Лабораторная работа. Metasploit

**Цель**: Изучение методов автоматизированного пентестинга. Приобретение навыков работы с Metasploit.

# Основные теоретические сведения

Тестирование на проникновение позволяет ответить на вопрос, как кто-то со злым умыслом может вмешаться в вашу сеть. Используя инструменты пентеста, «белые хакеры» и профессионалы в области безопасности могут на любом этапе разработки или развертывания исследовать сети и приложения на предмет недостатков и уязвимостей путем взлома системы.

Одним из таких средств пентеста является проект Metasploit. Этот фреймворк с открытым исходным кодом, созданный на Ruby, позволяет проводить тестирование с помощью командной строки или графического интерфейса. Его можно расширить, создавая собственные надстройки с поддержкой нескольких языков.

## Что такое Metasploit Framework и как он используется?

Metasploit Framework — это мощнейший инструмент, который могут использовать как киберпреступники, так и «белые хакеры» и специалисты по проникновению для исследования уязвимостей в сетях и на серверах. Поскольку это фреймворк с открытым исходным кодом, его можно легко настроить и использовать на большинстве операционных систем.

С помощью Metasploit пентестеры могут использовать готовый или создать пользовательский код и вводить его в сеть для поиска слабых мест. В качестве еще одного способа поиска угроз, после идентификации и документирования недостатков, эту информацию можно использовать для устранения системных недостатков и определения приоритетности решений.

## Краткая история Metasploit

Проект Metasploit был создан на языке Perl в 2003 году Эйч Ди Муром (H.D. Moore) при содействии основного разработчика Мэтта Миллера для использования в качестве портативного сетевого инструмента. Он был полностью переведен на язык Ruby к 2007 году, а в 2009 году лицензию приобрела Rapid7, и теперь этот инструмент остается частью ассортимента этой бостонской компании, специализирующейся на разработке систем обнаружения вторжений и инструментов эксплуатации уязвимостей систем удаленного доступа.

Этот фреймворк стал основным инструментом разработки эксплойтов и устранения уязвимостей. До Metasploit пентестерам приходилось выполнять все проверки вручную, используя различные инструменты, которые могли поддерживать или не поддерживать тестируемую платформу, а также вручную писать собственный код и внедрять его в сети. Дистанционное тестирование было чем-то экстраординарным, и это ограничивало работу специалиста по безопасности собственным регионом и местными компаниями, а организациям приходилось тратить целые состояния на собственных ИТ-консультантов или специалистов по безопасности.

## Кто использует Metasploit?

Благодаря широкому спектру применений и доступному открытому исходному коду Metasploit используется самыми разными людьми, от профессионалов кибербезопасности до хакеров. Metasploit полезен для всех, кому нужен простой в установке и надежный инструмент, выполняющий свою работу независимо от платформы или языка. Это программное обеспечение пользуется популярностью у хакеров и широко доступно, что мотивирует специалистов по безопасности изучать платформу Metasploit, даже если сами они ей не пользуются.

Современная версия Metasploit содержит свыше 1677 эксплойтов для более 25 платформ, включая Android, PHP, Python, Java, Cisco и другие. Фреймворк также содержит около 500 единиц информационного наполнения («пейлоад»), среди которых вы найдёте:

**Пейлоады для командной оболочки** — позволяют пользователям запускать сценарии или случайные команды на хосте.

**Динамические пейлады** — позволяют тестировщикам генерировать уникальные пейлоады для обхода антивирусного программного обеспечения.

**Пейлоады Meterpreter** — позволяют пользователям перехватывать управление монитором устройства с помощью контроллера видеопамяти, захватывать сеансы, а также скачивать или загружать файлы.

**Статические пейлоады** — позволяют устанавливать переадресацию портов и обмен данными между сетями.

Все, что вам нужно для использования Metasploit после его установки, — это получить информацию о цели либо путем сканирования портов, либо путем получения цифрового отпечатка операционной системы, либо с помощью сканера уязвимостей, чтобы найти способ проникнуть в сеть. Затем остается просто выбрать эксплойт и полезную нагрузку. В этом контексте эксплойт — это средство для выявления слабости в вашей сети или системе и использования этой уязвимости для получения доступа.

Платформа состоит из различных моделей и интерфейсов, которые включают: **msfconsole** на базе библиотеки curses, **msfcli** для всех функций msf из терминала или командной строки, **Armitag** — инструмент с графическим интерфейсом на Java, который используется для интеграции с MSF, а также веб-интерфейс сообщества Metasploit, поддерживающий удаленный пентест.

## Алгоритм работы с Metasploit Framework

Работа с модулем состоит из следующих шагов:

* Поиск подходящего модуля с помощью команды **search**.
* Выбор модуля с помощью команды **use**.
* Просмотр основной информации по модулю командой **info**
* Просмотр настроек выбранного модуля с помощью команд **show options** (продвинутые настройки — **show advanced**).
* Установка конкретной опции с помощью команды **set**. Самыми часто задаваемыми опциями являются **RHOSTS** и **LHOSTS**. В первом случае можно задать только один адрес цели, а во втором – множество.
* Установка подробного вывода с помощью команды **set verbose true** (если любопытно знать, что происходит).
* Запуск модуля с помощью команды **run**.

Практическая работа №4. Metasploit.

Часть 1. Сбор информации.

1. Для выполнения работы устанавливаем 2 виртуальные машины. Для атакующей машины используем готовую сборку Linux – Kali Linux, а для атакуемой машины используем специальную сборку Виртуальной машины metasploitable 2.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

1. Необходимо настроить сетевые адаптеры данных машин для работы в режиме сеть Nat. Возьмём значение изолированной сети “Nat” из прошлой лабораторной работы. Затем выбираем тип подключения «Сеть NAT» на обеих машинах:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

1. Зафиксируем сетевые настройки данных машин:

Первая машинa – kali-linux IP адрес – 192.168.10.12/27, MAC адрес – 08:00:27:c7:e1:36.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Вторая машина – Metasploitable. IP адрес – 192.168.10.13/27, MAC адрес – 08:00:27:9d:5e:1d.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Далее работаем в виртуальной машине Kali

1. Командой «msfconsole» в терминале Kali запускаем консоль Metasploit Framework:

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

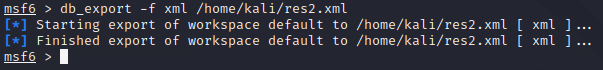
Автоматически созданное описание

1. Сканируем атакуемую систему. Для сканирования используем оболочку nmap встроенную в meta:

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

1. Результат сканирования нужно экспортировать в файл:



1. Далее полученный файл можно открыть для просмотра в OpenOffice, LibreOffice или MS Excel. Далее будем его использовать для поиска уязвимых сервисов.

Часть 2. Подбор паролей.

1. Проведем подбор пароль для сервиса СУБД PostgreSQL:

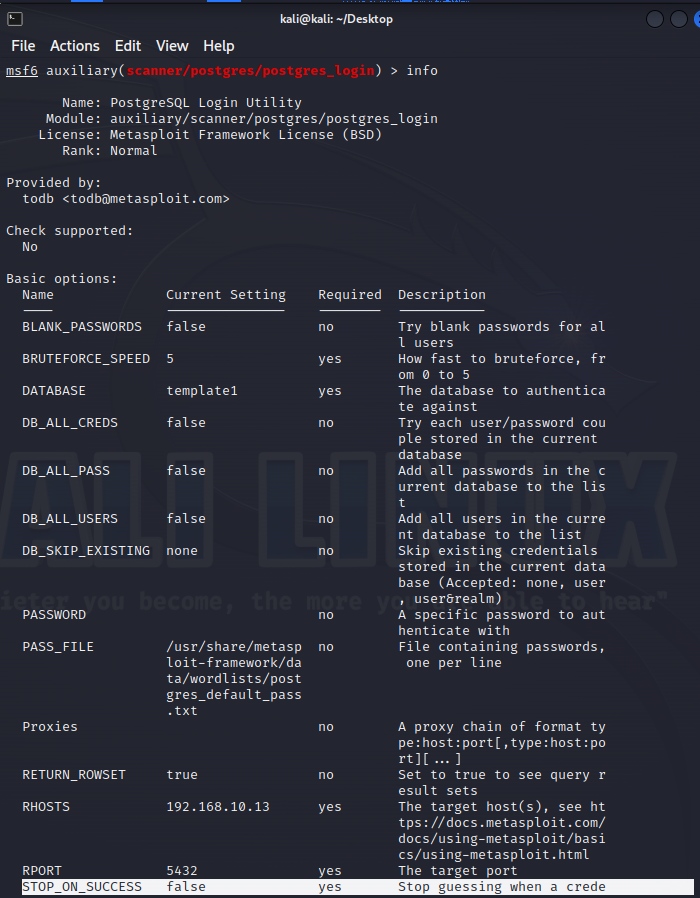


1. Укажем хост для тестирования и запустим сканирование:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Через команду info уточняем опцию, которая позволит остановить подбор паролей до первого удачного подбора:

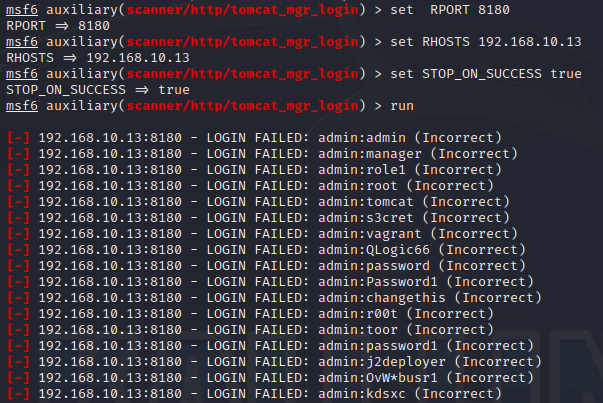


Устанавливаем эту опцию и вновь проводим сканирование:

Изображение выглядит как текст, снимок экрана, Шрифт

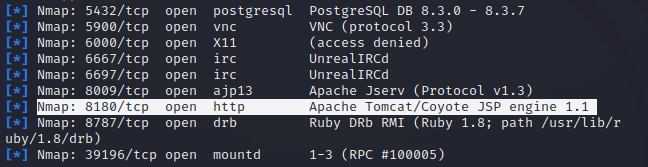
Автоматически созданное описание

1. Аналогичным образом проводим сканирование сервиса tomcat:



Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Из результата выполненного раннее в пункте 5 сканирования nmap можно понять, что сервис tomcat находится на порту 8180/tcp.

Часть 3. Эксплуатация и проведение атак.

1. Проверяем наличие готовой атаки в базе на сервис vsftpd:

Изображение выглядит как текст, Мультимедийное программное обеспечение, программное обеспечение, снимок экрана

Автоматически созданное описание

1. Выбираем для тестирования найденный модуль:



1. Проверяем, есть ли в базе готовые варианты полезной нагрузки для эксплуатации уязвимости:

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

1. Выбераем эту нагрузку для использования:

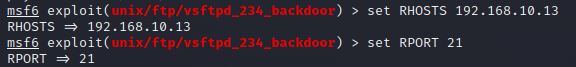


1. Узнаем, что сервис vsftpd запущен на порту 21 из nmap сканирования:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Устанавливаем параметр RHOSTS и RPORT:



И запускаем атаку:

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

1. Проверяем, что мы действительно получили доступ к удаленной машине командой «ip a»:

Изображение выглядит как текст, снимок экрана, Шрифт

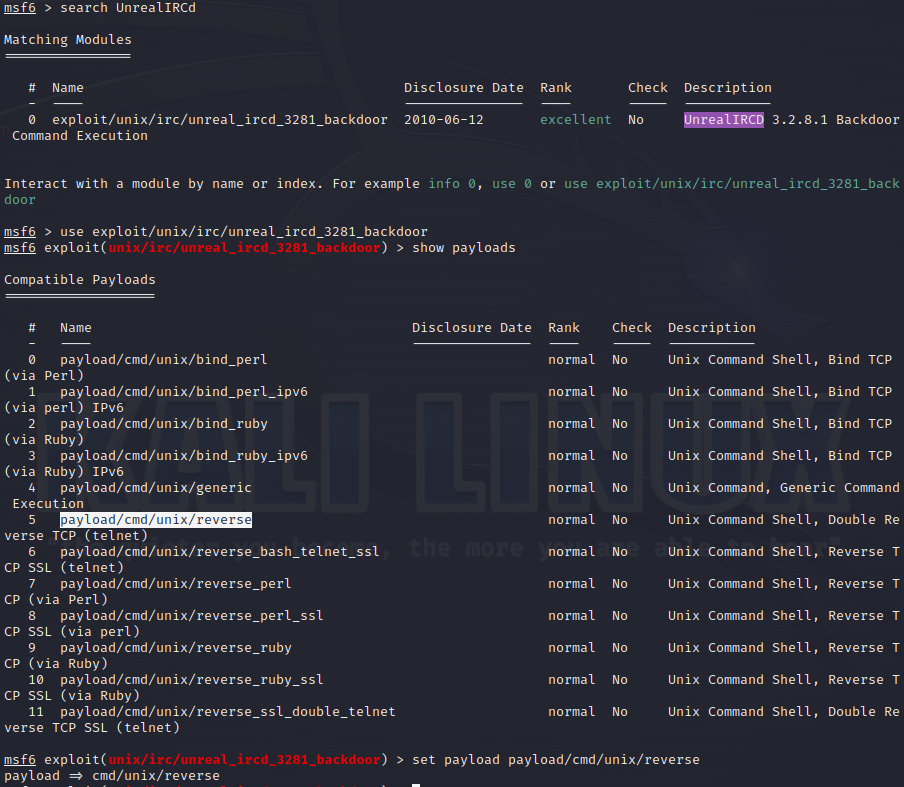
Автоматически созданное описание

1. Аналогичным образом проводим атаку на сервис «UnrealIRCd»:
2. Ишем порты для сервиса UnrealIRCd:

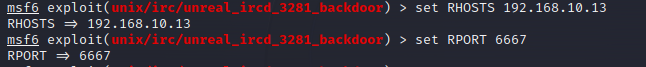
Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

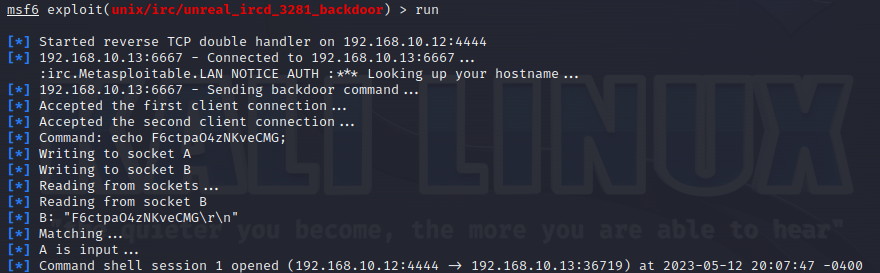
1. Выполняем те же действия, что и в пунктах 12-15:



1. Устанавливаем параметры:



1. Запускаем атаку:



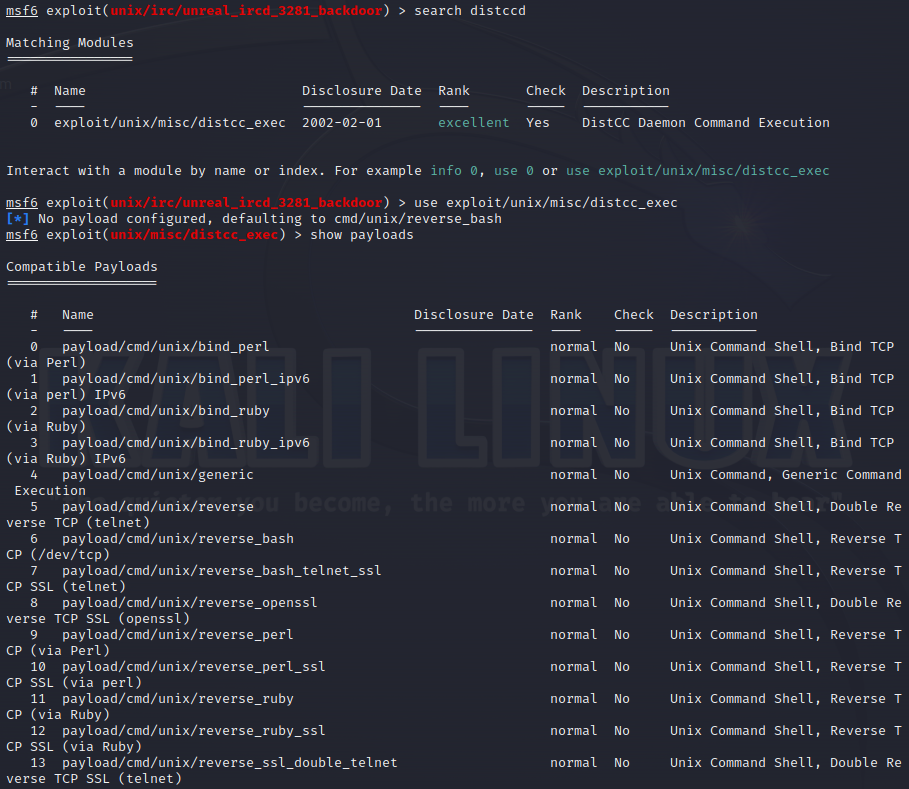
1. Убеждаемся, что действительно получен доступ к удаленной машине:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Часть 4. Самостоятельная-соревновательная.

Рассмотрим сервис distccd. Это 3632 порт на атакуемой машине. Повторяем все выполненные раннее действия:



Изображение выглядит как текст, снимок экрана, Шрифт, меню

Автоматически созданное описание

По результатам данной атаки, удалось успешно эксплуатировать уязвимость distccd\_exec в сервисе distccd, что позволило вновь получить удаленный доступ к атакуемой машине.