**Лабораторная работа №3 Оценка уязвимости**

**Предусловия:**

Для выполнения работ рекомендуется установить программу виртуализации для операционных систем VirtualBox [1] https://www.virtualbox.org/wiki/Downloads, на которую рекомендуется установить дистрибутив Kali Linux [2] https://www.kali.org/get -kali/#kali-virtualmachines – это Linux дистрибутив, созданный на основе Debian с открытым исходным кодом, предназначенный для решения различных задач информационной безопасности, таких как тестирование на проникновение, исследование безопасности и компьютерная криминалистика.

Установлена виртуальная машина с metasploitable 2 (https://sourceforge.net/projects/metasploitable/).

# Упражнение 1. Оценка уязвимости вручную

**Цель:**

понять принцип определения уязвимостей на основе версии сервиса **После окончания работы студент должен**

* знать: что такое служебный баннер и как его можно использовать для обнаружения уязвимых мест;
* уметь: провести вручную оценку уязвимости.

**Задание:**

1. Подключиться к FTP-серверу и определить его версию;
2. Используя информацию о версии, найдите, какие уязвимости существуют для данного FTP-сервера;
3. Проэксплуатируйте обнаруженный бэкдор на FTP-сервере
4. Выполните шаги 1-2 для веб-сервера. Для определения версии веб-сервера используйте NMAP.

## Технические инструменты для выполнения работы

* Kali Linux VM (Kali)
* Metasploitable 2 VM (target)

## Порядок выполнения работы

Определение версии FTP-сервера с помощью команды: nmap 192.168.0.105 –p21 –sV

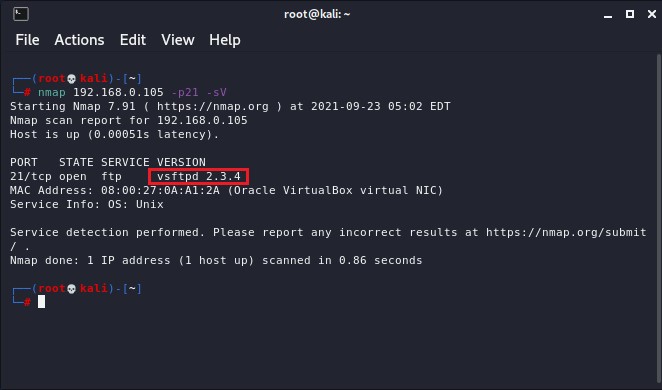


Рисунок 1. Определение версии ftp-сервера.

Поиск уязвимости определенной версии ftp-сервера.

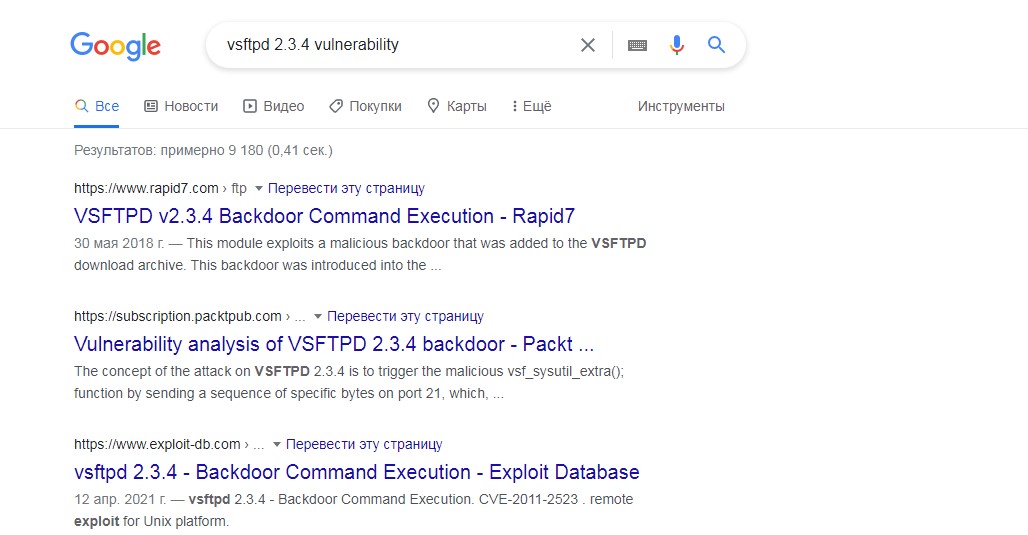


Рисунок 2. Поиск уязвимости определенной версии ftp-сервера.

Эксплуатация бэкдора. Добавьте “:)” в логине пользователя ftp, пароль – любое слово.

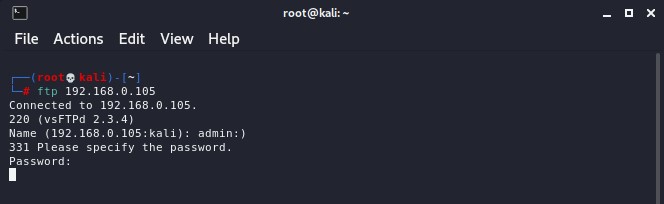


Рисунок 3. Эксплуатация бэкдора ftp-сервера.

После подключения к ftp-серверу открывается порт 6200 (сервер лицензий LM-X). Подключимся к уже открытому порту 6200. Для этого можно воспользоваться командой:

netcat 192.168.0.105 6200

Эта уязвимость позволяет выполнить любую системную команду. К примеру, выведем информацию о сетевых интерфейсах, для этого воспользуемся командой ifconfig.

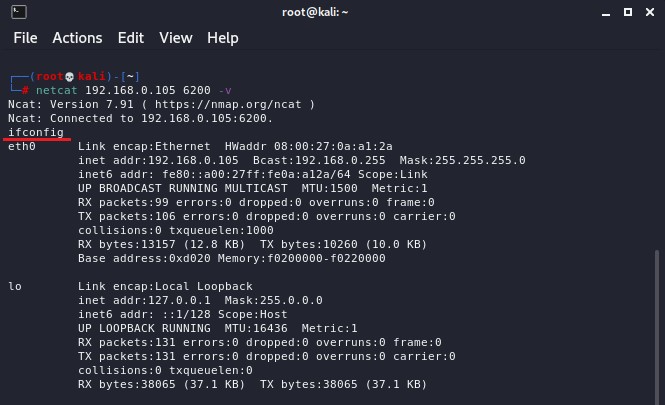


Рисунок 4. Эксплуатация уязвимости сервера лицензий LM-X на порту 6200.

Определение версии Веб-сервера Apache с помощью команды: nmap 192.168.1.100 –p80 – sV

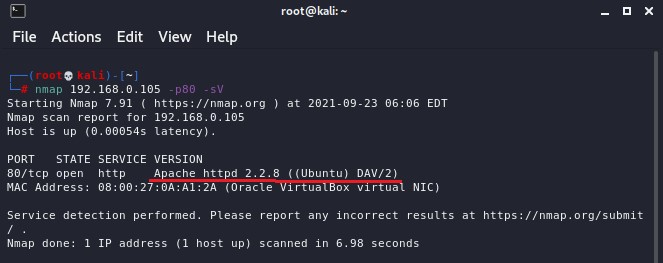


Рисунок 5. Определение версии веб-сервера Apache.

Поиск уязвимости определенной версии Веб-сервера Apache.

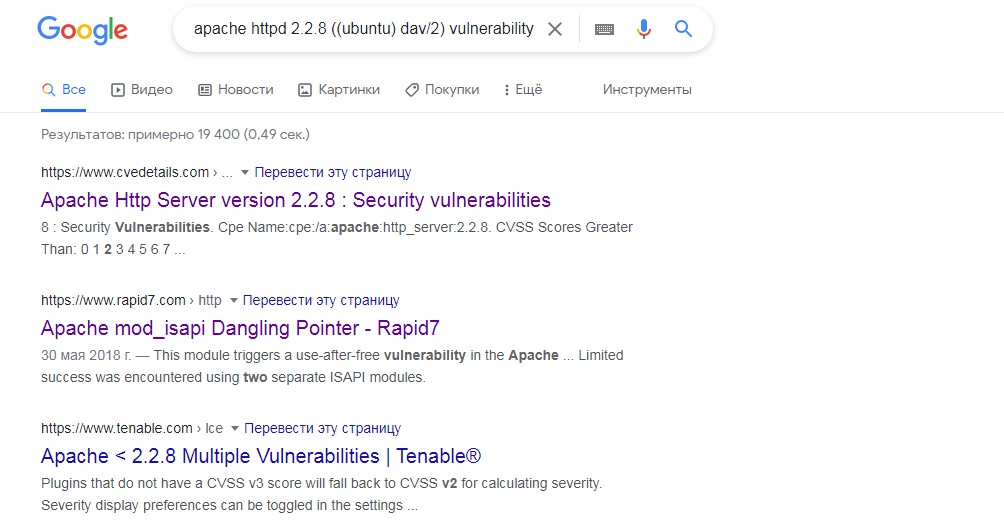


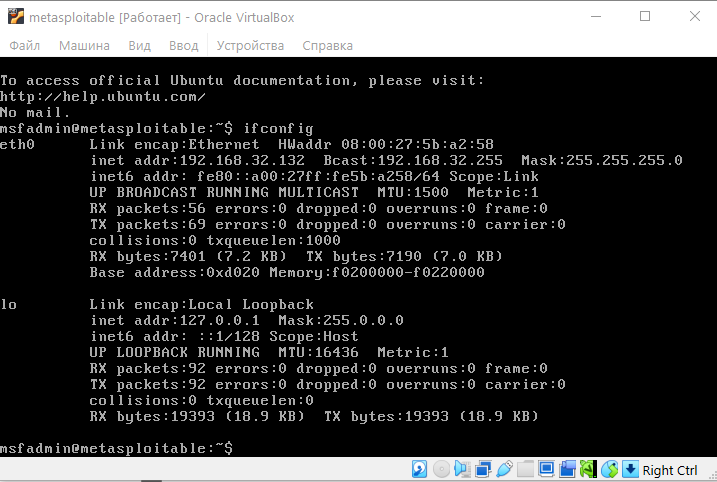
Рисунок 6. Поиск уязвимости определенной версии Веб-сервера Apache.

## ЗАДАНИЕ 1

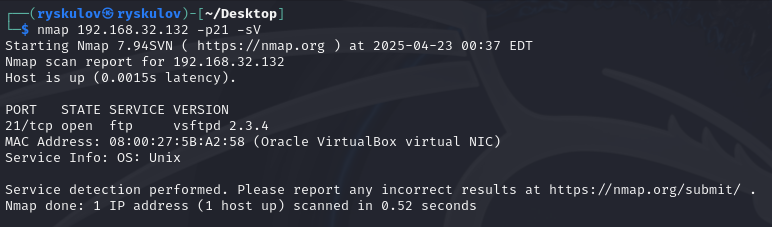
Подключиться к FTP-серверу и определить его версию. Используя информацию о версии, найдите какие уязвимости существуют для данного FTP-сервера. Проэксплуатируйте обнаруженный бекдор на FTP-сервере. Докажите это с помощью скриншотов.

**Ответ:**

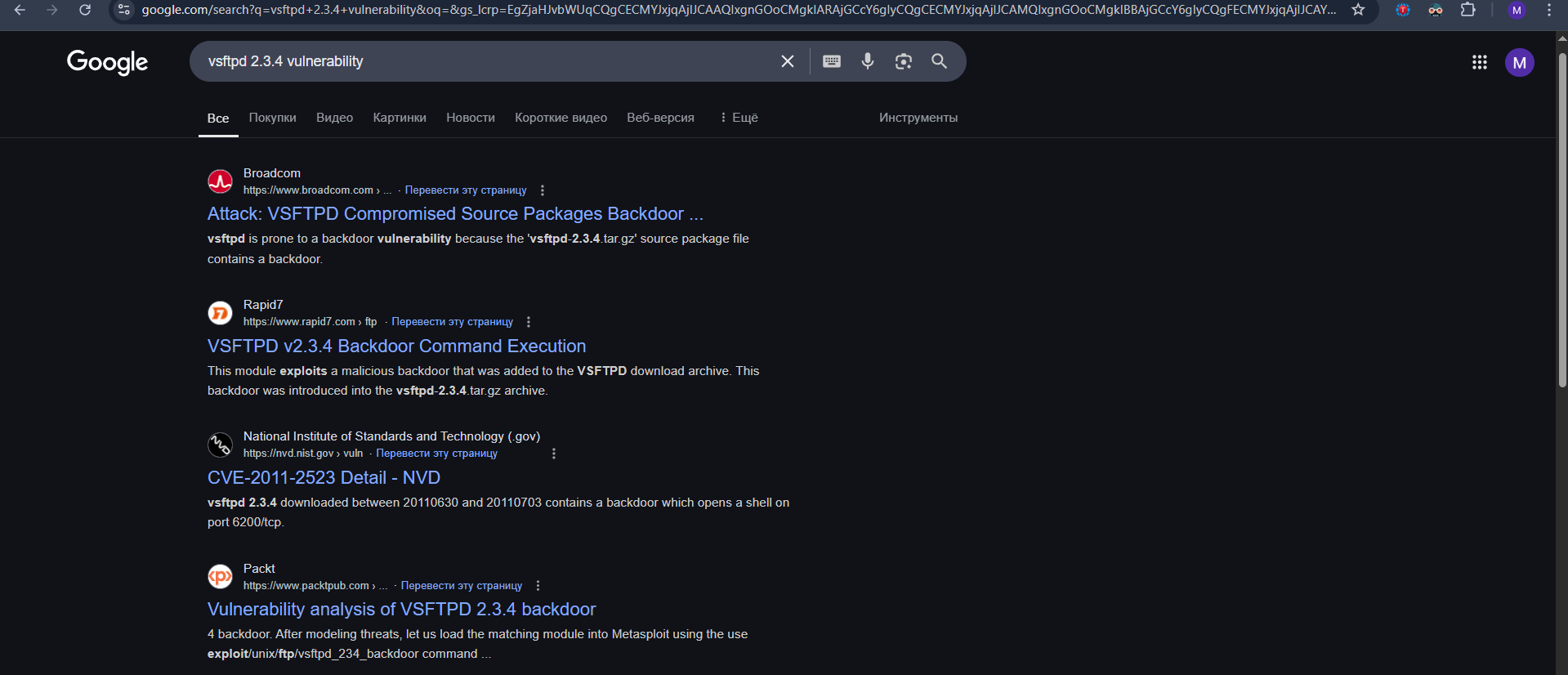
Ip-адрес metasploitable – 192.168.32.132:

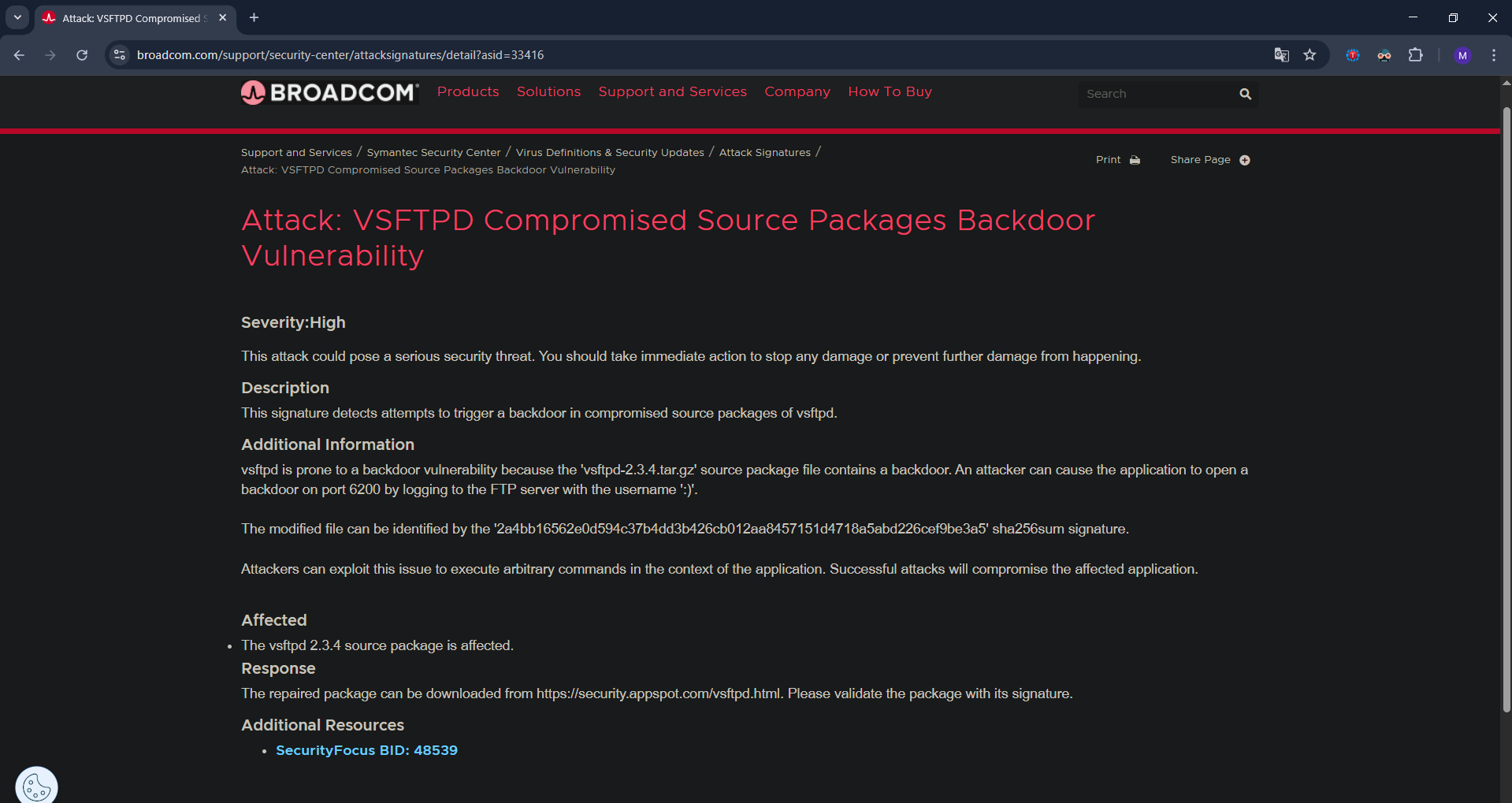


Определим версию FTP-сервера:



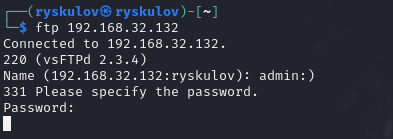
Найдем, какие уязвимости существуют для данного FTP-сервера:



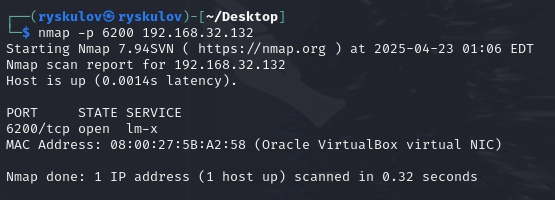


Проэксплуатируем бэкдор:

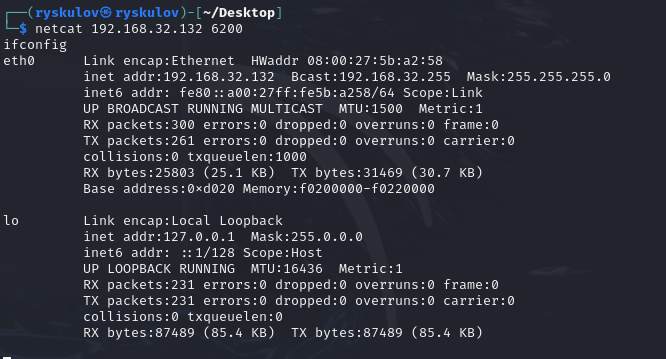
Подключимся к ftp-серверу:



Проверим, открылся ли порт 6200:



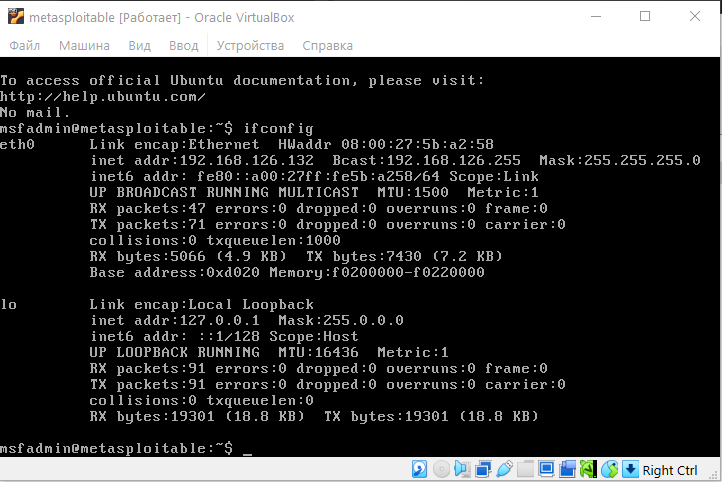
Подключимся к порту через netcat и воспользуемся сетевой командой ifconfig:



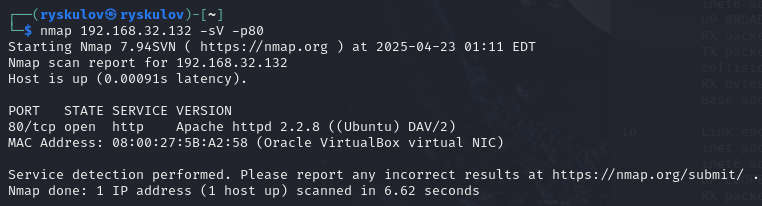
## ЗАДАНИЕ 2

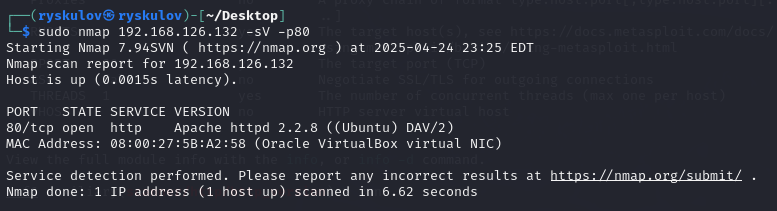
Подключиться к веб-серверу Apache и определить его версию. Используя информацию о версии, найдите, какие уязвимости существуют для данного Веб-сервера Apache. Проэксплуатируйте обнаруженный бэкдор на Веб-сервере Apache. Докажите это с помощью скриншотов.

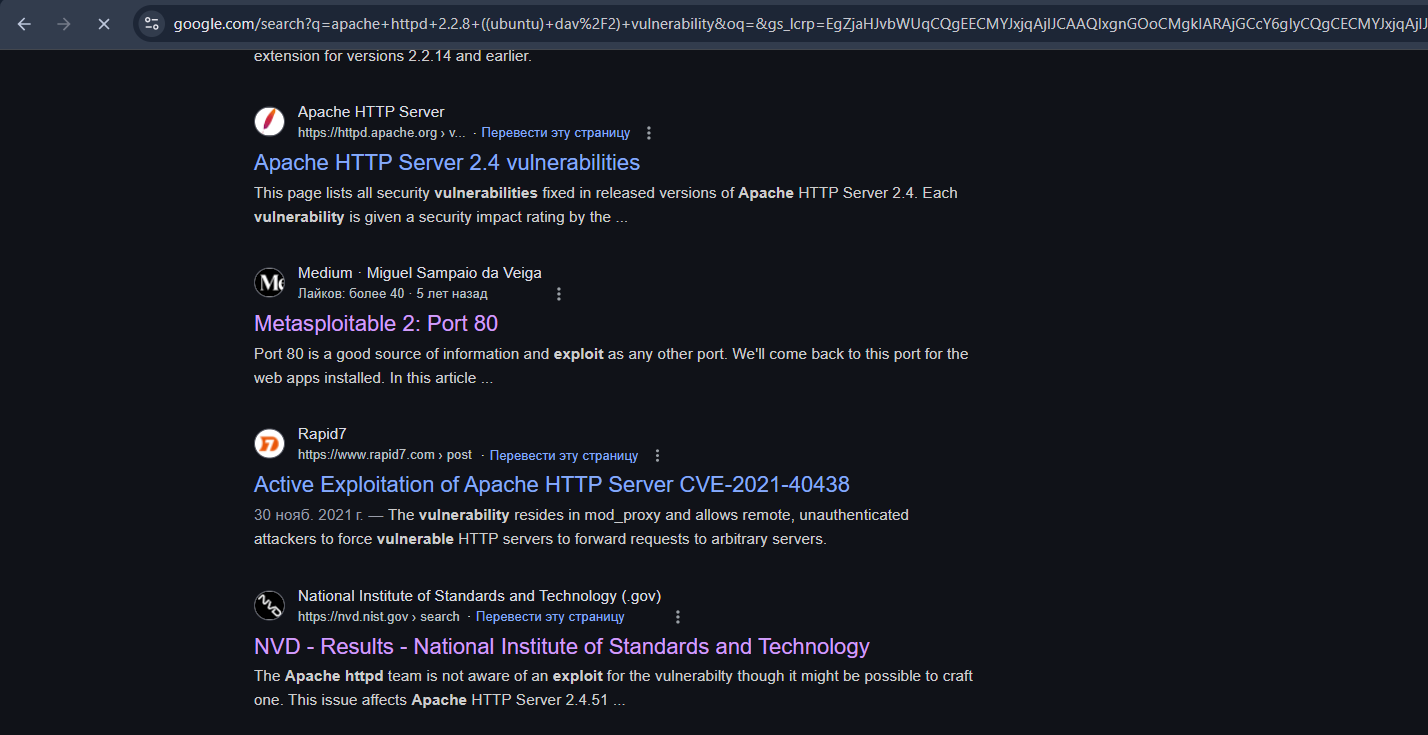
**Ответ:**

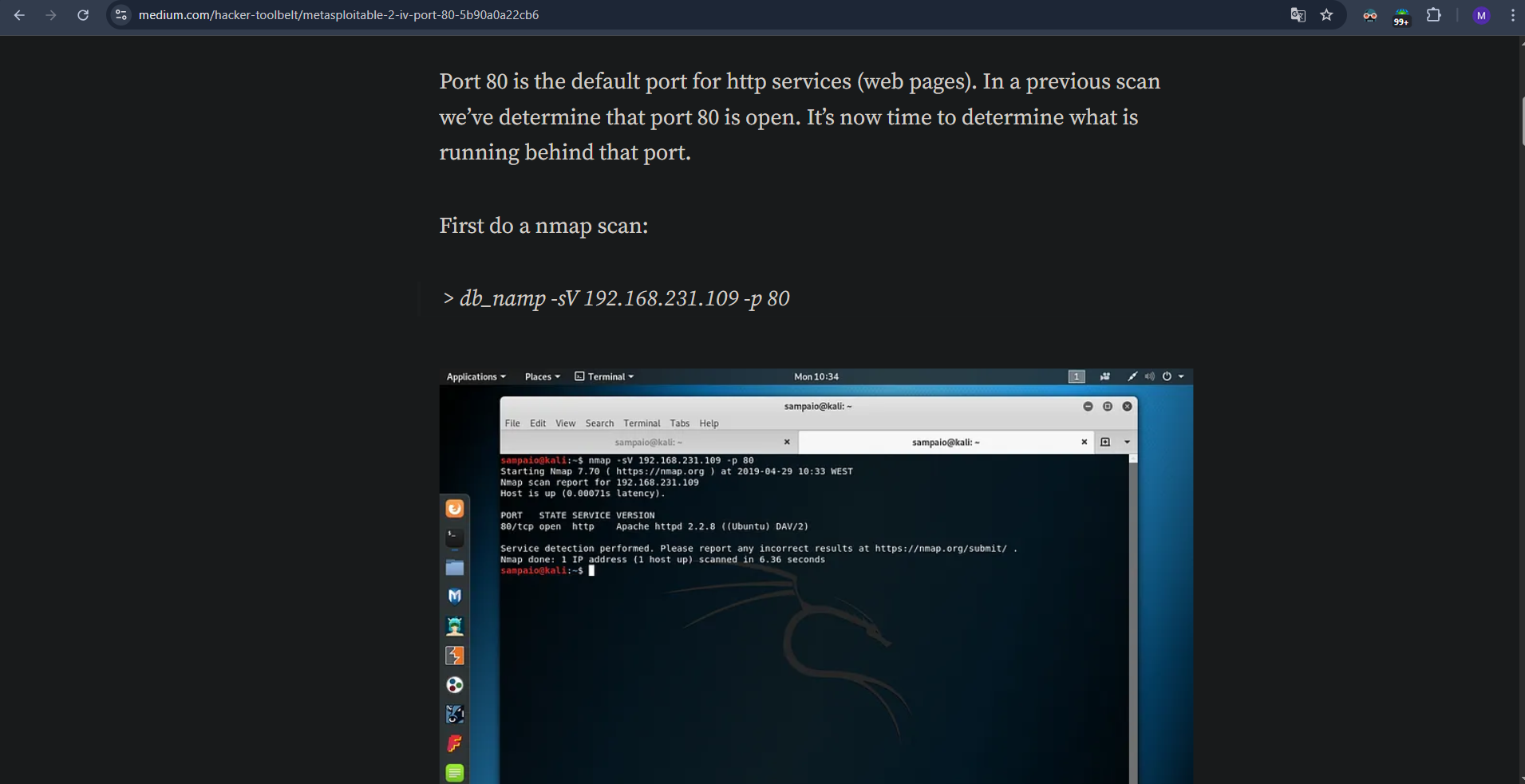


Определим версию Apache-сервера:

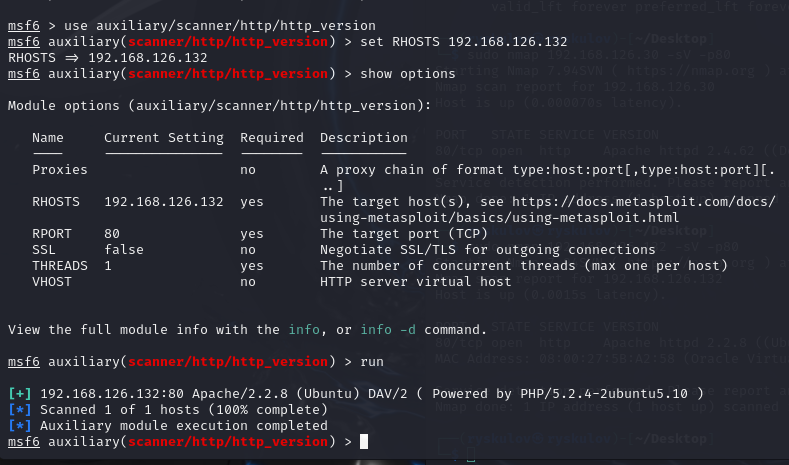








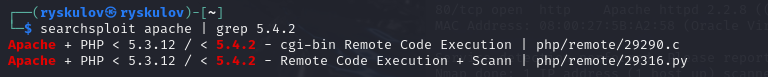
Узнаем версию PHP:



Убедимся, открыв в браузере:

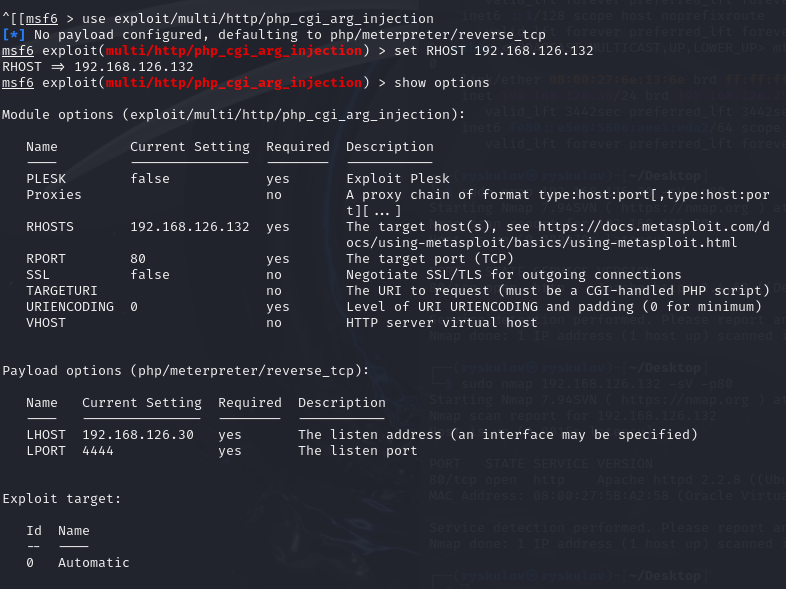


Найдем уязвимость в базе данных:

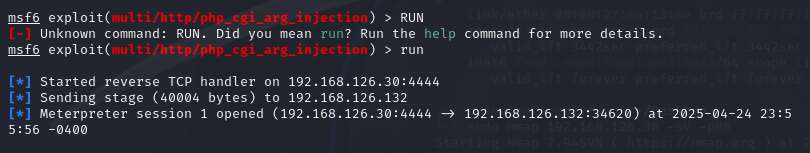


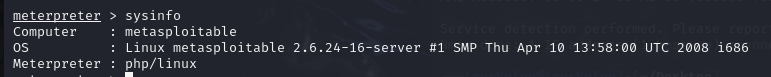
Обнаружили удаленное выполнение кода CGI.

Поэксплуатируем его:



Получили доступ к meterpreter:





# Упражнение 2. Использование сканеров уязвимостей для оценки уязвимости

**Цель:**

понять принципы использования сканеров уязвимостей

**После окончания работы студент должен**

* знать: как работают сканеры уязвимостей
* уметь: проводить оценку уязвимости с помощью сканеров уязвимостей.

**Задание:**

1. Установите и обновите сканер GVM в Kali Linux.
2. Произведите сканирование цели с помощью GVM.
3. Импортируйте результаты сканирования GVM в базу данных Metasploit.
4. 4. Установите и обновите сканер Nessus.
5. Произведите сканирование цели с помощью сканера Nessus.
6. Импорт результатов сканирования Nessus в базу данных Metasploit.
7. Сравните результаты работы сканеров.

## Технические инструменты для выполнения работы

* Kali Linux VM (Kali)
* Metasploitable 2 VM (цель)

## Ссылки

* <https://www.openvas.org/>
* <https://www.tenable.com/products/nessus>

## Порядок выполнения работы (сканер уязвимостей GVM)

OpenVAS – это сканер уязвимостей с открытым кодом. OpenVAS предназначен для активного мониторинга узлов сети на предмет выявления проблем, связанных с безопасностью, оценкой серьезности этих проблем и контроля их устранения.

После выпуска платформы OpenVAS 9 она была переименована в Greenbone Vulnerability Management (GVM).

1. Установите сканер GVM - Greenbone Vulnerability Management (ранее назывался OpenVAS), для этого выполните следующие команды:

sudo apt update sudo apt install gvm –y

1. Для настройки сканера в первый раз использование выполните команду: sudo gvm-setup
2. Сохраните пароль, созданный после настройки сканера GVM.



Рисунок 7. Логин и пароль сканера GVM.

1. Проверьте правильность настроек с помощью команды: sudo gvm-check-setup
2. Откройте браузер и перейдите по ссылке https://127.0.0.1:9392
3. Введите логин и пароль, сохраненные на шаге 3.

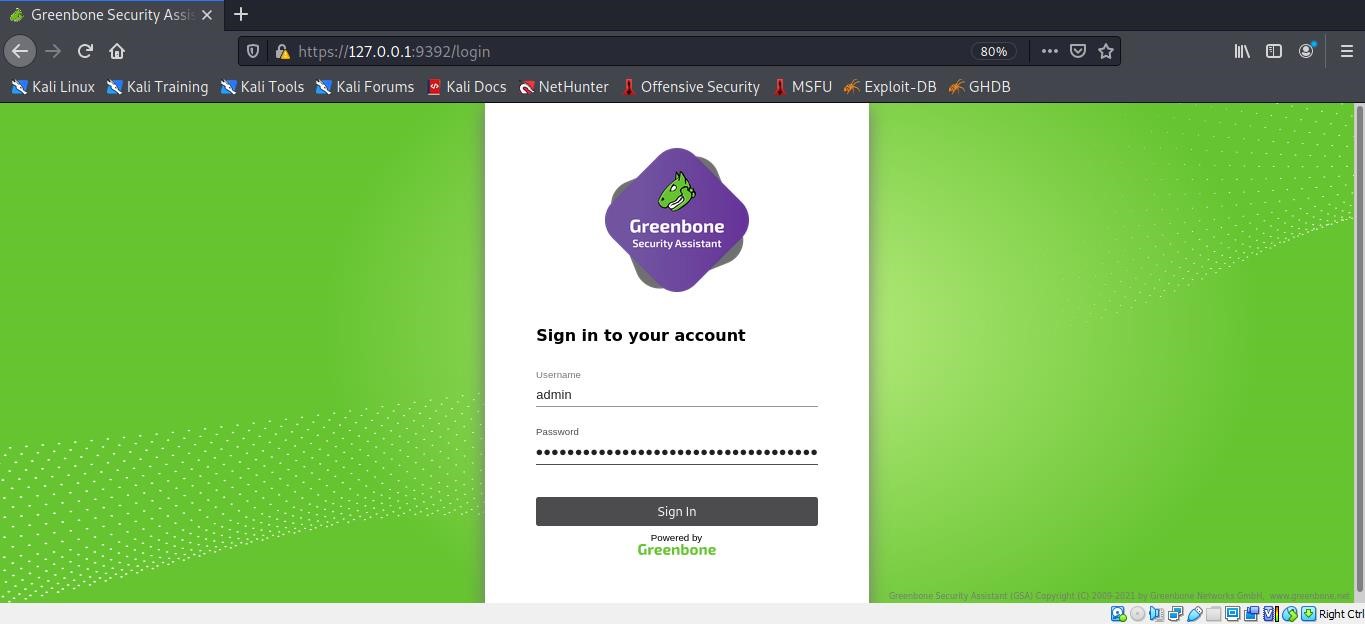
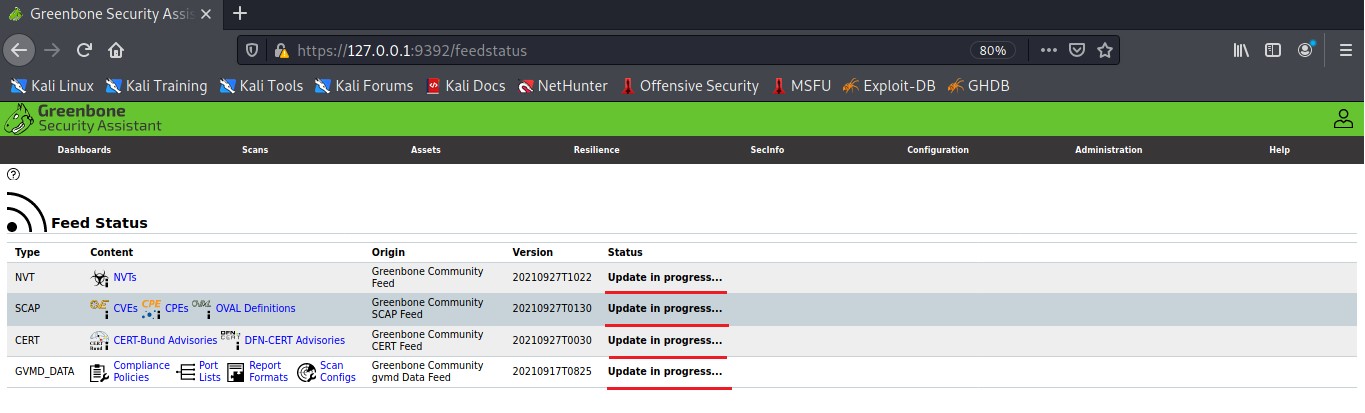


Рисунок 8. Страница входа сканера GVM.

1. Перейдите в меню Administration-Feed Status и дождитесь завершения обновления



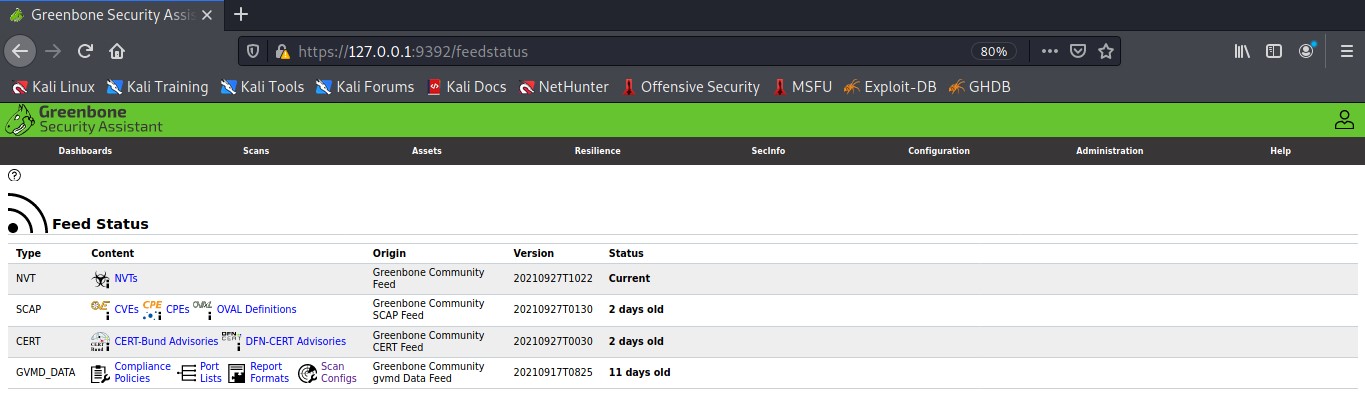


Рисунок 9. Обновление ленты сканера GVM.

1. Обновите Feed с помощью команды:

sudo gvm-feed-update



Рисунок 10. Обновление ленты сканера GVM.

1. Для запуска или остановки GVM используйте команды:

sudo gvm-start/stop

1. Запустіть швидке сканування своєї цілі.

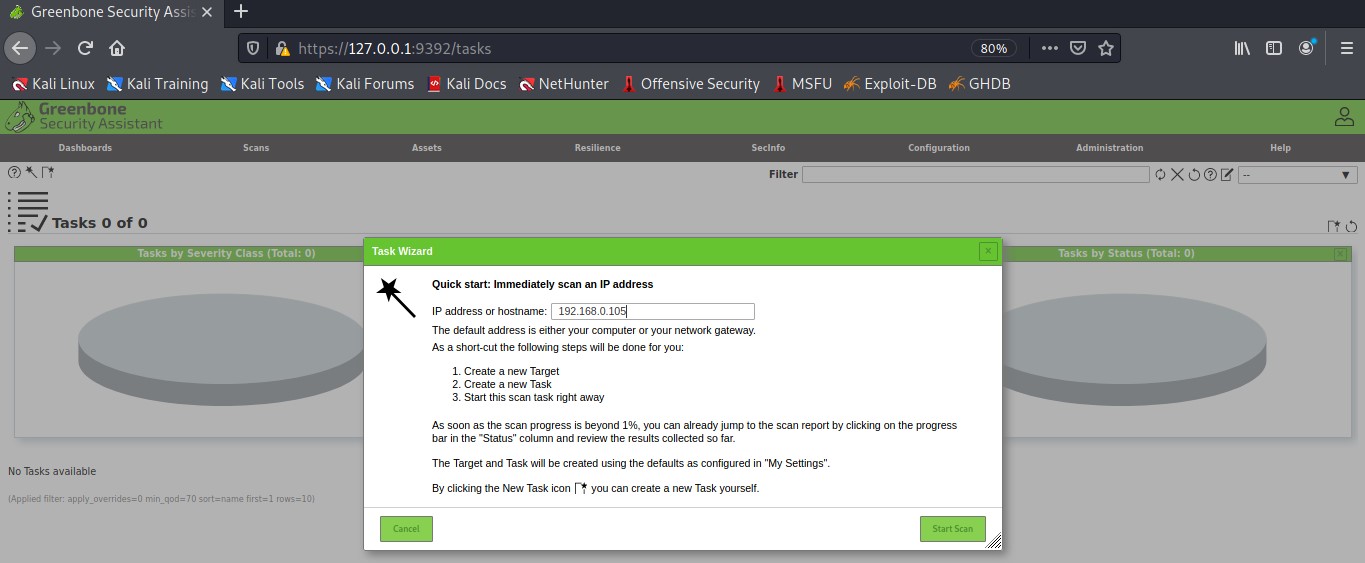


Рисунок 11. Сканер GVM, запуск быстрого сканирования.

1. Просмотрите результаты сканирования и экспортируйте отчет.

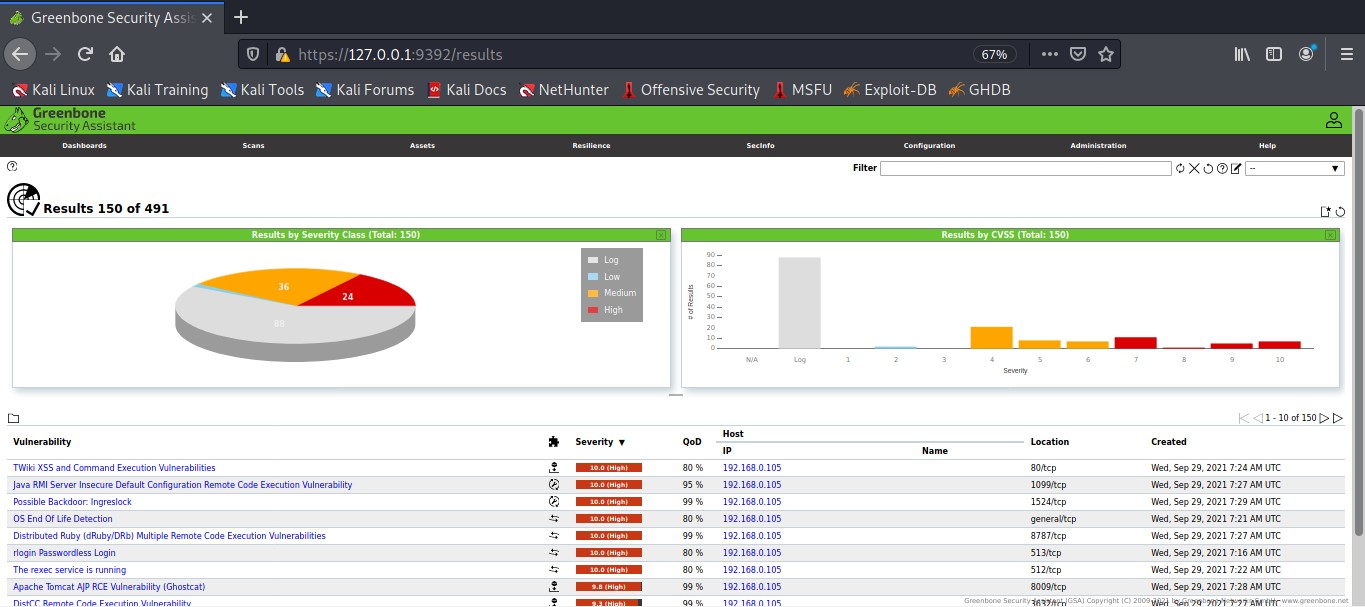


Рисунок 12. Сканер GVM, результаты сканирования.

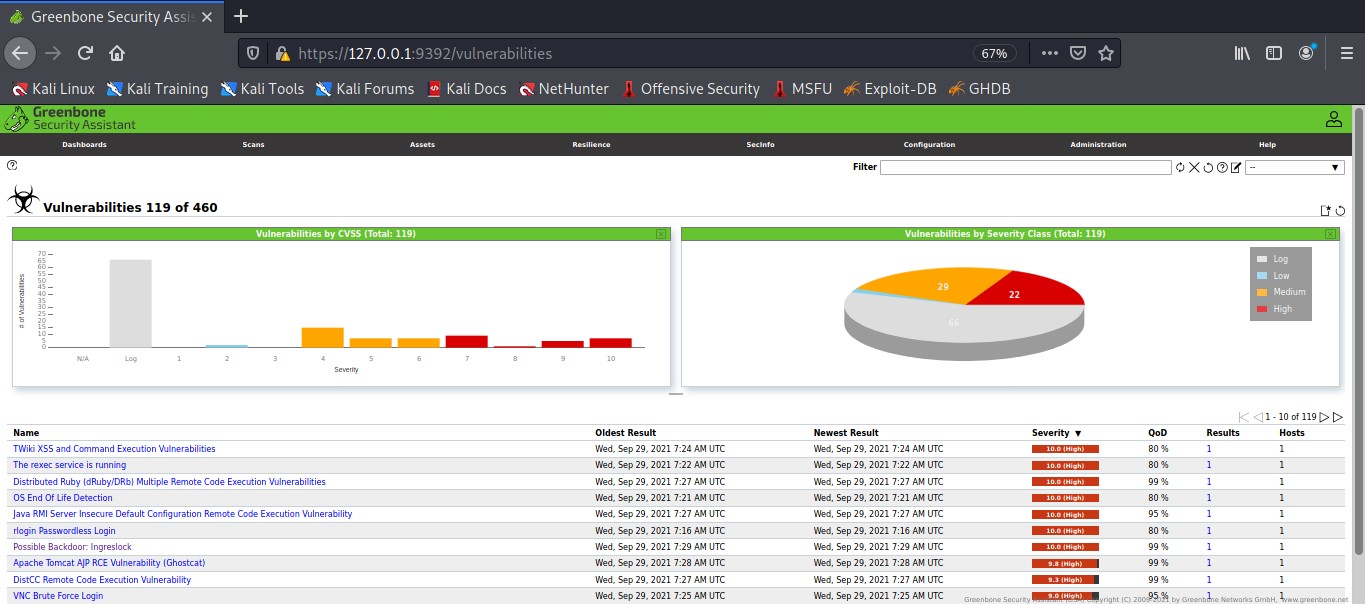


Рисунок 13. Сканер GVM, найденные уязвимости.

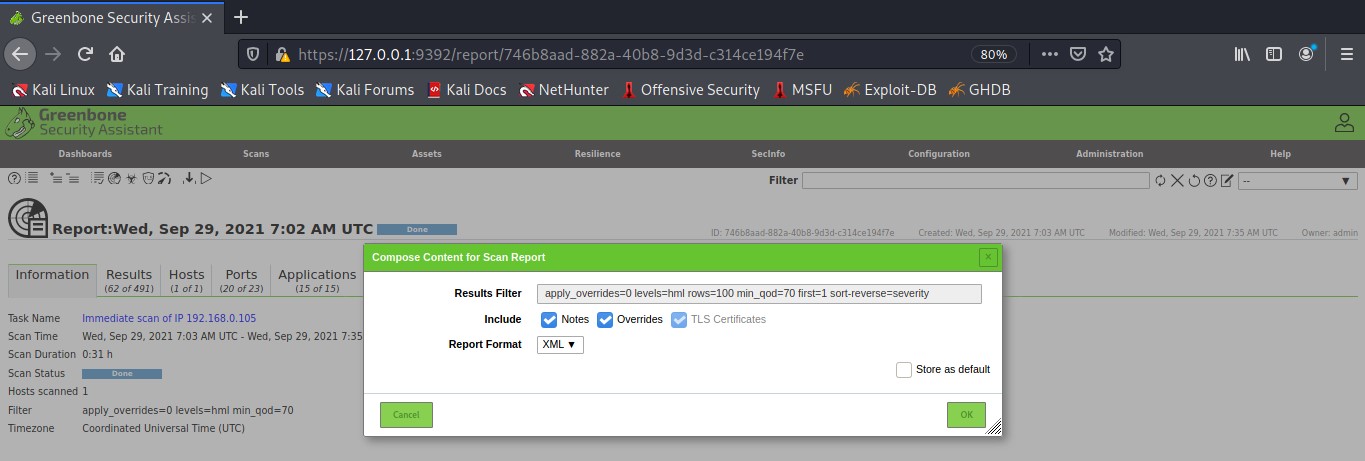


Рисунок 14. Сканер GVM, экспорт отчета.

## Порядок выполнения работы (сканер уязвимостей Nessus)

Загрузите текущую версию Nessus (пакет Debian).

**Nessus** – программа для автоматического поиска известных недостатков в защите информационных систем. Она способна выявить наиболее часто встречающиеся виды уязвимостей, например:

* Наличие уязвимых версий служб или доменов
* Ошибки конфигурации (например, отсутствие авторизации на SMTP-сервере)
* Наличие паролей по умолчанию, пустых или слабых паролей

База уязвимостей обновляется еженедельно, однако для коммерческих подписчиков есть возможность загружать новые плагины без семидневной задержки.

**ВНИМАНИЕ**

При отключенной опции «safe checks» некоторые тесты на уязвимости, которые используют Nessus, могут привести к нарушениям в работе сканированных систем.

Необходимо зарегистрироваться на сайте https://www.tenable.com/products/nessus/nessusessentials и получить код активации продукта.

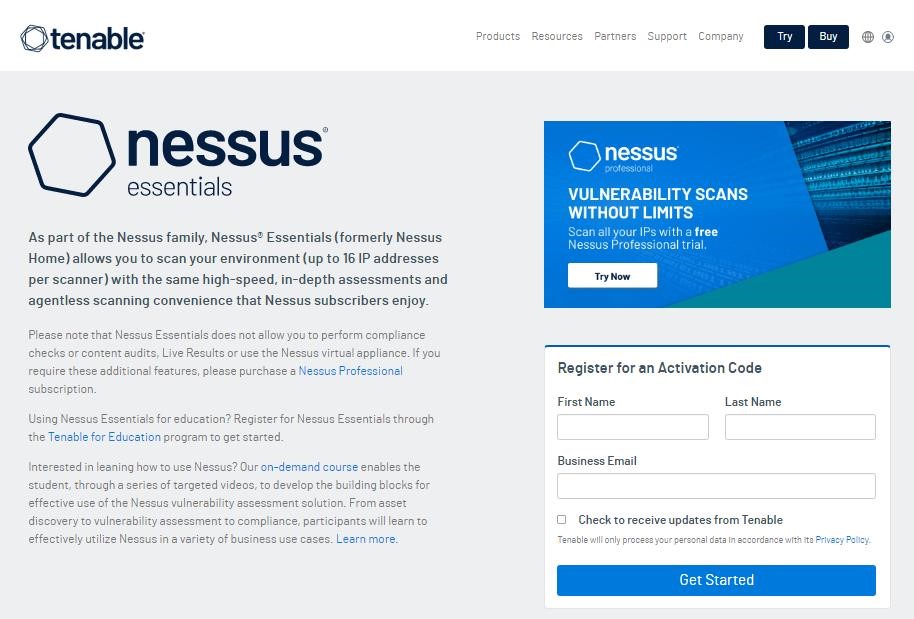


Рисунок 14. Сканер NESSUS, страница регистрации продукта.

Для использования версии для операционной системы Kali Linux скачать deb-пакет на странице https://www.tenable.com/downloads/nessus?loginAttempted=true и сохранить на компьютере, например в папке Downloads.

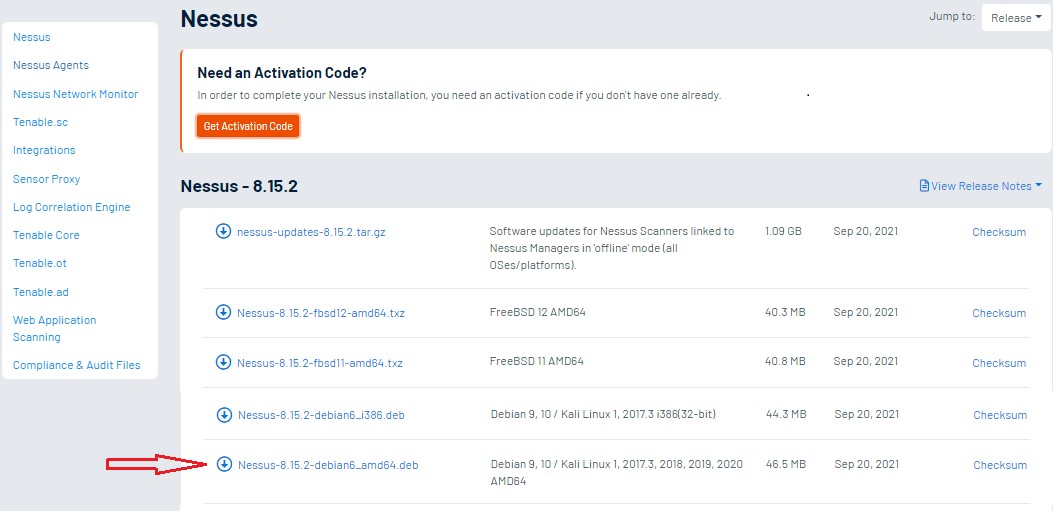


Рисунок 15. Сканер NESSUS, страница загрузки продукта.

Перейти в папку Downloads и установить сканер NESSUS с помощью команды:

sudo dpkg -i Nessus-8.15.2-debian6\_amd64.deb

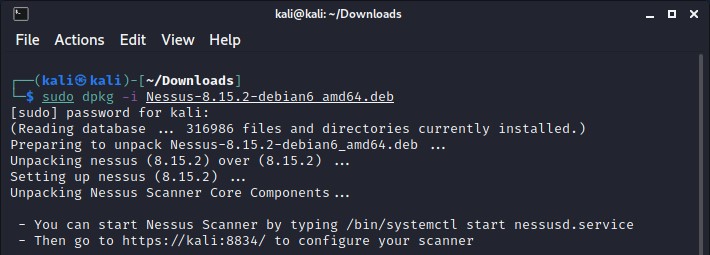


Рисунок 16. Сканер NESSUS, установка программы.

Запустить службу nessusd.service, выполнив команду:

sudo /bin/systemctl start nessusd.service

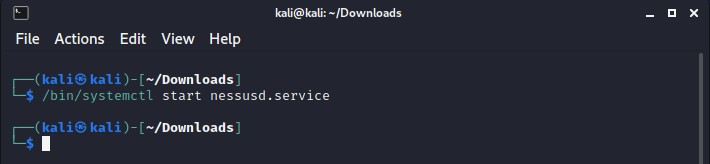


Рисунок 17. Сканер NESSUS, запуск службы nessusd.service.

Запустите сканер NESSUS, набрав в браузере https://kali:8834/

Создать Логин и Пароль, а также ввести код активации, полученный на предыдущих шагах. Войти в сканер NESSUS.

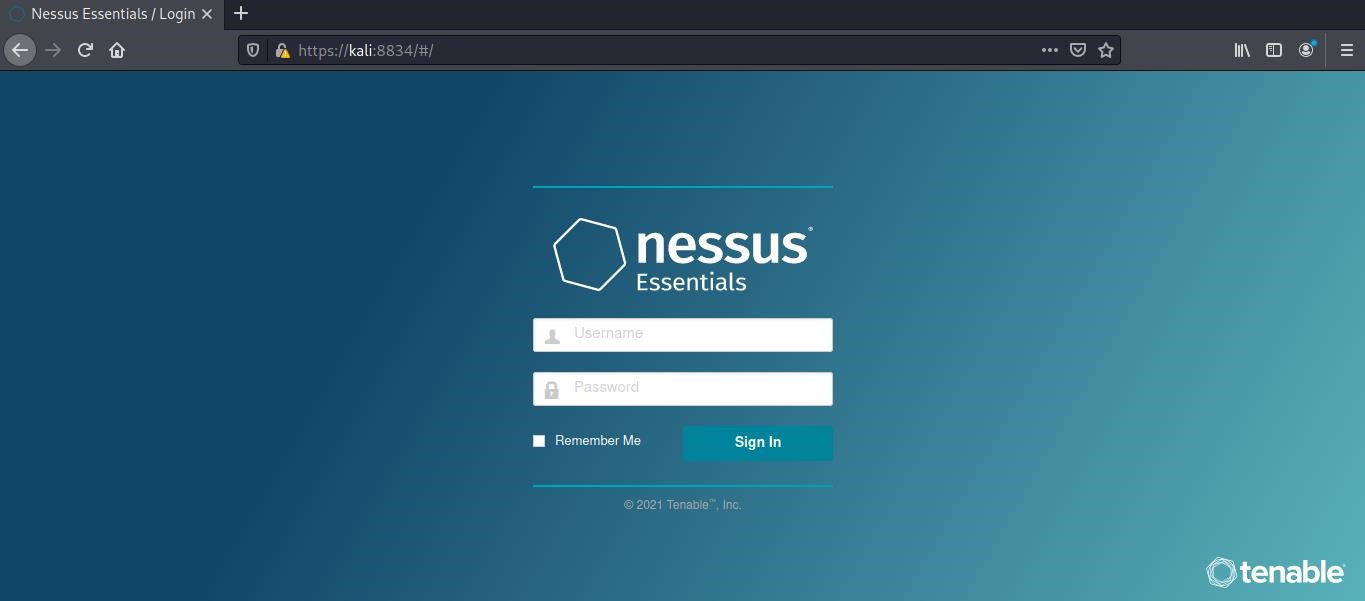


Рисунок 17. Сканер NESSUS, страница входа.

Выполнить полное сканирование цели (используйте расширенную политику)

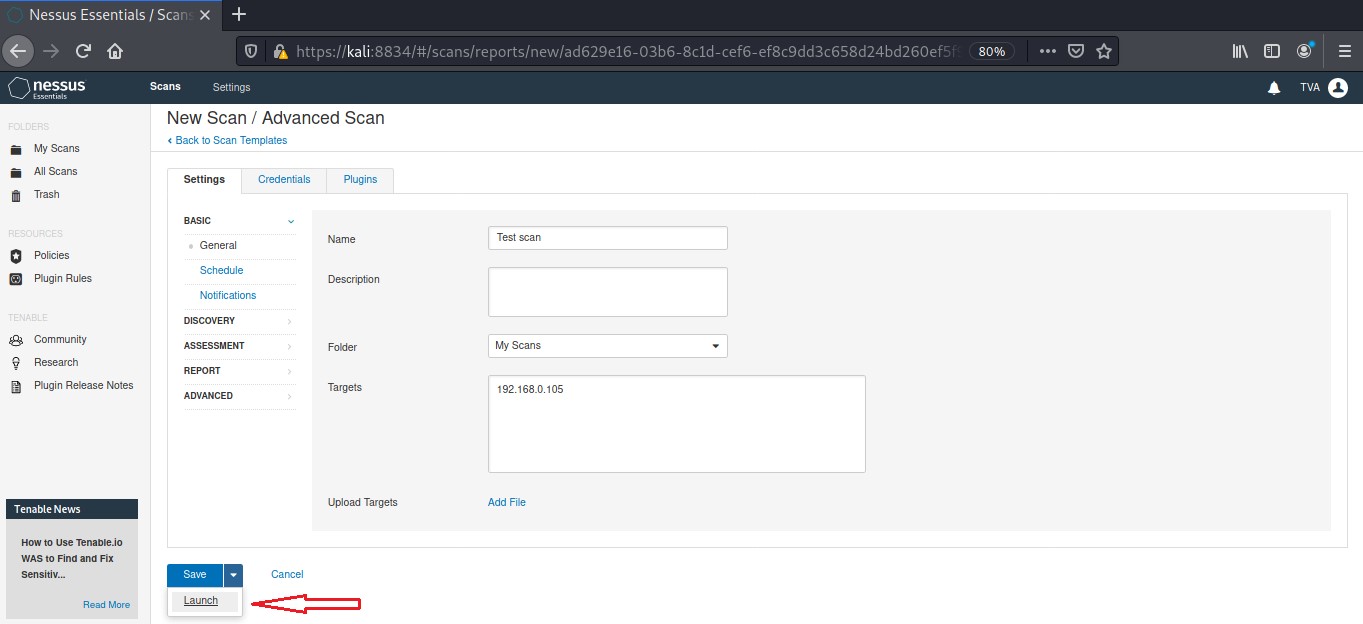


Рисунок 18. Сканер NESSUS, запуск расширенного сканирования.

Просмотрите результаты оценки уязвимости и экспортируйте ее в формате Nessus.

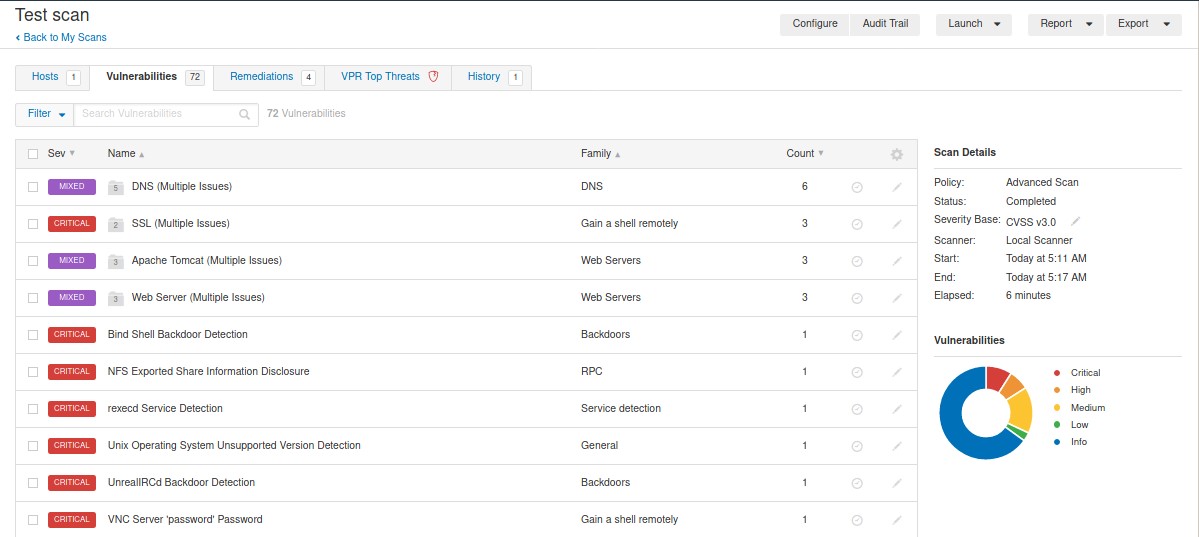


Рисунок 19. Сканер NESSUS, результат поиска уязвимостей.

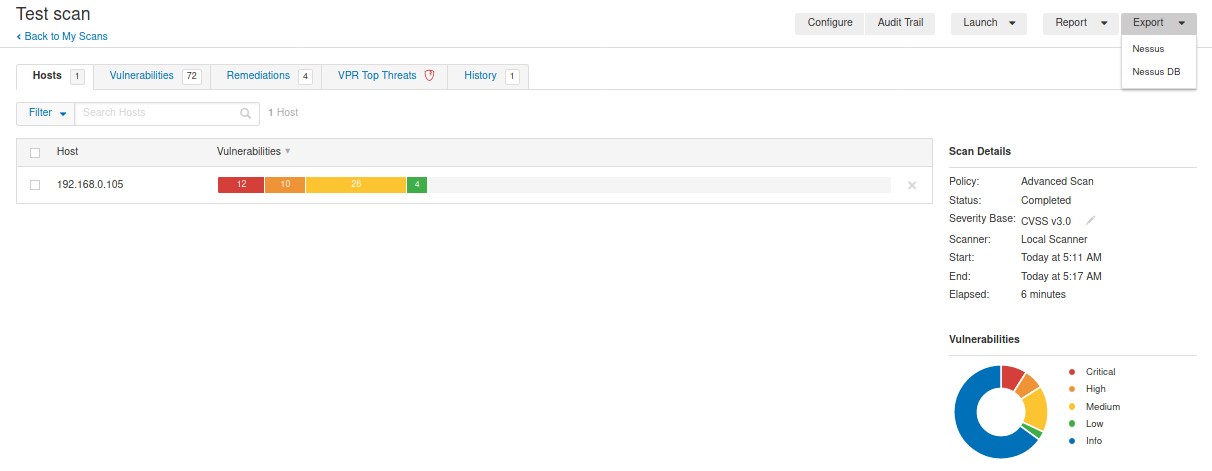


Рисунок 20. Сканер NESSUS, экспорт результатов.

Импорт результатов сканирования в базу данных Metasploit Framework.

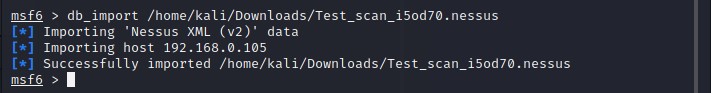


Рисунок 21. Фреймворк Metasploit, импорт результатов работы сканера NESSUS.

Metasploit Project – это проект сферы компьютерной безопасности, предоставляющий информацию об уязвимости системы и помогающий в тестах на проникновение. Metasploit был создан американским хакером H. D. Moore в 2003 году как портативное сетевое приложение написанное на Perl. К 2007 году Metasploit Framework был полностью перенесен на Ruby. 21 октября 2009 года Metasploit Project анонсировал, что его приобрела Rapid7, предприятие по кибербезопасности, предоставляющее услуги унифицированного управления уязвимостями.

Просмотр уязвимостей для конкретного порта.



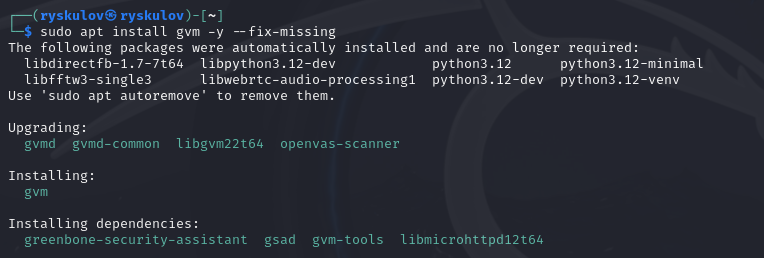
Рисунок 22. Фреймворк Metasploit, просмотр уязвимостей по 80 порту.

## ЗАДАНИЕ 1

Произведите сканирование цели с помощью GVM. Докажите это с помощью скриншотов.

**Ответ:**

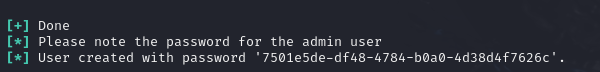
Устанавливаем GVM:



Настраиваем сканер:

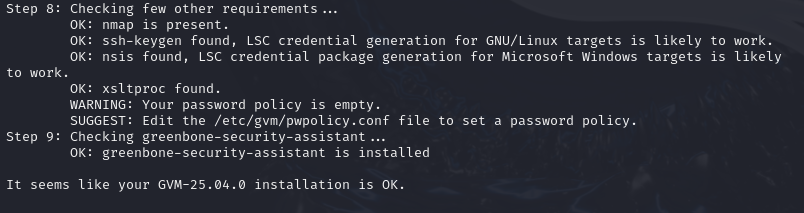


Получили пароль:

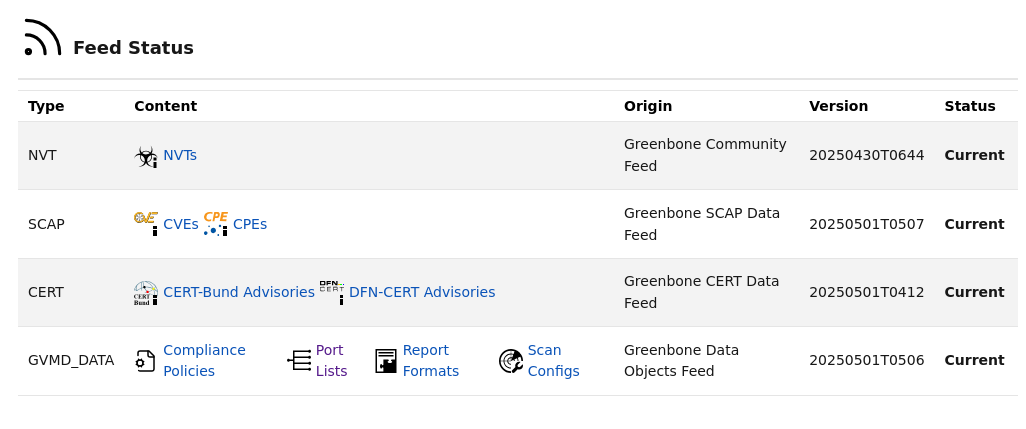


Проверим установку:

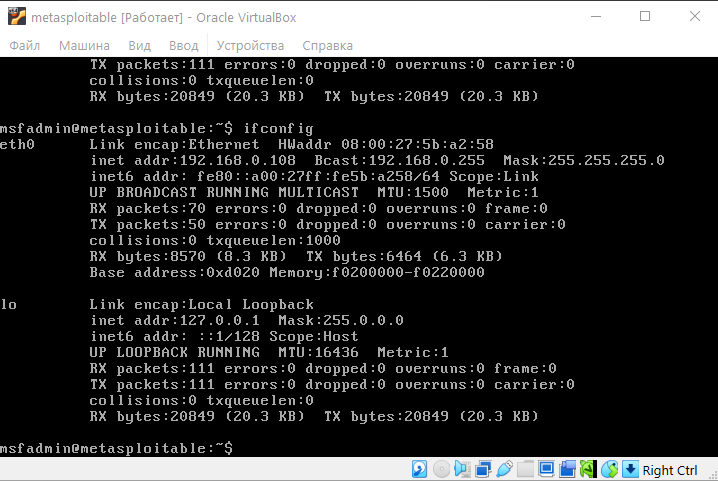




Дождался завершения обновлений:

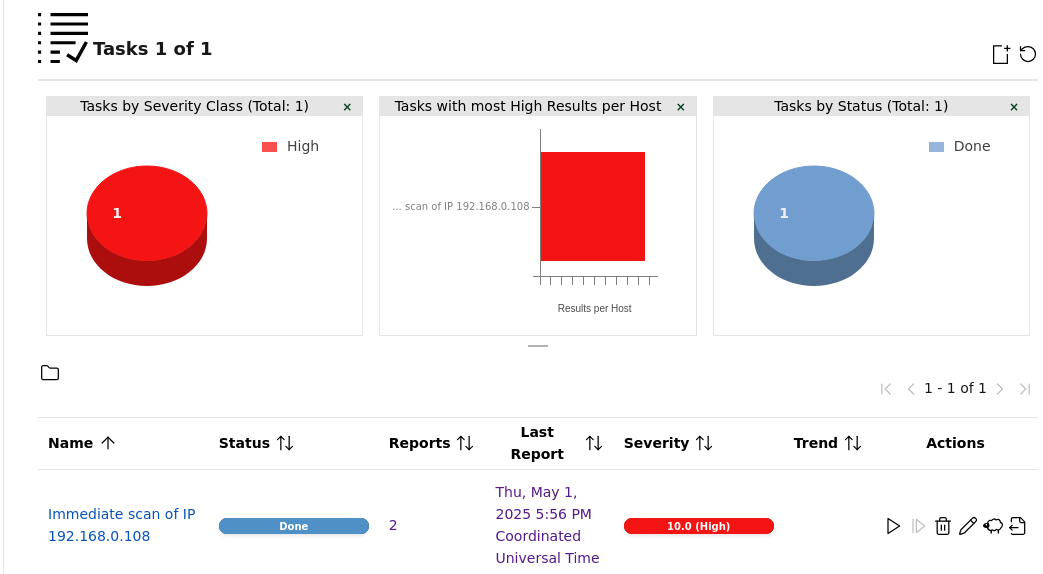


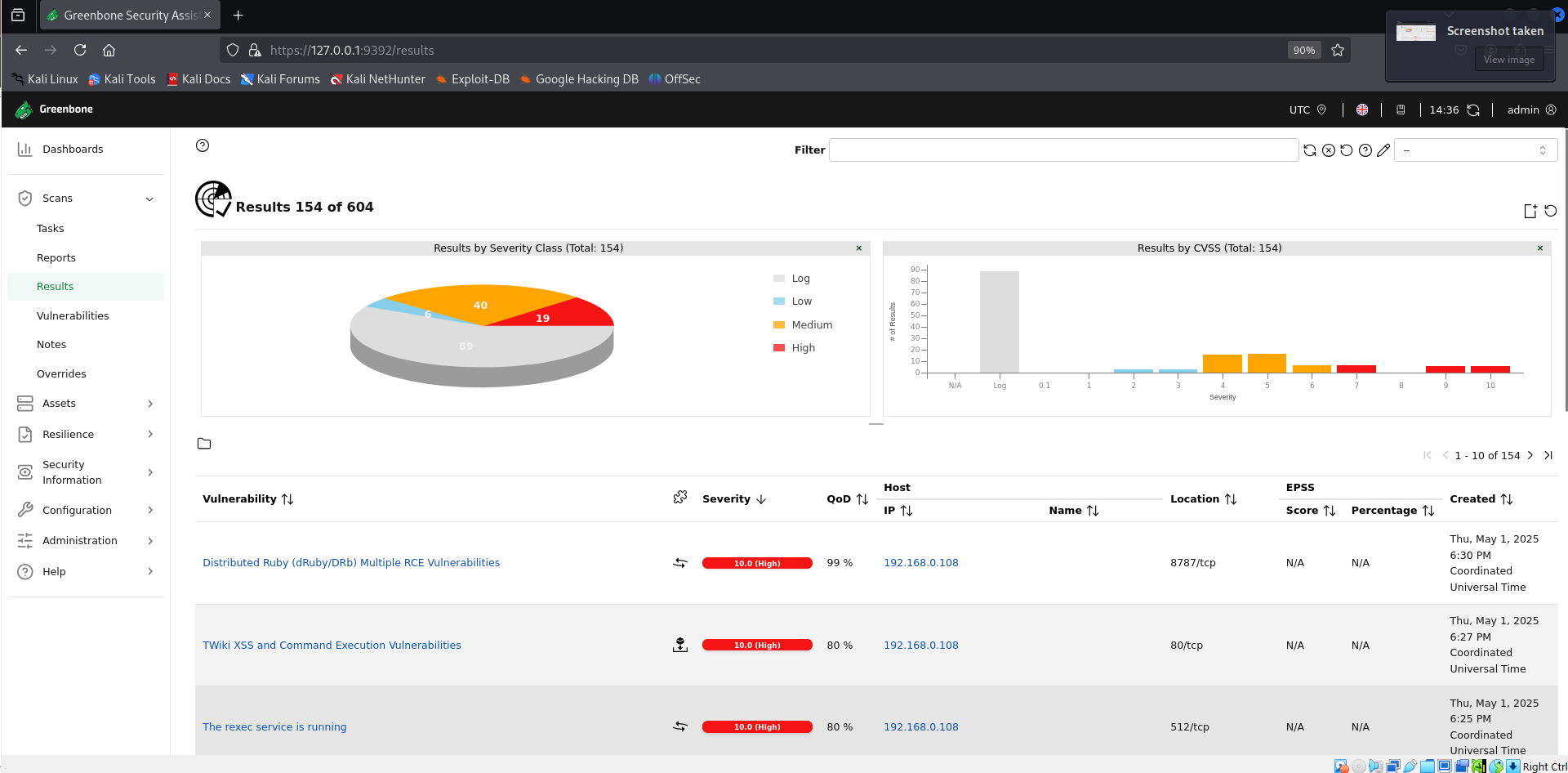
Запустим сканирование metasploitable:

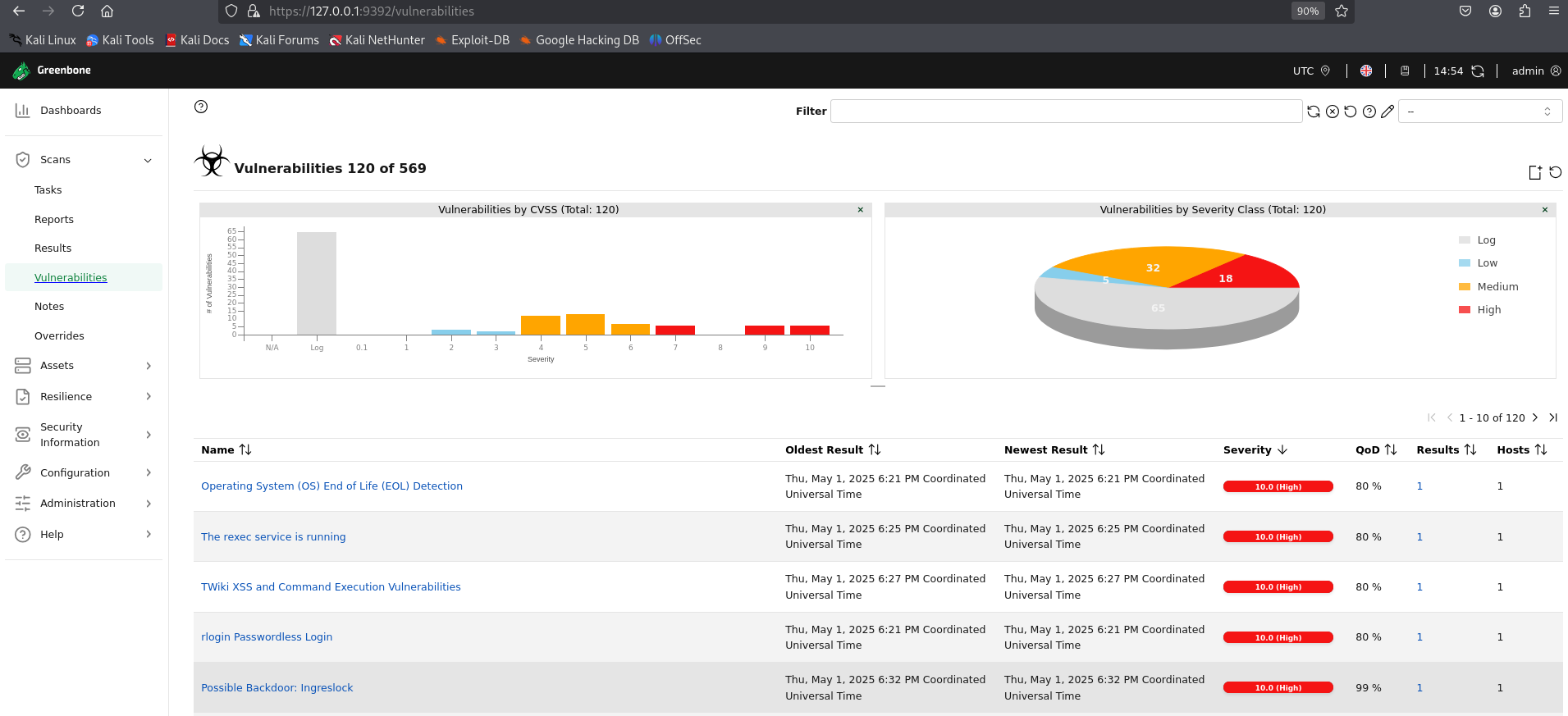




Сканирование завершено:



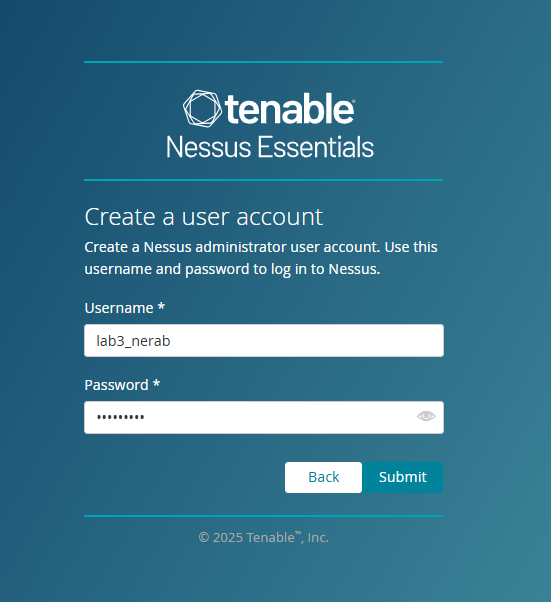


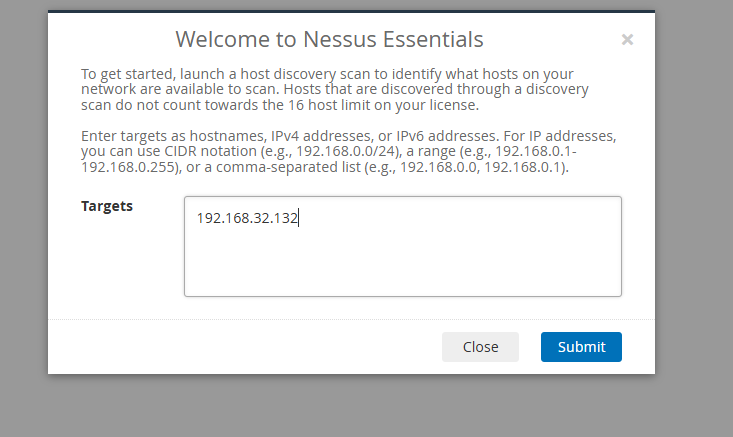


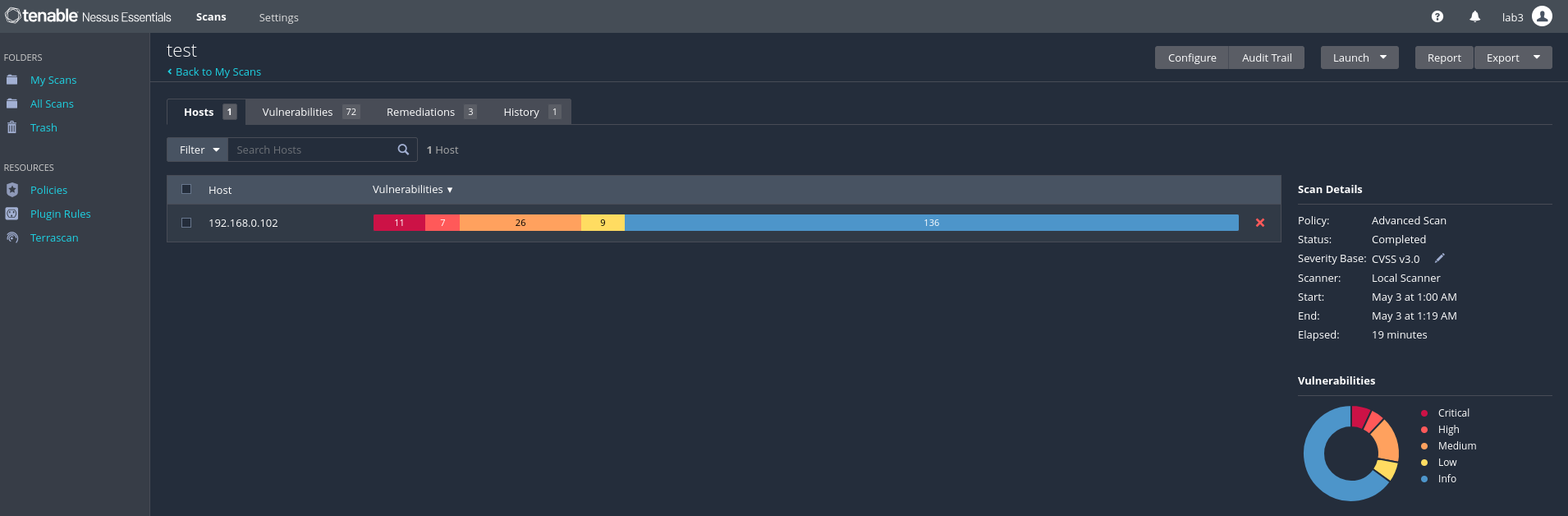
## ЗАДАНИЕ 2

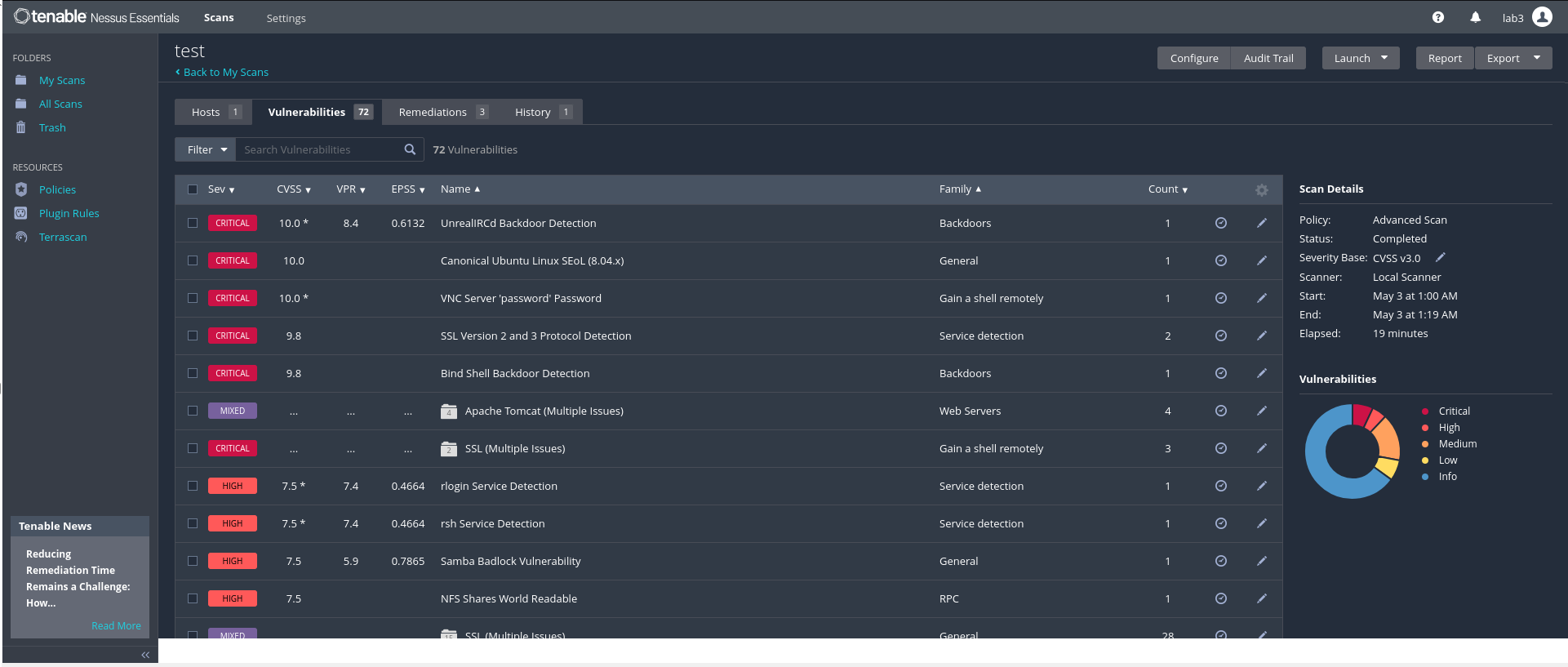
Произведите сканирование цели с помощью сканера Nessus. Докажите это с помощью скриншотов.

**Ответ:**

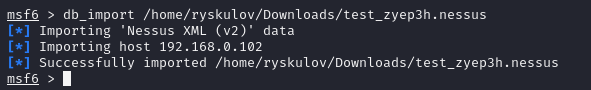




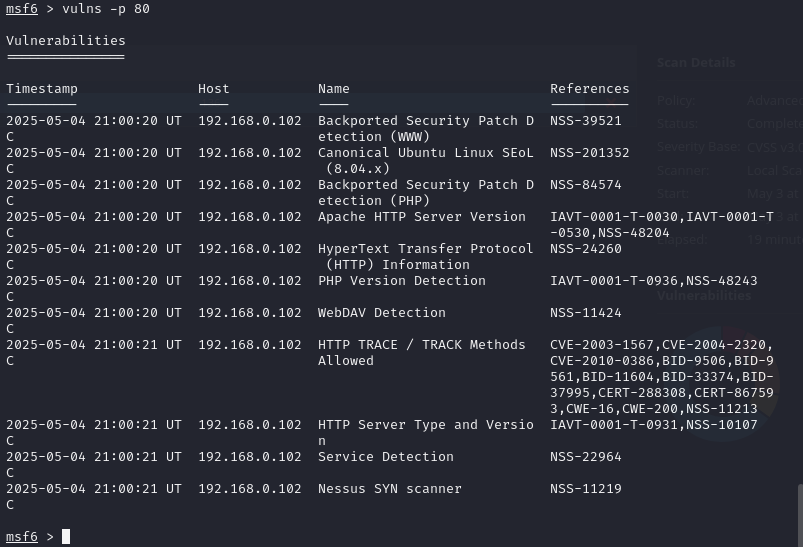




Импортируем результаты сканирования в базу данных metasploit framework:



Посмотрим уязвимости для 80 порта:



## ЗАДАНИЕ 3

Сравните результаты сканеров (результат предоставить в табличном виде) **Ответ:**

|  |  |  |
| --- | --- | --- |
| Критерий | GVM | NESSUS |
| Время сканирования | ~45 минут | ~20 минут |
| Количество уязвимостей | 65 | 72 |
| Выявлено портов | 18 | - |
| Операционная система | Ubuntu 8.04 | - |

# Упражнение 3. Обзор конфигурации

**Цель:**

понять, что такое обзор конфигурации

**После окончания работы студент должен**

* знать: подход к обзору конфигурации
* уметь: проводить обзор конфигурации **Задание:**
* Загрузите в Metasploitable 2 утилиту unix-privesc-check и проведите локальную проверку безопасности.
* Провести сканирование цели вместе с Nessus с "Политикой аудита исправлений учетных данных". Предоставьте учетные данные ssh.
* Произвести сканирование цели с помощью скрипта LinEnum.
* Сравните отчет, созданный Nessus, с отчетами, полученными утилитами unix-privesccheck и LinEnum.

## Технические инструменты для выполнения работы

* Kali Linux VM (Kali)
* Metasploitable 2 VM (цель)

## Ссылки

* <http://pentestmonkey.net/tools/audit/unix-privesc-check>
* <https://github.com/rebootuser/LinEnum>

## Порядок выполнения работы

Загрузите скрипт http://pentestmonkey.net/tools/audit/unix-privesc-check и запустите его. ./unix-privesc-check

Unix-privesc-checker-это сценарий, работающий в системных Unix (протестированный на Solaris, HPUX, разных Linux, FreeBSD). Он пытается найти “неправильные” конфигурации, которые могут позволить локальным непривилегированным пользователям повысить привилегии для других пользователей или получить доступ к локальным приложениям (например, базы данных). Скрипт легко загрузить и выполнить. Он может работать как от имени обычного пользователя, так и от имени пользователя root (от имени root, он сможет читать больше файлов).

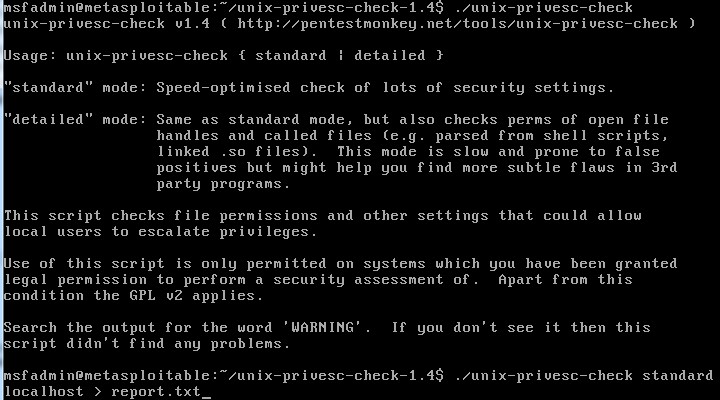


Рисунок 23. Запуск утилиты unix-privesc-check.

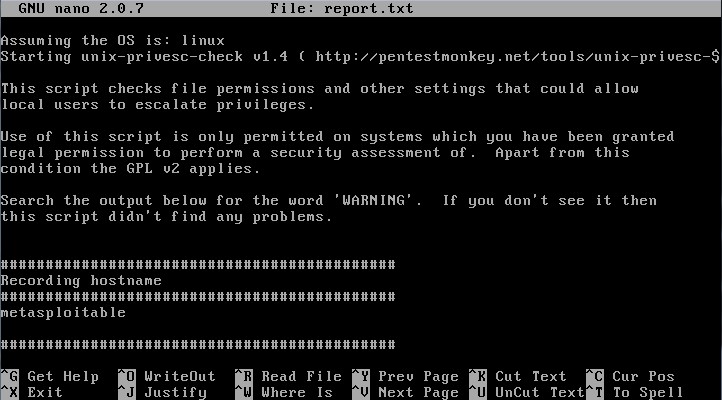


Рисунок 24. Обзор результата работы утилиты unix-privesc-check.

Выберите политику "Credential Patch Audit" (Аудит исправлений учетных данных)

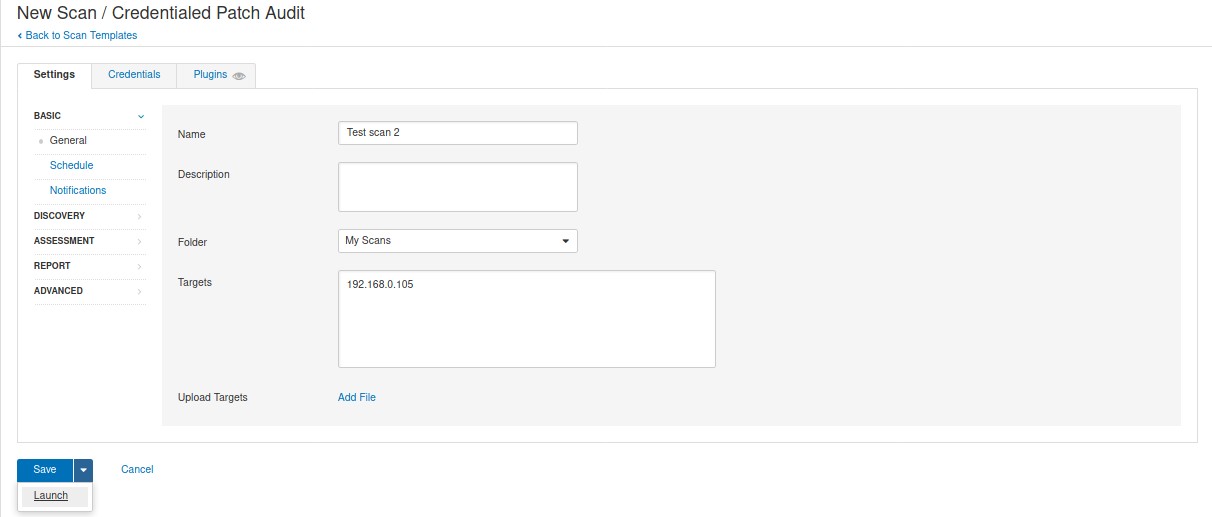


Рисунок 25. Сканер NESSUS, запуск аудита исправлений учетных данных.

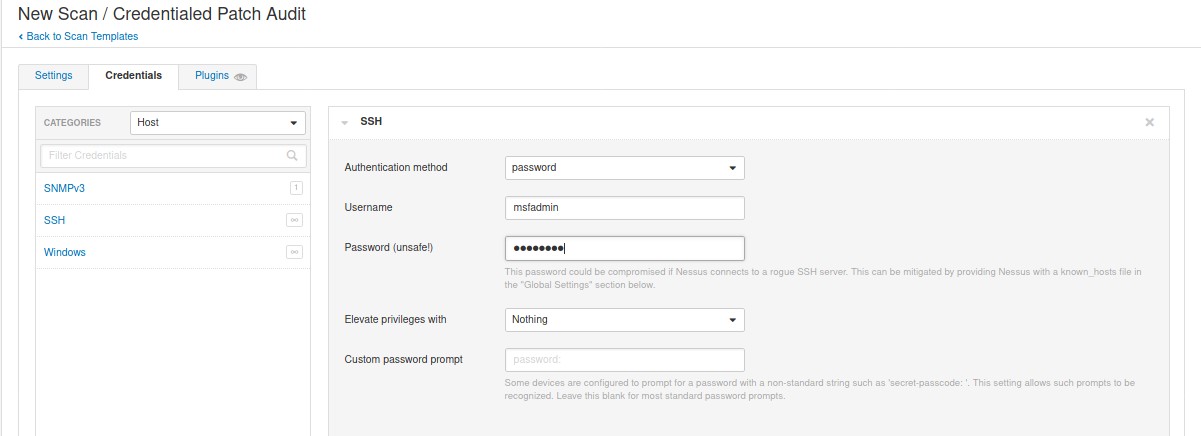


Рисунок 26. Сканер NESSUS, предоставление учетных данных ssh.

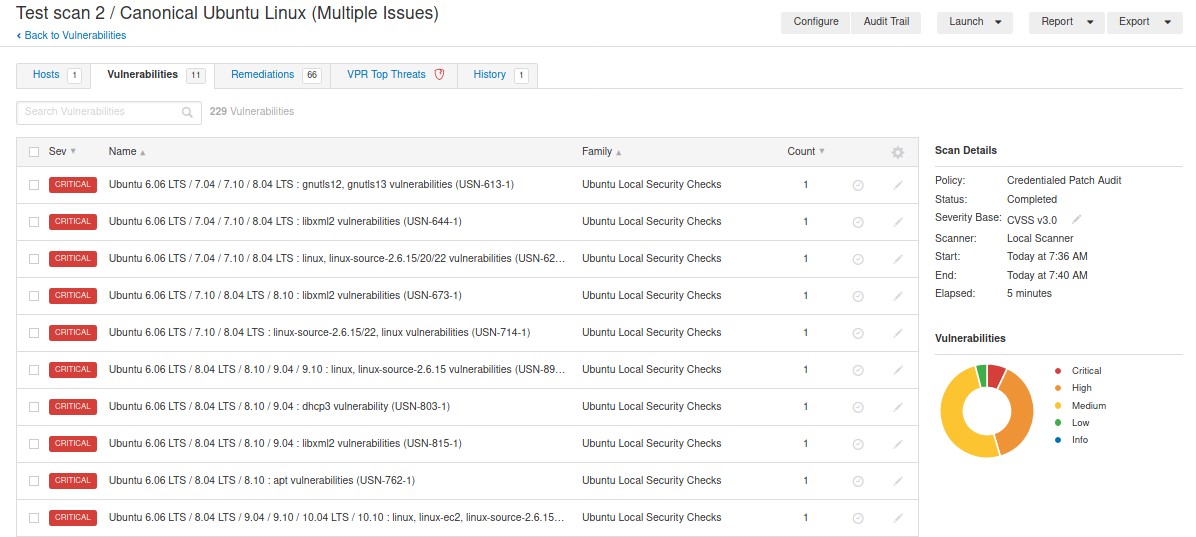


Рисунок 27. Сканер NESSUS, результат поиска уязвимостей.

С помощью скрипта LinEnum выполняют сбор всей возможной информации о системе, которая может привести к повышению привилегий.

Все проверки собраны в один файл Bash, который не требует дополнительных зависимостей. Программа не ищет и не пытается выполнять эксплойты – она только собирает информацию и делает проверки о возможных направлениях эскалации привилегий.

Для установки LinEnum в Kali Linux выполните следующие команды:

git clone https://github.com/rebootuser/LinEnum cd LinEnum/

Ознакомьтесь с основными аргументами из справки скрипта LinEnum ./LinEnum.sh –h

OPTIONS:

-k Enter keyword

(опциональный переключатель, с которым можно указать ключевое слово для поиска файлов из нескольких расширений) -e Enter export location

(устанавливает папку, куда будет сохранен отчет сканирования, например /tmp/export. Если эта папка не существует, она будет создана) -s Supply user password for sudo checks (INSECURE)

(использовать текущего пользователя с предоставленным паролем для проверки разрешений sudo)

-t Include thorough (lengthy) tests

(означает выполнять тщательные (медленные) тесты. Без этого переключателя будет выполнено стандартное 'быстрое' сканирование.) -r Enter report name

(определяет имя файла отчета. Отчет (.txt файл) будет сохранен в текущую рабочую директорию)

-h Displays this help text

(выводит справку на экран)

Используем Python для передачи файла нашей цели. Для этого используем модуль SimpleHTTPServer.

Модуль SimpleHTTPServer, поставляемый с Python, представляет собой простой HTTPсервер, предоставляющий стандартные обработчики запросов GET и HEAD

Этот модуль будет обслуживать содержимое любого каталога, в котором вы его запустите, на порту 8000.



Рисунок 28. Модуль SimpleHTTPServer, запуск HTTP-сервера.

Загрузим скрипт на целевую машину. Для этого используем каталог /var/tmp, хотя можно использовать любой каталог, доступный для записи.



Рисунок 29. Каталог /var/tmp на целевой машине..

Для передачи файлов используют такие утилиты как wget и curl, обычно они установлены в системах Linux и предлагают простой способ захвата файлов. Проверьте, что утилита wget установлена, с помощью какой команды: which wget.



Рисунок 30. Проверка наличия утилиты wget на целевой машине.

Используем утилиту wget для загрузки LinEnum на целевой хост.

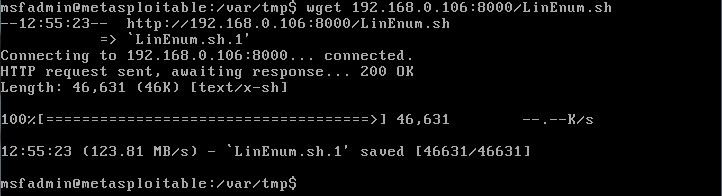


Рисунок 31. Загрузка скрипта LinEnum.sh с помощью утилиты wget на целевой хост.

Просмотрим разрешение, возможно ли выполнение скрипта?

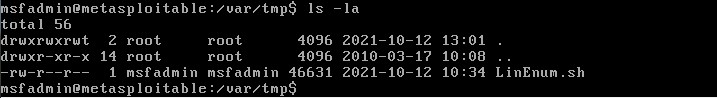


Рисунок 32. Проверка скрипта LinEnum.sh на исполнение.

Сделаем скрипт исполняемым с помощью команды chmod.

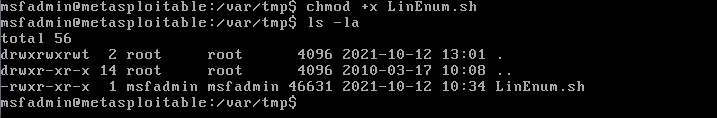


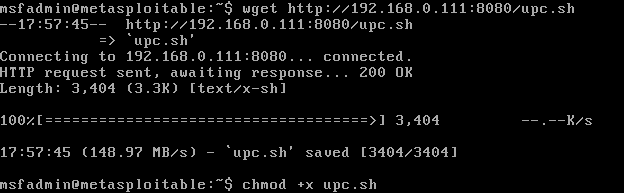
Рисунок 32. Делаем скрипт LinEnum.sh исполняемым.

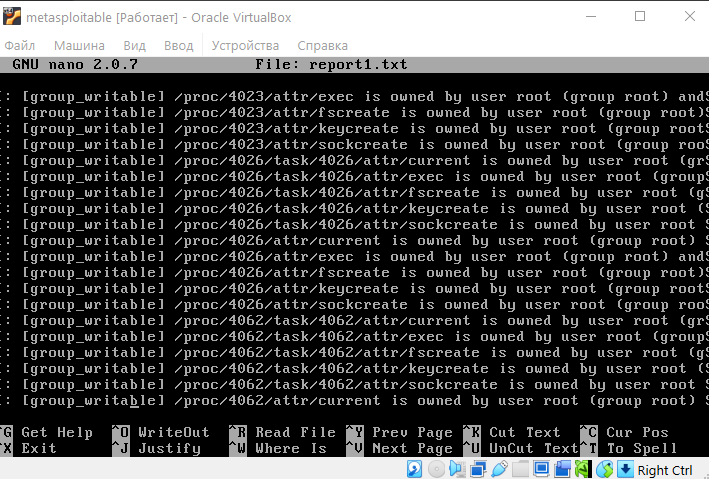
Теперь, когда все готово, остается только запустить LinEnum.

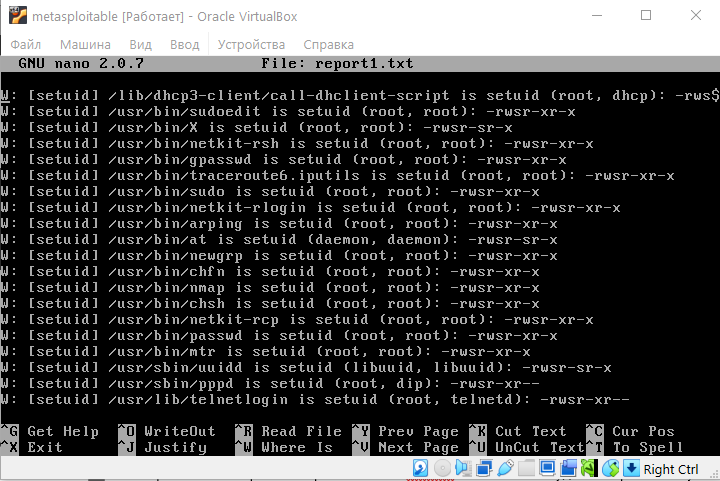
## ЗАДАНИЕ 1

Загрузить в Metasploitable 2 утилиту unix-privesc-check и провести локальную проверку безопасности. Докажите это с помощью скриншотов.

**Ответ:**





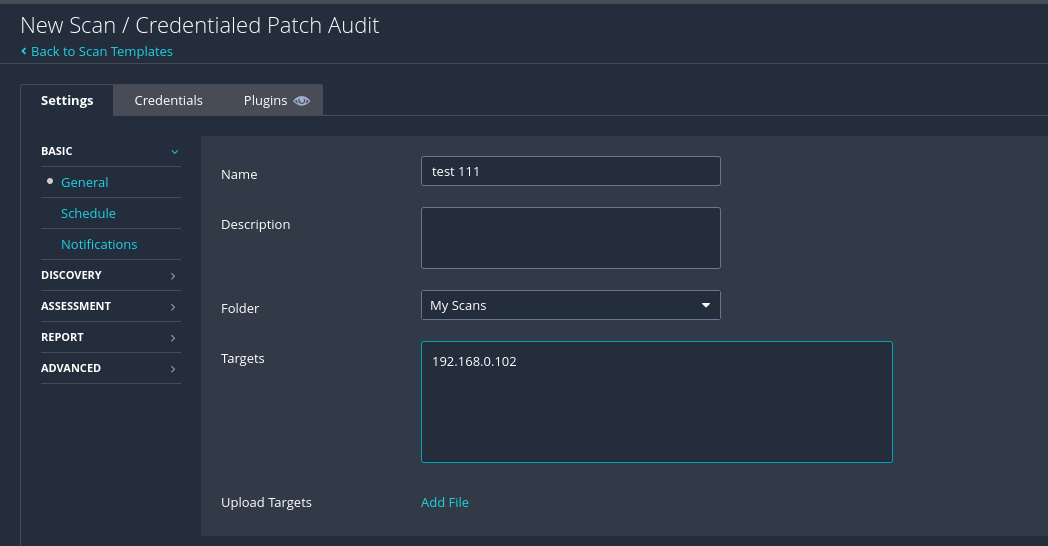


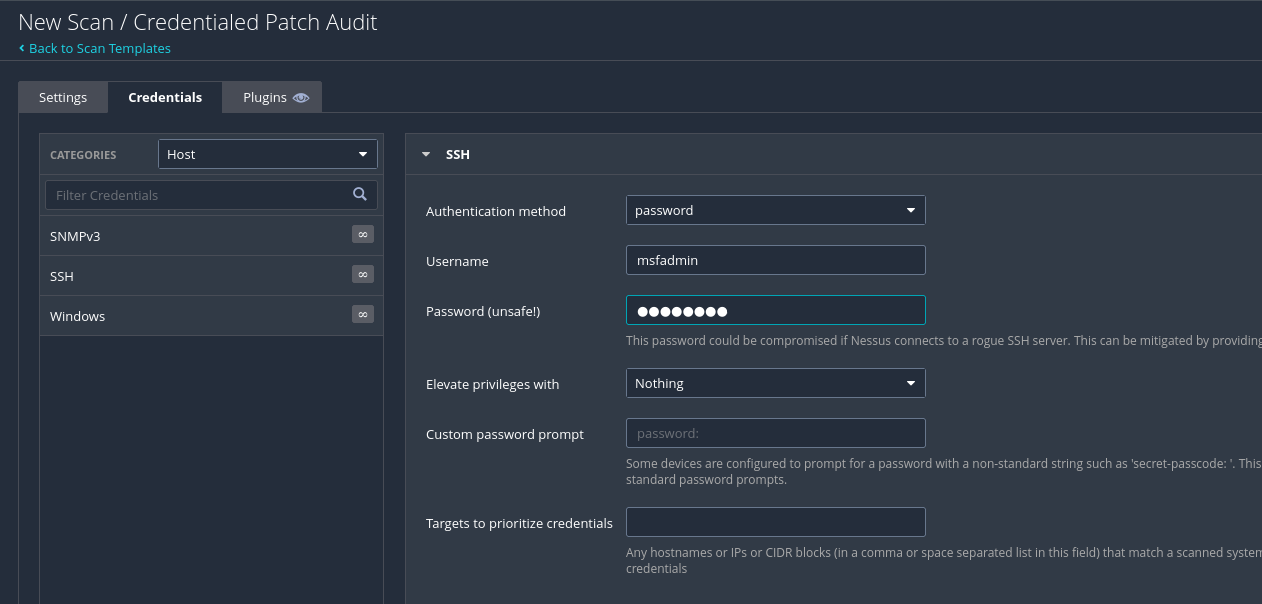
И еще много таких же строк. Не совсем понятные результаты.

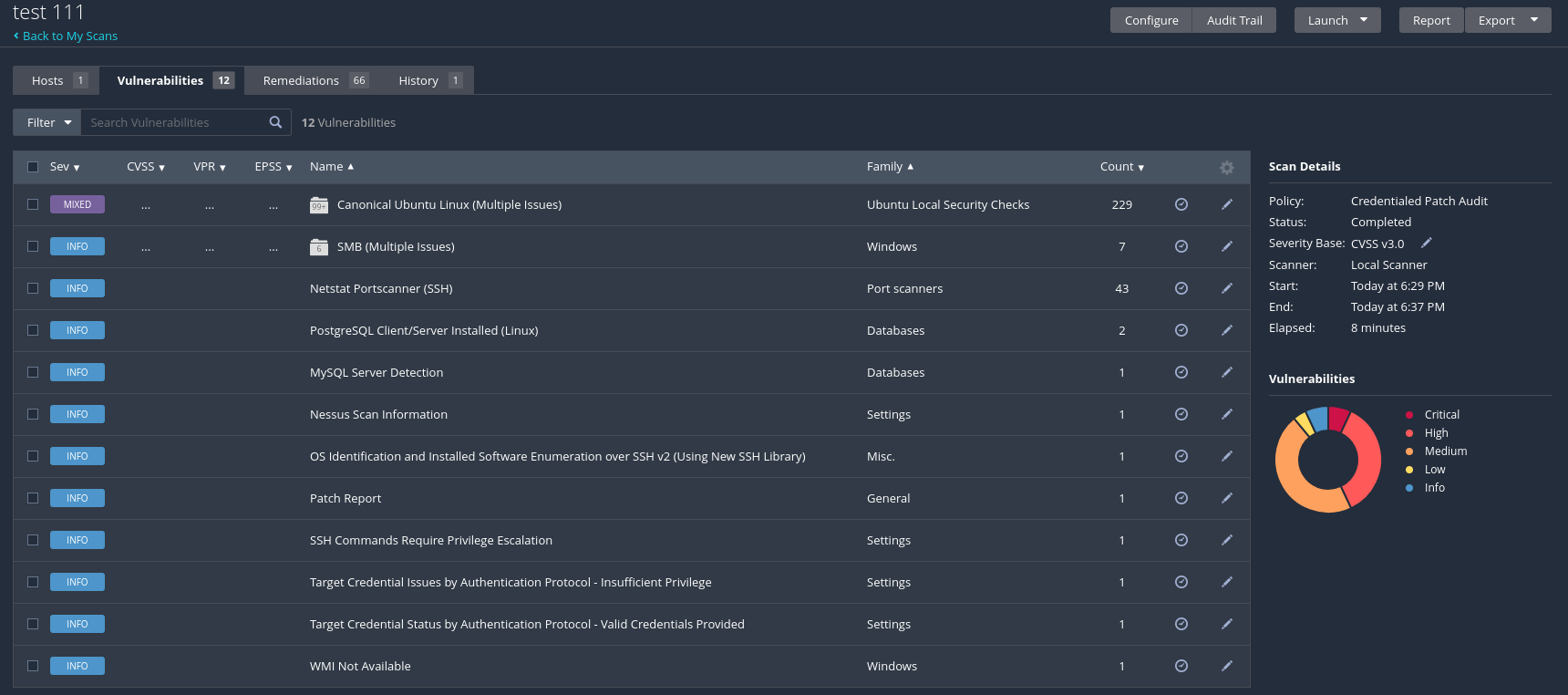
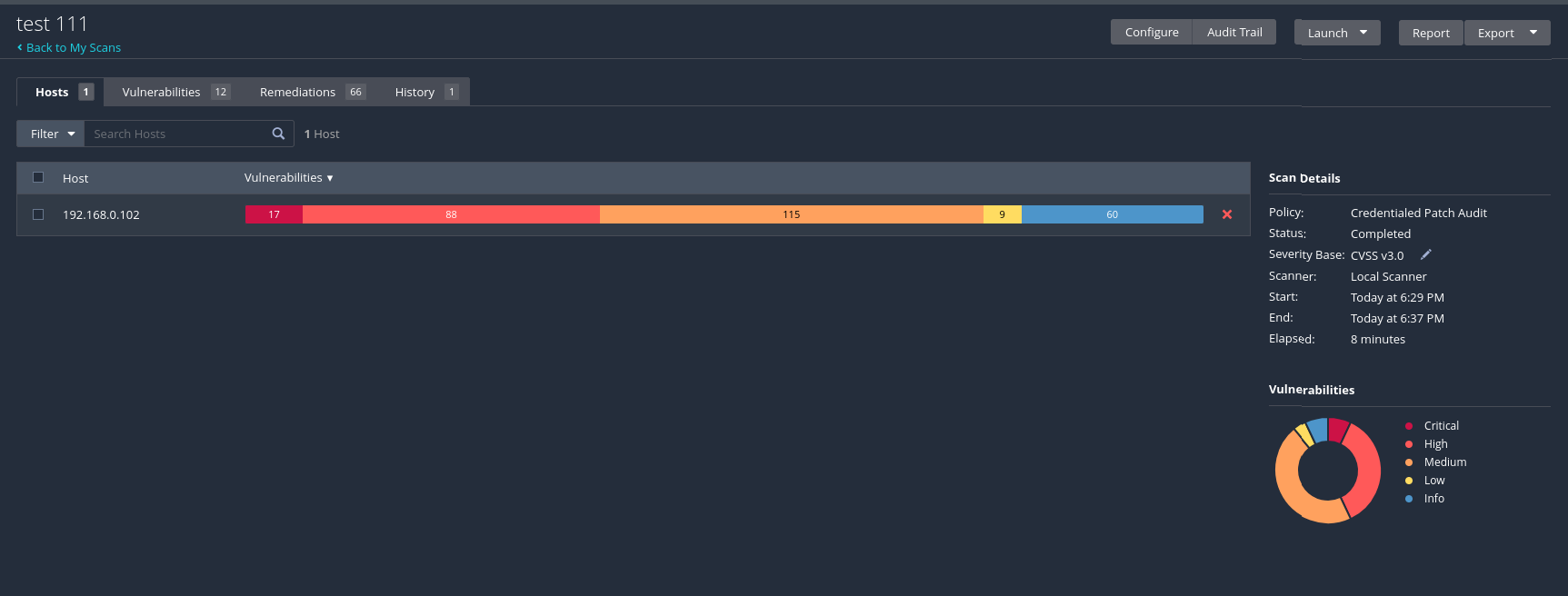
## ЗАДАНИЕ 2

Провести сканирование цели вместе с Nessus с "Политикой аудита исправлений учетных данных". Предоставьте учетные данные ssh. Докажите это с помощью скриншотов.

**Ответ:**





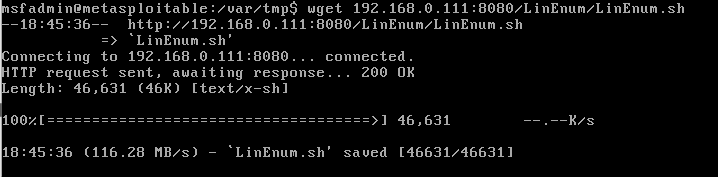


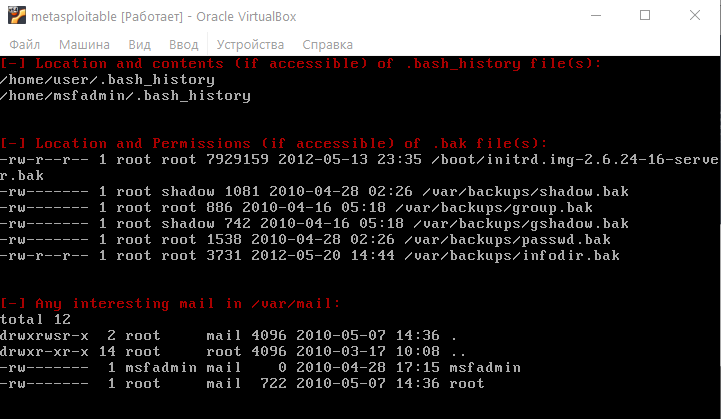


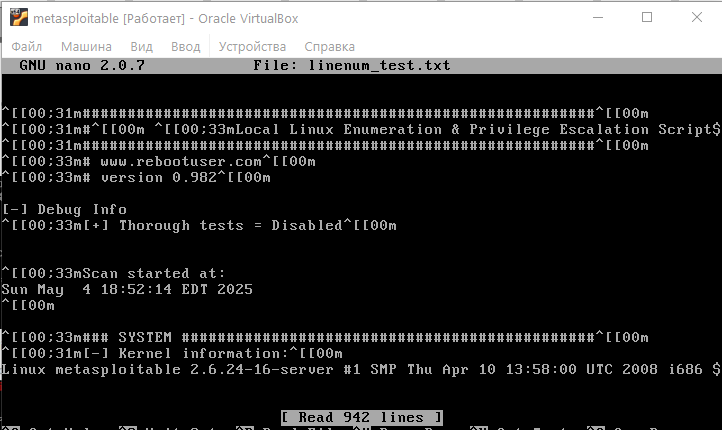
## ЗАДАНИЕ 3

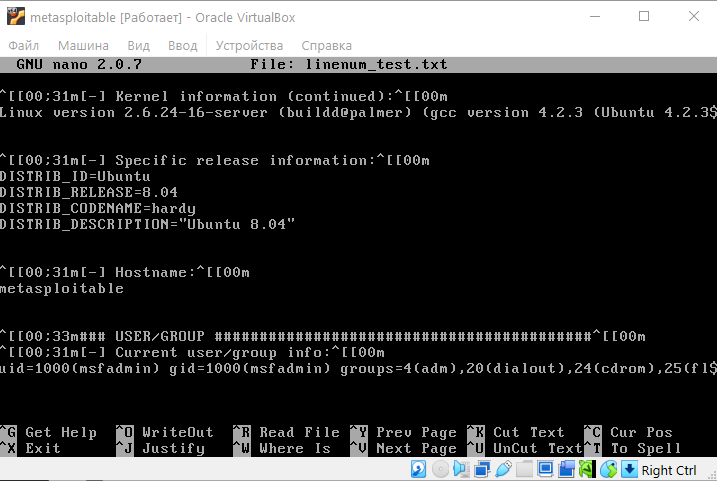
Произвести сканирование цели с помощью скрипта LinEnum. Докажите это с помощью скриншотов.

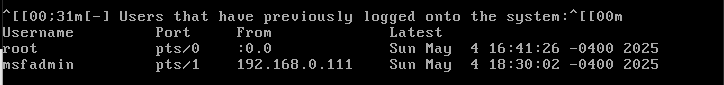
**Ответ:**

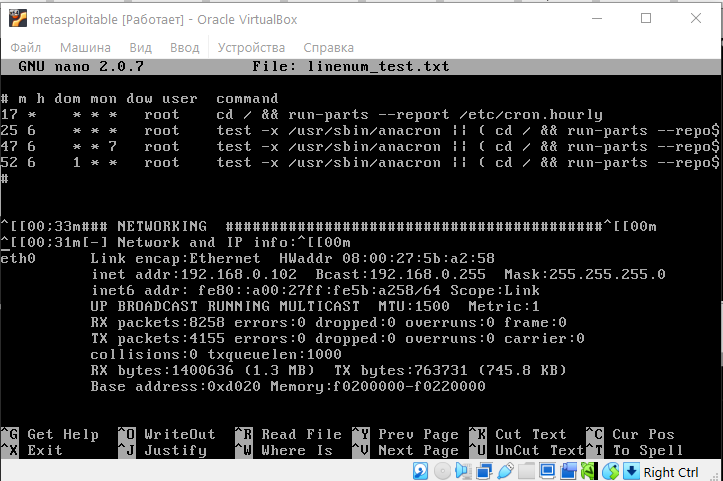


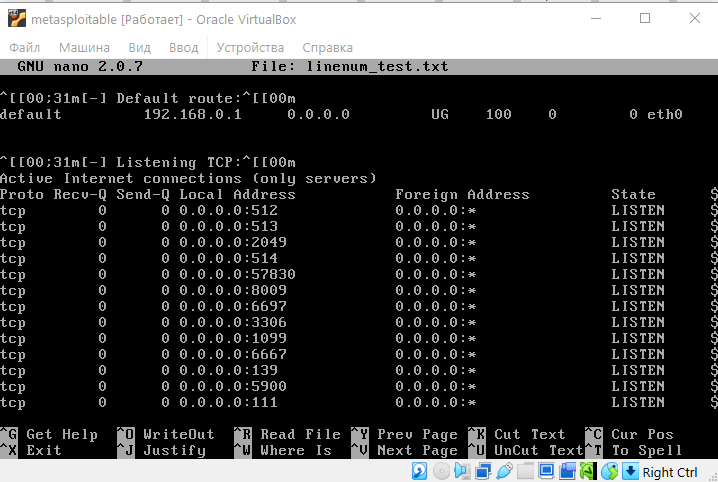












## ЗАДАНИЕ 4

Сравните отчет, созданный Nessus, с отчетами, полученными утилитами unix-privesc-check и LinEnum. Объясните с помощью сравнительной таблицы. **Ответ:**

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий сравнения | NESSUS | UNIX-PRIVESC | LINENUM |
| Время сканирования | 8 минут | 3-4 минуты | Несколько секунд |
| Информация про уязвимости | + | - | - |
| Информация про порты | + | возможно | + |
| Информация про сеть | - | - | + |

Нессус предоставил много информации, связанной с уязвимостями, также есть решения для них.  
Unix-privesc-check выдал много строк не совсем понятной информации.  
LinEnum выдал более общую информацию насчет целевой машины.