Введение в анализ сетевого трафика

Введение в сетевую безопасность. Классификация сетевых атак. Инструменты, которые нам понадобятся. Kali Linux. DVL. Анализ сетевого трафика. Wireshark, tcpdump.

[Введение в сетевую безопасность](#_p23rodvcxje9)

[Классификация сетевых атак](#_1jzykp3e9l15)

[Типичный сценарий атаки](#_7kep7whooz0m)

[Анализ сетевого трафика](#_hfszwk4smff2)

[Инструменты](#_efg01fnyun5d)

[Настройка сетевых интерфейсов в VMWare](#_3o2xdra1rg16)

[Настройка сетевых интерфейсов в VirtualBox](#_gdqm3x68utms)

[Введение в Kali Linux](#_z400n1edj3n7)

[Категории программ](#_d4wxiabkkv9x)

[Программы](#_b38f3kbumhx5)

[Работа в Wireshark](#_mxb8wqiw1jzn)

[tcpdump](#_y8azoudmgqay)

Практическое [задание](#_3tfrjxxltv85)

[Дополнительные материалы](#_jeeoh8oz8lfe)

# 

# Введение в сетевую безопасность

Алан Тьюринг, взломавший немецкую шифровальную машинку Enigma, считается одним из отцов-основателей вычислительной техники. Она развивалась во многом благодаря задачам шифрования и дешифрования.

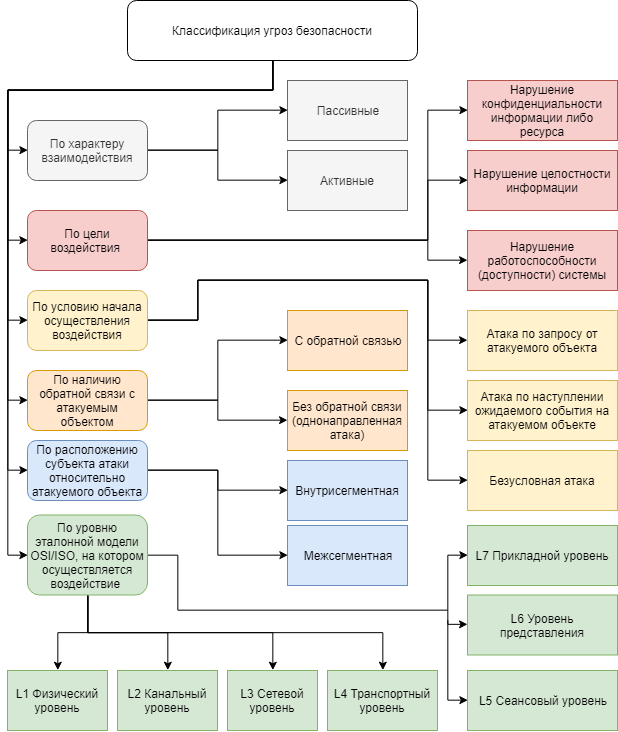
Приложения «уехали в облака», а смартфон есть у каждого, поэтому сейчас вопросы безопасности приобрели новое значение. Недостаточно запереть конфиденциальные данные в сейф. Нужно быть уверенным: переходя по сети, они не попадут к злоумышленникам, не будут изменены или утрачены. Система безопасности во многом строится на знаниях об уязвимостях и сетевых атаках. Они базируются преимущественно на архитектуре сетевого стека и сетевых протоколов.

Чтобы усвоить курс, понадобится хорошее знание стека TCP/IP. Проверьте себя по этому чек-листу:

* зачем нужен MAC-адрес, чем он отличается от IP-адреса?
* как компьютер-отправитель понимает, на какой MAC-адрес нужно адресовать фрейм, когда вы пытаетесь запинговать Google?
* что такое **gateway**?
* правильно ли, что маршрутизация выполняется на каждом компьютере?
* зачем нужны идентификатор пакета, флаги **Don’t Fragment**, **More Fragment** и смещение сегмента в заголовке IPv4-пакета?
* как работают протоколы:
  + **ARP**;
  + **DHCP**;
  + **DNS**?
* в чем разница между **TCP** и **UDP**?
* зачем нужен протокол **ICMP**?
* как устанавливается TCP-соединение?
* как отреагирует хост, если попытаться отправить сообщение на порт 80, не проводя процедуру «тройного рукопожатия» (если порт открыт и если закрыт)?
* что такое шифрование?
* что такое аутентификация, и чем она отличается от авторизации?
* чем отличаются симметричное и асимметричное шифрование?
* что такое SSL-сертификат?
* что такое сетевой концентратор (хаб, hub), как он работает?
* что такое сетевой мост (бридж, bridge), как он работает?
* что такое сетевой коммутатор (свитч, switch), как он работает?
* что такое атака Man In The Middle (MITM, «человек посередине»)?

Если затрудняетесь с ответом на любой из вопросов — повторите материал, прежде чем приступать к курсу.

# Классификация сетевых атак



Все угрозы можно условно разделить на два подмножества — в зависимости от объекта, подвергающегося воздействию:

* ***удаленные атаки на инфраструктуру*** — под инфраструктурой сети будем понимать сложившуюся систему организации отношений между объектами сети, сервисные службы, которые используются в ней, и ее протоколы;
* ***удаленные атаки на телекоммуникационные службы.***

По характеру выделяют:

* пассивные воздействия;
* активные воздействия.

Пассивное не оказывает непосредственного влияния на работу распределенной вычислительной системы, но способно нарушить ее политику безопасности.

Активное воздействие может изменять конфигурацию, нарушать работоспособность и принятую политику безопасности сети.

По цели воздействия:

* нарушение конфиденциальности информации либо ресурсов системы;
* нарушение целостности информации;
* нарушение работоспособности (доступности) системы.

По условию начала воздействия:

* атака после запроса от атакуемого объекта;
* атака после наступления ожидаемого события на атакуемом объекте;
* безусловная атака.

По наличию обратной связи с атакуемым объектом:

* с обратной связью;
* однонаправленная атака — без обратной связи.

По расположению субъекта атаки относительно атакуемого объекта:

* внутрисегментное воздействие — субъект и объект атаки находятся в одном сегменте;
* межсегментная атака — субъект и объект атаки находятся в разных сегментах.

По уровню эталонной модели OSI/ISO, на котором осуществляется воздействие:

* физический;
* канальный;
* сетевой;
* транспортный;
* сеансовый;
* представительный;
* прикладной.

## Типичный сценарий атаки

1. Внешняя разведка.
2. Внутренняя разведка.
3. Атака.
4. Скрытие следов.
5. Получение прибыли.

## Анализ сетевого трафика

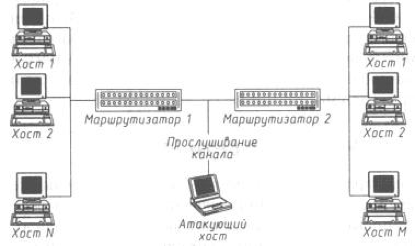
Для этого используются анализаторы трафика, или снифферы (от англ. to sniff — нюхать).

Сетевой анализатор трафика — это программа или программно-аппаратное устройство, которое перехватывает и анализирует (или только анализирует) сетевой трафик, предназначенный для других узлов.

Примеры программ: **Wireshark (tshark)**, **tcpdump** и другие. Все работают на базе библиотеки **libpcap**.

Трафик можно перехватывать:

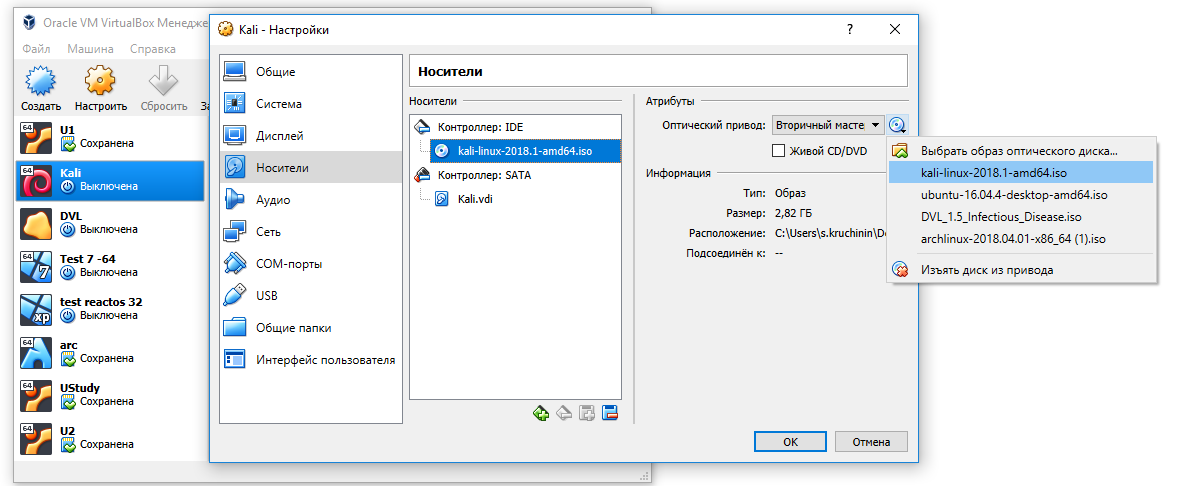
* **обычным «прослушиванием» сетевого интерфейса.** Метод эффективен, если использовать его в сегменте концентраторов вместо коммутаторов. Иначе на сниффер попадают лишь отдельные кадры;
* **подключением сниффера в разрыв канала;**
* **программным или аппаратным** **ответвлением** **трафика** и направлением его копии на сниффер;
* **через атаку на канальном или сетевом уровне**. Трафик жертвы или весь трафик сегмента перенаправляется на сниффер, а потом возвращается по надлежащему адресу.



*Схема анализа сетевого трафика*

# Инструменты

* **VirtualBox** или **VmWare** для запуска в машин-злоумышленников и машин-жертв;
* **Kali Linux** — сборка Linux с набором утилит сетевой безопасности:
  + **Kali Linux** можно не устанавливать, а запускать в режиме Live CD;
  + если есть достаточные навыки, можно самостоятельно поставить дополнительные инструменты и работать на том Linux, который вы используете (например, Ubuntu). . **Wireshark** можно запускать даже в Windows.
* Машины-жертвы:
  + **DVL** — специальная сборка Linux с уязвимостями. Достаточно запустить в режиме Live CD, установка не требуется;
  + свежеустановленная в виртуальной машине Ubuntu последней версии;
  + желательно в виртуалке иметь Windows XP/Windows 7/Windows 8/Windows 10. Если работаете в этой ОС, можно воспользоваться хостовой машиной в качестве «жертвы».



*Выбор носителя в* ***VirtualBox*** *для запуска* ***Kali Linux*** *в режиме Live CD*

Для работы потребуется настроить виртуальные машины, чтобы они:

* имели доступ в интернет;
* находились в одном сегменте сети:
* были доступны с хостовой машины.

## Настройка сетевых интерфейсов в VMWare

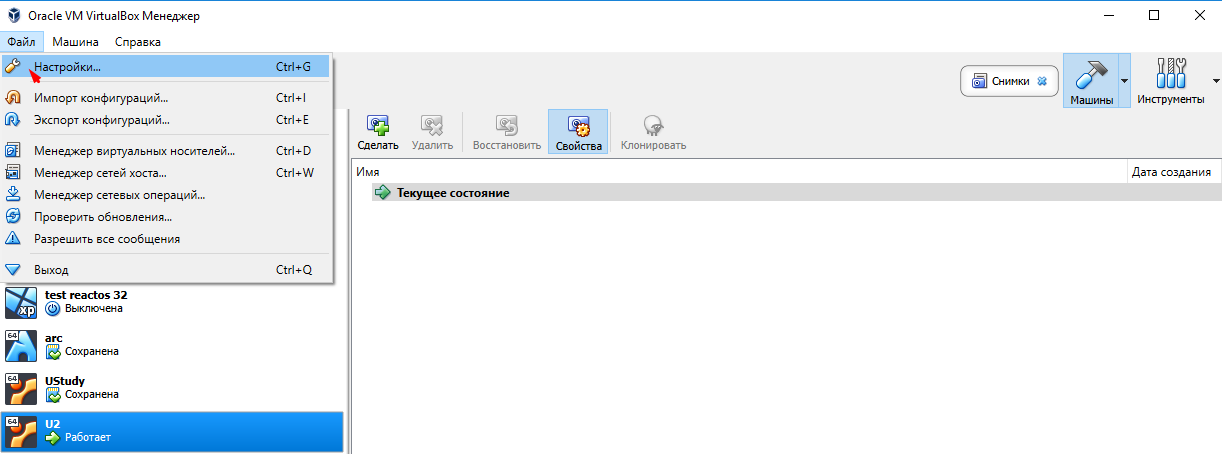
Как установить **VMWare** и настроить виртуальную машину, подробно разбиралось в курсе Linux. Как тип сетевого подключения достаточно выбрать сеть **NAT**. **VMWare** работает таким образом, чтобы внутренний IP-адрес был доступен с хостовой машины — поэтому дополнительных сетевых интерфейсов не требуется.

В качестве альтернативы можно воспользоваться подключением типа **Bridge**. Тогда виртуальные машины будут находиться в той же сети, что и хостовая с ее соседями. Недостатки такого подхода:

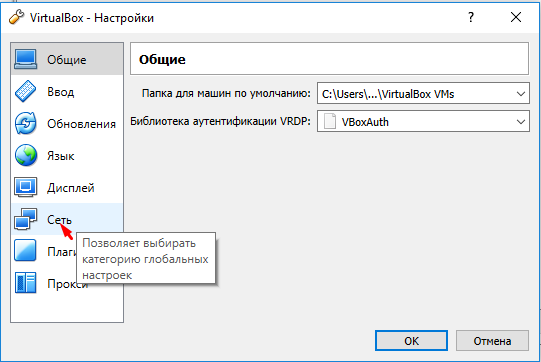
* при переключении с **Ethernet** на **Wi-Fi** виртуалки становятся недоступны и теряют доступ к интернету;
* отправлять в действующую сеть поддельные запросы — не лучшая идея.

## Настройка сетевых интерфейсов в VirtualBox

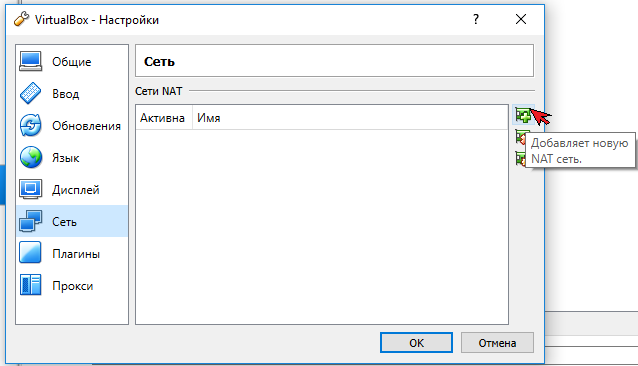
Прежде чем приступить к настройке, зайдем в настройки **VirtualBox**.



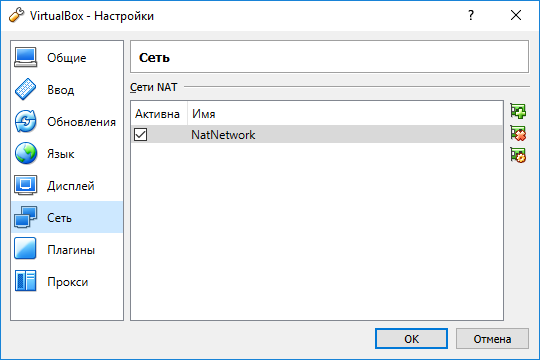
Перейдем в раздел «Сеть»:



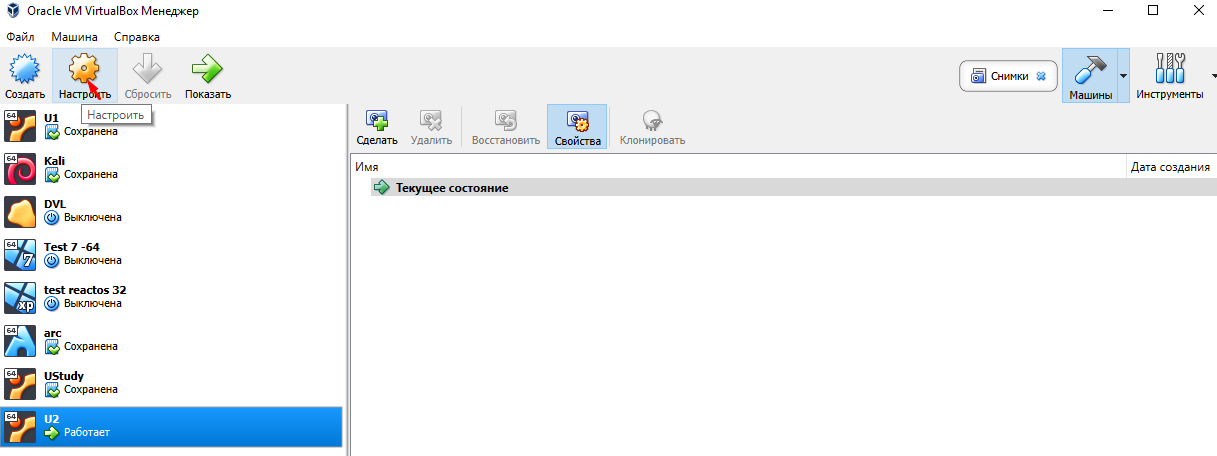
Увидим список сетей **NAT** (или пустой). Жмем «Добавить новую NAT-сеть»:



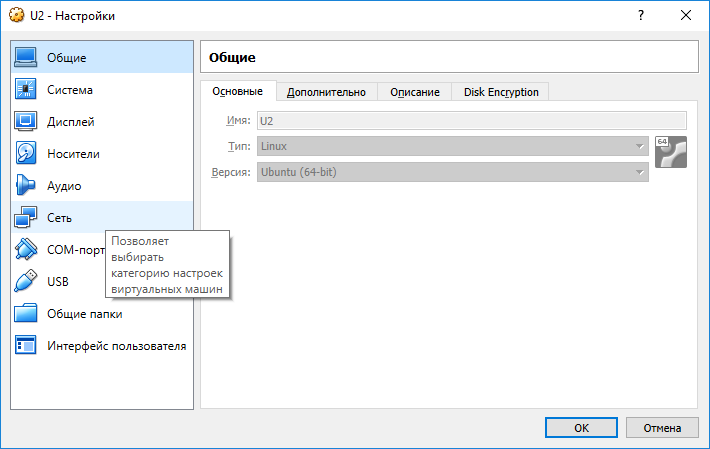
Добавили NAT-сеть, которая понадобится в дальнейшем:



Теперь заходим в настройки виртуальной машины:



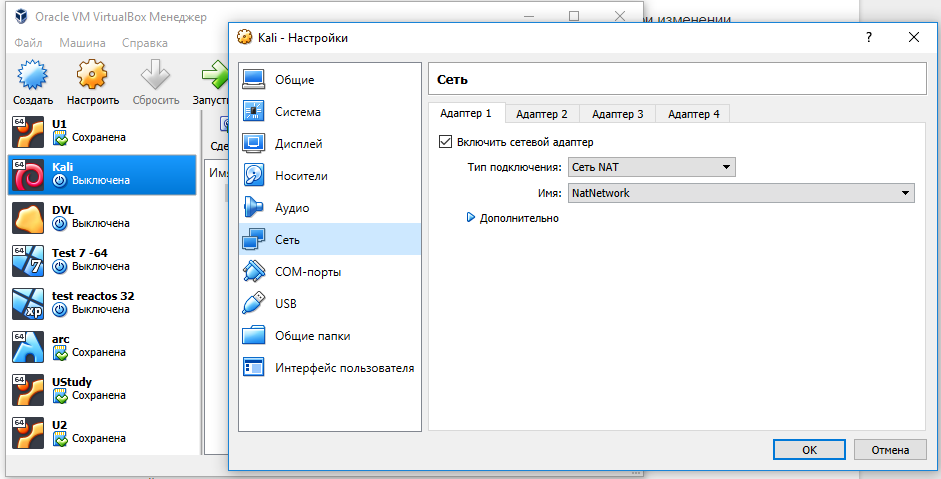
Переходим в раздел «Сеть»:



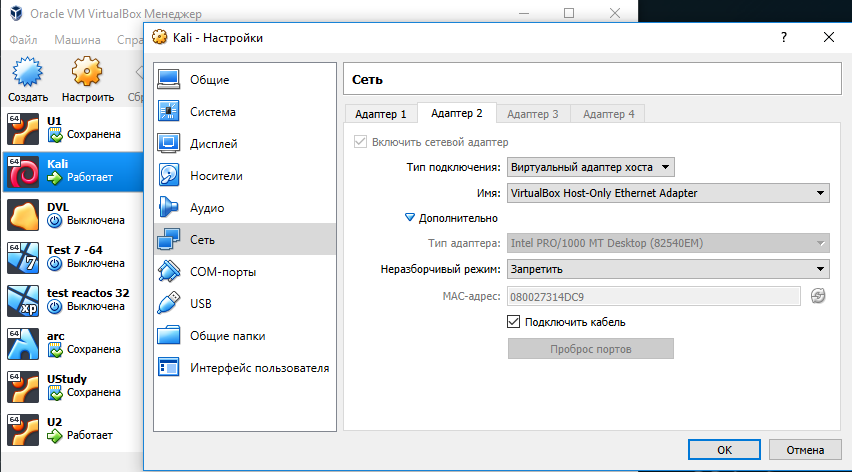
В **VirtualBOX** сеть **NAT** не дает автоматически возможность получить доступ к виртуальной машине. Поэтому остаются варианты:

* сетевой мост (**Bridge**) — с его проблемами: потерей доступа при изменении сетевого подключения и поддельными запросами в действующую сеть;
* два сетевых интерфейса — сеть NAT (для организации сети между виртуальными машинами с доступом в интернет) и адаптер виртуального хоста (для доступа к виртуальной машине с машины хоста — например, по **ssh**).

Посмотрим, как это выглядит. Указываем ранее созданную сеть NAT:



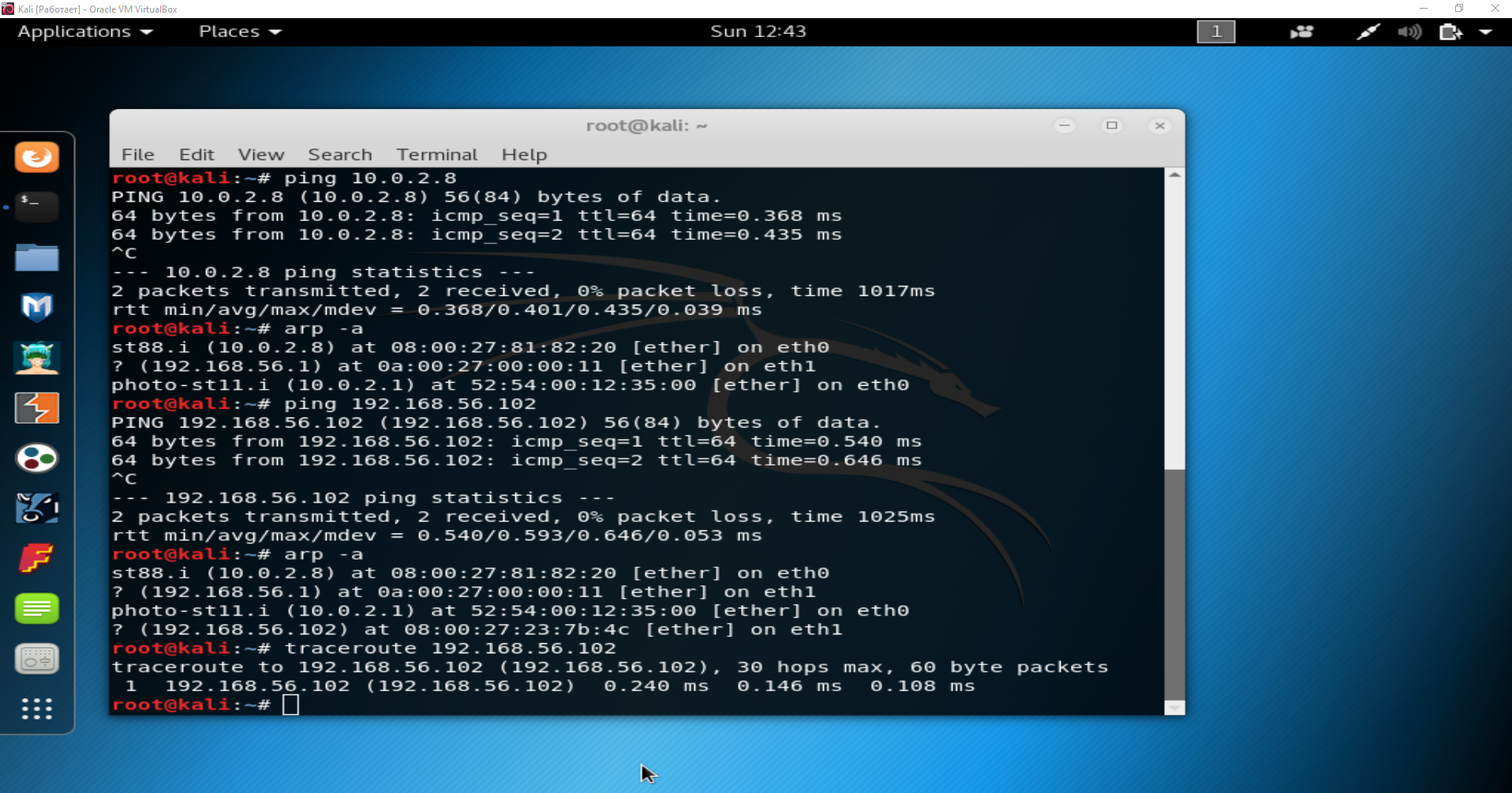
Первый адаптер — сеть NAT:



Второй — виртуальный адаптер хоста. Проверьте в «Дополнительно», чтобы кабель был подключен:

# 

**eth0** служит для доступа во внешнюю сеть, а **eth1** — с хостовой машины.



По arp-таблицам видим — можно начинать исследовать уязвимости уровня L2.

# Введение в Kali Linux

**Kali Linux** — сборка Linux для комплексного аудита безопасности. Основана на **Debian 7** и включает множество программ и утилит, ориентированных на поиск и анализ уязвимостей.

Для работы с **Kali Linux** можно использовать **VMWare** или **VirtualBox**. Систему можно установить и использовать в режиме Live CD (в **VMWare**/**VirtualBox**, на CD или загрузочной флешке).

Тип операционной системы: Kali Linux собран на основе Debian 7, 64-битный режим (если дистрибутив 64-битный).

Необходимо 20 Gb жесткого диска и не меньше 1 GB Ram.

Логин в LiveCD — **root**, пароль — **toor**.

В случае установки, а не Live CD, следует создать пользователя и при необходимости применить **sudo**.

## Категории программ

* **Information Gathering** — сбор информации;
* **Vulnerable Analysis** — анализ уязвимостей;
* **Web Application Analysis** — анализ веб-уязвимостей (поиск CMS, краулеры, сканирование уязвимостей);
* **Database Assessment** — работа с базами данных;
* **Password Attacks** — атаки на пароли;
* **Wireless Attacks** — работа с беспроводными сетями Wi-Fi, Radio;
* **Reverse Engineering** — реверс инжиниринг, дебаггинг, дизассемблер;
* **Exploitation Tools** — эксплойты, эксплуатация уязвимостей;
* **Sniffing & Spoofing** — снифферы и спуферы, анализ трафика и работа с ним;
* **Post Exploitation** — эксплуатация **XSS** (**cross-site scripting**);
* **Forensis** — расследование инцидентов в области информационной безопасности (анализ дисков, восстановление данных, сокрытие данных и каналов связи, реверс-инжиниринг **malware**);
* **Reporting Tools** — утилиты для отчетов;
* **Social Engineering Tools** — социальная инженерия;
* **System Services** — системные сервисы.

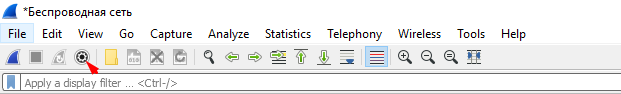
## Программы

* **aircrack-ng** — программа для аудита Wi-Fi сетей;
* **burpsuite** — прокси-сканер для аудита веб-уязвимостей;
* **hydra** — подбор пароля в реальном режиме (почты, ssh);
* **john (john ripper)** — утилита для оффлайн-подбора паролей. Разработчик — Саша Кисляк, проект **OpenWall.com**;
* **maltegoce** — утилита для конкурентной разведки: сбора данных из открытых источников, социальных сетей;
* **metasploit** — фреймворк, который делает уязвимости удобными для эксплуатации злоумышленником;
* **nmap** — сканер портов и операционных систем. Еще используют **zmap**, ориентированный на большую производительность, но его в составе нет **Kali**. Если потребуется, ее можно поставить самостоятельно.
* **owasp-zap** — краулер, многофункциональный «перочинный нож» для веба;
* **sqlmap** — программа для тестинга SQL-инъекций;
* **ettercap** — набор для атак **MITM** («человек посередине»).
* **wireshark** — сетевой сниффер.

# Работа в Wireshark

Можно использовать Wireshark из комплекта Kali Linux или отдельно установленный.

Открываем вкладку **Capture Interfaces** — в панели инструментов кружок, четвертый слева:



* **Input** — сетевые интерфейсы;
* **Output** — формат сохраняемого файла. Можно указать размер файла, чтобы сохранять трафик в файлы по 1 Гб, например;
* **Options** — дополнительные опции, возможности остановить захват после определенного числа пакетов или и времени.

Меню: **Analyze — Enabled Protocols**: весь набор протоколов, которые понимает Wireshark .

Информация в основном окне:

1. Все пакеты.
2. Информация о пакете.
3. Байты пакета.

Если в списке кликнуть правой кнопкой мыши на захваченный пакет, станет доступно поле **Packet Comment**. Если кликнуть правой кнопкой мыши на один из захваченных пакетов в списке, станет доступно поле **Packet Comment.** Оно позволяет добавить произвольный комментарий, который будет отображаться в поле инфоракфьумации о пакете наряду с заголовком. Можно сделать Mark-пакет, без комментария, — бывает полезно.

Чтобы найти прокомментированные пакеты, следует нажать на кружок в левом нижнем углу окна. Это пригодится для домашнего задания.



В **Statistics — Resolved Address** видим список встреченных IP-адресов и доменных имен, которые в них разрешаются.

Фильтр устроен так, что в нем можно использовать все поля в заголовках.

К свойствам пакета обращаются через точку. Например:

|  |
| --- |
| frame.marked==true |

или

|  |
| --- |
| frame.marked eq true |

Синтаксис

* **==** или **eq**;
* **!=** или **ne**;
* **>** или **gt**;
* **<** или **lt**;
* **>=** или **ge**;
* **<=** или **le**.

Фильтруем по ip-адресу:

|  |
| --- |
| ip.addr == 192.168.129.129 |

Рассмотрим, как исключить адрес. Такой способ не будет работать корректно:

|  |
| --- |
| ip.addr!=192.168.129.129 |

А такой будет:

|  |
| --- |
| !(ip.addr == 192.168.129.129) |

или

|  |
| --- |
| (ip.src != 192.168.129.129)&&(ip.dst != 192.168.129.129) |

Более сложные условия:

|  |
| --- |
| ip.src==192.168.129.19&& tcp.port==80 |

Можно применить такой вариант:

|  |
| --- |
| ip.src==192.168.129.19 and tcp.port==80 |

Чтобы использовать условие «ИЛИ», можно сделать так:

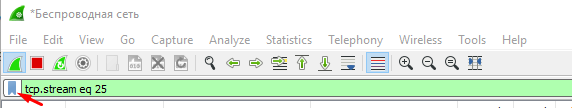
|  |
| --- |
| ip.src==192.168.129.19||192.168.129.20 |

Или

|  |
| --- |
| ip.src==192.168.129.19 or 192.168.129.20 |

Верхний регистр не работает: **WireShark Case Sensitive**.

Если кликнуть на значок закладки слева от фильтра, можно дать ему название и сохранить для дальнейшего использования.



Чтобы найти доступные фильтры, кликните на значок закладки.

Рассмотрим полезную функцию: **Analyze — Follow Stream — TCP**. Возьмем в качестве примера сайт lib.ru, он не поддерживает **HTTPS**.

Включаем в фильтре **http**:

|  |
| --- |
| http |

Видим запросы и ответы.

Зайдем на **samlib.ru** и найдем отправку данных с помощью метода **POST**:

|  |
| --- |
| http.request.method==POST |

Можем искать по строке:

|  |
| --- |
| http contains "password" |

Научимся выполнять поиск по произвольному слову:

|  |
| --- |
| data-text-lines contains "copyright" |

То же можно сделать через меню **FInd — Find Packet.**

Дальше отследим сессию целиком. Нажимаем **Analyze — Follow Stream — TCP**.

Запросы в рамках данного потока будут отмечены красным, а ответы — синим.

Отфильтруются только те пакеты, которые относятся к данной сессии. Автоматически сформируется фильтр, например:

|  |
| --- |
| tcp.stream eq 17 |

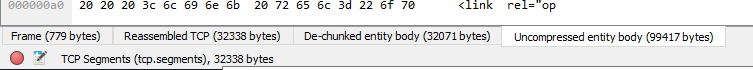
В потоке данных можем видеть, что их тип — **text/html**, но в тексте идут кракозябры. Напоминает шифрование, но это компрессия. О ней свидетельствует заголовок с типом сжатия **gzip.**

Если кликнем на пакет, содержащий HTTP-ответ, увидим, что WireShark собрал целую сессию из фрагментов. Это помечается отдельной вложенностью: **[25 Reassembled TCP Segments].**

В таком случае внизу будут отображаться вкладки:

* **Reassembled TCP**, которая восстанавливает из сегментов поток данных;
* и **Uncompressed entity body**, если данные сжаты.

Увидим, что данные не зашифрованы.



Для любого заголовка в протоколе можно использовать в контекстном меню (правый клик мыши) **Apply As A Filter — Selected**. Тогда фильтр сформируется автоматически.

Если кликнем на заголовке **host: samlib.ru**, получим фильтр:

|  |
| --- |
| http.host == "lib.ru" |

Опция **File — Save Objects — HTTP** позволяет сохранить данные, **html**, **css**, **js**, картинки и загруженные файлы.

# tcpdump

**tcpdump** — консольная утилита, позволяющая слушать трафик.

Если работать на удаленной машине, может быть удобнее утилита **tcpdump**, которая действует в терминале.Она позволяет сохранить данные в файл **pcap** — который используется и в Wireshark благодаря общей библиотеке **libpcap**. Затем утилита передает данные по **ssh** и на анализ в Wireshark.

Рассмотрим пример, как слушать трафик на сетевом интерфейсе **eth0**:

|  |
| --- |
| tcpdump -i eth0 |

Пример: слушаем tcp-трафик на сетевом интерфейс **eth0**:

|  |
| --- |
| tcpdump -i eth0 -p tcp |

Слушаем tcp- и icmp-трафик на сетевом интерфейс **eth0**:

|  |
| --- |
| tcpdump -i eth0 -p tcp or icmp |

Слушаем tcp- и icmp-трафик на сетевом интерфейс **eth0**, не преобразуя доменные имена в IP-адреса

|  |
| --- |
| tcpdump -ni eth0 -np tcp or icmp |

Разберемся с **tcpdump**:

|  |
| --- |
| man tcpdump |

Запуск с сохранением дампа:

|  |
| --- |
| tcpdump -i any -nnvv dst port 23 -w /tmp/test.pcap |

Анализ с регулярным выражением:

|  |
| --- |
| tcpdump port http or port ftp or port smtp or port imap or port pop3 -l -A | egrep -i 'pass=|pwd=|log=|login=|user=|username=|pw=|passw=|passwd=|password=|pass:|user:|username:|password:|login:|pass |user ' --color=auto --line-buffered -B20 |

Перенаправление tcpdump-потока с помощью **ssh**:

|  |
| --- |
| tcpdump -i any ! host 192.168.1.2 -s 0 | ssh someone@192.168.1.2 "cat > dump.txt" |

# Практическое задание

1. Ознакомиться с кодировкой **base64**.
2. Провести аудит сетевого трафика:
   * Скачать дамп **.pcap**, приложенный к уроку;
   * Проанализировать трафик, найти секретное послание;
   * Отчет приложить к домашнему заданию\*.

*\* Задание сложное, совместная работа не возбраняется.*

1. Ознакомиться со статьями 28, 272, 273, 274 Уголовного кодекса по ссылке «УК с комментариями» в «Дополнительных материалах». Грань между легальным и нелегальным применением инструментов очень тонкая.

# Дополнительные материалы

1. <https://ru.wikipedia.org/wiki/Base64>.
2. Уголовный кодекс с комментариями: <http://rfuk.ru/head_28.html>.
3. <https://www.wireshark.org/docs/wsug_html/>.
4. <http://www.tcpdump.org/>.
5. <http://bit.ly/1DOvIkD>.
6. **Перенаправление tcpdump-потока с помощью ssh**: <https://splittingelectrons.wordpress.com/2012/02/23/how-to-redirect-tcpdump-output-to-a-remote-server/>.

**Использованные материалы**

1. <https://ru.wikipedia.org/wiki/Base64>.
2. Уголовный кодекс с комментариями: <http://rfuk.ru/head_28.html>.
3. <https://www.wireshark.org/docs/wsug_html/>.
4. <http://www.tcpdump.org/>.
5. <http://bit.ly/1DOvIkD>.
6. **Перенаправление tcpdump-потока с помощью ssh**: <https://splittingelectrons.wordpress.com/2012/02/23/how-to-redirect-tcpdump-output-to-a-remote-server/>.