САНКТ-ПЕТЕРБУРГСИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Кафедра Компьютерных Систем и Программных Технологий

ОТЧЕТ

по лабораторной работе №4

Тема: «Утилита для исследования сети и сканер портов Nmap»

Дисциплина: «Методы и средства защиты информации»

Выполнил: студент гр. 53501/2
Майоров А.П.
Ломтев Д.С.
Хазан Н.А.

Преподаватель
Вылегжанина К.Д.

Санкт-Петербург
2015

Содержание

1 Задание

а) Проверить поиск активных хостов

б) Определить открытые порты

в) Определить версии сервисов

г) Изучить файлы nmap-services, nmap-os-db, nmap-service-probes

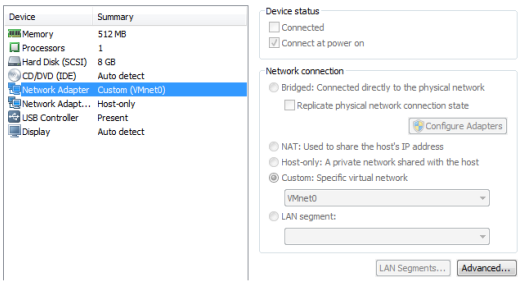
д) Добавить новую сигнатуру службы в файл nmap-service-probes (для этого создать минимальный tcp server, добиться, чтобы при сканировании nmap указывал для него название и версию)

е) Сохранить вывод утилиты в формате xml

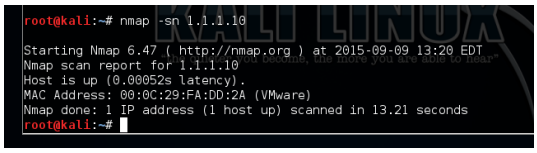
ж) Исследовать различные этапы и режимы работы nmap с использованием утилиты Wireshark

2 Выполнение

2.1 Начальные настройки

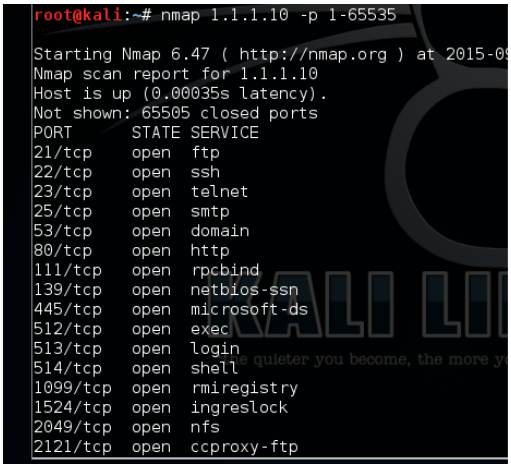


2.2 Проверить поиск активных хостов



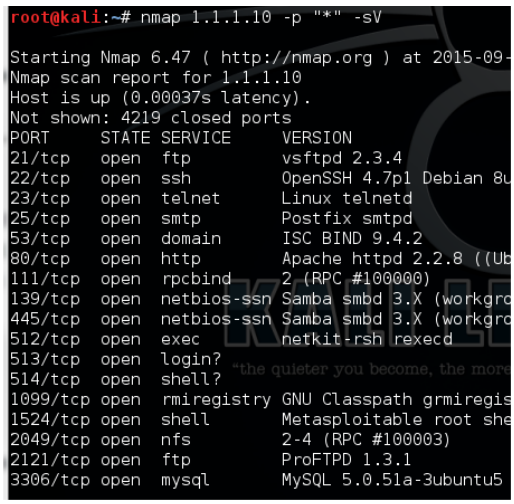
2.3 Определить открытые порты

Для сканирования открытых портов предварительно запустим Metaspoit. Его адрес в сети — 1.1.1.10. Просканируем весь диапазон его портов:



2.4 Определить версии сервисов

Вновь будем сканировать Metasploit:



2.5 Изучить файлы nmap-services, nmap-os-db, nmap-service-probes

а) *nmap-services*

Структура данного файла представлена в виде таблицы с тремя колонками:

1) Имя сервиса
2) Номер и тип порта
3) Как часто данный порт встречается.

Фрагмент файла:

```
root@kali:~# cat /usr/share/nmap/nmap-services | more
# THIS FILE IS GENERATED AUTOMATICALLY FROM A MASTER - DO NOT EDIT.
...
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
...
```

б) *nmap-os-db*

Содержит набор отпечатков для каждой ОС представленных различными директивами. Генерируются шесть пакетов специального вида, которые посылаются целевой машине с перерывом в 100 мс. Для получения результатов теста используются директивы:

— SEQ — результаты последовательного анализа
— OPS — флаги пакетов, полученных в ответ
— WIN — размер окон
— T1 — данные касательно ответа на первый пакет

Также отпечаток может содержать директивы T2–T7 посылающие пакеты различного вида. Например, без указания флагов, с указанием флагов SYN, FIN, URG, PSH; а также пакеты другого вида.
Кроме того, существует возможность тестировать указанный хост с помощью UDP пакетов (директива U1), а также множество других возможностей.
Модификация данного файла достаточно сложна и, как правило, производиться крайне редко.

```
  root@kali:~# cat /usr/share/nmap/nmap-os-db | more
# Nmap OS Fingerprinting 2nd Generation DB. -*- mode: fundamental; -*-
...
root@kali:~# cat /usr/share/nmap/nmap-os-db | more
# Nmap OS Fingerprinting 2nd Generation DB. -*- mode: fundamental; -*-
...
```

в) *nmap-service-probes* Основные директивы, используемые в файле.

- Probe <протокол> <имя> q«посылаемая строка>"
  Указывает nmap, какие данные отправлять в процессе определения служб
- match <название сервиса> <шаблон> [<версия>]
  Указывает nmap на то, как точно определить службу, используя полученный ответ на запрос.
- softmatch <название сервиса> <шаблон> [<версия>]
  Аналогичен match, но не прекращает сопоставление в случае успеха.
- totalwaitms <миллисекунды>
  Время ожидания

```
root@kali:~# cat /usr/share/nmap/nmap-service-probes | more
# Nmap service detection probe list -*- mode: fundamental; -*-
...
match 1c-server m|^S\xf5\xc6\x1a{| p/1C:Enterprise business management server/
...
```

2.6 Добавить новую сигнатуру службы в файл nmap-service-probes

Напишем простой tcp-сервер, который просто ждет подключения клиента и отправляет ему сообщение.

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>

int main(int argc, char**argv)
{
   int listenfd;
   int connfd;
   int msgsize;

   struct   sockaddr_in servaddr;
   struct   sockaddr_in cliaddr;

   socklen_t clilen;
   pid_t    childpid;
   char     mesg[1000];

   listenfd = socket(AF_INET, SOCK_STREAM, 0);
   bzero(&servaddr, sizeof(servaddr));

   servaddr.sin_family = AF_INET;
   servaddr.sin_addr.s_addr = htonl(INADDR_ANY);       /* ADDR: ANY! */
   servaddr.sin_port = htons(2404);                    /* PORT: 2404 */
   bind(listenfd, (struct sockaddr *)&servaddr, sizeof(servaddr));

   listen(listenfd, 1024);

   for(;;)
   {
      clilen = sizeof(cliaddr);
      connfd = accept(listenfd, (struct sockaddr *)&cliaddr, &clilen);

      if ((childpid = fork()) == 0)
      {
         close (listenfd);
```

```
        for(;;)
        {
           msgsize = recvfrom(connfd, mesg, 1000, 0, (struct sockaddr
               *)&cliaddr, &clilen);
           if (!strncmp(mesg,"version", 7))
           {
               strcpy(mesg, "1.0.0\n");
               msgsize = strlen(mesg);
           }
           sendto(connfd, mesg, msgsize, 0, (struct sockaddr *)&cliaddr,
               sizeof(cliaddr));

        }

     }
     close(connfd);
  }
}
```

Впишем следующие строчки в файл nmap-service-probes:

```
# Simple TSP server.
Probe TCP simple-tcp-server-ver q|version\r\n|
rarity 9
ports 2404
match stcps m|^1\.0\.0$| p/Simple TCP Server/ v/1.0.0-3/
```

В результате получим:

```
root@kali:~# nmap -sV -p 2404 1.1.1.10
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-06 15:52 EDT
Nmap scan report for eth0 (1.1.1.11)
Host is up (0.00020s latency).
PORT    STATE SERVICE VERSION
2404/tcp open stcps  Simple TCP Server 1.0.0-3
MAC Address: B8:E8:56:10:2B:BE (Apple)
```

2.7 Сохранить вывод утилиты в формате xml

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

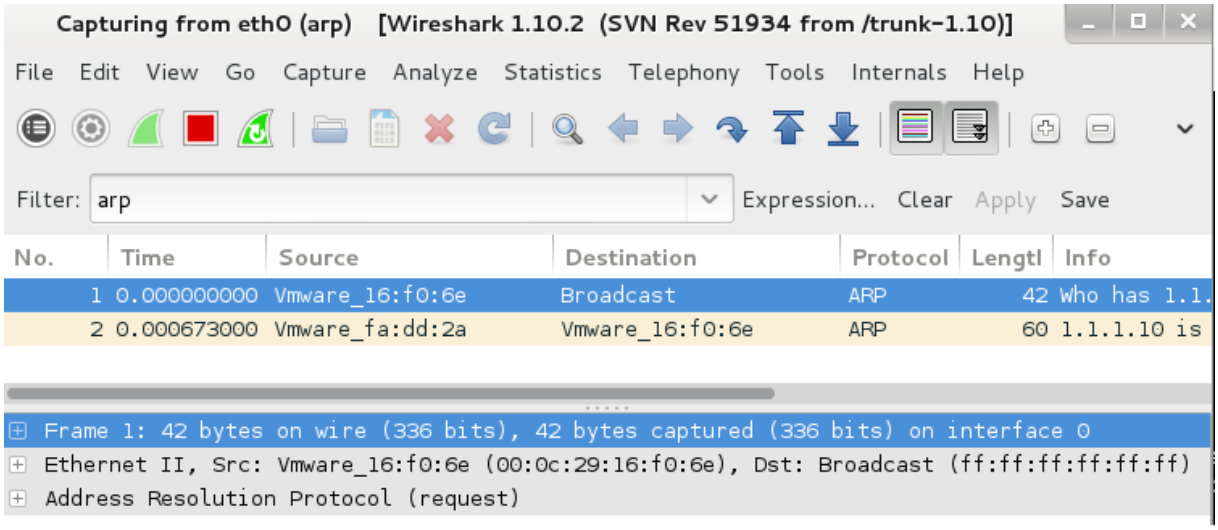
```
root@kali:~# nmap -sn -oX output.xml 1.1.1.10
```

После этого просмотрим содержимое файла:

```
root@kali:~# cat output.xml
<?xml version="1.0"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/bin/../share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 6.47 scan initiated Sat Jun 6 19:15:55 2015 as: nmap -sn -oX
    output.xml 1.1.1.10 -->
<nmaprun scanner="nmap" args="nmap -sn -oX output.xml 192.168.1.37"
    start="1433632555" startstr="Sat Jun 6 19:15:55 2015" version="6.47"
    xmloutputversion="1.04">
<verbose level="0"/>
...
```

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

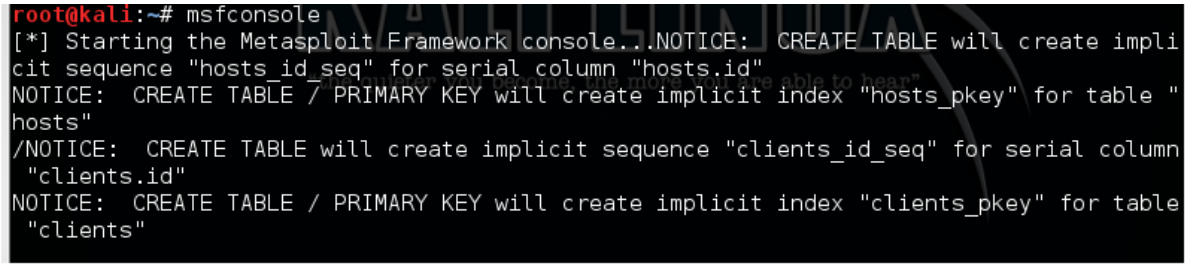
Сканирование локальной сети проводится при помощи ARP запросов. Чтобы анализировать пакеты, порождаемые утилитой nmap в wireshark в фильтре напишем «arp», чтобы ловить только arp пакеты, а в терминале напишем какую-нибудь nmap команду:



2.9 Просканировать виртуальную машину Metasploitable2 используя db nmap из состава metaspoit-framework

Сначала включим postgresql и metasploit:

Перейдем в консоль metasploit framework console, будем использовать любую команду nmap, но вместо него будем использовать db nmap. Все результаты будут занесены в базу данных. Просканируем Metasploit:



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

2.10.1 Первая запись

```
Probe TCP NULL q||
totalwaitms 6000
```

Запись теста с отправкой null-запроса. В данной записи будет отправляться пустой запрос по протоколу TCP. С ожиданием ответа в 6 секунд.

2.10.2 Вторая запись

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

Если пользователь будет использовать nmap с ключем -sV, и после отправки нулевого теста с сервера приедет выражение mSxf5xc6x1a, тогда в колонке он увидит наименование сервиса 1c-server и 1C:Enterprise business management server.

2.10.3 Третья запись

```
Probe UDP AndroMouse q|AMSNIFF|
rarity 9
ports 8888
```

Протокол — UDP, название — AndroMouse, посылаемая строка — AMSNIFF. rarity указывает на ожидание ответа результатов от этого теста, в данном случае она очень маленькая — 9. Взаимодействие происходит по порту 8888.

2.10.4 Четвертая запись

```
Probe UDP FreelancerStatus
   q|\x00\x02\xf1\x26\x01\x26\xf0\x90\xa6\xf0\x26\x57\x4e\xac\xa0\xec\xf8\x68\xe4\x8d\x21|
rarity 9
ports 2302
```

Протокол — UDP, название — FreelancerStatus, посылаемая строка — сообщение в определенной кодировке (скорее всего оно отображается по-другому). rarity указывает на ожидание ответа результатов от этого теста, в данном случае она очень маленькая — 9. Взаимодействие происходит по порту 8888.

2.10.5 Пятая запись

```
# Offset Type   Value                Comment
# 0-1    uint16 0xBEF4               Class: connection
# 2-3    uint16 0x0004               Type: login reply
```

Комментарии, однострочные комментарии пишутся со знаком «#» в начале строки.

3 Выводы

В ходе выполнения данной лабораторной работы были изучены основные возможности nmap, а именно, научился определять активные активные хосты, сканировать порты, определять версии сервисов. Были изучены основные файлы, используемые для определения версий сервисов. Изучена вохможность сохранения результатов в xml файл.

Кроме этого была рассмотрена версия db nmap, которая сохраняет результаты в БД. Данная возможность позволяет без использования сети в дальнейшем использовать информацию о конфигурации сети.