Министерство науки и высшего образования Российской Федерации федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

**Основы кибербезопасности**

Практическая работа №4

**Выполнил**:

студент группы К34211

Фисенко Максим Вячеславович

**Проверил**:

преподаватель практики, КТН

Назаров Михаил Сергеевич

Санкт-Петербург

2024

**Оглавление**

[**Введение** 3](#_Toc184923332)

[**Содержание отчета** 4](#_Toc184923333)

[Задание 1. Работа в лаборатории для тестирования и поиска уязвимостей 4](#_Toc184923334)

[Задание 2. Работа с инструментом *Nmap* 6](#_Toc184923335)

[Опция *-A* 7](#_Toc184923336)

[Опция *-sT* 7](#_Toc184923337)

[Опция -*sS* 8](#_Toc184923338)

[Опция *-sN* 8](#_Toc184923339)

[Опции *-sM, -sA, -sW* 8](#_Toc184923340)

[Опция *-sO* 9](#_Toc184923341)

[Задание 3. Установка Metasploit 10](#_Toc184923342)

[**Вывод по работе** 12](#_Toc184923343)

# **Введение**

В данной практической работе было необходимо изучить типовой алгоритм работы с нарушителями информационных систем. В ходе практической работы необходимо было приобрести практические навыки по использованию инструментов сканирования информационных систем, а также научиться идентифицировать узлы в информационной системе.

Для выполнения данной практической работы использовался персональный компьютер на операционной системе *Windows* с подключенной через *WSL* дистрибутивом *Linux Ubuntu*.

# **Содержание отчета**

## **Задание 1. Работа в лаборатории для тестирования и поиска уязвимостей**

Первым делом для выполнения данной практической работы необходимо было создать и запустить все необходимые *Docker*-контейнеры, а также проверить их связность. Работа велась в сети *pentest*, созданной в ходе выполнения предыдущей практической работы. В первую очередь был создан контейнер *metasploitable1* (рисунок 1). Аналогичным способом был создан контейнер *metasploitable2*.

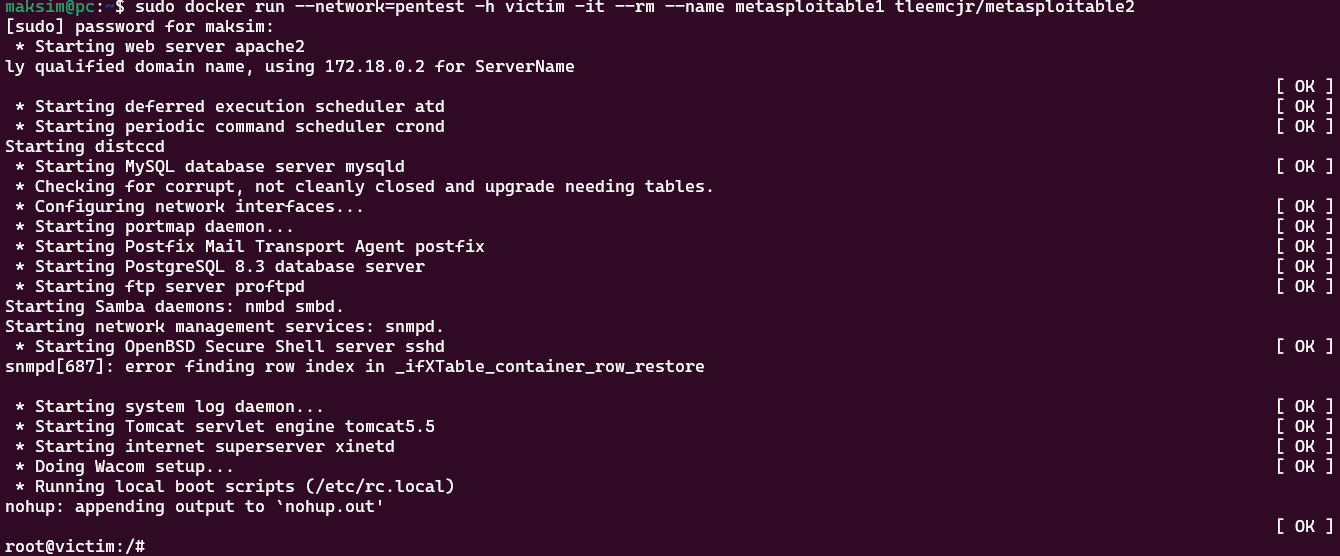


Рисунок 1 – Запуска контейнера *metasploitable1*

Затем было необходимо создать контейнер, имитирующий нарушителя информационной системы, на образе *kalilinux/kali-rolling*. В этот контейнер была установлена утилита *ping*, необходимая для работы с сетью.

Далее было необходимо узнать *ip*-адреса всех созданных контейнеров, что и было сделано с помощью утилиты *ifconfig*. Так, на рисунке 2 видно, что *ip*-адрес контейнера *metasploitable1* – 172.18.0.2. Адреса контейнеров *metasploitable2* и *kalibox* – 172.18.0.3 и 172.18.0.4 соответственно.

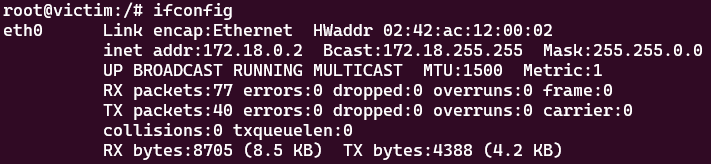


Рисунок 2 – *Ip*-адрес контейнера *metasploitable1*

После нахождения *ip*-адресов контейнеров с помощью утилиты *ping* была проверена их связность. Как видно на рисунке 3, контейнер *kalibox* видит оба контейнера, а значит, на данном этапе всё сделано правильно.

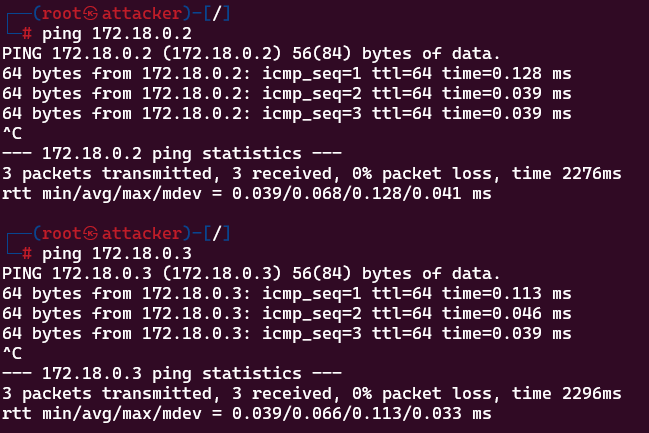


Рисунок 3 – Проверка связности контейнеров

Затем в контейнер *kalibox* была установлена утилита *fping*. Данная утилита делает то же самое, что и утилита *ping*, однако отличается от нее тем, что в одной строке можно сразу указать несколько *ip*-адресов, что более удобно при работе с несколькими контейнерами.

Первым делом данной утилиты были просканированы все 3 контейнера и, как и ожидалось, был получен ответ, говорящий о том, что все 3 узла доступны (рисунок 4).

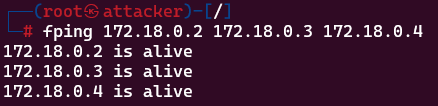


Рисунок 4 – Доступность трех узлов

Затем в качестве аргументы утилите была передана целая сеть – 172.18.0.0/16, и данная утилита начала сканировать всю сеть, обращаясь к каждому узлу.

Первые 4 узла, как и ожидалось, были доступны (рисунок 5). Узлы 172.18.0.2–172.18.0.4 – это *ip*-адреса запущенных контейнеров, доступность которых уже была проверена на предыдущем шаге, а *ip*-адрес 172.18.0.1 использует *Docker* для выхода из сети в *pentest* во внешнюю сеть, поэтому он тоже оказался доступным.

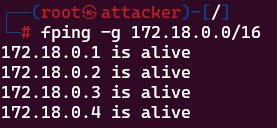


Рисунок 5 – Доступные узлы в сети

Все же остальные *ip*-адреса в подсети оказались недоступны, ведь кроме трех контейнеров больше в данной сети запущено ничего не было (рисунок 6). Как и ожидалось, утилита не могла получить от них ответ. Так как узлов в сети было достаточно много, в какой-то момент работу утилиты пришлось прервать.

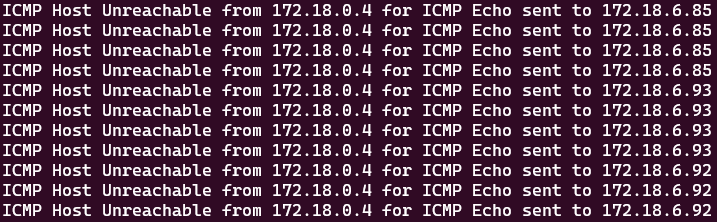


Рисунок 6 – Недоступность остальных узлов

## **Задание 2. Работа с инструментом *Nmap***

Для выполнения данного задания первым делом в контейнер *kalibox* была установлена утилита *Nmap*. *Nmap* – это инструмент, предназначенный для сканирования сетей. После установки утилиты необходимо было, воспользовавшись ей, просканировать узел информационной сети. В качестве узла был взят *metasploitable2* с *ip*-адресом 172.18.0.3. Данная утилита обладает следующим синтаксисом: *nmap <опция сканирования> <цель сканирования>*. Ниже приведены результаты сканирования узла 172.18.0.3 с различными опциями.

### **Опция *-A***

Данная опция включает полное сканирование. Утилита выводит в консоль всю полученную информацию об узле, включая операционную систему и детальную информацию о сервисах, работающих на портах (рисунок 7).

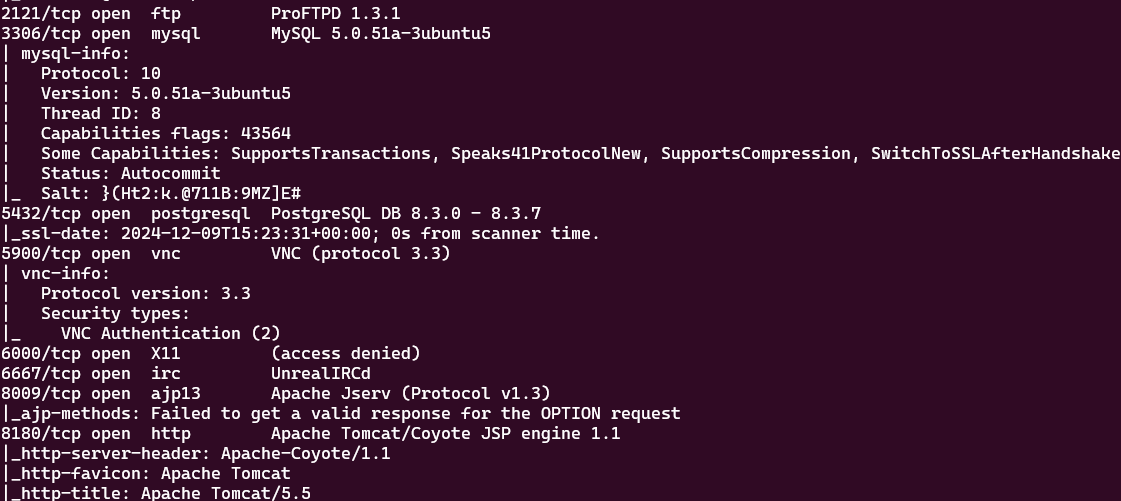


Рисунок 7 – Работа утилиты *nmap* с опцией *-A*

### **Опция *-sT***

Работая с данной опцией, утилита проверяет порты узла, устанавливая *TCP*-соединение и использую трехстороннее рукопожатие. В консоль выводится список открытых, закрытых и фильтрованных портов, а также идентифицируются все службы, работающие на этих портах (рисунок 8).

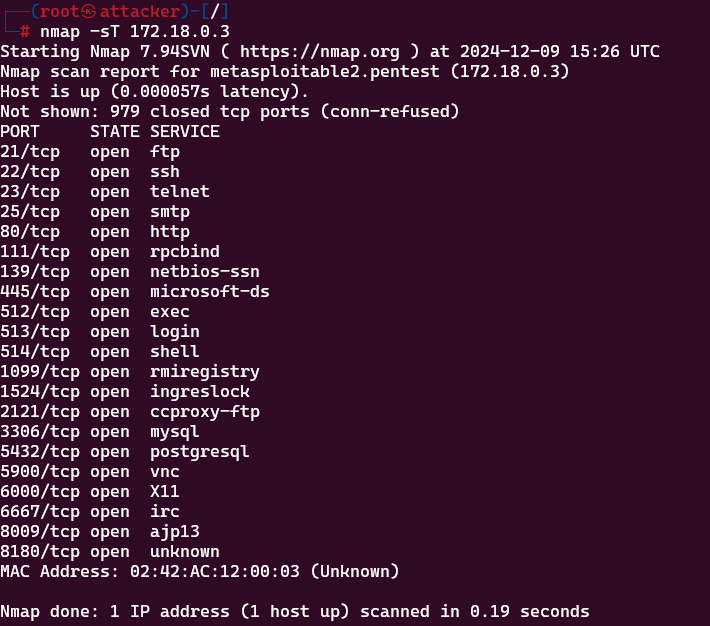


Рисунок 8 – Работа утилиты *nmap* с опцией *-sT*

### **Опция -*sS***

При данной опции выполняется *SYN*-сканирование, которое отличается от *TCP*-сканирования тем, что оно не использует трехстороннее рукопожатие, вследствие чего оно работает быстрее и является менее заметным. Как видно на рисунке 9, вывод совпадает с тем, что было при опции *-sT*.

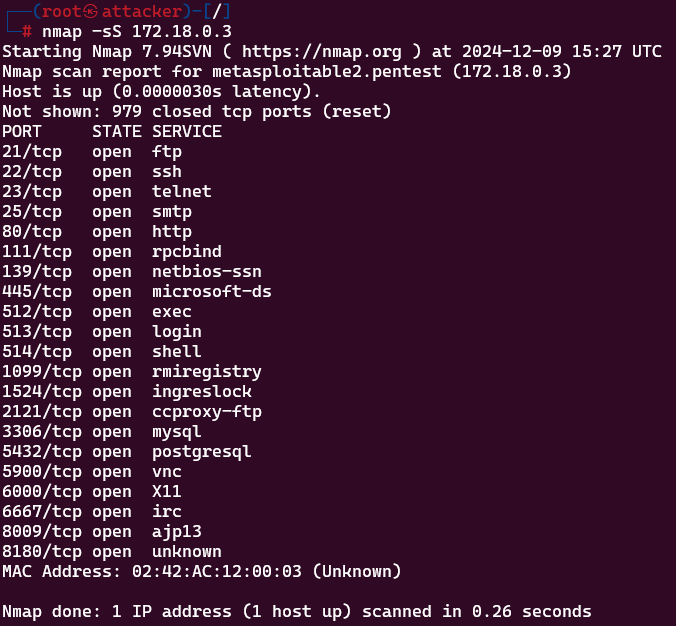


Рисунок 9 – Работа утилиты *nmap* с опцией *-sS*

### **Опция *-sN***

При данной опции выполняется *TCP NULL*-сканирование, которое отличается от простого *TCP*-сканирования тем, что у отправленных *TCP*-пакетов нет установочных флагов. Результат такого сканирования был аналогичен результатам сканирований с опциями *-sT* и *-sS*, представленных выше.

### **Опции *-sM, -sA, -sW***

Используя данные опции, утилита принимает *TCP*-сканирование Маймона, *TCP ACK*-сканирование и *TCP Window*-сканирование соответственно. Однако при использовании данных опции в лабораторной работе был получен одинаковый результат: в консоль выводилось сообщение о том, что ни один порт просканировать не получилось (рисунок 10). Вероятнее всего, это связано с тем, что файрволл *kalibox* блокирует такие типы подключений.

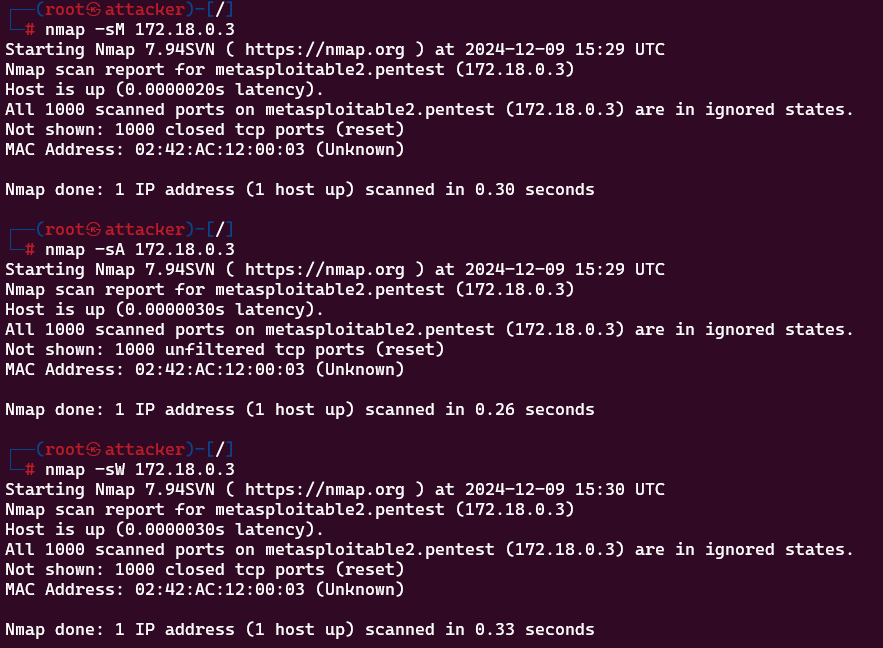


Рисунок 10 – Работа утилиты *nmap* с опциями *-sM, -sA, -sW*

### **Опция *-sO***

Данная опция позволяет узнать характеристики операционной системы, установленной на устройстве (рисунок 11).

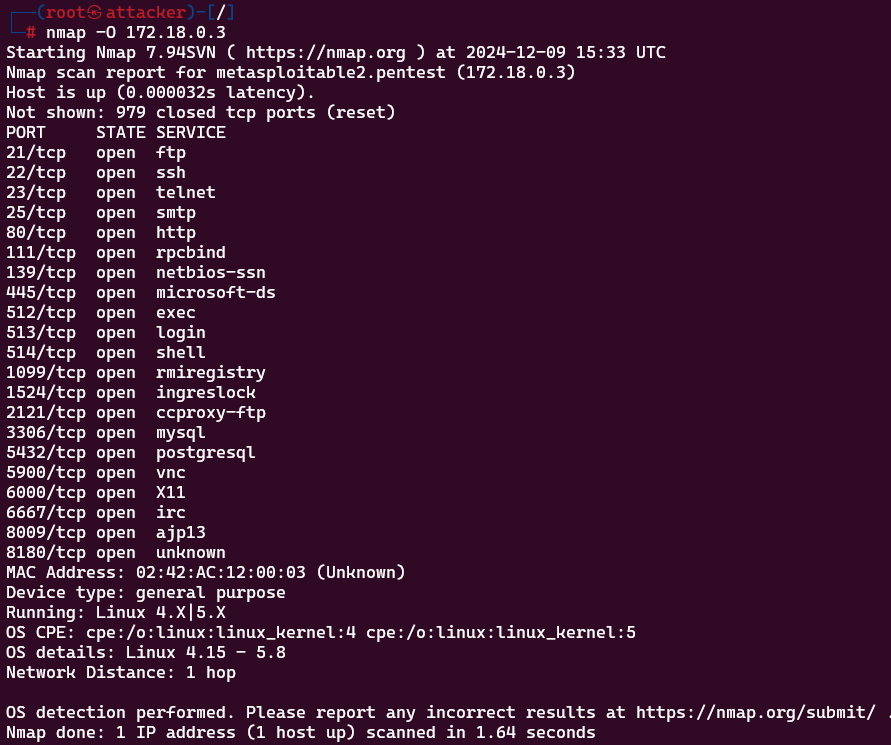


Рисунок 11 – Работа утилиты *nmap* с опцией *-sO*

## **Задание 3. Установка Metasploit**

Для выполнения данного задания первым делом с помощью команды из инструкции к лабораторной работе был установлен и запущен контейнер на основе образа *Metasploit* (рисунок 12).

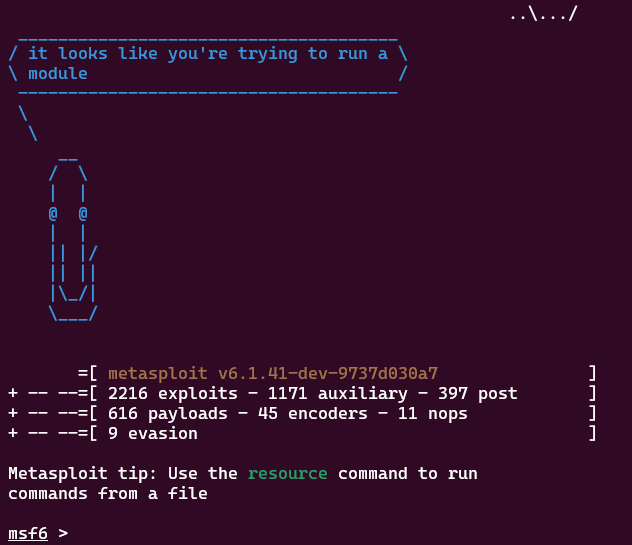


Рисунок 12 – Установленный *Metasploit*

Далее в этом контейнере была запущена утилита *db\_nmap*. Можно сказать, что данная утилита является той же утилитой *nmap*, которая была использована в предыдущем задании. В данном случае утилита была запущена без явного указания на тип сканирования, в результате чего был получен вывод в консоль, аналогичный выводу в предыдущем задании (рисунок 13).

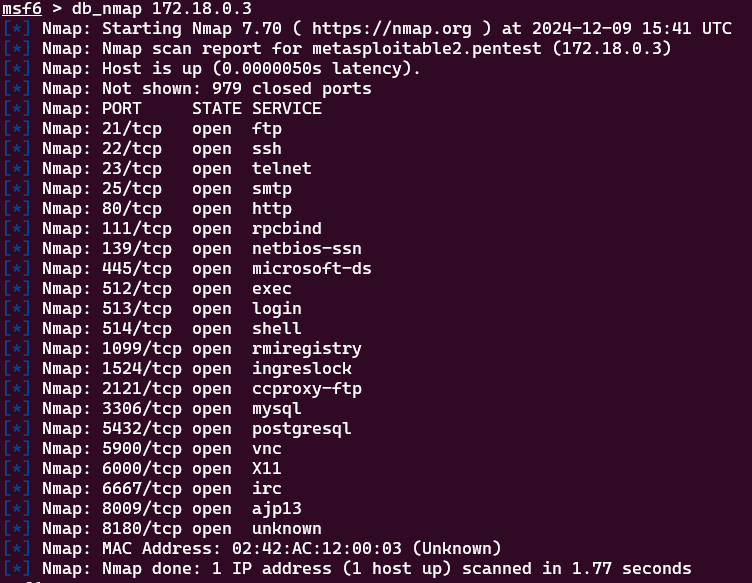


Рисунок 13 – Работы утилиты *db\_nmap*

Затем с помощью команды search *http*, которая используется для поиска модулей, связанных с *http* и, таким образом, позволяет найти эксплойты, связанные с *http* (рисунок 14).

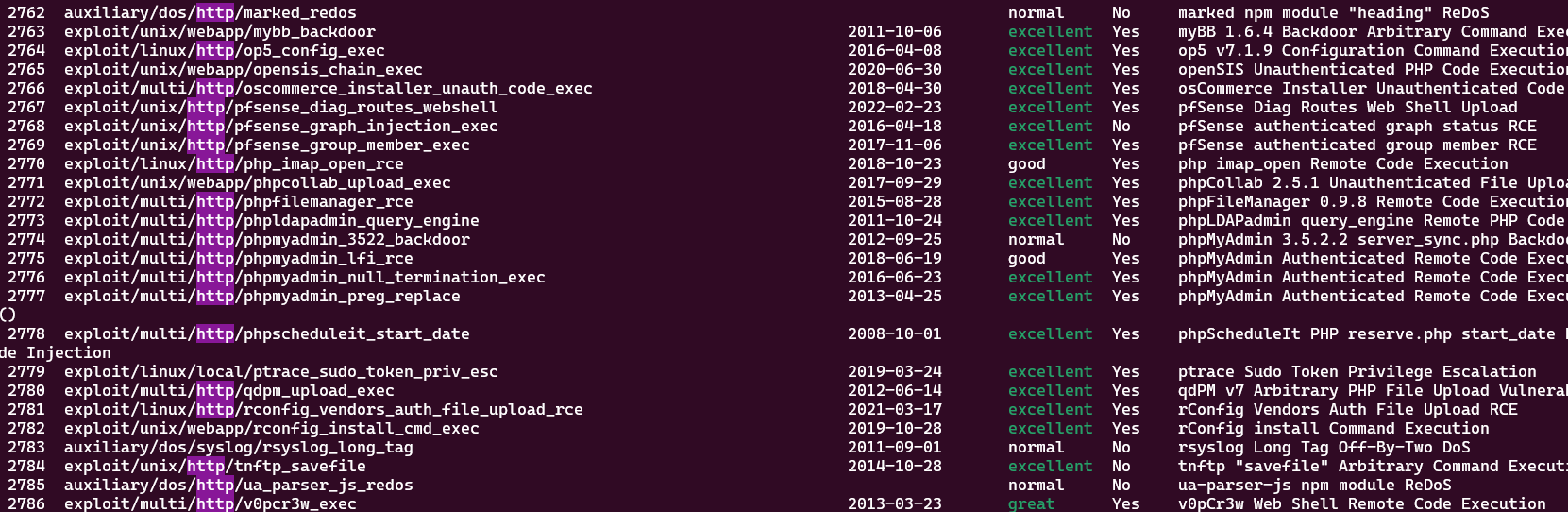


Рисунок 14 – Поиск эксплойтов с *http*

После этого один из таких эксплойтов был выбран с помощью команды *use*. Затем для данного типа атаки был с помощью *set* *RHOSTS* был указан *ip*-адрес контейнера, атаку на который планировалось провести (рисунок 15).

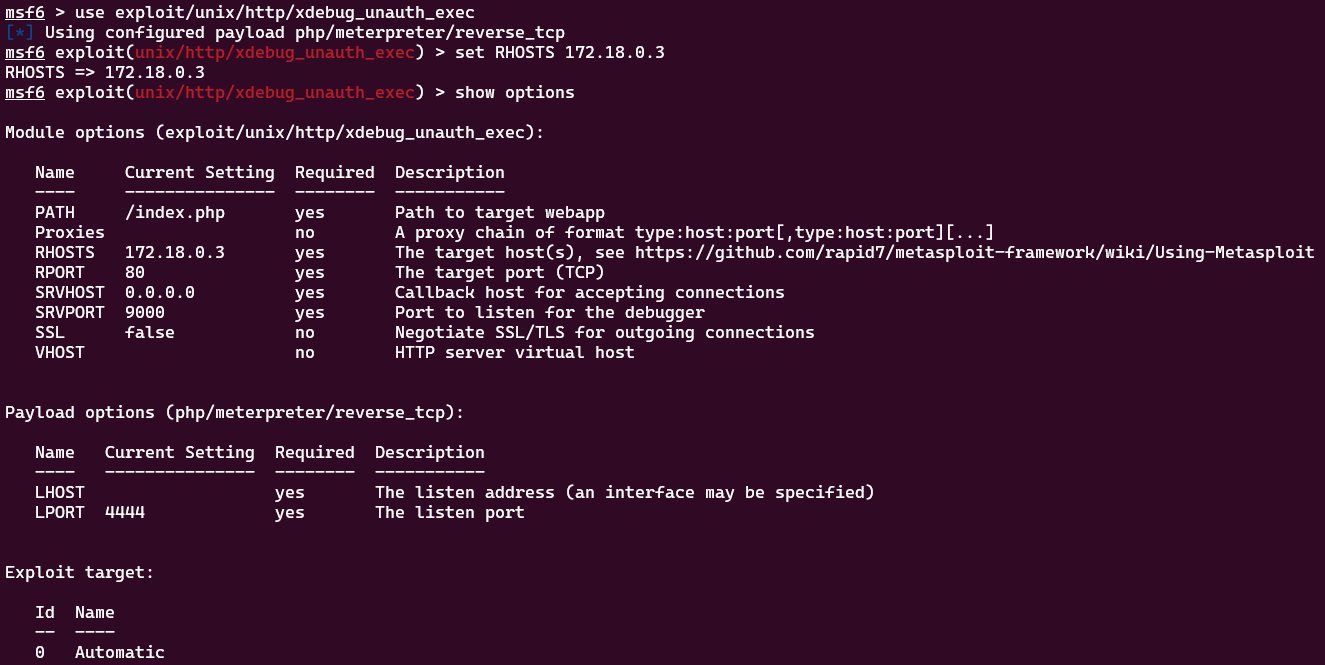


Рисунок 15 – Настройка атаки

Наконец, была введена команда *run*, которая запускает атаку на выбранный хост, однако сама атака выполнена не было, так как не удалось подтвердить параметр *LHOST* выбранной атаки (рисунок 16), в результате чего эксплойт использован не был.

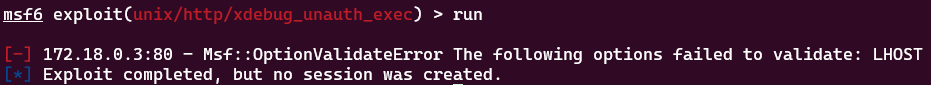


Рисунок 16 – Попытка запуска атаки

# **Вывод по работе**

В результате выполнения данной практической работы был изучен типовой алгоритм работы с нарушителями информационных систем, а также были приобретены практические навыки по использованию инструментов сканирования информационных систем. В ходе выполнения практической работы была проведена работа в лаборатории для поиска уязвимостей, затем была проведена работа с утилитой *Nmap* и, наконец, была произведена установка *Metasploit*.