
Host is up (0.0013s latency).
MAC Address: 08:00:27:3B:18:A4 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.202.3
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 28.13 seconds
```

В результате сканирования, утилита нашла 3 активных хоста: адрес первого хоста - адрес адаптера виртуальной сети, второй - адрес виртуальной машины `metasploitable`, а третий адрес - адрес текущей машины.

### 4 Сканирование портов хоста

Для сканирования портов хоста, утилите `nmap` необходимо передать адрес хоста, например:

```
root@kali:~# nmap 192.168.202.2
Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-20 16:42 EDT
Nmap scan report for 192.168.202.2
Host is up (0.00022s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
```

```

53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:3B:18:A4 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 13.35 seconds

```

Как видно из вывода, на исследуемой виртуальной машине открыты 21 порт (ftp), 22 порт (ssh), 23 порт (telnet), 80 порт (http), а так же ряд других портов.

## 5 Исследование служебных файлов nmap-services, nmap-os-db, nmap-service-probes

Служебные файлы для утилиты nmap по умолчанию располагаются в директории `"/usr/share/nmap"`.

### 5.1 Файл nmap-services

Служебный файл nmap-services содержит в себе описание назначения стандартных портов. Сам файл имеет структуру таблицы со следующими столбцами: имя\_сервиса, номер\_порта/название\_протокола, частота, комментарии.

Для известных, зарезервированных номеров портов, файл содержит подробное описание, например:

```

ftp-data 20/sctp 0.000000 # File Transfer [Default Data]
ftp-data 20/tcp 0.001079 # File Transfer [Default Data]
ftp-data 20/udp 0.001878 # File Transfer [Default Data]
ftp 21/sctp 0.000000 # File Transfer [Control]
ftp 21/tcp 0.197667 # File Transfer [Control]
ftp 21/udp 0.004844 # File Transfer [Control]
ssh 22/sctp 0.000000 # Secure Shell Login
ssh 22/tcp 0.182286 # Secure Shell Login
ssh 22/udp 0.003905 # Secure Shell Login

```

Для "свободных"номеров портов, файл так же содержит записи, но они не несут никакой полезной информации, что ожидаемо, так как на этих портах запускаются пользовательские сервисы, и они не закреплены ни за одним приложением.

```
unknown 26860/udp 0.000654
unknown 26861/udp 0.000654
unknown 26866/udp 0.001307
unknown 26868/udp 0.001307
```

## 5.2 Файл nmap-os-db

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

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

Для того, что бы утилита nmap выводила операционную систему, необходимо задать ключ -O:

```
root@kali:~# nmap -O 192.168.202.2
Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-20 17:51 EDT
Nmap scan report for 192.168.202.2
Host is up (0.00064s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
```

```
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:3B:18:A4 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
```

Как видно из вывода утилиты, на хосте располагается операционная система Linux с версией ядра 2.6.

### 5.3 Файл `nmap-service-probes`

Данный файл содержит сигнатуры для определения сервисов, прослушивающих тот или иной порт. Как правило, это относится к известным службам, например SMTP - почтовый сервер, или DNS, или какой-либо другой широко используемый сервис. Данные о сервисах задаются при помощи нескольких директив:

- Exclude <port specification>
- Probe <protocol> <probenam> <probestring>
- match <service> <pattern> [<versioninfo>]

## 6 Добавление собственной сигнатуры в файл `nmap-service-probes`

Для того, что бы добавить собственную сигнатуру, создадим небольшой сервер, который мы будем идентифицировать при помощи утилиты nmap. Код сервера представлен ниже:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>

int main()
{

char str[100];
char *rec_srt="Hello, client (MyServer 1.1)\n";

int listen_fd, comm_fd;

struct sockaddr_in servaddr;

listen_fd = socket(AF_INET, SOCK_STREAM, 0);
```

```

bzero( &servaddr, sizeof(servaddr));

servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = htons(INADDR_ANY);
servaddr.sin_port = htons(22000);

bind(listen_fd, (struct sockaddr *) &servaddr, sizeof(servaddr));

listen(listen_fd, 10);

comm_fd = accept(listen_fd, (struct sockaddr*) NULL, NULL);

while(1)
{

bzero( str, 100);

read(comm_fd,str,100);

printf("Echoing back - %s",str);

write(comm_fd, rec_srt, strlen(rec_srt)+1);

}
}

```

Сервер слушает порт 22000 и на входящее сообщение отправляет приветственное сообщение со своей версией.

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

```

Probe TCP MyServer q|\x02Hi|
rarity 1
ports 22000
match testServer m/^Hello, client \((\w*) ([\d.]*)\) / p/$1/ v/$2/

```

В данных строках мы описываем, какое сообщение будем отправлять для идентификации, на какой порт, а так же какой ответ мы будем ожидать.

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

```

root@kali:~/test# nmap 192.168.202.2 -p 22000 -sV

```

```

Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-20 19:33 EDT
Nmap scan report for 192.168.202.2
Host is up (0.00048s latency).
PORT      STATE SERVICE  VERSION
22000/tcp  open  testServer MyServer 1.1
MAC Address: 08:00:27:3B:18:A4 (Oracle VirtualBox virtual NIC)

```



Service detection performed. Please report any incorrect results  
at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 20.05 seconds

Как видно из вывода, утилита корректно определила наш сервер.

## 7 Сохранение вывода утилиты nmap в формате XML

Для сохранения вывода утилиты nmap в формате XML необходимо при запуске указать ключ -oX например:

```
nmap -T4 -A -p 1-5000 -oX - 192.168.202.2 > out.xml
```

В результате запуска, утилитой nmap будут просканированы порты с 1 по 5000 хоста с адресом 192.168.202.2, и вывод будет сохранен в формате XML. Содержимое файла out.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/bin/./share/nmap/nmap.xsl"
type="text/xsl"?>
<!-- Nmap 7.01 scan initiated Mon Mar 21 04:37:35 2016 as: nmap -T4
-A -p 1-5000 -oX - 192.168.202.2 -->
<nmaprun scanner="nmap" args="nmap -T4 -A -p 1-5000 -oX -
192.168.202.2" start="1458549455" startstr="Mon Mar 21 04:37:35
2016" version="7.01" xmloutputversion="1.04">
<scaninfo type="syn" protocol="tcp" numservices="5000"
services="1-5000"/>
<verbose level="0"/>
<debugging level="0"/>
<host starttime="1458549455" endtime="1458549492"><status
state="up" reason="arp-response" reason_ttl="0"/>
<address addr="192.168.202.2" addrtype="ipv4"/>
<address addr="08:00:27:3B:18:A4" addrtype="mac" vendor="Oracle
VirtualBox virtual NIC"/>
<hostnames>
</hostnames>
<ports><extraports state="closed" count="4982">
<extrareasons reason="resets" count="4982"/>
</extraports>
<port protocol="tcp" portid="21"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="ftp"
product="vsftpd" version="2.3.4" ostype="Unix" method="probed"
conf="10"><cpe>cpe:/a:vsftpd:vsftpd:2.3.4</cpe></service><script
id="ftp-anon" output="Anonymous FTP login allowed (FTP code
230)"/></port>
<port protocol="tcp" portid="22"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="ssh"
product="OpenSSH" version="4.7p1 Debian 8ubuntu1"
```

```

extrainfo="protocol 2.0" ostype="Linux" method="probed"
conf="10"><cpe>cpe:/a:openbsd:openssh:4.7p1</cpe><cpe>cpe:/o:linux:linux_kernel</cpe></ser
  id="ssh-hostkey" output="#xa; 1024
60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)&#xa; 2048
56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)"><table>
<elem key="fingerprint">600fcfe1c05f6a74d69024fac4d56ccd</elem>
<elem
key="key">AAAAAB3NzaC1kc3MAAACBALz4hsc8a2SrQ4n1W960qV8xwBG0JC+jI7fWxm5METIJH4tKr/xUTwsTYEYn
<elem key="type">ssh-dss</elem>
<elem key="bits">1024</elem>
</table>
<table>
<elem key="fingerprint">5656240f211ddea72bae61b1243de8f3</elem>
<elem
key="key">AAAAAB3NzaC1yc2EAAAABIwAAAQEAstqnuFMB0Zv03WTEjP4TUDjgWkIVNdTq6kboEDjte0fc65TlI7sR
<elem key="type">ssh-rsa</elem>
<elem key="bits">2048</elem>
</table>
</script></port>
<port protocol="tcp" portid="23"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="telnet"
product="Linux telnetd" ostype="Linux" method="probed"
conf="10"><cpe>cpe:/o:linux:linux_kernel</cpe></service></port>
<port protocol="tcp" portid="25"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="smtp"
product="Postfix smtpd" hostname=" metasploitable.localdomain"
method="probed"
conf="10"><cpe>cpe:/a:postfix:postfix</cpe></service><script
id="smtp-commands" output="metasploitable.localdomain, PIPELINING,
SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITIME,
DSN, "/><script id="ssl-cert" output="Subject:
commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There
is no such thing outside US/countryName=XX&#xa;Not valid before:
2010-03-17T14:07:45&#xa;Not valid after:
2010-04-16T14:07:45"><table key="subject">
<elem key="stateOrProvinceName">There is no such thing outside
US</elem>
<elem key="organizationName">OCOSA</elem>
<elem key="countryName">XX</elem>
<elem key="localityName">Everywhere</elem>
<elem key="organizationalUnitName">Office for Complication of
Otherwise Simple Affairs</elem>
<elem key="commonName">ubuntu804-base.localdomain</elem>
<elem key="emailAddress">root@ubuntu804-base.localdomain</elem>
</table>
<table key="issuer">
<elem key="stateOrProvinceName">There is no such thing outside
US</elem>
<elem key="organizationName">OCOSA</elem>
<elem key="countryName">XX</elem>

```

```

<elem key="localityName">Everywhere</elem>
<elem key="organizationalUnitName">Office for Complication of
Otherwise Simple Affairs</elem>
<elem key="commonName">ubuntu804-base.localdomain</elem>
<elem key="emailAddress">root@ubuntu804-base.localdomain</elem>
</table>
<table key="pubkey">
<elem key="type">rsa</elem>
<elem key="bits">1024</elem>
</table>
<elem key="sig_algo">sha1WithRSAEncryption</elem>
<table key="validity">
<elem key="notBefore">2010-03-17T14:07:45</elem>
<elem key="notAfter">2010-04-16T14:07:45</elem>
</table>
<elem key="md5">dcd9ad906c8f2f7374af383b25408828</elem>
<elem key="sha1">ed093088706603bfd5dc237399b498da2d4d31c6</elem>
<elem key="pem">-&#45;&#45;&#45;&#45;BEGIN
CERTIFICATE-&#45;&#45;&#45;&#45;&#xa;MIIDWzCCAsQCCQD6+TpMf7a5zDANBgkqhkiG9w0BAQUFADC8TELM
CERTIFICATE-&#45;&#45;&#45;&#45;&#xa;</elem>
</script><script id="ssl-date" output="2016-03-21T08:38:05+00:00;
+1s from scanner time."><elem key="delta">1</elem>
<elem key="date">2016-03-21T08:38:05+00:00</elem>
</script></port>
<port protocol="tcp" portid="53"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="domain"
product="ISC BIND" version="9.4.2" method="probed"
conf="10"><cpe>cpe:/a:isc:bind:9.4.2</cpe></service><script
id="dns-nsid" output="&#xa; bind.version: 9.4.2"><elem
key="bind.version">9.4.2</elem>
</script></port>
<port protocol="tcp" portid="80"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="http"
product="Apache httpd" version="2.2.8" extrainfo="(Ubuntu) DAV/2"
method="probed"
conf="10"><cpe>cpe:/a:apache:http_server:2.2.8</cpe></service><script
id="http-server-header" output="Apache/2.2.8 (Ubuntu)
DAV/2"><elem>Apache/2.2.8 (Ubuntu) DAV/2</elem>
</script><script id="http-title" output="Metasploitable2 -
Linux"><elem key="title">Metasploitable2 - Linux</elem>
</script></port>
<port protocol="tcp" portid="111"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="rpcbind"
version="2" extrainfo="RPC #100000" method="probed"
conf="10"/><script id="rpcinfo" output="&#xa; program version
port/proto service&#xa; 100000 2 111/tcp
rpcbind&#xa; 100000 2 111/udp rpcbind&#xa; 100003
2,3,4 2049/tcp nfs&#xa; 100003 2,3,4 2049/udp
nfs&#xa; 100005 1,2,3 46259/tcp mountd&#xa; 100005
1,2,3 48241/udp mountd&#xa; 100021 1,3,4 42298/udp

```

```

nlockmgr&#xa; 100021 1,3,4 47669/tcp nlockmgr&#xa; 100024
1 47986/tcp status&#xa; 100024 1 51797/udp
status&#xa;"><table key="100021">
<table key="tcp">
<table key="version">
<elem>1</elem>
<elem>3</elem>
<elem>4</elem>
</table>
<elem key="port">47669</elem>
</table>
<table key="udp">
<table key="version">
<elem>1</elem>
<elem>3</elem>
<elem>4</elem>
</table>
<elem key="port">42298</elem>
</table>
</table>
<table key="100000">
<table key="udp">
<table key="version">
<elem>2</elem>
</table>
<elem key="port">111</elem>
</table>
<table key="tcp">
<table key="version">
<elem>2</elem>
</table>
<elem key="port">111</elem>
</table>
</table>
<table key="100024">
<table key="tcp">
<table key="version">
<elem>1</elem>
</table>
<elem key="port">47986</elem>
</table>
<table key="udp">
<table key="version">
<elem>1</elem>
</table>
<elem key="port">51797</elem>
</table>
</table>
<table key="100003">
<table key="tcp">

```

```

<table key="version">
<elem>2</elem>
<elem>3</elem>
<elem>4</elem>
</table>
<elem key="port">2049</elem>
</table>
<table key="udp">
<table key="version">
<elem>2</elem>
<elem>3</elem>
<elem>4</elem>
</table>
<elem key="port">2049</elem>
</table>
</table>
<table key="100005">
<table key="tcp">
<table key="version">
<elem>1</elem>
<elem>2</elem>
<elem>3</elem>
</table>
<elem key="port">46259</elem>
</table>
<table key="udp">
<table key="version">
<elem>1</elem>
<elem>2</elem>
<elem>3</elem>
</table>
<elem key="port">48241</elem>
</table>
</table>
</script></port>
<port protocol="tcp" portid="139"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="netbios-ssn"
product="Samba smbd" version="3.X" extrainfo="workgroup: WORKGROUP"
method="probed"
conf="10"><cpe>cpe:/a:samba:samba:3</cpe></service></port>
<port protocol="tcp" portid="445"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="netbios-ssn"
product="Samba smbd" version="3.X" extrainfo="workgroup: WORKGROUP"
method="probed"
conf="10"><cpe>cpe:/a:samba:samba:3</cpe></service></port>
<port protocol="tcp" portid="512"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="exec"
product="netkit-rsh rexecd" ostype="Linux" method="probed"
conf="10"><cpe>cpe:/a:netkit:netkit</cpe><cpe>cpe:/o:linux:linux_kernel</cpe></service></p>
<port protocol="tcp" portid="513"><state state="open"

```

```

reason="syn-ack" reason_ttl="64"/><service name="login"
method="table" conf="3"/></port>
<port protocol="tcp" portid="514"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="shell"
product="Netkit rshd" method="probed"
conf="10"><cpe>cpe:/a:netkit:netkit_rsh</cpe></service></port>
<port protocol="tcp" portid="1099"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="java-rmi"
product="Java RMI Registry" hostname="localhost" method="probed"
conf="10"/></port>
<port protocol="tcp" portid="1524"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="shell"
product="Metasploitable root shell" method="probed"
conf="10"/></port>
<port protocol="tcp" portid="2049"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="nfs" version="2-4"
extrainfo="RPC #100003" method="probed" conf="10"/></port>
<port protocol="tcp" portid="2121"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="ftp"
product="ProFTPD" version="1.3.1" ostype="Unix" method="probed"
conf="10"><cpe>cpe:/a:proftpd:proftpd:1.3.1</cpe></service></port>
<port protocol="tcp" portid="3306"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="mysql"
product="MySQL" version="5.0.51a-3ubuntu5" method="probed"
conf="10"><cpe>cpe:/a:mysql:mysql:5.0.51a-3ubuntu5</cpe></service><script
id="mysql-info" output="&#xa; Protocol: 53&#xa; Version:
.0.51a-3ubuntu5&#xa; Thread ID: 8&#xa; Capabilities flags:
43564&#xa; Some Capabilities: Speaks41ProtocolNew, Support41Auth,
ConnectWithDatabase, SupportsCompression, SupportsTransactions,
SwitchToSSLAfterHandshake, LongColumnFlag&#xa; Status:
Autocommit&#xa; Salt: *'&#xa;n/&gt;*UMc,0)|V{*&#xa;p"><elem
key="Protocol">53</elem>
<elem key="Version">.0.51a-3ubuntu5</elem>
<elem key="Thread ID">8</elem>
<elem key="Capabilities flags">43564</elem>
<table key="Some Capabilities">
<elem>Speaks41ProtocolNew</elem>
<elem>Support41Auth</elem>
<elem>ConnectWithDatabase</elem>
<elem>SupportsCompression</elem>
<elem>SupportsTransactions</elem>
<elem>SwitchToSSLAfterHandshake</elem>
<elem>LongColumnFlag</elem>
</table>
<elem key="Status">Autocommit</elem>
<elem key="Salt">*'&#xa;n/&gt;*UMc,0)|V{*&#xa;p</elem>
</script></port>
<port protocol="tcp" portid="3632"><state state="open"
reason="syn-ack" reason_ttl="64"/><service name="distccd"
product="distccd" version="v1" extrainfo="(GNU) 4.2.4 (Ubuntu

```

```

4.2.4-1ubuntu4)" method="probed" conf="10"/></port>
</ports>
<os><portused state="open" proto="tcp" portid="21"/>
<portused state="closed" proto="tcp" portid="1"/>
<portused state="closed" proto="udp" portid="31595"/>
<osmatch name="Linux 2.6.9 - 2.6.33" accuracy="100" line="53435">
<osclass type="general purpose" vendor="Linux" osfamily="Linux"
osgen="2.6.X"
accuracy="100"><cpe>cpe:/o:linux:linux_kernel:2.6</cpe></osclass>
</osmatch>
</os>
<uptime seconds="105" lastboot="Mon Mar 21 04:36:27 2016"/>
<distance value="1"/>
<tcpsequence index="205" difficulty="Good luck!"
values="1B8A9A35,1B16770E,1B201DD6,1B2C7610,1BF024EA,1B8225C1"/>
<ipidsequence class="All zeros" values="0,0,0,0,0,0"/>
<tcptssequence class="100HZ"
values="24A4,24AE,24B9,24C3,24CD,24D8"/>
<hostscript><script id="nbstat" output="NetBIOS name:
METASPLOITABLE, NetBIOS user: &lt;unknown&gt;, NetBIOS MAC:
&lt;unknown&gt; (unknown)"><table key="names">
<table>
<elem key="flags">1024</elem>
<elem key="suffix">0</elem>
<elem key="name">METASPLOITABLE</elem>
</table>
<table>
<elem key="flags">1024</elem>
<elem key="suffix">3</elem>
<elem key="name">METASPLOITABLE</elem>
</table>
<table>
<elem key="flags">1024</elem>
<elem key="suffix">32</elem>
<elem key="name">METASPLOITABLE</elem>
</table>
<table>
<elem key="flags">33792</elem>
<elem key="suffix">0</elem>
<elem key="name">WORKGROUP</elem>
</table>
<table>
<elem key="flags">33792</elem>
<elem key="suffix">30</elem>
<elem key="name">WORKGROUP</elem>
</table>
</table>
<elem key="server_name">METASPLOITABLE</elem>
<table key="statistics">
<elem>00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</elem>

```

```

<elem>00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</elem>
<elem>00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</elem>
</table>
<table key="mac">
<elem key="manuf">unknown</elem>
<elem key="address">&lt;unknown&gt;</elem>
</table>
<elem key="user">&lt;unknown&gt;</elem>
</script><script id="smb-os-discovery" output="&#xa; OS: Unix
(Samba 3.0.20-Debian)&#xa; NetBIOS computer name: &#xa;
Workgroup: WORKGROUP&#xa; System time:
2016-03-21T04:38:05-04:00&#xa;"><elem key="os">Unix</elem>
<elem key="lanmanager">Samba 3.0.20-Debian</elem>
<elem key="domain">WORKGROUP\x00</elem>
<elem key="server"></elem>
<elem key="date">2016-03-21T04:38:05-04:00</elem>
</script></hostscript><trace>
<hop ttl="1" ipaddr="192.168.202.2" rtt="1.17"/>
</trace>
<times srtd="1168" rttvar="862" to="100000"/>
</host>
<runstats><finished time="1458549492" timestr="Mon Mar 21 04:38:12
2016" elapsed="38.04" summary="Nmap done at Mon Mar 21 04:38:12
2016; 1 IP address (1 host up) scanned in 38.04 seconds"
exit="success"/><hosts up="1" down="0" total="1"/>
</runstats>
</nmaprun>

```

## 8 Исследование работы утилиты nmap при помощи wireshark

При сканировании сети на наличие доступных хостов утилита nmap обращается к DNS серверу для того, чтобы получить список узлов в данной подсети:

```

7 6.528600840 192.168.202.3 192.168.20.3 DNS 86 Standard
query 0x58e6 PTR 2.202.168.192.in-addr.arpa
8 9.029153317 192.168.202.3 192.168.20.3 DNS 86 Standard
query 0x58e7 PTR 2.202.168.192.in-addr.arpa
9 13.074352796 192.168.202.3 8.8.4.4 DNS 86 Standard query
0x58e8 PTR 3.202.168.192.in-addr.arpa
10 15.576831906 192.168.202.3 8.8.4.4 DNS 86 Standard query
0x58e9 PTR 3.202.168.192.in-addr.arpa

```

После получения ответов от DNS сервера, утилита составляет карту доступных узлов в сети.

При простом анализе открытых портов, утилита nmap пытается установить соединение с доступными портами. Если на запрос установления соединения приходит пакет с флагами [SYN, ACK], то порт считается от-



крытым. После этого утилита берет информацию об этом порте из файла `nmap-services`. Ниже приведен пример сканирования 80 порта (HTTP).

```
1 0.000000000 192.168.202.3 192.168.202.2 TCP 58 54248 -> 80
[SYN] Seq=0 Win=1024 Len=0 MSS=1460
2 0.002266788 192.168.202.2 192.168.202.3 TCP 60 80 -> 54248
[SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
3 0.002281592 192.168.202.3 192.168.202.2 TCP 54 54248 -> 80
[RST] Seq=1 Win=0 Len=0
```

Как видно из вывода утилиты `wireshark`, при сканировании порта, `nmap` посылает пакет с флагом SYN (флаг запроса на установление соединения), после чего, если в ответ приходит ответ с флагами [SYN, ACK], это означает, что сервер пытается остановить соединение на этом порте. Это означает что порт открыт, и утилита `nmap` разрывает соединение, посылая пакет с флагом RST (флаг сброса соединения).

Попробуем просканировать закрытый порт. Из предыдущих выводов утилиты `nmap` видно, что порт 112 закрыт. Попробуем просканировать этот порт, указав его при запуске утилиты при помощи ключа `-p`:

```
root@kali:~# nmap 192.168.202.2 -p 112
```

```
Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-21 05:54 EDT
Nmap scan report for 192.168.202.2
Host is up (0.00041s latency).
PORT      STATE SERVICE
112/tcp    closed mcidas
MAC Address: 08:00:27:3B:18:A4 (Oracle VirtualBox virtual NIC)
```

```
Nmap done: 1 IP address (1 host up) scanned in 13.22 seconds
```

Вывод в утилите `wireshark`:

```
15 46.978256397 192.168.202.3 192.168.202.2 TCP 58 51420
-> 112 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
16 46.978812034 192.168.202.2 192.168.202.3 TCP 60 112
-> 51420 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

Как видно из вывода утилиты, утилита так же пытается установить соединение по протоколу TCP, посылая пакет с флагом SYN. Так как порт закрыт, в ответ приходит пакет с флагами [RST, ACK]. После этого утилита считает, что порт закрыт, находит информацию об этом порте в файле `nmap-services` и выводит информацию о нем, однако статус порта будет `closed`.

## 9 Просканировать виртуальную машину Metasploitable2 используя `db_nmap` из состава `metasploit-framework`

Перед началом сканирования необходимо включить `postgresql` и выполнить команду `msfdb init` для инициализации базы данных:

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

```
root@kali:~# msfconsole
```

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

```
msf > db_nmap 192.168.202.2
[*] Nmap: Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-27 18:12 EDT
[*] Nmap: Nmap scan report for 192.168.202.2
[*] Nmap: Host is up (0.00023s latency).
[*] Nmap: Not shown: 977 closed ports
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 21/tcp    open  ftp
[*] Nmap: 22/tcp    open  ssh
[*] Nmap: 23/tcp    open  telnet
[*] Nmap: 25/tcp    open  smtp
[*] Nmap: 53/tcp    open  domain
[*] Nmap: 80/tcp    open  http
[*] Nmap: 111/tcp   open  rpcbind
[*] Nmap: 139/tcp   open  netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds
[*] Nmap: 512/tcp   open  exec
[*] Nmap: 513/tcp   open  login
[*] Nmap: 514/tcp   open  shell
[*] Nmap: 1099/tcp  open  rmiregistry
```

```

[*] Nmap: 1524/tcp open  ingreslock
[*] Nmap: 2049/tcp open  nfs
[*] Nmap: 2121/tcp open  ccproxy-ftp
[*] Nmap: 3306/tcp open  mysql
[*] Nmap: 5432/tcp open  postgresql
[*] Nmap: 5900/tcp open  vnc
[*] Nmap: 6000/tcp open  X11
[*] Nmap: 6667/tcp open  irc
[*] Nmap: 8009/tcp open  ajp13
[*] Nmap: 8180/tcp open  unknown
[*] Nmap: MAC Address: 08:00:27:3B:18:A4 (Oracle VirtualBox virtual NIC)
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 13.39 seconds

```

## 10 Описать работу пяти записей из файла nmap-service-probes

- Рассмотрим распознавание сервиса Samba

```
139/tcp open netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
```

Найдем соответствующую строку в файле

```
match netbios-ssn m=~\0\0\0.\xffSMB\0\0\0\0\x88..\0\0[-\w. ]*\0+@
\x06\0\0\x01\0\x11\x06\0.*(?:[^\0]|[\^_A-Z0-9-]\0)((?:[-\w]\0){2,50})=s
p/Samba smbd/ v/3.X/ i/workgroup: $P(1)/
```

Как и было описано выше, строка состоит из директивы `match`, названия сервиса и шаблона. Шаблон состоит из регулярного выражения и строки для печати. К выражениям взятым в скобках, при печати можно обращаться как к параметрам. Данная директива сопоставляет ответ с регулярным выражением:

```
~\0\0\0.\xffSMB\0\0\0\0\x88..\0\0[-\w. ]*\0+
@\x06\0\0\x01\0\x11\x06\0.*(?:[^\0]|[\^_A-Z0-9-]\0)((?:[-\w]\0){2,50})
```

При этом, выражение подставленное вместо указанного ниже может быть использовано в качестве параметра при печати. Остальные игнорируются т.к. внутри скобок указан знак вопроса.

```
((?:[-\w]\0){2,50})
```

Последняя строка определяет результат при совпадении. Ключ `p` указывает имя продукта, ключ `v` - версию, а `i` - дополнительную информацию. При выводе дополнительной информации также используется вспомогательная функция `P()`, которая удаляет все непечатаемые символы из параметра.

```
p/Samba smbd/ v/3.X/ i/workgroup: $P(1)/
```

- Probe TCP NULL q

Данная директива используется для тестирования TCP портов, ее название NULL. Это связано с тем, что она не передает никакой запрос серверу.

- totalwaitms 6000

Данная строка означает, что максимальное время ожидания ответа равно 6000 миллисекунд.

- Рассмотрим сопоставление для telnet

```
match telnet m|\xff\xfd\x18\xff\xfd \xff\xfd#\xff\xfd'$| p/Linux
telnetd/ o/Linux/ cpe:/o:linux:linux_kernel/a
```

Сравнивает ответ с последовательностью байт 0xff, 0xfd, 0x18, 0xff, 0xfd, 0xff, 0xfd, '#', 0xff, 0xfd, '', конец строки. В случае успеха возвращает имя продукта Linux telnetd, ОС - Linux, cpe (Common platform enumeration) - o:linux:linux-kernel

- Добавление собственных строк:

```
Probe TCP MyServer q|\x02Hi|
rarity 1
ports 22000
match testServer m/^Hello, client \((\w*) ([\d.]*)\) / p/$1/ v/$2/
```

В данных строках мы описываем, какое сообщение будем отправлять (Hi) для идентификации, на какой порт, а так же какой ответ мы будем ожидать (Hello, client, имя сервиса и его версия).

## 11 Описать работу скрипта из состава Nmap

Выбран скрипт mysql-empty-password. Проверки для серверов MySQL с пустым паролем для root или anonymous. Пример использования:

```
nmap -sV -script=mysql-empty-password <target>
```

Результат работы:

```
3306/tcp open  mysql
| mysql-empty-password:
|   anonymous account has empty password
|_  root account has empty password
```

Код скрипта:

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

```

description = [[
Checks for MySQL servers with an empty password for <code>root</code> or
<code>anonymous</code>.
]]

---
-- @output
-- 3306/tcp open  mysql
-- | mysql-empty-password:
-- |   anonymous account has empty password
-- |_  root account has empty password

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

-- Version 0.3
-- Created 01/15/2010 - v0.1 - created by Patrik Karlsson <patrik@cquire.net>
-- Revised 01/23/2010 - v0.2 - revised by Patrik Karlsson, added anonymous
account check
-- Revised 01/23/2010 - v0.3 - revised by Patrik Karlsson, fixed abort bug
due to try of loginrequest

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

action = function( host, port )

local socket = nmap.new_socket()
local result = {}
local users = {"", "root"}

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

for _, v in ipairs( users ) do
local status, response = socket:connect(host, port)
if( not(status) ) then return stdnse.format_output(false, "Failed to
connect to mysql server") end

status, response = mysql.receiveGreeting( socket )
if ( not(status) ) then
stdnse.debug3("%s", SCRIPT_NAME)
socket:close()
return response
end

status, response = mysql.loginRequest( socket, { authversion = "post41",
charset = response.charset }, v, nil, response.salt )

```

```

if response.errorcode == 0 then
table.insert(result, string.format("%s account has empty password",
( v==" and "anonymous" or v ) ) )
if nmap.registry.mysqlusers == nil then
nmap.registry.mysqlusers = {}
end
nmap.registry.mysqlusers[v==" and "anonymous" or v] = ""
end
socket:close()
end

return stdnse.format_output(true, result)

end

```

В скрипте производится управление сокетом - установка и закрытие соединения с хостом, указанным в параметрах. Метод `loginRequest` осуществляет попытку аутентификации, после чего анализируются полученные ответы. В случае, если аутентификация завершилась успешно, значит пароль не установлен.

## 12 Выводы

В ходе данной лабораторной работы были изучены основные возможности утилиты `nmap`: сканирование открытых портов и доступных хостов, определение версий сервисов. Так же были рассмотрены основные служебные файлы, которые используются для работы утилиты `nmap`: файлы конфигураций и скрипты. Так же было рассмотрено расширение `db_nmap`, которая позволяет сохранять результаты сканирования в базу данных.