

Отчет по лабораторной работе №1 : L^AT_EX, Git, GPG

Дедков Сергей

2015

Содержание

1	Цель работы	2
2	Ход работы	2
2.1	Провести поиск активных хостов	2
2.2	Определить открытые порты	2
2.3	Определить версии сервисов	2
2.4	Изучить файлы nmap-services, nmap-os-db, nmapservice-probes	3
2.5	Добавить новую сигнатуру службы в файл nmap-service-probes (для этого создать минимальный tcp server, добиться, чтобы при сканировании nmap указывал для него название и версию)	6
2.6	Сохранить выводы утилиты в формате xml	8
2.7	Исследовать различные этапы и режимы работы nmap с использованием утилиты Wireshark	9
2.8	Просканировать виртуальную машину Metasploitable2 используя nmapdb из состава metasploit-framework	9
2.9	Выбрать пять записей из файла nmap-service-probes и описать их работу. Выбрать один скрипт из состава Nmap и описать его работу	9

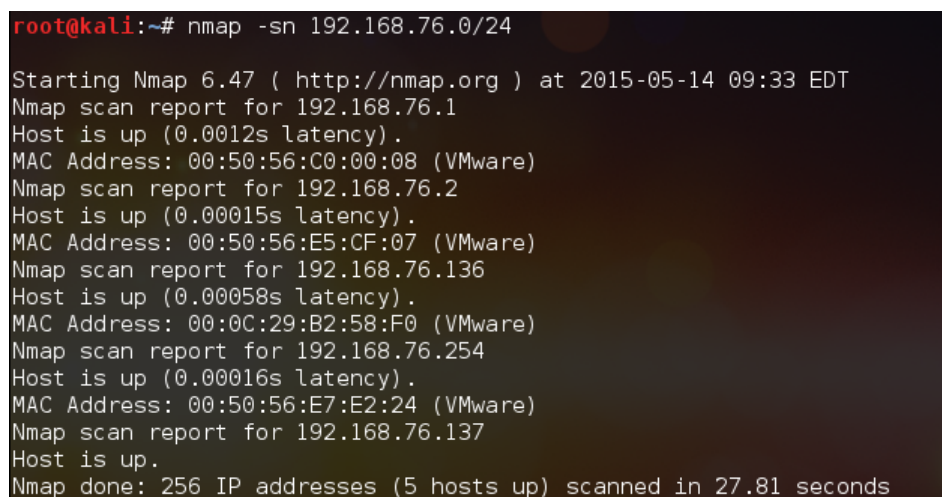
1 Цель работы

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

2 Ход работы

2.1 Провести поиск активных хостов

Так как наш компьютер находится в подсети 192.168.76.0/24 выполним поиск в ней(для определения воспользовались командой ifconfig). Для этого воспользуемся командой nmap 192.168.76.0/24. В результате видим 5 хостов, один из них - виртуальная машина с запущенной metasploit, а именно с ip адресом 192.168.76.136. См. рисунок 1.



```
root@kali:~# nmap -sn 192.168.76.0/24
Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-14 09:33 EDT
Nmap scan report for 192.168.76.1
Host is up (0.0012s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.76.2
Host is up (0.00015s latency).
MAC Address: 00:50:56:E5:CF:07 (VMware)
Nmap scan report for 192.168.76.136
Host is up (0.00058s latency).
MAC Address: 00:0C:29:B2:58:F0 (VMware)
Nmap scan report for 192.168.76.254
Host is up (0.00016s latency).
MAC Address: 00:50:56:E7:E2:24 (VMware)
Nmap scan report for 192.168.76.137
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 27.81 seconds
```

Рис. 1: Поиск хостов

2.2 Определить открытые порты

Для определения открытых портов достаточно просто ввести nmap 192.168.76.136 (сканируются порты до 1024). Или же воспользоваться опцией -p, например nmap -p "*"192.168.76.136. Данной командой просканируются все порты, если необходимо задать диапазон достаточно указать его вместо "*". После ввода команды увидим результат, как на рисунке 2.

2.3 Определить версии сервисов

Чтобы определить версии сервисов необходимо воспользоваться командой nmap с ключем sV следующим образом: nmap -sV 192.168.76.136. На

```

root@kali:~# nmap 192.168.76.136

Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-14 11:10
Nmap scan report for 192.168.76.136
Host is up (0.00026s 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: 00:0C:29:B2:58:F0 (VMware)

```

Рис. 2: Поиск портов

рисунке 3 можно увидеть результат.

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

Рассмотрим файл nmap-services. Для этого введем команду

```
vim /usr/share/nmap/nmap-services.
```

Файл служит для быстрого поиска, напрмер с ключем -F. В файле в каждой строчке задаются сервисное название или сокращение, число порта и протокол, определенный разделом, частота порта мера того, как часто порт был найдет открытым во время сканирования. Пример файла можно увидеть на рисунке 4.

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

```

root@kali:~# nmap -sV 192.168.76.136

Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-14 11:26 EDT
Nmap scan report for 192.168.76.136
Host is up (0.00023s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login
514/tcp   open  tcpwrapped
1099/tcp  open  rmiregistry  GNU Classpath grmiregistry
1524/tcp  open  shell        Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          Unreal ircd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:B2:58:F0 (VMware)
Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploit; Unix, Linux; CPE: cpe:/o:linux:linux_kernel

```

Рис. 3: Определение версий сервисов

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

Синтаксис:

- Probe <protocol> <probenamе> <probesendstring> - директива probe (тест) - указывает nmap, какие данные отправлять в процессе определения служб
- match <service> <pattern> <productname> <version> <device> <h?????> <info> <OS> - указывает nmap на то, как точно определить службу, используя полученный ответ на запрос, отправленный предыдущей директивой probe. Эта директива используется

```
# Fields in this file are: Service name, portnum/protocol, open-frequency, optional comments
#
tcpmux 1/tcp 0.001995 # TCP Port Service Multiplexer [rfc-1078]
tcpmux 1/udp 0.001236 # TCP Port Service Multiplexer
compressnet 2/tcp 0.000013 # Management Utility
compressnet 2/udp 0.001845 # Management Utility
compressnet 3/tcp 0.001242 # Compression Process
compressnet 3/udp 0.001532 # Compression Process
unknown 4/tcp 0.000477
rje 5/udp 0.000593 # Remote Job Entry
unknown 6/tcp 0.000502
echo 7/sctp 0.000000
echo 7/tcp 0.004855
echo 7/udp 0.024679
unknown 8/tcp 0.000013
discard 9/sctp 0.000000 # sink null
discard 9/tcp 0.003764 # sink null
```

Рис. 4: Файл nmap-services

```
MatchPoints
SEQ(SP=25%GCD=75%ISR=25%TI=100%CI=50%II=100%SS=80%TS=100)
OPS(O1=20%O2=20%O3=20%O4=20%O5=20%O6=20)
WIN(W1=15%W2=15%W3=15%W4=15%W5=15%W6=15)
ECN(R=100%DF=20%T=15%TG=15%W=15%O=15%CC=100%Q=20)
T1(R=100%DF=20%T=15%TG=15%S=20%A=20%F=30%RD=20%Q=20)
T2(R=80%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T3(R=80%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T4(R=100%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T5(R=100%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T6(R=100%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T7(R=80%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
U1(R=50%DF=20%T=15%TG=15%IPL=100%UN=100%RIPL=100%RID=100%RIPCK=100%RUO
IE(R=50%DFI=40%T=15%TG=15%CD=100)
```

Рис. 5: Файл nmap-os-db

в случае, когда полученный ответ полностью совпадает с шаблоном. При этом тестирование порта считается законченным, а при помощи дополнительных спецификаторов nmap строит отчет о названии приложения, номере версии и дополнительной информации, полученной в ходе проверки

- `softmatch <service> <pattern> <productname> <version> <device> <h?????> <info> <OS>` - имеет аналогичный формат директиве `match`. Основное отличие заключается в том, что после совпадения принятого ответа с одним из шаблонов `softmatch`, тестирование будет продолжено с использованием только тех тестов, которые относятся к определенной шаблоном службе. Тестирование порта будет идти до тех пор, пока не будет найдено строгое соответствие (`match`) или не закончатся все тесты для данной службы

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

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

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

```
match tcp-server m|^111| v/1.0.X/ p/Dedkov S.V./ i/It's works /
```

Код сервера:

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

namespace ExampleTcpListener_Console
{
    class ExampleTcpListener
    {
        static void Main(string[] args)
        {
            TcpListener server = null;
            try
            {
                int MaxThreadsCount = Environment.ProcessorCount * 4;
                Console.WriteLine(MaxThreadsCount.ToString());
                ThreadPool.SetMaxThreads(MaxThreadsCount, MaxThreadsCount);
                ThreadPool.SetMinThreads(2, 2);

                Int32 port = 9596;
```

```

        IPAddress localAddr = IPAddress.Parse("192.168.137.1");
        int counter = 0;
        server = new TcpListener(localAddr, port);

        server.Start();

        while (true)
        {

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

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

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

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

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

    TcpClient client = client_obj as TcpClient;

    data = null;

    NetworkStream stream = client.GetStream();

    int i;

    data = "111";

```

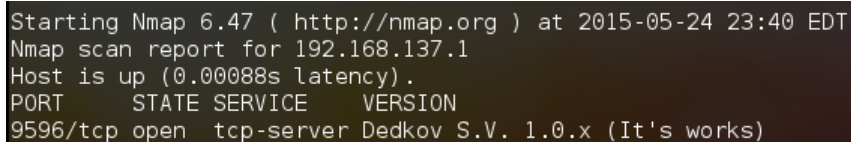
```

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

        client.Close();
    }
}

```

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



```

Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-24 23:40 EDT
Nmap scan report for 192.168.137.1
Host is up (0.00088s latency).
PORT      STATE SERVICE      VERSION
9596/tcp  open  tcp-server  Dedkov S.V. 1.0.x (It's works)

```

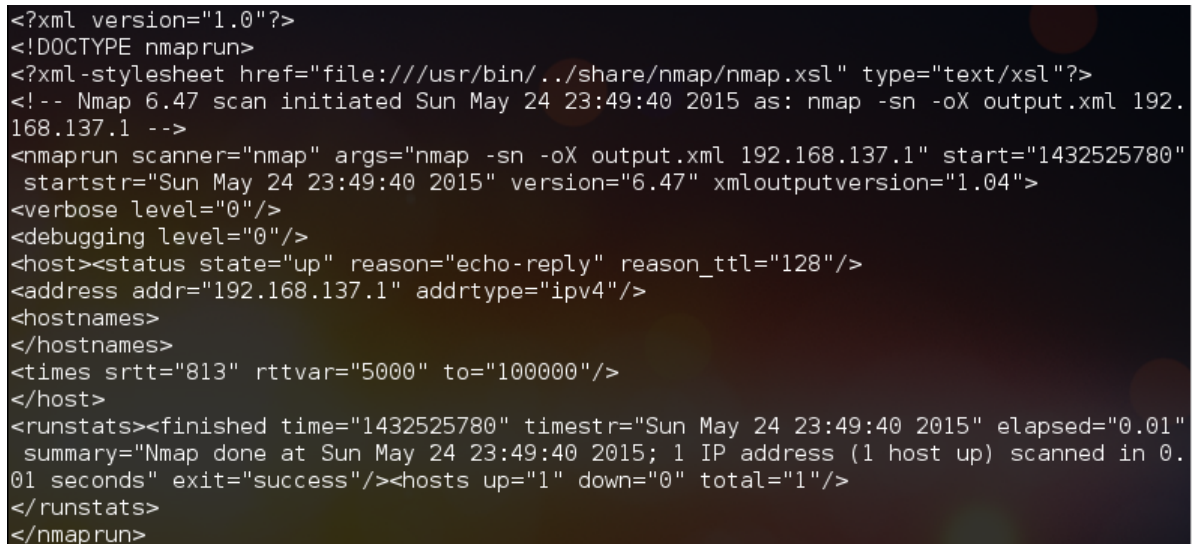
Рис. 6: Вывод информации о сервисе

2.6 Сохранить выводы утилиты в формате xml

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

```
nmap -sn -oX output.xml 192.168.137.1
```

Результат можно увидеть на рисунке 7:



```

<?xml version="1.0"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/bin/./share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 6.47 scan initiated Sun May 24 23:49:40 2015 as: nmap -sn -oX output.xml 192.168.137.1 -->
<nmaprun scanner="nmap" args="nmap -sn -oX output.xml 192.168.137.1" start="1432525780" startstr="Sun May 24 23:49:40 2015" version="6.47" xmloutputversion="1.04">
  <verbose level="0"/>
  <debugging level="0"/>
  <host><status state="up" reason="echo-reply" reason_ttl="128"/>
  <address addr="192.168.137.1" addrtype="ipv4"/>
  <hostnames>
  </hostnames>
  <times srtt="813" rttvar="5000" to="100000"/>
</host>
<runstats><finished time="1432525780" timestr="Sun May 24 23:49:40 2015" elapsed="0.01" summary="Nmap done at Sun May 24 23:49:40 2015; 1 IP address (1 host up) scanned in 0.01 seconds" exit="success"/><hosts up="1" down="0" total="1"/>
</runstats>
</nmaprun>

```

Рис. 7: output.xml

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

Запустим утилиту Wireshark, выберем интерфейс eth0 и нажмем Start. Будет произведено сканирование сети и вывод передаваемых по сети пакетов. Пропингуем с ОС Metasploitable2 основную ОС и проверим, будет ли утилита Wireshark видеть подобную активность сети? Как видно из рисунка 14, утилита отображает информацию об активности в сети. Так же можно посмотреть информацию о соединении и передаваемых пакетах, а в некоторых случаях и перехватить cookie.

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

Запустив metasploit, мы можем воспользоваться командой nmap_db

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

- Первая запись

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

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

Рис. 8: Запись 1

- Вторая запись

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

- Третья запись

```
match lc-server m|^S\xf5\xc6\x1a{| p/1C:Enterprise business management server/
```

Рис. 9: Запись 2

В данной записи на сервер отправляется запрос по протоколу TCP в котором

```
#####NEXT PROBE#####  
Probe TCP GetRequest q|GET / HTTP/1.0\r\n\r\n|  
rarity 1  
ports 1,70,79,80-85,88,113,139,143,280,497,505,514,515,540,554,591,620,631,783,888,898,  
900,901,993,995,1026,1080,1042,1214,1220,1234,1311,1314,1344,1503,1610,1611,1830,1900,2  
001,2002,2030,2064,2160,2306,2396,2525,2715,2869,3000,3002,3052,3128,3280,3372,3531,368  
9,3872,4000,4444,4567,4660,4711,5000,5427,5060,5222,5269,5280,5432,5800-5803,5900,6103,  
6346,6544,6600,6699,6969,7002,7007,7070,7100,7402,7776,8000-8010,8080-8085,8088,8118,81  
81,8443,8880-8888,9000,9001,9030,9050,9080,9090,9999,10000,10001,10005,11371,13013,1366  
6,13722,14534,15000,17988,18264,31337,40193,50000,55555  
sslports 443,4443
```

Рис. 10: Запись 3