МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»

ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра компьютерной инженерии и моделирования

**Анализ уязвимости в сети с помощью инструментов Kali Linux**

Отчет по курсовой работе

по дисциплине «Компьютерные сети»

студента 2 курса группы ИВТ-б-о-202(1)  
Шор Константина Александровича

Направления подготовки 09.03.01«Информатика и вычислительная техника»

|  |  |  |
| --- | --- | --- |
| Научный руководитель  старший преподаватель кафедры компьютерной инженерии и моделирования | \_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | \_\_\_\_\_\_\_\_\_\_\_ |

Симферополь, 2022

**РЕФЕРАТ**

«Анализ уязвимости в сети с помощью инструментов Kali Linux» - Симферополь: ФТИ КФУ им. В. И. Вернадского, 2022. – 25с., 15 ил., 7 ист.

Объект исследования – инструменты Kali Linux, которые анализируют уязвимости для пентестинга и предотвращения возможного использования уязвимостей системы.

Цель работы – узнать какие бывают скарны сети и научится ими пользоваться.

Реализация проекта происходила в условиях домашней сети. Использовалась виртуальная машина, на которой была установлена Kali Linux. Анализ производился над маршрутизатором, Windows 10 и Windows 8.1 c отключенным брандмауэром.

KALI LINUX, УЯЗВИМОСТИ, NMAP, METASPLOIT, OPNEVAS, VIRTUAL OS.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 5](#_Toc101197300)

[ГЛАВА 1 KALI LINUX 6](#_Toc101197301)

[**1.1.** **О Kali Linux** 6](#_Toc101197302)

[**1.2.** **Инструменты Kali Linux** 7](#_Toc101197303)

[**1.2.1.** **Information Gathering (Сбор информации)** 7](#_Toc101197304)

[**1.2.2.** **Vulnerability Analysis (Анализ на уязвимости)** 8](#_Toc101197305)

[**1.2.3.** **Web Application Analysis (Анализ Веб приложения)** 9](#_Toc101197306)

[**1.2.4.** **Database Assessment (Оценка базы данных)** 9](#_Toc101197307)

[**1.2.5.** **Password Attack (Атаки на пароли)** 9](#_Toc101197308)

[**1.2.6.** **Wireless Attacks (атаки на беспроводные сети)** 10](#_Toc101197309)

[**1.2.7.** **Reverse Engineering (Реверсная инженерия)** 11](#_Toc101197310)

[**1.2.8.** **Exploitation Tools (Инструменты эксплоита)** 11](#_Toc101197311)

[**1.2.9.** **Sniffing & Spoofing (Сниффинг и спуфинг)** 11](#_Toc101197312)

[**1.2.10.** **Post Exploitation (Закрепление успеха)** 12](#_Toc101197313)

[**1.2.11.** **Forensics (Оценка)** 12](#_Toc101197314)

[**1.2.12.** **Reporting Tools (Инструменты отчёта)** 13](#_Toc101197315)

[**1.2.13.** **Social Engineering Tools (Инструменты социальной инженерии)** 13](#_Toc101197316)

[**1.3.** **Как работать в терминале:** 13](#_Toc101197317)

[Глава 2 АНАЛИЗ С ПОМОЩЬЮ ИНСТРУМЕНТОВ 14](#_Toc101197318)

[**2.1.** **Nmap** 14](#_Toc101197319)

[**2.1.1.** **Host Discovery (Обнаружение хостов)** 14](#_Toc101197320)

[**2.1.2.** **Port Scanning Techniques (Сканирование портов)** 15](#_Toc101197321)

[**2.1.3.** **Service and Version Detection (Обнаружение служб и их версий)** 16](#_Toc101197322)

[**2.1.4.** **OS Detection (Определение ОС)** 16](#_Toc101197323)

[**2.1.5.** **Timing and Perfomance (Опции управления временем)** 17](#_Toc101197324)

[**2.1.6.** **Firewall/IDS Evasion and Spoofing (Обход брандмауэра)** 17](#_Toc101197325)

[**2.1.7.** **Output (Вывод результатов)** 17](#_Toc101197326)

[**2.1.8.** **Nmap Scripting Engine** 18](#_Toc101197327)

[**2.2.** **OpenVAS** 20](#_Toc101197328)

[**2.3.** **Metasploit** 21](#_Toc101197329)

[**2.1.1.** **Exploit (Эксплоит)** 22](#_Toc101197330)

[**2.1.2.** **Payload (Полезная нагрузка)** 22](#_Toc101197331)

[**2.1.3.** **Post (Послеэксплуатационый)** 22](#_Toc101197332)

[**2.1.4.** **Encoder** 22](#_Toc101197333)

[**2.1.5.** **NOP** 22](#_Toc101197334)

[**2.1.6.** **Auxiliary** 22](#_Toc101197335)

[ЗАКЛЮЧЕНИЕ 24](#_Toc101197336)

[ЛИТЕРАТУРА 25](#_Toc101197337)

# ВВЕДЕНИЕ

На сегодняшний день информационная безопасность является актуальным направлением начиная с силовых структур и заканчивая огромными предприятиями. Информация становится таким же ценным товаров, как природные ресурсы, её можно обменивать, продавать, покупать и использовать. История любой компании зависит от обеспечения безопасности данных, которая она использует. Одним из направлений защиты безопасности данных является анализ на уязвимости.

Уязвимость – это слабое место информационного актива или средства управления и контроля, которые могут быть использованы хакерами. Если простым языком, то речь идёт о ошибках и недостатках программ, которые негативно влияют на безопасность.

Сама по себе уязвимость не представляет никакой опасности, она лишь является воротами для исполнения угроз. Самые частые причины возникновения уязвимости – ошибка проектирования и человеческий фактор.

Тестирование на проникновение – эффективный метод анализа защищённости. Он позволяет выявить уязвимые места в корпоративной инфраструктуре и получить объективную оценку ее текущего уровня защищенности. В ходе тестирования на проникновение моделируются действия потенциального нарушителя, осуществляющего атаки как со стороны сети. Такой подход позволяет воссоздать ситуацию, наиболее приближенную к реальным условиям и устранить недостатки защиты.

# ГЛАВА 1 KALI LINUX

* 1. **О Kali Linux**

Kali полностью повторяет сбору BackTrack Linux и придерживается стандартов разработки Debian. Была пересмотрена вся инфраструктура и инструменты, также Kali перешла на Git. Предназначена для проведения тестирования на проникновение и других процедур информационной безопасности.

Проект был создан в 2012 году, когда разработчики Offensive Security решили обновить свой проект под названием Linux BackTrack, который поддерживался вручную и мог быть превращен в настоящий дистрибутив Debian со всеми необходимыми инструментами. Спустя некоторое время, они решили создать Kali поверх дистрибутива Debian, который известен своим качеством и стабильностью.

В течение нескольких лет после выпуска версии Kali 1.0 произвела много обновлений благодаря новой версии ядра, расширив диапазон доступных приложений. Пользователи всегда могут создавать свои собственные образы в реальном времени, что является уникальной особенностью дистрибутива.

В 2015 году, когда был выпущен Debian 8, велась работа по портированию Kali Linux. Хотя Kali Linux не использует GNOME (вместо использования GNOME Fallback), оболочка была улучшена в новой версии. Кроме того, были добавлены некоторые расширения GNOME для реализации отсутствующих функций, особенно меню «Приложение». После всех разработок дистрибутив Kali Linux 2.0 был выпущен в августе 2015 года. GNOME – стандартный рабочий стол на Kali:

* Более 300 инструментов для проведения тестирования на проникновение
* Бесплатный и всегда будет бесплатным
* Git дерево с открытым источником кода
* FHS совместимый (позволяет найти исполняемые файлы, файлы поддержки, библиотеки и т.д.)
* Обширная поддержка беспроводных устройств
* Специальное ядро пропатчено от инъекций
* Безопасная среда разработки
* GPG подписанные пакеты и репозитории (все пакеты подписываются каждым отдельным разработчиком)
* Многоязычность
* Полностью настраиваемый
* Дизайн single user, root access
* Сетевые сервисы отключены по умолчанию
* Поддержка ARMEL и ARMHF. Kali в настоящие время доступна для следующим ARM-устройств:
* Rk3306 mk/ss808
* Raspberry
  1. **Инструменты Kali Linux**

**Разделы инструментов:**

* + 1. **Information Gathering (Сбор информации)**

Инструменты разведки, используются для сбора данных по целевой сети и устройствам:

* **DNS Analysis** – анализ DNS:
* **IDS/IPS identification –** определение наличия Intrusion Detection System (Систем обнаружения вторжения) или Intrusion Prevention System (Систем предотвращения вторжения):
* **Live Host identification –** сканирование на наличие хостов (компьютеров и прочего оборудования) в сети и подсети:
* **Network & Port Scanners –** сканирование портов на устройстве:
* **OSINT Analysis – (**Open-Source Intelligence) сбор и сопоставление данных из интернета:
* **Route Analysis –** анализ маршрутов:
* **SMB Analysis –** анализ Server Message Block (сетевых протокол прикладного уровня) в Windows:
* **SMTP Analysis –** анализ передачи электронной почты.
* **SNMP Analysis –** анализ маршрутизаторов, серверов, коммутаторов, принтеров, рабочих станций, модемов:
* **SSL Analysis –** анализ приборов, использующих протокол шифрования SSL (Secure Sockets Layer – уровень защищённых сокетов), который кодирует данные:

**Дополнительные инструменты** –dmitry, ike-scan, legion, netdiscover, nmap, recon-ng, spiderfoot.

* + 1. **Vulnerability Analysis (Анализ на уязвимости)**

Инструменты, сфокусированные на оценки уязвимостей. Обычно, они основываются на информации, полученной с помощью инструментов для разведки (Information Gathering):

* **Fuzzing Tools –** фаззинг - один из методов тестирования ПО методом «чёрного ящика», который заключается в автоматизированном поиске ошибок с помощью инъекций деформированных данных. Ошибки в работе программы (зависание, прекращение работы) – свидетельствует о нахождении уязвимости:
* **VoIP Tools –** инструмент проверки безопасности, проверяет, может ли компьютер имитировать поведение IP телефона. Он быстро автоматизирует VLAN Hop в Voice VLAN:

**Дополнительные инструменты** – legion, nikto, nmap, unix-privesc-check**.**

* + 1. **Web Application Analysis (Анализ Веб приложения)**

Проверка и использование уязвимостей в веб- серверах.

* **CMS & Framework Identification –** определение Content Management System (Система управления контентом). Узнав версию CMS, можно использовать уже существующие уязвимости на сате:
* **Web Application Proxies –** программы (прокси), которые находятся между браузером и веб сайтом, перехватывая весь трафик между ними:
* **Web Crawlers & Directory Bruteforce –** получение определённой информации с веб-сайтов с помощью бота или автоматизированного скрипта и перебор каталогов, файлов в веб-приложениях:
* **Web Vulnerability Scanners –** сканирование, разведка, проникновение в веб-приложения:

**Дополнительные инструменты** – burpsuite, commix, skipfish, sqlmap, wpscan

* + 1. **Database Assessment (Оценка базы данных)**

Инструменты, используемые для создания проектирование и редактирования фалов баз данных, а также использование такой уязвимости, как SQL-инъекция.

**Дополнительные инструменты** - SQLite database browser, Sqlmap.

* + 1. **Password Attack (Атаки на пароли)**

Инструменты для взлома паролей.

* **Offline Attacks –** для непосредственной физической атаки на пароли, с целью получить более высокие привилегии:
* **Online Attacks –** не всегда получается физически атаковать устройство, и именно поэтому существует онлайн атака. Инструменты этого раздела используют множество протоколов: FTP, HTTP, HTTPS, MySQL, Oracle, IMAP, VNC и другие. Однако атаки этого типа являются «шумными»:
* **Passing the Hash (PtH) Tools –** программы, использующие технологию перехвата хэша паролей. После перехвата можно просто передать хэш для аутентификации и получить доступ к системе. Интересно то, что хэш расшифровывать не нужно, поскольку он остаётся статичным до тез пор пока пароль не будет изменён. Как правильно хэш получают путём обхода активной памяти, но есть и другие способы:
* **Password Profiling & Wordlists –** раздел, в котором создаётся настроенный список паролей, который подбирается к каждой личности индивидуально. Корректно созданный список паролей значительно уменьшает время перебора и увеличивает шансы на успех атаки:

**Дополнительные инструменты** – hashcat, john, medusa, ncrack, ophcrack.

* + 1. **Wireless Attacks (атаки на беспроводные сети)**

Сигнал Wi-fi может быть обнаружен кем угодно – это делает устройства Wi-fi очень уязвимыми. Взлом включает в себя перехват и взлом хешированного пароля. Стоит отметить, что у провайдеров и маршрутизаторов по умолчанию включена WPS (Wi-fi Protected Setup) защита. Протокол WPS отдает в сеть половину PIN кода, что существенно уменьшает количество вариантов ключа при брутфорсе. Примерно на перебор уходить до 11 часов, но это ещё не всё, у большинства роутеров PIN код вшит. Поэтому узнав его один раз можно получать доступ к роутеру независимо был сменён пароль WPA (Wi-fi Protected Access) или нет:

* **802.11 Wireless Tools** – инструменты, которые осуществляют реализацию атаки перебором WPS, а также взлом и восстановление ключей WPE (Wired Equivalent Privacy) и WPA:
* **Bluetooth Tools** – программы для автоматизации подмены или клонирования имени, класса и адреса Bluetooth. Эти функции помогают скрываться на виду у всех:

**Дополнительные инструменты –** aircrack-ng, kismet, pixiewps, reaver, wifite.

* + 1. **Reverse Engineering (Реверсная инженерия)**

Полезный набор инструментов, способных анализировать работу программы существует статическое и динамическое обратное проектирование. В статическом происходит анализ ассемблерного кода и результаты его функций. Динамический анализ – запускает код и наблюдает результат.

**Дополнительные инструменты –** clang, clang++, NASM shell, radare2.

* + 1. **Exploitation Tools (Инструменты эксплоита)**

После всех вышеперечисленных инструментов (сканирование, сбор информации, поиск уязвимости) настаёт основной этап взлома – использование уязвимости. Найти уязвимость – это конечно хорошо, но научится её использовать это уже совсем другое. Вред приложению наносит не сама уязвимость, а злоумышленник, который её использует.

**Дополнительные инструменты –** crackmapexec, **metasploit framework,** msf payload creator, serchsploit, social engineering toolkit, sqlmap.

* + 1. **Sniffing & Spoofing (Сниффинг и спуфинг)**

Сниффинг – процесс отслеживания всех пакетов данных, которые уходят в сеть. Используется для мониторинга и устранения неполадок, но злоумышленники используют с целью кражи конфиденциальных данных.  
Спуфинг – процесс ввода поддельной информации в трафик. Осуществляется путём отправки пакетов с неверным источником. Лучшая борьба со спуфинг использование цифровой печати.

* **Network Sniffers** - инструменты для анализа сетевого трафика, протоколов, обратной разработки и откладки сети:
* **Spoofing & MITM** – программы для подделки DNS, атаки на зашифрованные сетевые соединения SSL/TLS, воспроизведения реального фонового трафика:

**Дополнительные инструменты –** ettercap-graphicat, macchanger, minicom, mitmproxy, responder, wireshark

* + 1. **Post Exploitation (Закрепление успеха)**

Это этап проникновения, целью которого является сохраннее доступа к объекту. Обеспечивается поддержка доступа и получения более привилегированного уровня:

* **OS Backdoors –** инструменты для переноса и шифрования данных:
* **Tunneling & Exfiltration –** создание всевозможного тунелинга (соединения с удалённым узлом), преобразования исполняемого файла в пакетный и наоборот, общение с сервером, который ничего не знает о клиенте:
* **Web Backdoors –** инструменты, которые имитируют соединения типа Telnet и используют инъекции:

**Дополнительные инструменты –** mimikatz, powershell empire, powersploit.

* + 1. **Forensics (Оценка)**

Помогают определить, как была проведена атака и как на неё следует реагировать. Всё это можно назвать цифровая криминалистика. На данный момент цифровой мир заполнен вредоносными программами и вирусами, одним из важнейших навыков является умение защитить себя от них:

* **Forensic Carving Tools –** инструменты по восстановлению, удалению и разделению любых файлов:
* **Forensic Imaging Tools –** поддержка разных файлов изображений:
* **PDF Forensics Tools –** анализ и сканирование PDF документов:
* **Sleuth Kit Suite –** исследование образов дисков, фалов системы (NTFS, FAT, FFS, EXT2FS):

**Дополнительные инструменты –** ffind, fls, fsstat, hfind, icat-sleuthkit, ils-sleuthkit, img\_cat, img\_stat, istat, jcat, jlc, mactime-sleuthkit, mmcat, mmks, mmstat, sigfind, sorter, srch\_strings, tsk.

* + 1. **Reporting Tools (Инструменты отчёта)**

Для отчётов.

Это методы доставки информации, найденной во время исполнения проникновения

* + 1. **Social Engineering Tools (Инструменты социальной инженерии)**

Программы, которые помогают работать с людьми с целью получения информации для дальнейшего взлома.

* 1. **Как работать в терминале:**

Графическое окружение сделано очень хорошо, но несмотря на это приходится довольно часто работать с командной строкой. Для этого мы можем использовать программу «Terminal». Особенности работы:

* Регистр имеет значение (Folger и folder не одно и тоже)
* Папки и файлы, начинающиеся с точки, считаются скрытыми. (.folger)
* Tab дописывает команды
* История терминала сохраняется
* Прерывание команд осуществляется сочетанием Ctrl-C, Ctrl-D, Ctrl-Z.
* Для получения документации (мануала) используется команда **man**. (man lc)

# Глава 2 АНАЛИЗ С ПОМОЩЬЮ ИНСТРУМЕНТОВ

1. **Nmap**

**Nmap** – бесплатная программа на Kali Linux, которая имеет открытый код и уже предустановлена. В большинстве своём она используется для сканирования портов и сканирования при помощи скриптов на уязвимости, но также имеет функционал для определения ОС и обхода брандмауэре путём спуфинга. Она содержит большое количество команд, которые предназначены для конкретных случаев.

Статус портов:

1. open: Приложение принимает на порт пакет или запросы на соединение.
2. closed: Порт отвечает на запросы, но не используется ни каким приложением.
3. filtered: Запросы не доходят до порта, невозможно определить открыт он или закрыт.
4. unfiltered: Порт доступен, но Nmap не может его определить.
5. open|filtered: Возникает если открытый порт не отвечает
6. closed|filtered: Не определено закрыт порт или фильтруется.

Я рассмотрю часть из них, со всеми можно ознакомится через флаг -h. Команды можно комбинировать, как и диапазон сети, предусмотрено сохранение и чтение из разных типов файла:

1. **Host Discovery (Обнаружение хостов)**

Cоставление списков хостов сети, с которыми в дальнейшем можно работать.

-sP: Пинг сканирование – определяет активные хосты

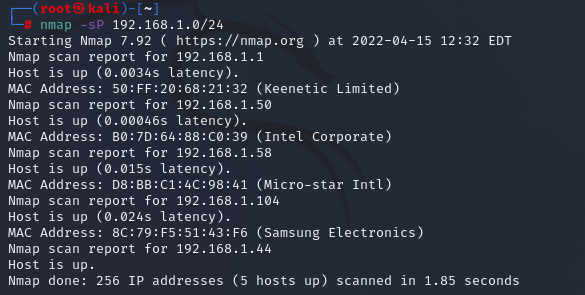


Рисунок 2.1.1 Обнаружение хостов

--traceroute: показывает путь до таргета

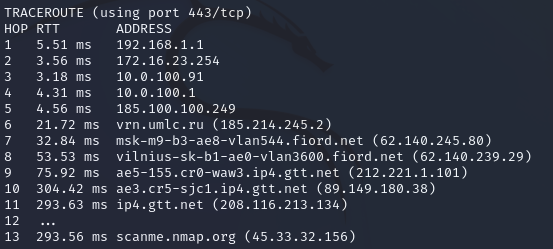
****

Рисунок 2.1.2 Демонстрация пути до цели

1. **Port Scanning Techniques (Сканирование портов)**

**-**sS: TCP соединение.

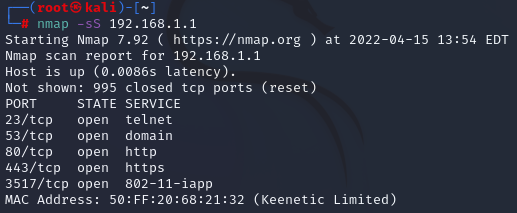


Рисунок 2.1.3 Сканирование TCP портов

**-**sU:UDP соединение**.** Так как UDP сканируется очень медленно использовалось уточнение какие именно порты нужно просканировать

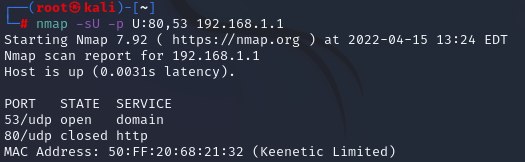
****

Рисунок 2.1.4 Сканирование UDP портов

**-**sA: Определение правил брандмауэре.

1. **Service and Version Detection (Обнаружение служб и их версий)**

-sV: определение информации о порте

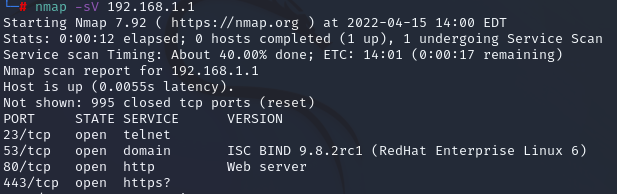


Рисунок 2.1.5 Обнаружение версий

1. **OS Detection (Определение ОС)**

-O:определение ОС. Странно, но Nmap определила, что у меня стоит Windows XP на 91%

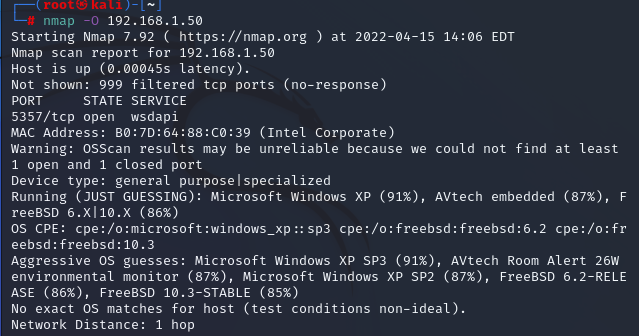
****

Рисунок 2.1.6 Определение Операционной системы

1. **Timing and Perfomance (Опции управления временем)**

Не буду рассматривать ручное управление временем, потому что это слишком сложное и узконаправленное, а рассмотрю шаблоны управления временем.

- T: всего их 6. Скорость идёт по нарастанию номера, каждый используется в конкретных случаях.

1. **Firewall/IDS Evasion and Spoofing (Обход брандмауэра)**

Нету уникальной формулы для обхода брандмауэра. Каждая ситуация уникальная и Nmap лишь даёт инструменты, которое помогут в этом.

-f: включает фрагментацию пакетов. Nmap, этим флагом, будет разбивать TCP заголовок для всех сканирований на небольшие фрагментированные пакеты.

-D: Создание фиктивных хостов. Цель будет думать, что все перечисленные хосты её сканируют

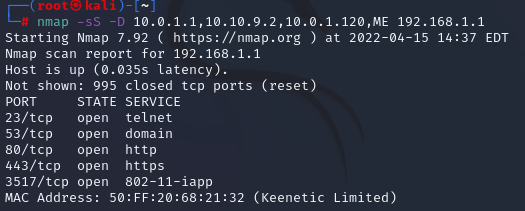


Рисунок 2.1.7 Пример создания фиктивных хостов

--spoof-mac: Подмена mac адреса.

1. **Output (Вывод результатов)**

Для анализа больших объёмов данных можно использовать сохранение данных в файл:

-oN: Вывод терминала

-oX: XML вывод.

-oG: Вывод в одну строчку

Нельзя не отметить функции:

-A: агрессивное сканирование – определение ОС и версий, сканирование при помощи скриптов и трассировки.

-v: динамический вывод – пошагово выводить работу в консоли.

1. **Nmap Scripting Engine**

Кто-то думает, что на этом возможности Nmap заканчиваются и они в корне будут не правы. Кроме сканирования портов этот инструмент имеет **NES** (Скриптовой движок) – движок, которой позволяет написать на Lua свой функционал сканирования, с помощью которого можно провести более обширный анализ. Существуют также уже написанные скрипты (более 500), которые, для удобства, разбили на категории:

* **auth –** сбор учётных данных (всё тоже сканирование)
* **broadcast –** для обнаружения не перечисленных хостов путём широковещательной передачи.

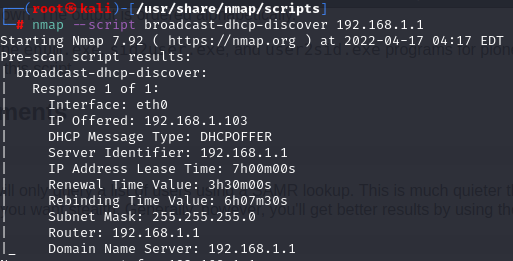
****

Рисунок 2.1.8 Обнаружение хостов широковещательной передачей

* **brute –** для проведения брутфорса (полного перебора).
* **default –** базовой набор скриптов (-sC -A)
* **discovery –** сбор учётных, регистрационных и статистических данных
* **dos –** проверка на уязвимость отказа в обслуживании

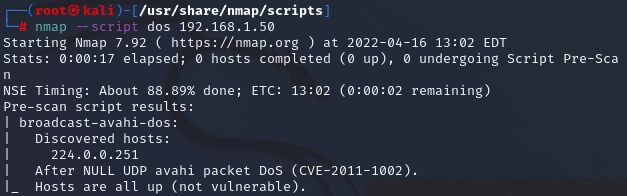
****

Рисунок 2.1.9 Анализ на отказ в обслуживание

Можно увидеть, что скрип нашел уязвимость CVE-2011-1002. Это отказ в обслуживание в Avahi. Уязвимость существует из-за не обработки UDP диаграмм и позволяет произвести DoS-атаку.

* **exploit –** эксплуатации известных уязвимостей
* **external –** анализ при помощи сторонних баз данных
* **fuzzer –** для выявления уязвимостей сервера, путём отправки неожиданных и рандомных полей в каждом пакете.
* **intrusive –** самые агрессивные скрипты, мой инстинкт самосохранения намекнул, что лучше пропустить эту категорию во избежание проблем.
* **malware –** проверка на заражение вирусами
* **safe –** сбор общей информации и исследование сети
* **version –** определение версии
* **vuln –** проверка известной уязвимости

Скриптов очень много, как и их способов разнообразного применения, поэтому разбирать каждый не вижу смысла. Проанализировав большинство категорий на своей сети, я получил, только наличие уязвимости CVE-2011-1002.

1. **OpenVAS**

**OpenVAS –** сканер уязвимостей и средство управления ими с открытым кодом. С помощью него можно мониторить узлы сети на наличие проблем с безопасностью и оценивать серьёзность проблем.

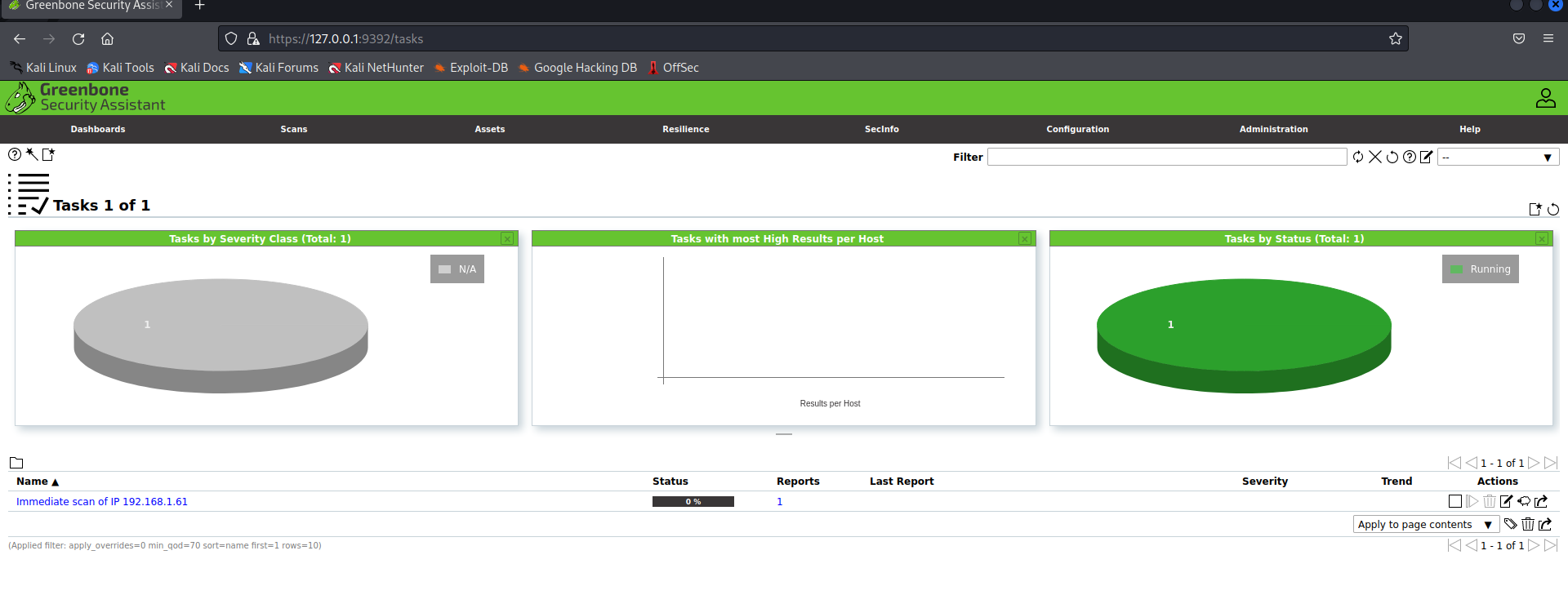
Он достаточно долго настраивается, а потом ещё дольше доканчивает недостающие обновления. Взаимодействие идёт через веб-браузер с интерфейсом: 

Рисунок 2.2.1 Интерфейс OpneVAS

Для сканирование была установлена Windows 8.1 с отключенным брандмауером.

После долгого анализа были выяснены следующие результаты:

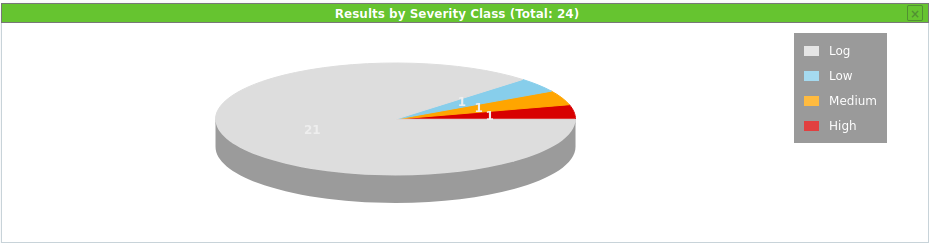
****

Рисунок 2.2.2 Часть результата анализа на уязвимости

Сканер имеет более подробное описание, но оно не помещается в word. Поэтому импортируем его в metasploit и проанализируем уже там.

1. **Metasploit**

**Metasploit –** фреймворк, который имеет широкий спектр применения в сфере информационной безопасности. Он имеет открытый исходный код, а также предустановлен в Kali Linux. На сегодняшний момент инструмент содержит более 1600 эксплоитов для более чем 20 платформ таких как: Android, PHP, Java, Cisco и других. Библиотека REX является сердцем metasploit, она требуется для работы с сокетами, протоколами, работы с кодировками, форматирование текста. На ней основывается библиотека MSF Core, благодаря которой осуществляется работа с API.

У metasploit большой функционал, но я покажу лишь его необходимую часть. После того как мы вошли в программу, нужно ввести пароль и создать базу данных. После в базу импортируем полученные xml файлы и теперь мы можем с ними работаь, для того чтобы посмотреть сервисы и уязвимсоти воспользуемся командами services и vulns:

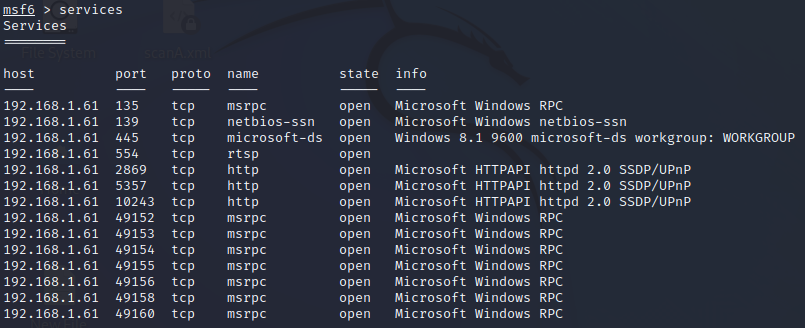


Рисунок 2.3.1 Импортированные данные портов

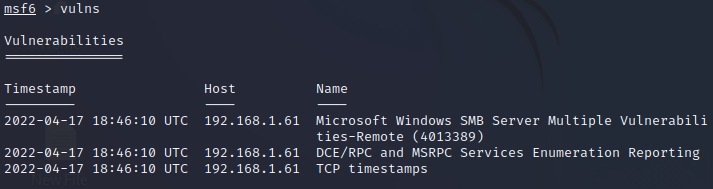


Рисунок 2.3.2 Импортированные данные уязвимостей

1. **E****xploit (****Эксплоит)**

Код, эксплуатирующий определенную уязвимость на целевой системе.

1. **Payload (Полезная нагрузка)**

Установка соединения. Используется для загрузки какой-то нагрузки в систему для того, чтобы остаться там. Они разбиты по типу операционной системы. Дополнительно его используют для скачивания модулей и конечным соединением.

1. **Post (Послеэксплуатационый)**

Код, который запускается после успешного проникновения. Необходим для отключения файрволла, антивируса и дополнительное защитное ПО с целью облегчения повторного захода в систему.

1. **E****ncoder**

Инструменты для маскировки модулей от антивирусов.

1. **N****OP**

Ассемблерная инструкция, которая не производит никаких действий. Используется для корректировки необходимого размера в файлах

1. **A****uxiliary**

Модули для сканирования и анализа трафика. Просканировав нашу Windows 8.1 можно обнаружить что она имеет уязвимость:

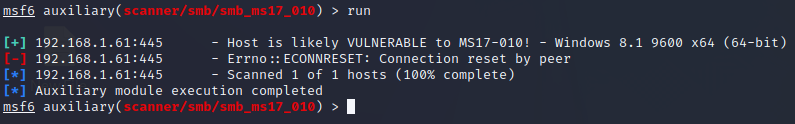


Рисунок 2.3.3 Обнаружение уязвимости SMB

Теперь наша уязвимость добавлена в базу и с ней уже можно дальше работать и эксплуатировать.

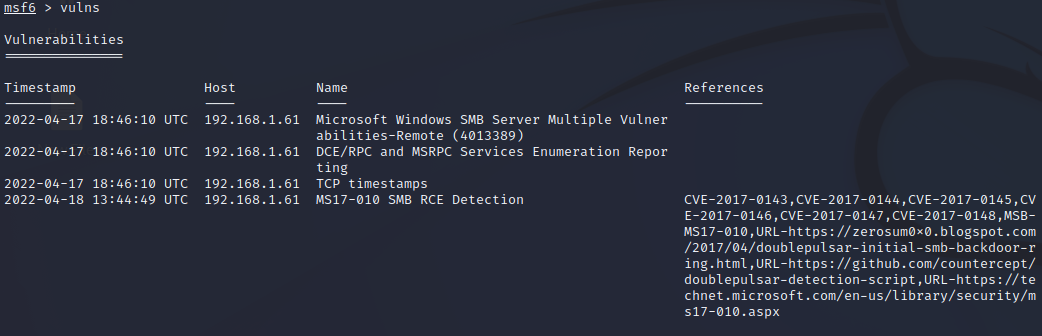


Рисунок 2.3.4 Добавление уязвимости в базу

# ЗАКЛЮЧЕНИЕ

В ходе написания курсовой работы была проделана большая работа. Я познакомился с Kali Linux, изучил её историю и понял в целом как она работает. Изучил все категории и подкатегории инструментов, расписал какие инструменты чаще всего используется в том или ином случае. Освоил терминал Kali. Также стоит отметить, что я выбрал три инструмента, которые по-моему мнению, лучше всего подходили для начала роботы с этим дистрибутивом, а именно Nmap, OpenVAS, Metasploit.

Я изучил каждый инструмент с нуля, прочитал всю документацию Nmap и Metasploit. А потом провел эксперименты с каждой командой и их флагами. Хочу отметить, что это очень мощные утилиты, функционал которых не останавливается на одном месте и совершенствуется с каждым днём. Благодаря им можно с хорошей точностью определить бреши в системе и предотвратить их использование. Так, проводя эксперимент, я обнаружил, с помощью Nmap, в своей сети уязвимость CVE-2011-1002, которая возникает из-за не обработки UDP диаграмм и позволяет произвести DoS-атаку (отказ в обслуживание).

В середине работы я столкнулся с тем, что анализировать свою сеть почти нет смысла, так как уязвимостей почти нет. Поэтому я установил Windows 8.1 на виртуальную машину и отключил там брандмауэр. С результатами можно ознакомится в отчёте OpenVAS и Metasploit. Это было сделано для полной демонстрации функционала софта, но надо учесть, что не все уязвимости были обнаружены. Что свидетельствует о том, что нет универсальной кнопки для обнаружения уязвимостей и что не все скрипты работают идеально. Тоже самое работает и в обратном случае нет уникального способа защитить себя от взлома. Тут и раскрывается вся магия безопасности, каждый случай имеет свой индивидуальный подход.

# ЛИТЕРАТУРА

1. Алексей Милосердов Данил Гриднев «Тестирование на проникновение с помощью Kali Linux 2.0» - 2015 - 348с.
2. Offensive-security «Kali Linux Revealed» - 394с.
3. Ric Messier «Learning Kali Linux» - 2018 – 584с.
4. Документация по OC Kali Linux – URL: <https://www.kali.org/tools>
5. Документация по Nmap – URL: <https://nmap.org>
6. Документация по Metasploit – URL: <https://docs.rapid7.com/metasploit/>
7. Документация по OpenVAS – URL: <https://www.greenbone.net/en/documents/>