Лекция от 11.09.24



**Модель OSI (Open System Interconnection)**, или эталонная модель взаимодействия открытых систем описывает, как устройства в локальных и глобальных сетях обмениваются данными и что происходит с этими данными. Её предложили в 1984 году инженеры из Международной организации по стандартизации (ISO), которая работала над единым стандартом передачи данных по интернету.

При этом сама по себе эталонная модель — не стандарт интернета, как, например, TCP/IP; её можно сравнить с фреймворками в мире языков программирования: в OSI «из коробки» доступны разные веб-стандарты — UDP, HTTP, FTP, Telnet и другие. Всего таких протоколов — более 100 штук.

Модель OSI включает семь слоёв, или уровней, — причём каждый из них выполняет определённую функцию: например, передать данные или представить их в понятном для человека виде на компьютере. Кстати, у каждого слоя — свой набор протоколов.

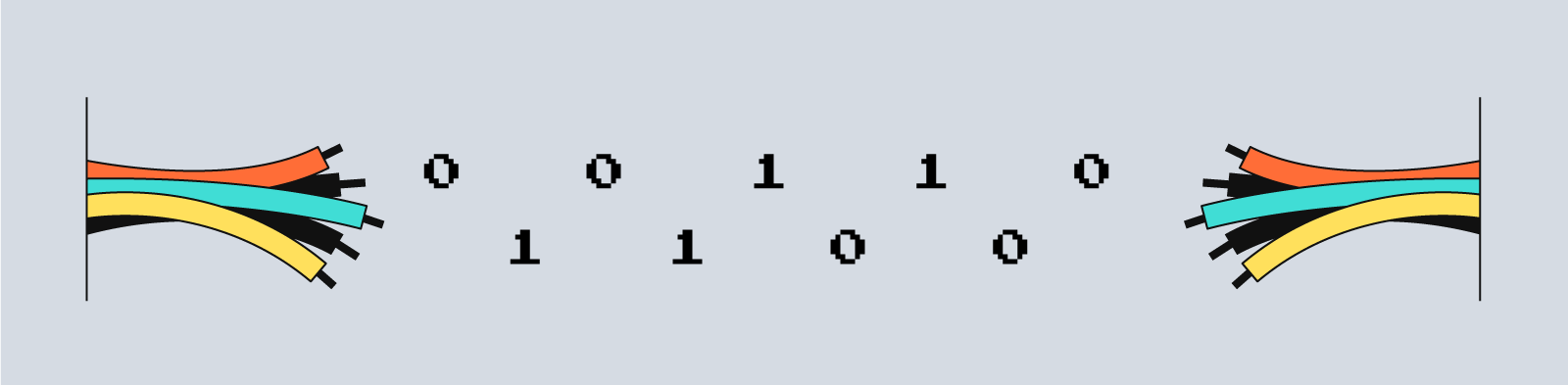
Самый нижний слой отвечает за физическое представление данных, то есть за то, как данные передаются по проводам или с помощью радиоволн, а самый верхний — за то, как приложения взаимодействуют с сетью.

Нижний слой оперирует такими понятиями, как «тип кабеля» или «тип коннектора», а верхний — такими, как HTTP или API.

## ****1-й уровень OSI — физический**** ****(L1, physical layer)****

На самом нижнем уровне модели OSI данные представляют собой физические объекты — ток, свет или радиоволны. Они передаются по проводам или с помощью беспроводных сигналов.

Этот слой работает с кабелями, контактами в разъёмах, модуляцией сигнала, кодированием единиц и нулей и другими низкоуровневыми штуками. По сути, первый уровень — это уровень проводов и физических способов передачи сигнала. Минимальная абстракция.



Самый известный протокол на физическом уровне — Ethernet. Он описывает, как сигналы кодируются и передаются по проводам. Кроме него есть Bluetooth, Wi-Fi и ИК-порт, которые также содержат инструкции для передачи данных.

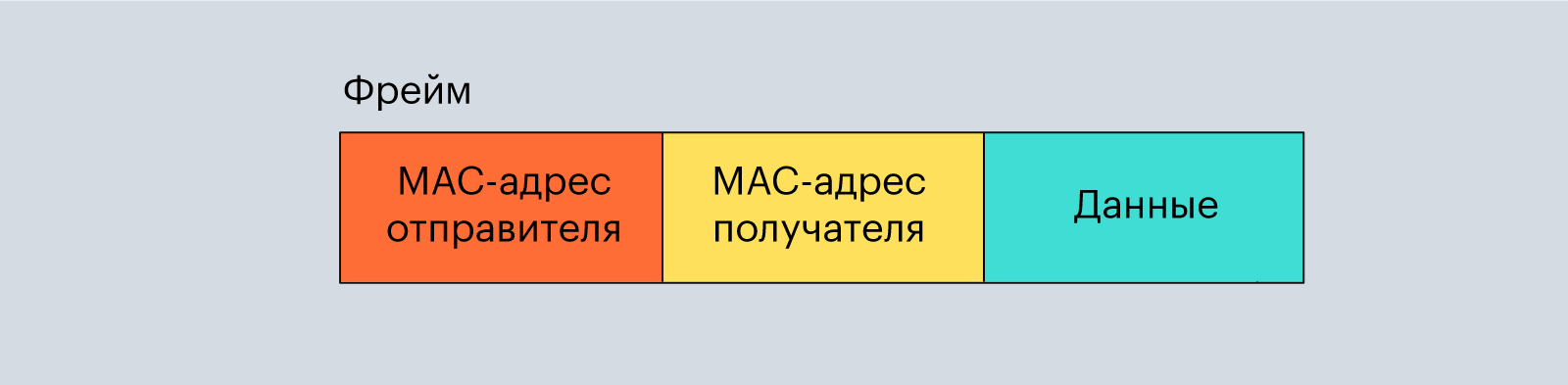
Устройства физического уровня — концентраторы и репитеры. Они работают с физическим сигналом «втупую» и не вникают в его логику: получили данные — передали их дальше по проводу.



## ****2-й уровень OSI — канальный**** ****(L2, data link layer)****

Над физическим уровнем располагается канальный. Его задача — проверить целостность полученных данных и исправить ошибки. Этот уровень «поумнее» предыдущего: он уже понимает, что разные амплитуды напряжений отвечают разным битам — нулям и единицам. А ещё канальный уровень умеет кодировать сигналы в биты и передавать их дальше.

Полученные с нижнего уровня данные делятся на фреймы, или кадры. Каждый фрейм состоит из служебной информации — например, адреса отправителя и адреса получателя, — а также самих данных.

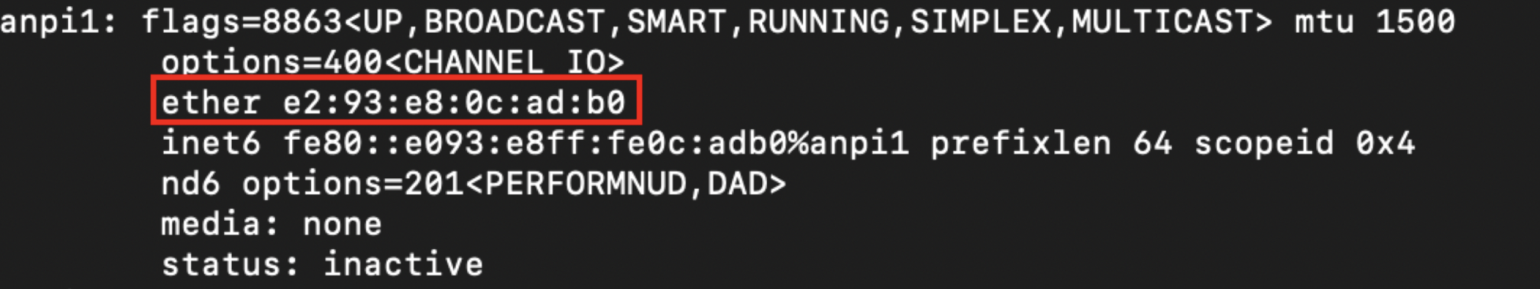


Получается что-то вроде почтового конверта. На лицевой стороне у него написано, от кого пришло письмо, а внутри находится само письмо (в нашем случае данные).

Лицевая сторона конверта — это MAC-адрес устройства, которое отправило нам информацию. Он нужен, чтобы идентифицировать устройства в локальной сети, состоит из 48 или 64 бит и выглядит примерно так:



Ещё один важный факт о MAC-адресах: когда на заводе собирают ноутбук или смартфон, ему сразу же присваивают определённый MAC-адрес, который потом уже никак нельзя поменять. MAC-адрес настольных ПК зашит в сетевую карту, поэтому его можно изменить, только заменив эту самую карту.



С помощью команды ifconfig можно узнать MAC-адрес вашего Macbook или компьютера на Linux. В Windows нужно ввести команду ipconfig.

Канальный уровень не так прост — он делится ещё на два подуровня:

* уровень управления логическим каналом — LLC (logical link control);
* уровень управления доступом к среде — тот самый MAC (media access control).

Первый подуровень нужен для взаимодействия с верхним уровнем, сетевым, а второй — для взаимодействия с нижним, физическим.

Устройства канального уровня — коммутаторы и мосты. Они нужны, чтобы передавать фреймы нужному адресату. Протоколы канального уровня — PPP, CDP.

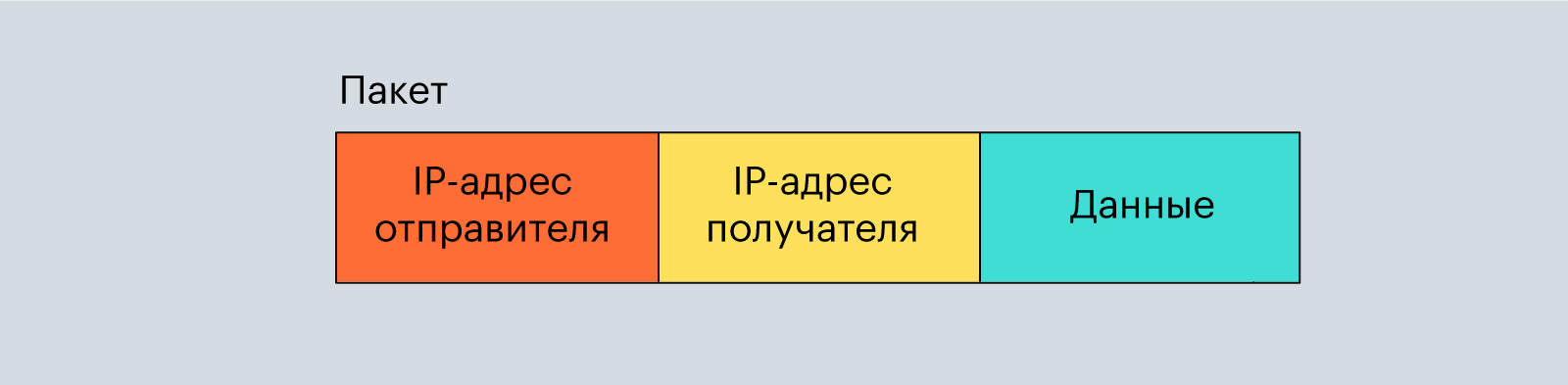
## ****3-й уровень OSI — сетевой**** ****(L3, network layer)****

Этот уровень отвечает за маршрутизацию данных внутри сети между компьютерами. Здесь уже появляются такие термины, как «маршрутизаторы» и «IP-адреса».

Маршрутизаторы позволяют разным сетям общаться друг с другом: они используют MAC-адреса, чтобы построить путь от одного устройства к другому.

Данные на сетевом уровне представляются в виде пакетов. Такие пакеты похожи на фреймы из канального уровня, но используют другие адреса получателя и отправителя — IP-адреса.

Чтобы получить IP-адрес обоих устройств (отправителя и получателя), существует протокол ARP (address resolution protocol). Он умеет конвертировать MAC- в IP-адрес и наоборот.



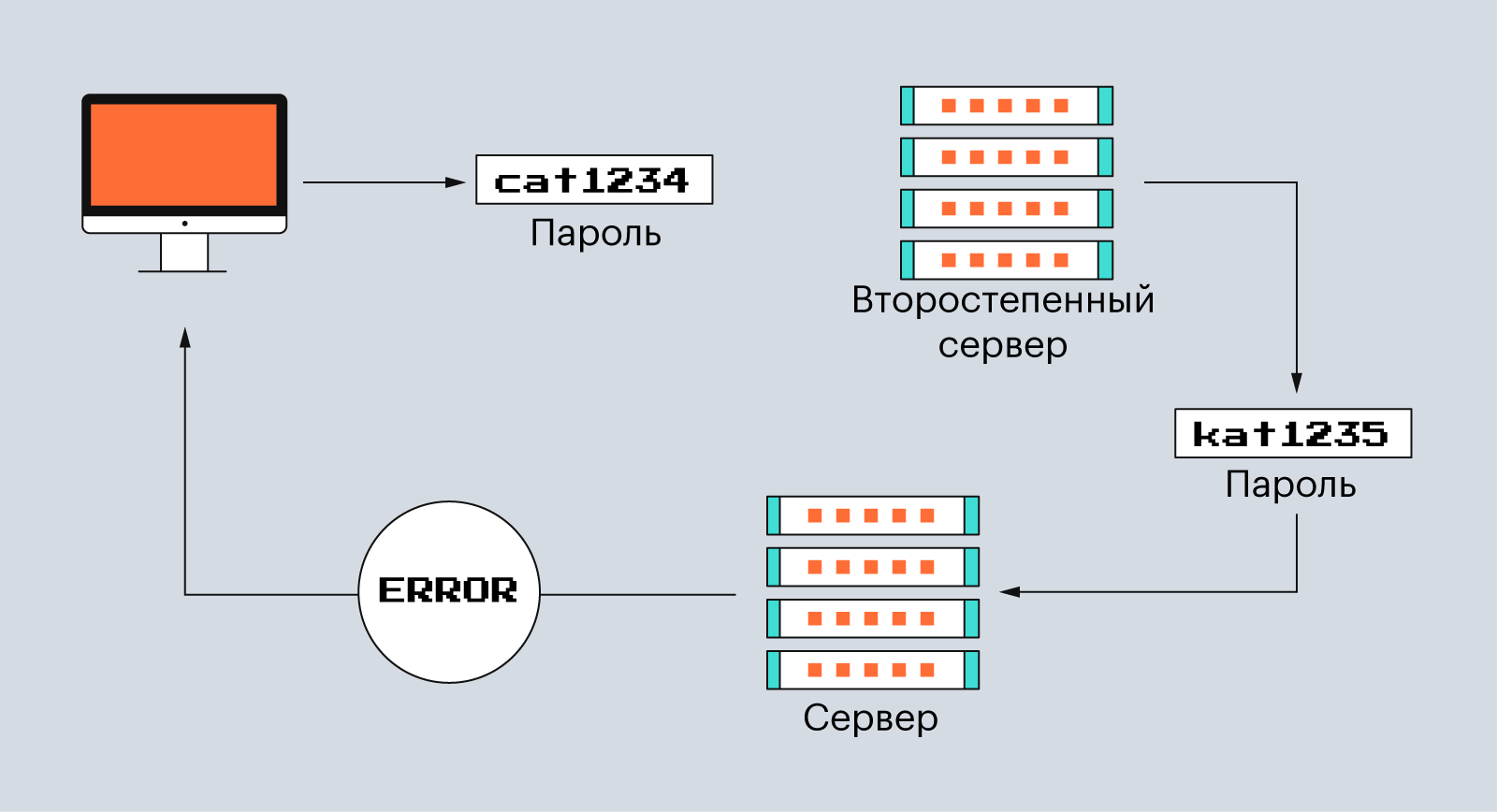
Примерно так выглядят пакеты

## ****4-й уровень OSI — транспортный**** ****(L4, transport layer)****

Из названия понятно, что на этом уровне происходит передача данных по сети. Так и есть. Два главных протокола здесь — TCP и UDP. Они как раз и отвечают за то, как именно будут передаваться данные.

TCP (Transmission Control Protocol) — это протокол, который гарантирует доставку данных в корректном виде. Он жёстко следит за каждым битом информации, но работает гораздо медленнее UDP.

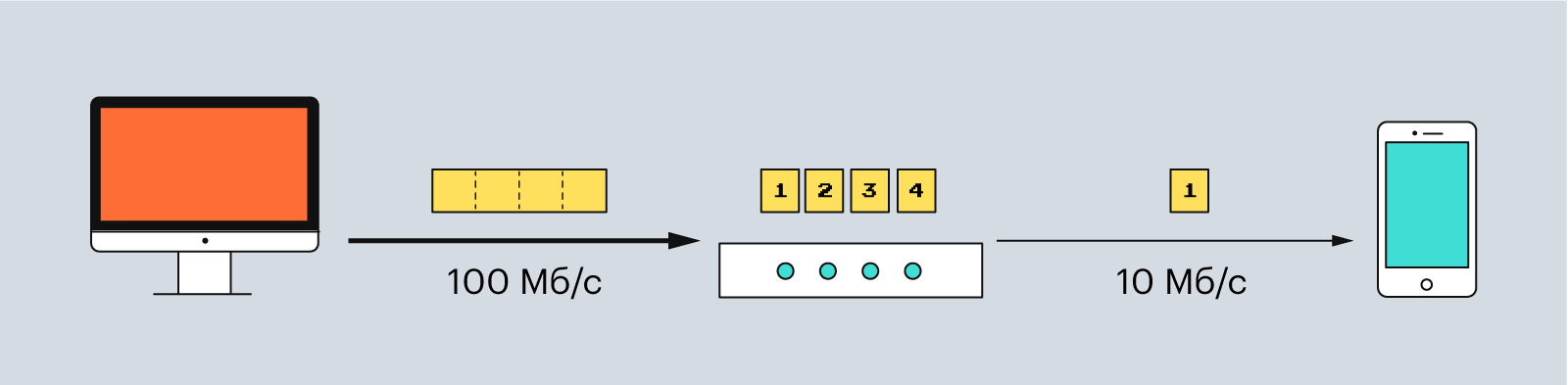
Например, когда вы вводите логин и пароль при входе в социальную сеть, очень важно, чтобы все символы отправились в определённой последовательности. Если какие-то потеряются или изменятся, вы просто не сможете авторизоваться. Поэтому протокол TCP использует разные методы проверок — например, контрольные суммы.



А вот в видео или аудио небольшие потери некритичны, зато важна скорость передачи данных. Для таких задач как раз и придумали протокол UDP (user datagram protocol). Он уже не проверяет цельность битов, его задача — как можно быстрее передать данные с одного устройства на другое.

В протоколе TCP данные делятся на сегменты. Каждый сегмент — часть пакета. Сегменты нужны, чтобы передавать информацию по сети, учитывая её пропускную способность.

Например, если вы передаёте данные с компьютера, у которого пропускная способность 100 Мб/c, на смартфон с пропускной способность 10 Мб/c, то данные разделятся так, чтобы не застревать в самом медленном устройстве.



Ещё сегментация важна для надёжности. Один большой пакет может потеряться или направиться не тому адресату. А маленькие пакеты снижают риск подобных ошибок и даже позволяют проверять их количество. Если какой-то сегмент не получилось доставить, протокол TCP может запросить его у отправителя снова. Так обеспечивается надёжность.

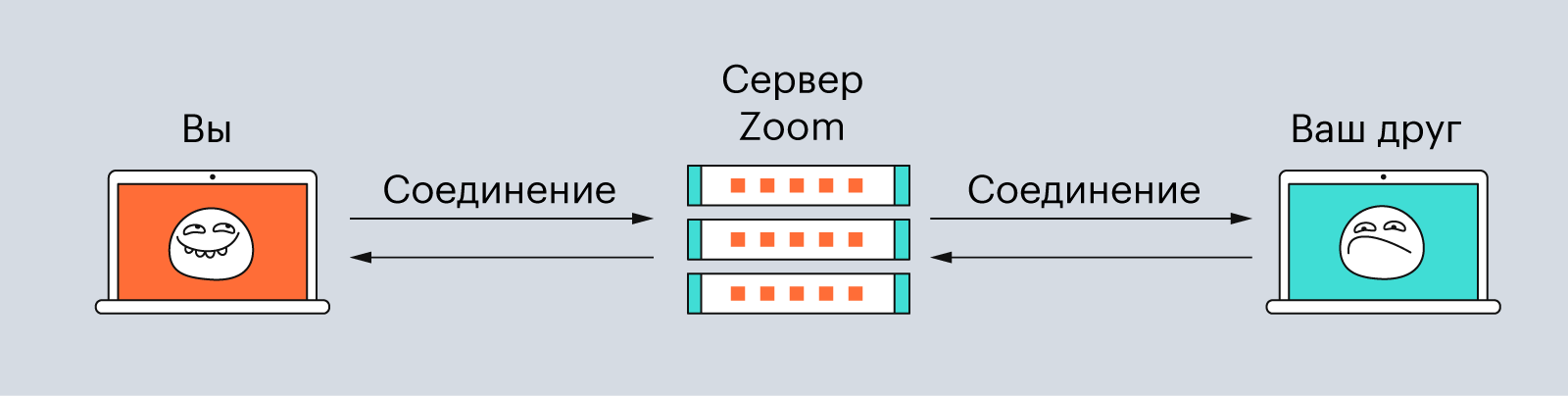
В UDP данные делятся на датаграммы — это примерно то же, что и пакет, только датаграммы автономны. Каждая датаграмма имеет всё необходимое, чтобы дойти до получателя. Поэтому они не зависят от сети и могут доставляться по разным маршрутам и в произвольном порядке.

## ****5-й уровень OSI — сеансовый**** ****(L5, session layer)****

Начиная с этого уровня и выше, данные имеют уже нормальный вид — например, привычных нам JPEG- или MP3-файлов. Задача сети на этих уровнях — представить информацию в понятном для человека виде и сделать так, чтобы пользователь мог её как-то «потрогать».

Сеансовый уровень управляет соединениями, или сессиями. Типичный пример — звонок по Skype или Zoom. Когда вы звоните другому человеку, между вашими компьютерами устанавливается соединение, по которому передаются аудио и видео. Если такое соединение разорвать, то и ваш звонок прервётся.

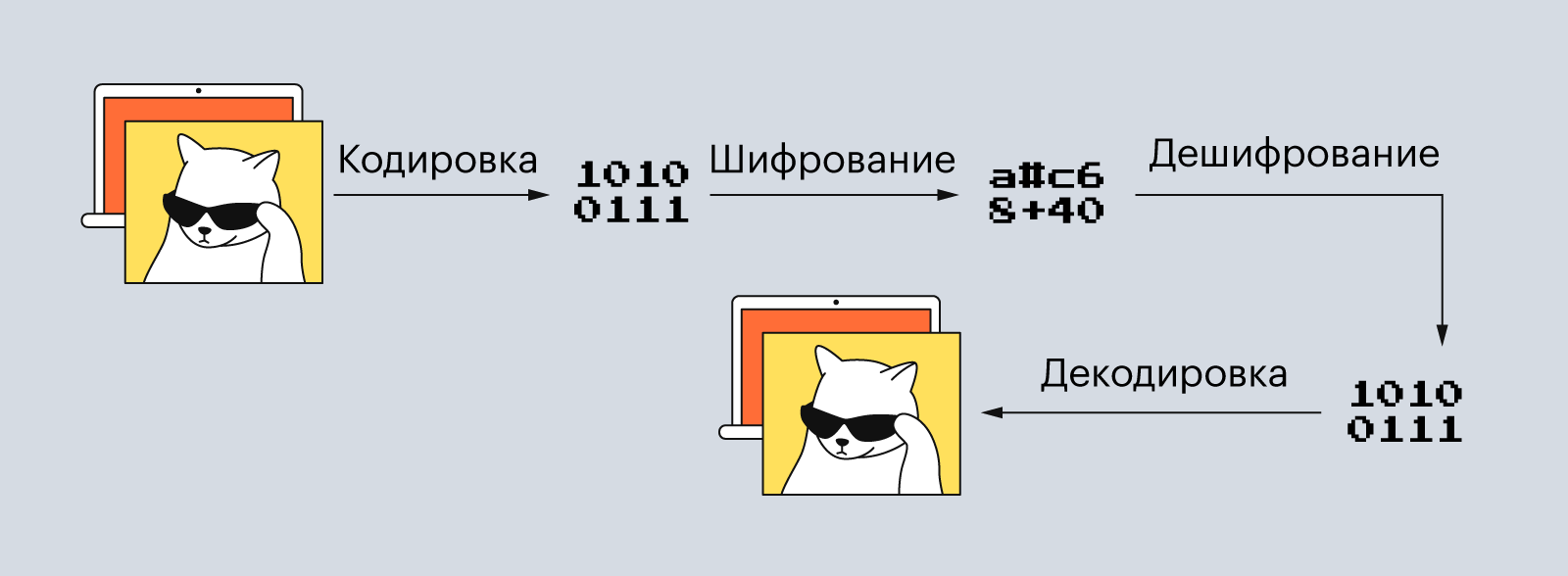
На сеансовом уровне очень важно, чтобы соединение правильно установилось и поддерживалось. То есть механизмы протоколов должны проверить, что у обоих собеседников есть нужные кодеки и сигнал между устройствами присутствует.



## ****6-й уровень OSI — уровень представления данных**** ****(L6, presentation layer)****

На этом уровне происходит преобразование форматов данных — их кодирование и сжатие. Например, полученные данные могут превратиться в GIF- или MP4-файл. То же самое происходит и в обратном порядке: когда пользователь отправляет файл другому человеку, данные сначала конвертируются в биты и сжимаются, а потом уже передаются на транспортный уровень.

Помимо кодировки и сжатия на уровне представления, данные могут шифроваться — если, конечно, это необходимо.



## ****7-й уровень OSI — прикладной**** ****(L7, application layer)****

Последний уровень модели OSI — прикладной. На нём находятся сетевые службы, которые помогают без проблем сёрфить в интернете.

Прикладной уровень похож на некий графический интерфейс для всей модели OSI — с его помощью пользователь взаимодействует с другими уровнями, даже не подозревая об этом. Этот интерфейс называется сетевым.

Самые популярные из сетевых интерфейсов — это HTTP, HTTPS, FTP и SMTP. А «устройства» здесь — это уже программы: Zoom, Telegram, браузеры.



## ****Как на практике работает сетевая модель OSI****

В начале статьи мы задались вопросом: а как передаются сообщения в Telegram? Настало время на него ответить — и показать весь процесс передачи данных по модели OSI.

Мы хотим отправить сообщение нашему другу. Печатаем текст и нажимает кнопку «Отправить», а дальше перемещаемся внутрь компьютера.

**Прикладной уровень.**Приложение Telegram работает на прикладном уровне модели OSI. Когда мы печатаем текст сообщения и нажимаем кнопку «Отправить», эти данные передаются на сервер мессенджера, а оттуда — нашему другу.

Весь процесс проходит через API разных библиотек — например, для HTTP-запросов. Интерфейсы позволяют без лишних проблем обмениваться данными и не погружаться в то, как они представлены на низком уровне. Всё, что нужно знать, — это какую функцию вызвать и какие переменные туда передать.

**Уровень представления.**Здесь данные должны преобразоваться в унифицированный формат, чтобы их можно было передавать на разные устройства и операционные системы. Например, если мы отправляем сообщение c Windows на macOS, данные должны быть в читаемом для компьютеров Apple виде. Такая же ситуация и с другими устройствами.

Раз мы собираемся передать данные на другой компьютер, их нужно перевести в бинарный формат. После этого начнётся сам процесс передачи по сети.

**Сеансовый уровень.**Чтобы данные успешно передались сначала на сервер Telegram, а затем к нашему другу, приложению нужно установить соединение, или сеанс. Он обеспечивает синхронизацию между устройствами и восстанавливает связь, если она прервалась.

Благодаря сеансам вы можете видеть, что собеседник что-то печатает или отправляет вам картинки или видео. Но главная задача этого соединения — обеспечить стабильное соединение для передачи данных.

**Транспортный уровень.**Когда соединение установлено и данные унифицированы, пора передавать их. Этим занимается транспортный уровень.

Здесь данные разбиваются на сегменты и к ним добавляется дополнительная информация — например, номер порта и контрольные суммы. Всё это нужно, чтобы данные дошли до пользователя в целостности.

**Сетевой уровень.**Теперь данным нужно найти маршрут к устройству нашего друга, а затем отправить их по нему. Поэтому данные упаковываются в пакеты и к ним добавляются IP-адреса.

Чтобы получить IP-адрес устройств, которым нужно отправить пакеты, маршрутизаторы (устройства сетевого уровня) обращаются к ARP. Этот протокол быстро найдёт адрес получателя и отдаст его нам.

**Канальный уровень.**Здесь данные передаются от одного MAC-адреса к другому. Изначальный текст делится на фреймы — с заголовками и контрольными суммами для проверки целостности данных.

**Физический уровень.**И на самом нижнем уровне данные в виде электрических сигналов передаются по проводам, кабелям или по радиоволнам. Тут только одна задача — как можно быстрее откликаться на сигналы свыше.

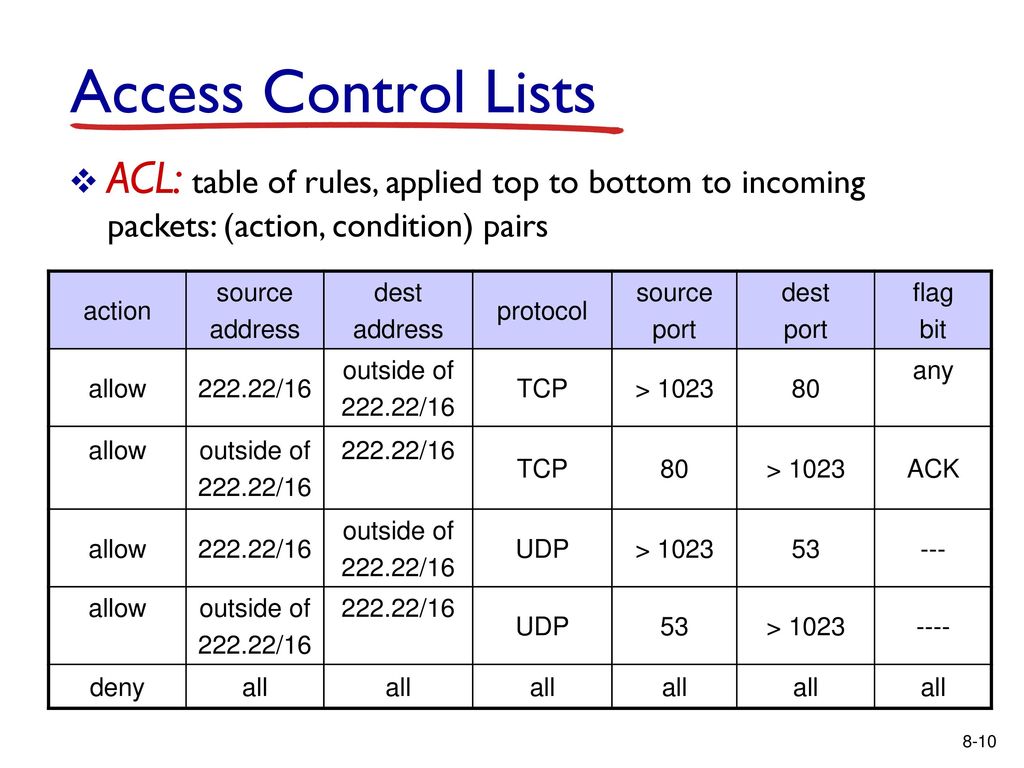
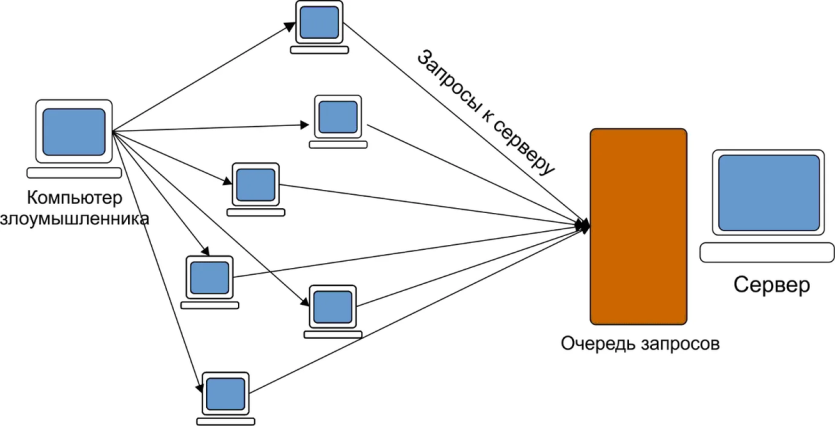
После прохождения всех уровней модели OSI сообщение успешно доставляется на устройство нашего друга. Правда, в реальности это занимает всего миллисекунды.

Модель OSI описывает, как работает весь интернет: как электрические сигналы преобразуются в картинки с котиками и как устройства обмениваются этими данными.

Модель включает семь уровней:

* физический;
* канальный;
* сетевой;
* транспортный;
* сеансный;
* представления;
* прикладной.

На каждом уровне находятся определённые протоколы, которые помогают данным перемещаться или превращаться в удобный для пользователей формат.



ACL (Access Control List) — это набор текстовых выражений, которые что-то разрешают, либо что-то запрещают. Обычно ACL разрешает или запрещает IP-пакеты, но помимо всего прочего он может заглядывать внутрь IP-пакета, просматривать тип пакета, TCP и UDP порты.

Также ACL существует для различных сетевых протоколов (IP, IPX, AppleTalk и так далее). В основном применение списков доступа рассматривают с точки зрения пакетной фильтрации, то есть пакетная фильтрация необходима в тех ситуациях, когда у вас стоит оборудование на границе Интернета

Возможности расширенных ACL богаче стандартных, кроме того, они могут расширяться дополнительными технологиями:

Dynamic ACL — ACL, в котором некоторые строчки до поры до времени не работают, но когда администратор подключается к маршрутизатору по telnet-у, эти строчки включаются, то есть администратор может оставить для себя «дыру» в безопасности для отладки или выхода в сеть.

Reflexive ACL — зеркальные списки контроля доступа, позволяют запоминать, кто обращался из нашей сети наружу (с каких адресов, с каких портов, на какие адреса, на какие порты) и автоматически формировать зеркальный ACL, который будет пропускать обратный трафик извне вовнутрь только в том случае, если изнутри было обращение к данному ресурсу.

TimeBased ACL, как видно из названия, это ACL, у которых некоторые строчки срабатывают только в какое-то время. Например, с помощью таких ACL легко настроить, чтобы в офисе доступ в интернет был только в рабочее время.

Все ACL (и стандартные, и расширенные) можно задавать по-разному: именованным и нумерованным способом. Первый предпочтительнее, так как позволяет затем редактировать ACL, в случае же использования нумерованного способа, ACL можно только удалить целиком и заново создать, либо дописать очередную строчку в конец.

5. Применение ACL

ACL используются для решения различных задач, связанных с управлением и фильтрацией трафика в сети. Основные сценарии применения ACL:

Фильтрация трафика:

Блокировка ненужного трафика (например, запрет доступа к определенным сайтам или сервисам).

Разрешение доступа только для конкретных устройств или подсетей.

Настройка NAT (Network Address Translation):

Использование ACL для выбора трафика, который будет подвергаться трансляции IP-адресов.

Обеспечение безопасности:

Защита сети от внешних атак (например, блокировка входящего трафика с определенных IP-адресов).

Ограничение доступа к критически важным ресурсам (например, серверам или маршрутизаторам).

Управление доступом:

Разграничение доступа для разных пользователей или групп пользователей.

Каждый интерфейс может иметь только один ACL на входящий и один на исходящий трафик.

Если требуется больше правил, используйте расширенные ACL.

Логика размещения ACL:

ACL должны быть размещены как можно ближе к источнику трафика, который они фильтруют.

Например, если нужно запретить доступ из Интернета к внутренней сети, ACL следует разместить на внешнем интерфейсе маршрутизатора.

Резервное копирование ACL:

Регулярно сохраняйте конфигурацию ACL, чтобы избежать потери настроек в случае сбоя.

Преимущества и недостатки ACL

Преимущества:

Гибкость настройки (можно фильтровать трафик по множеству параметров).

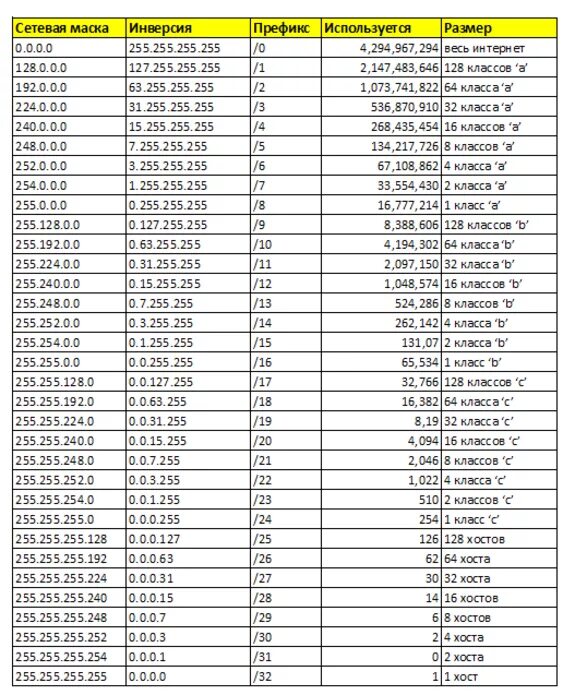
Простота реализации.

Широкое применение в сетях различного масштаба.

Недостатки:

Ограниченная производительность на высоконагруженных сетях.

Сложность управления большим количеством правил.



Необходимость точной настройки, чтобы избежать ошибок (например, запрета всего трафика).



AAA (Authentication Authorization and Accounting) — система аутентификации авторизации и учета событий, встроенная в операционную систему Cisco IOS, служит для предоставления пользователям безопасного удаленного доступа к сетевому оборудованию Cisco. Она предлагает различные методы идентификации пользователя, авторизации, а также сбора и отправки информации на сервер.

Однако мало того, что ааа по умолчанию выключена; конфигурация этой системы — дело довольно запутанное. Недочеты в конфигурации могут привести либо к нестабильному, небезопасному подключению, либо к отсутствию какого-либо соединения в принципе

Преимущество конфигурации aaa в том, что она содержит множество методов аутентификации (в отличие от предыдущего случая). Включение aaa происходит путем добавления команды aaa new-model в режиме глобальной конфигурации. Далее предстоит выбор методов аутентификации. Все методы организуются в списки, которым присваивается либо значение default, либо конкретное имя списка (list-name). Таким образом, на разные типы линий (aux, vty, con...) можно «повесить» разные методы аутентификации, рСлужба World Wide Web (WWW, Всемирная паутина, веб-служба) предоставляет доступ к связанным между собой электронным документам, хранящимся на различных веб-серверах в сети Интернет. Эта самая популярная

служба Интернета, образующая его единое информационное пространство.

Отдельные электронные документы, составляющие пространство Всемирной

паутины, называются веб-страницами (например, HTML-файл, содержащий

ссылки на другие объекты разного типа). Группы веб-страниц, объединенных

тематически, а также связанных между собой ссылками и обычно хранящихся на одном веб-сервере называют веб-сайтами (веб-узлами).

Сетевая веб-служба WWW работает по принципу клиент-сервер. Вебсервер – это программа, хранящая веб-страницы и связанные с ними объекты в папках компьютера1

, на котором она запущена, и обеспечивающая доступ к этим объектам по URL-адресам.

Веб-клиент (браузер) представляет собой приложение, которое устанавливается на компьютере конечного пользователя и служит для загрузки и

просмотра веб-страниц. Одной из важных функций браузера является поддержка пользовательского графического интерфейса для поиска, просмотра

веб-страниц, навигации между уже просмотренными страницами, хранения

истории посещений. Помимо этого, веб-браузер предоставляет пользователю возможность манипулирования страницами: сохранение их в файле на

диске своего компьютера, печать на принтере, поиск информации в пределах страницы, добавление страницы в закладки, изменение кодировки, просмотр исходного кода и др.

В сети Интернет символьные и цифровые адреса применяются одновременно. Между доменным именем и IP-адресом, относящимся к одному и

тому же сетевому узлу, нет никакой функциональной зависимости. Компьютеры, имена которых относятся к одному и тому же домену, могут иметь абсолютно независимые друг от друга IP-адреса. В процессе передачи данных

доменный адрес преобразуется в IP-адрес.

Установлением соответствия между символьными именами и цифровыми адресами занимается служба DNS – система доменных имен (Domain

Name System). В сети выделяются специальные компьютеры, называемые

серверами имен или DNS-серверами, на которых хранятся таблицы соответствия друг другу доменного имени и IP-адреса. Например:

pnu.edu.ru – 85.142.70.252

ege.edu.ru – 62.76.166.20

Таким образом, DNS представляет собой, с одной стороны, распределенную между серверами имен базу данных, а с другой – протокол прикладного уровня, организующий взаимодействие между компьютерами

Рассмотрим модель сетевого взаимодействия двух участников (абонентов). В литературе по защите информации их принято обозначать английским

буквами A и B, и называть именами Алиса и Боб соответственно. В качестве

таких участников сетевого взаимодействия могут выступать обычные пользователи на конечных узлах (например, желающие обменяться письмами по

электронной почте), программные модули (например, веб-браузер и вебсервер) или аппаратные устройства (например, маршрутизаторы, обменивающиеся таблицами маршрутизации в системе DNS).

Алиса отправляет некоторое сообщение (информационное или управляющее) Бобу по открытому информационному каналу связи (рис. 3.1). Открытый канал связи подразумевает доступность для прослушивания другим

лицам, отличным от отправителя Алиса и получателя Боба. Обычно «другим

лицом» является злоумышленник E, целью которого является нанесение

вреда абонентам.

‒ конфиденциальность – гарантия того, что только отправитель и

предполагаемый получатель способны понять содержимое передаваемых

сообщений. Поскольку злоумышленники могут перехватывать сообщения,

они должны быть зашифрованы;

‒ аутентификация – подтверждение того, что информация получена

из законного источника, и получатель действительно является теми, за кого

себя выдает.

‒ целостность данных – гарантия того, что информация при хранении

или передаче не изменилась (случайно или намеренно);

‒ управление доступом – гарантия того, что пользователи, пытающиеся получить доступ к ресурсам, смогут сделать это только в том случае, если

обладают соответствующими правами доступа и осуществляют этот доступ

определенным образом.

Для обеспечения данных свойств безопасной связи применяются различные механизмы и сервисы безопасности, которые представляют собой

программные и программно-аппаратные средства (например, программы

или устройства для шифрования, протоколы обмена сообщениями и др.).

1. Основные понятия AAA

AAA — это система управления доступом к сетевым ресурсам, которая обеспечивает три основные функции:

1. \*\*Аутентификация (Authentication)\*\*:

- Проверка подлинности пользователя (например, проверка логина и пароля).

2. \*\*Авторизация (Authorization)\*\*:

- Определение прав доступа пользователя к определенным ресурсам или функциям.

3. \*\*Учет (Accounting)\*\*:

- Сбор информации о действиях пользователя (например, время подключения, выполненные команды).

2. Преимущества AAA

1. \*\*Гибкость\*\*:

- Поддержка различных методов аутентификации (локальная база, RADIUS, TACACS+ и др.).

2. \*\*Безопасность\*\*:

- Увеличение уровня защиты сети за счет строгого контроля доступа.

3. \*\*Управление доступом\*\*:

- Возможность разграничения прав доступа для разных пользователей или групп пользователей.

4. \*\*Сбор данных\*\*:

- Возможность отслеживания действий пользователей для целей аудита и анализа.

3. Включение AAA на оборудовании Cisco

Для включения AAA на маршрутизаторе или коммутаторе Cisco используется команда:

После этого можно настраивать методы аутентификации, авторизации и учета.

4. Методы аутентификации в AAA

AAA поддерживает несколько методов аутентификации, которые можно комбинировать в списки:

1. Локальная аутентификация:

- Используется локальная база пользователей, настроенная на устройстве.

- Пример:

```

aaa authentication login DEFAULT local

```

2. Аутентификация через RADIUS:

- Используется внешний сервер RADIUS для проверки подлинности.

- Пример:

```

aaa authentication login MY\_RADIUS\_LIST group radius

```

3. \*\*Аутентификация через TACACS+\*\*:

- Используется внешний сервер TACACS+ для проверки подлинности.

- Пример:

```

aaa authentication login MY\_TACACS\_LIST group tacacs+

```

4. \*\*Комбинированная аутентификация\*\*:

- Используется несколько методов (например, сначала RADIUS, затем локальная база).

- Пример:

```

aaa authentication login COMBINED\_LIST group radius local

```

5. Методы авторизации в AAA\*\*

Авторизация определяет, какие действия или команды может выполнять пользователь после успешной аутентификации.

Современный интернет работает посредством OSI — сетевой инфраструктуры на основе протоколов OSI/ISO. Эта инфраструктура позволяет устройствам в сети взаимодействовать друг с другом. Взаимодействие происходит на разных уровнях, каждый из которых работает по правилам протоколов. Всего выделяют семь уровней взаимодействия: физический, канальный, сетевой, транспортный, сеансовый, представления и прикладной. Разберёмся, что они из себя представляют.

DDoS-атака может обрушиться на каждый из семи уровней, но чаще всего их инициируют на сетевом и транспортном уровне — **низкоуровневые атаки**, а также на сеансовом и прикладном — **высокоуровневые атаки**. Рассмотрим, что из себя представляют типы атак на прикладном уровне модели OSI и других обозначенных уровнях.

Типы DDoS-атак

**Низкоуровневые атаки**:

* **Атаки на сетевом уровне** OSI представляют из себя «забивание» канала. Примером может быть CMP-флуд — атака, которая использует ICMP-сообщения, которые снижают пропускную способность атакуемой сети и перегружают брандмауэр. Хост постоянно «пингуется» нарушителями, вынуждая его отвечать на ping-запросы. Когда их приходит значительное количество, пропускной способности сети не хватает и ответы на запросы приходят со значительной задержкой. Для предотвращения таких DDoS-атак можно отключить обработку ICMP-запросов посредством Firewall или ограничить их количество, пропускаемое на сервер.
* **Атаки транспортного уровня** выглядят как нарушение функционирования и перехват трафика. Например, SYN-флуд или Smurf-атака (атака ICMP-запросами с изменёнными адресами). Последствия такой DDoS-атаки — превышение количества доступных подключений и перебои в работе сетевого оборудования.

**Высокоуровневые атаки**:

* **На сеансовом уровне** атакам подвергается сетевое оборудование. Используя уязвимости программного обеспечения Telnet-сервера на свитче, злоумышленники могут заблокировать возможность управления свитчем для администратора. Чтоб избежать подобных видов атак, рекомендуется поддерживать прошивки оборудования в актуальном состоянии.
* Высокоуровневые **атаки прикладного уровня** ориентированы на стирание памяти или информации с диска, «воровство» ресурсов у сервера, извлечение и использование данных из БД. Это может привести к тотальной нехватке ресурсов для выполнения простейших операций на оборудовании. Наиболее эффективный способ предупреждения атак – своевременный мониторинг состояния системы и программного обеспечения.

Несмотря на то что разработчики ПО устраняют проблемы с безопасностью и регулярно выпускают обновления, злоумышленники постоянно совершенствуют возможности DDoS — придумывают новые способы привести системы к отказу.

Чтобы защитить свой проект от атак DDoS на низком уровне, стоит увеличить ёмкость канала или делать профилактику и глубокий анализ сетевой инфраструктуры. Чтобы защититься от высокоуровневых атак, рекомендуется регулярно анализировать TCP-клиентов и TCP-пакеты на сервере, а также использовать постоянный мониторинг состояния системы в целом и программного обеспечения в частности.

Снифферы (sniffers) — это программы, способные перехватывать и анализировать сетевой трафик.

Они полезны в тех случаях, когда нужно извлечь из потока данных какие-либо сведения (например, пароли) или провести диагностику сети.

Перехватить трафик через сниффер можно следующими способами:

путём прослушивания в обычном режиме сетевого интерфейса;

подключением в разрыв канала;

перенаправлением трафика;

посредством анализа побочных электромагнитных излучений;

при помощи атаки на уровень канала и сети, приводящей к изменению сетевых маршрутов.

Поток данных, перехваченный сниффером, подвергается анализу, что позволяет:

выявить паразитный трафик;

обнаружить активность вредоносных и нежелательных программ;

произвести перехват любого зашифрованного или незашифрованного трафика пользователя для извлечения паролей и других ценных данных.

.18.09.2024

Можно заметить, что инструменты сниффинга трафика незаменимы в мобильном тестировании, предоставляя возможность оперативно и грамотно подходить к исследованиям сетевых взаимодействий, осуществляемых приложениями. К числу таких инструментов, как уже указано, относятся Fiddler, HTTP Catcher, Charles Proxy, Burp Suite, Packet Capture, Reqable и Proxyman, каждый из которых обладает своим набором характеристик и особенностей, которые следует учитывать в процессе их применения.

Cовокупность удобства конкретного инструмента, cоответсвия вашим потребностям и подходящая ценовая политика, то из чего складывается представление об идеальном инструменте или группе инструментов для вас. Финансовый аспект использования снифферов варьируется от полностью бесплатных решений до платных продуктов, предлагающих расширенные функциональные возможности и техническую поддержку. Также стоит упомянуть различие в доступности – наличие мобильных или десктопных версий, масштабируемость и адаптивность инструментов сниффинга к требованиям конкретного проекта у разных инструментов разная. Кроме того, правильное использование снифферов в сочетании с методами тестирования может привести к повышению общего уровня безопасности и стабилизации производительности мобильного приложения.



Обозреваемая информация представляет собой анализ инструментов с целью рассказать о новых и старых инструментах, предназначенных для проведения тестирования. Рассмотрим более подробно, как и зачем используются снифферы трафика в мобильном тестировании, а также некоторые продвинутые методы их применения.

Зачем нужны снифферы в мобильном тестировании?

Снифферы трафика играют важную роль в мобильном тестировании, предоставляя тестировщикам возможность глубже погружаться в сетевые процессы приложения. Эти инструменты не только помогают анализировать и диагностировать сетевые проблемы, но и обеспечивают высокий уровень безопасности данных, выявляя скрытые угрозы и уязвимости. Рассмотрим более подробно, как и зачем используются снифферы трафика в мобильном тестировании, а также некоторые продвинутые методы их применения.

Анализ сетевого трафика

Одной из основных функций снифферов трафика является анализ сетевого трафика. Снифферы позволяют тестировщикам видеть, какие данные передаются между мобильным приложением и сервером, включая заголовки запросов, тело запросов и ответы. Это дает возможность проверить:

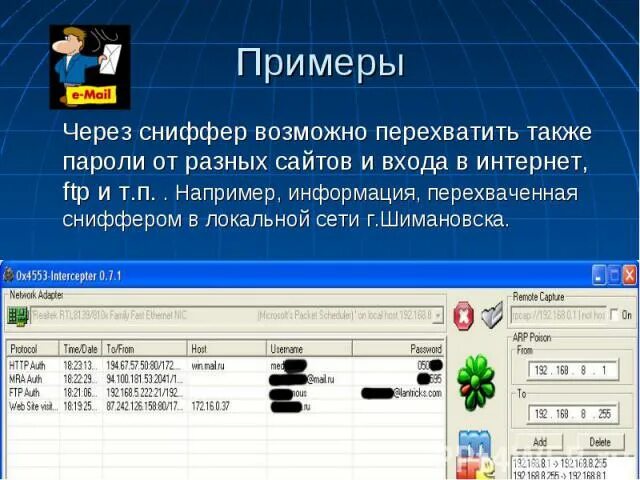
Соответствие протоколов: Проверка того, что приложение использует правильные протоколы для передачи данных, например, HTTPS вместо HTTP, чтобы обеспечить безопасное соединение.

Заголовки запросов и ответы: Анализ информации, передаваемой в заголовках, помогает понять, какие данные приложение раскрывает о пользователе и устройстве. Это также помогает выявлять неправильное или избыточное использование заголовков, которые могут представлять угрозу безопасности.

Оптимизация данных: Анализ трафика помогает понять, не передаются ли излишние данные, что может замедлять приложение или увеличивать нагрузку на сеть.

Обнаружение уязвимостей

Снифферы трафика помогают находить уязвимости в мобильных приложениях, связанные с неправильной передачей или хранением данных. Это особенно важно в условиях, когда конфиденциальная информация, такая как пароли, номера кредитных карт или персональные данные, может быть передана или сохранена без надлежащего шифрования. Некоторые продвинутые применения включают:



Выявление слабого шифрования: Снифферы позволяют проверить, используется ли сильное шифрование при передаче данных. Если данные передаются без шифрования или с использованием устаревших алгоритмов, это представляет угрозу безопасности.

Инъекции и атаки на API: Тестировщики могут использовать снифферы для симуляции различных видов атак, таких как SQL-инъекции или атаки на API. Это позволяет выявить уязвимости в коде и структуре приложения.

Анализ поведения при утечке данных: Снифферы помогают понять, что происходит с данными в случае утечки. Например, можно проверить, как приложение реагирует на несанкционированный доступ и какие данные могут быть скомпрометированы.

Диагностика проблем

Снифферы трафика полезны для диагностики различных проблем в работе мобильных приложений. Они позволяют тестировщикам выявлять ошибки в работе API, неправильную маршрутизацию запросов, проблемы с производительностью и другие технические неполадки. Продвинутые способы применения включают:

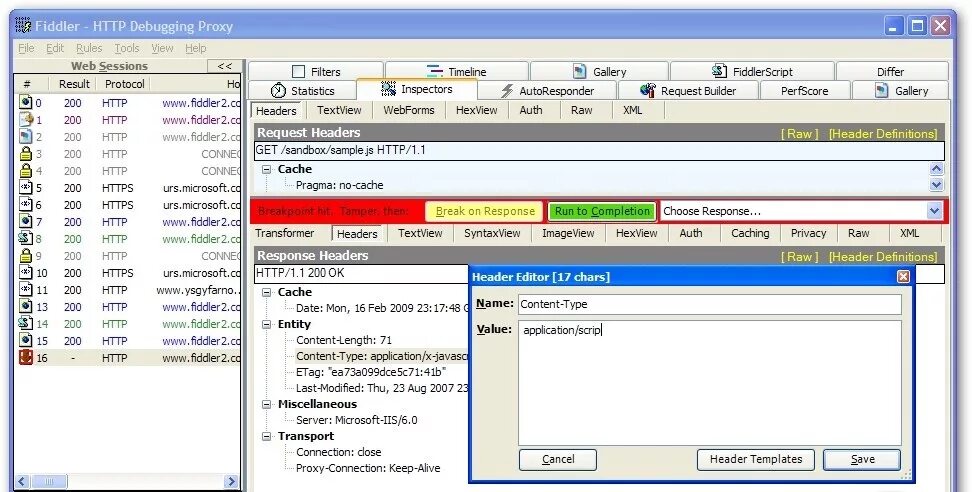
Анализ производительности сети: Снифферы помогают выявлять узкие места в сетевых запросах, такие как задержки или ошибки передачи данных, что позволяет оптимизировать производительность приложения.

Имитация сетевых условий: Снифферы, такие как Charles Proxy, позволяют имитировать различные сетевые условия, включая низкую скорость соединения или высокую задержку, чтобы протестировать работу приложения в реальных условиях.

Трассировка проблем с маршрутизацией: Снифферы помогают отслеживать маршрут запросов и ответов через различные серверы и узлы сети, что позволяет выявлять проблемы с маршрутизацией или конфигурацией сети.

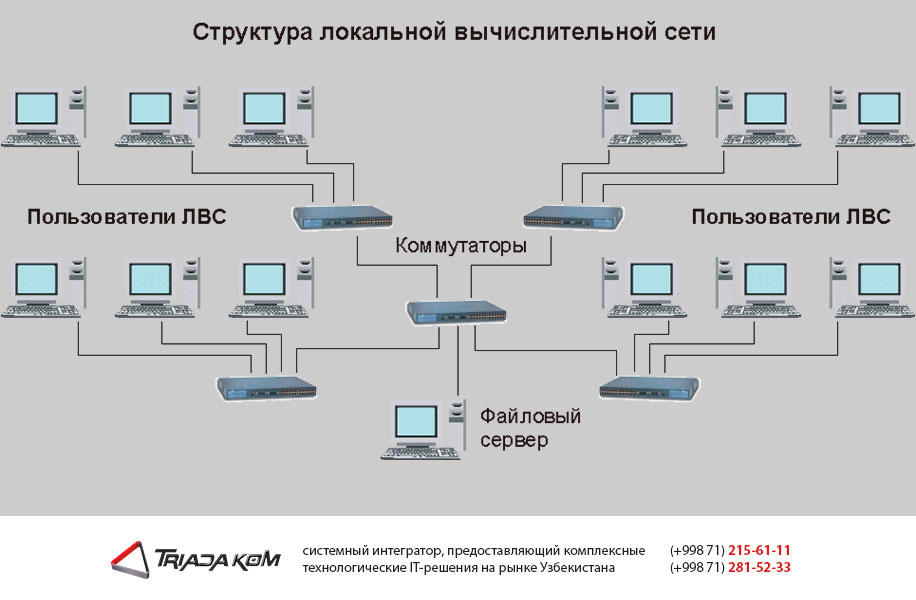
Перейдём непосредственно к обзору инструментов.

Fiddler



Fiddler — один из самых популярных инструментов для анализа HTTP/HTTPS трафика. Он поддерживает множество возможностей, полезных для тестирования мобильных приложений, классическое решение, по которому есть туториалы и большое сообщество:

Перехват и анализ трафика: Fiddler позволяет перехватывать все HTTP/HTTPS запросы и ответы, проходящие через устройство, на котором установлено приложение.



Дебаггинг: Инструмент предоставляет возможность редактирования запросов и ответов в реальном времени, что полезно для выявления и исправления ошибок.

Поддержка SSL: Fiddler может работать с зашифрованным трафиком, что делает его незаменимым при тестировании приложений, использующих HTTPS.

Поддержка gRPC: Fiddler может перехватывать и анализировать трафик, использующий протокол gRPC, что важно для микросервисных архитектур.

Функции командной работы: Fiddler Everywhere включает инструменты для совместной работы, такие как возможность делиться перехваченными сессиями и комментировать их, что облегчает работу в команде.

Поддержка современных протоколов: Fiddler поддерживает работу с HTTP/2 и WebSocket, что позволяет анализировать трафик современных веб-приложений.

AutoResponder: Эта функция позволяет автоматически отвечать на запросы предопределенными ответами, что полезно для симуляции различных сценариев работы приложения.

Форма решения: Fiddler представлен в виде десктопного приложения и поддерживает интеграцию с мобильными устройствами для перехвата их трафика.

Платность: Fiddler доступен в бесплатной версии с основными функциями. Однако для более продвинутых возможностей, таких как автоматизация тестирования и глубокий анализ, может потребоваться Fiddler Everywhere — платная версия с подпиской.

HTTP Catcher

HTTP Catcher — это мобильное приложение для iOS, которое позволяет перехватывать и анализировать сетевой трафик прямо на устройстве. Любим и используется мной довольно часто из-за ряда преимуществ:

Интуитивный интерфейс: HTTP Catcher имеет простой и удобный интерфейс, который облегчает работу с трафиком даже для новичков.

Поддержка HTTPS: Приложение может перехватывать зашифрованный трафик, что позволяет проверять безопасность данных, передаваемых между приложением и сервером.

Фильтрация и поиск: HTTP Catcher позволяет легко находить нужные запросы и ответы благодаря встроенным функциям фильтрации и поиска.

Форма решения: HTTP Catcher доступен в виде мобильного приложения для iOS.

Платность: HTTP Catcher доступен в App Store бесплатно с базовыми функциями. Однако для использования всех возможностей, таких как продвинутая фильтрация и экспорт данных, потребуется приобретение подписки.

Charles Proxy

Charles Proxy — еще один популярный инструмент для анализа трафика, который поддерживает работу с мобильными устройствами:

Поддержка HTTPS: Charles Proxy позволяет перехватывать и расшифровывать HTTPS трафик, что делает его отличным инструментом для анализа безопасности.

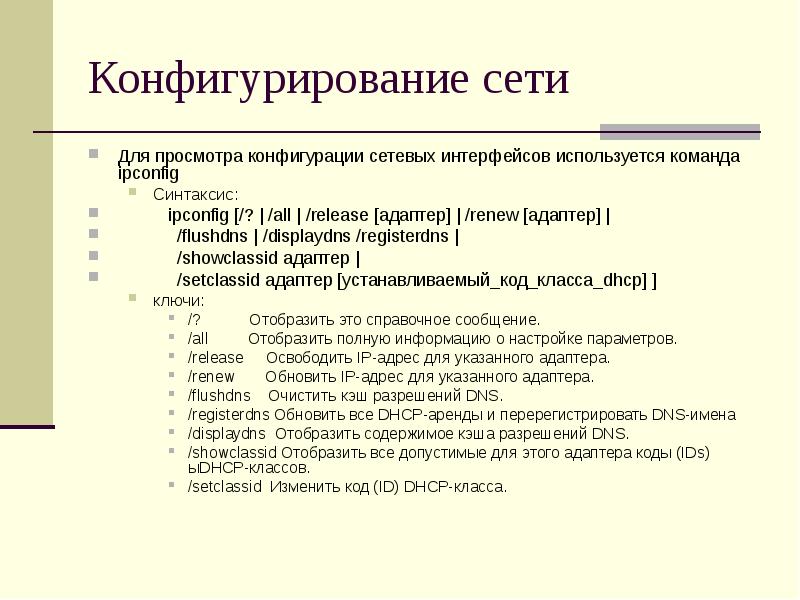
Имитация условий сети: Charles Proxy позволяет симулировать различные сетевые условия, такие как низкая скорость соединения, высокая задержка или потеря пакетов. Это помогает протестировать, как приложение будет работать в реальных условиях, когда соединение не всегда идеально, и выявить потенциальные проблемы с производительностью.

Интуитивный интерфейс: Charles Proxy предоставляет удобные инструменты для анализа и редактирования трафика, что делает его отличным выбором для мобильного тестирования.

Запись и воспроизведение сессий: Charles Proxy позволяет записывать сессии трафика и воспроизводить их позже, что полезно для регрессионного тестирования и анализа прошлых действий в приложении.

Форма решения: Charles Proxy доступен в виде десктопного приложения с поддержкой интеграции мобильных устройств через прокси-сервер.

Платность: Charles Proxy является платным инструментом с разовой лицензией. Бесплатной версии нет, однако доступна пробная версия с ограниченными возможностями.



Дилемма оператора  
  
Одна из проблем автоматизации сетей состоит в том, что у нас нет стандартизованных интерфейсов для настройки сетевого оборудования различных производителей. Другими словами, сложно сконфигурировать многовендорную сеть с помощью программных средств, когда на каждом устройстве этот процесс отличается.  
  
При управлении через командную строку (Command Line Interface — CLI), сетевые операторы сталкиваются со значительными отличиями в её синтаксисе на оборудовании разных производителей. Например, команды, которыми протокол BGP настраивается на устройстве Cisco под управлением операционной системы IOS, не подойдут для выполнения той же задачи на устройстве Juniper под управлением JUNOS. Переход с командной строки на конфигурирование оборудования через интерфейсы прикладного программирования (API) не решает проблему. Интерфейсы, предлагаемые производителями сетевого оборудования, по реализации и функциональности отличаются друг от друга не меньше CLI. Отчасти это обусловлено спецификой сетевых операционных систем. Не во все сетевые ОС закладывалась возможность использовать API, и производителям бывает непросто встроить их поддержку в свои продукты.  
  
Конфигурационные модели

Один из вариантов реализации программной настройки сетевого оборудования – конфигурационные модели. Модель данных определяет структуру, синтаксис и семантику данных, которые мы загружаем на сетевое оборудование. При их передаче, в связке с моделью данных действует протокол, который определяет их кодирование и сам механизм передачи на устройство. Другими словами, модель определяет правила, с помощью которых мы задаем параметры конфигурации сетевого оборудования. А транспортный протокол передаёт эти данные на оборудование. Конфигурационную модель можно сравнить с протоколом SNMP, который изначально был разработан для управления сетевыми устройствами, но в силу ряда причин используется в основном для мониторинга. — прим. ред.

Для нас, привыкших рассматривать всякое сетевое устройство, исходя из интерфейса его командной строки, «модель» представляется в виде строк конфигурации. К примеру, «interface gigabitethernet 0/1» или «router bgp 65432» открывает пользователю Cisco IOS режим конфигурации Ethernet-интерфейса или процесса BGP-маршрутизации. А транспортом в этом случае выступает протокол SSH.  
  
Многие понимают, что будущее конфигурирования сетевых устройств остаётся за интерфейсами прикладного программирования. Но отойдя от командной строки, мы не ушли от главной проблемы: унификация настройки оборудования теперь затрудняется не синтаксисом командной строки, а моделями данных для протоколов вроде NETCONF, созданными на базе описательных языков моделирования. Эти модели часто пишут на базе языка YANG и они, как и синтаксис командной строки, сильно разнятся от производителя к производителю. В результате, пользователи остаются всё с той же проблемой: на разном железе процесс выполнения одной и той же задачи по-прежнему отличается.  
  
Индустрия сталкивается с проблемой: мы хотим автоматизировать процесс конфигурирования сети, но не можем предугадать, какие модели и интерфейсы для этого понадобятся. Звучит очень знакомо. Вспомним о протоколе SNMP. SNMP — общепринятый стандарт. Все сетевые устройства поддерживают те или иные элементы баз MIB. Тем не менее, наиболее важные параметры уникальны для каждого производителя оборудования и требуют использования специфичных баз MIB.  
  
Сможем ли мы облегчить этот конфигурационный кошмар, перейдя с CLI и SNMP на NETCONF (или любой другой способ передачи конфигурации) с YANG-моделями?  
  
Производители, естественно, будут отстаивать уникальность своих решений, а также абсолютную необходимость использования моделей, которые соответствуют их конкретным возможностям. И их можно понять. Было бы несправедливо критиковать производителей за их стремление отличаться от конкурентов.  
  
В свою очередь, сетевые операторы имеют такое же право отстаивать идею, что сети есть сети. BGP это BGP, ISIS это ISIS, OSPF это OSPF, а интерфейс Ethernet (по большей части) это интерфейс Ethernet, и т.д. И какая разница, конфигурируем мы железку Cisco, Juniper, Brocade или что-то ещё. Если на каждой из них нужно настроить одну и ту же функцию, несложно догадаться, что хотелось бы, чтобы она везде настраивалась одинаково! Как можно автоматизировать конфигурирование сети, когда, стоит нам только отвернуться, как где-то уже нужно подправить скрипт, добавить модуль, подождать, пока инструмент автоматизации получит патч от производителя, или потребуются ещё какие-нибудь пляски с бубном?  
  
OpenConfig и IETF приходят к стандартным сетевым моделям  
  
Сложившаяся ситуация беспокоит не только автора. У IETF есть несколько проектов разной степени завершенности, направленных на разработку стандартных сетевых моделей на базе YANG. К примеру: модель для управления сетевыми интерфейсами RFC7223, модель для инвентаризации и управления RFC7317, модель для настройки протокола SNMP RFC7407, а также модель для настройки протокола OSPF. Но для кого-то работа IETF продвигается слишком медленно, а кому-то просто не подходят модели, которые они предлагают.  
  
В связи с этим, несколько крупных компаний решили приблизить переход на стандартные сетевые модели, организовав проект OpenConfig. На главной странице OpenConfig пишет о себе так:

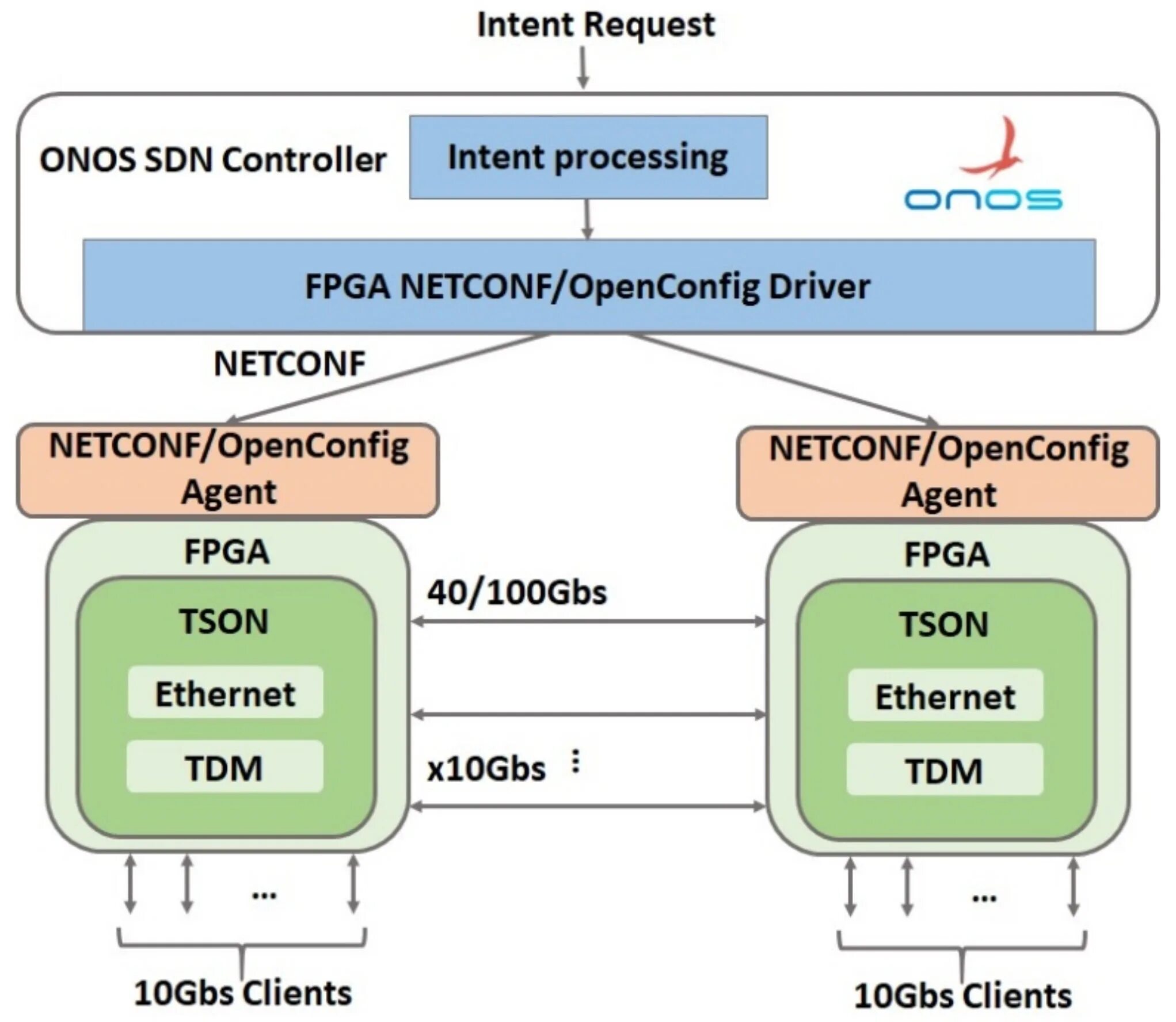
«OpenConfig — это неформальная рабочая группа операторов сетей, объединенных общей целью — перевести сети на более динамичную программируемую инфраструктуру путем внедрения принципов программно-определяемых сетей, таких как декларативная конфигурация и управление на основе моделей. Основное направление нашей работы — разработка единых моделей данных для конфигурации и управления, поддержка которых изначально встроена в программные и аппаратные платформы различных производителей оборудования»

Кто же эти компании? По состоянию на январь 2016, среди участников на  OpenConfig числились: Google, AT&T, British Telecom, Microsoft, Facebook, Comcast, Verizon, Level3, Cox Communications, Yahoo!, Apple, Jive Communications, Deutsche Telekom/TeraStream, и Bell Canada. Другими словами, КРУПНЫЕ поставщики услуг с огромными сетями и центрами обработки данных, которым нужно поддерживать постоянный рост и обеспечивать динамичные изменения в своих сетях, желательно, с минимизацией различных препятствий в их работе.  
  
На заре своей деятельности «неформальная рабочая группа» OpenConfig весьма продуктивно работала, создавая модели для BGP, MPLS и управления интерфейсами. Сейчас в доступе есть разные модели (например, модель для сетевой телеметрии). Все они опубликованы на Github, и быстро эволюционируют, ведь именно для быстрой эволюции OpenConfig и создавался. На странице FAQ OpenConfig пишут: «На данный момент формального руководства у проекта нет. Мы полагаемся на то, что у участников здесь общие цели и видение, их действия прозрачны и по общим вопросам они стараются поддерживать согласие».  
  
Конечно, если начинать размышлять о проекте вроде OpenConfig, обнаружатся два главных повода для беспокойства:

Будут ли производители сетевого оборудования поддерживать OpenConfig?

Не пренебрегает ли OpenConfig стандартами IETF?

Рассмотрим первый — поддержку производителями. Для производителей сетевого оборудования главный вопрос здесь состоит в том, насколько сложно им будет приспособить существующие сетевые ОС к работе на основе моделей. Например, для компании Juniper поддержка моделей IETF и OpenConfig не составит труда. JUNOS и так, по сути своей, управляется моделями. К тому же, Juniper намекает (к сожалению, ссылок на эти намеки автор не дает — прим. ред.), что поддержка OpenConfig будет в ближайших версиях JUNOS (возможно даже в 16.1). Компания Cisco анонсировала поддержку OpenConfig для IOS-XR (NCS, ASR, CRS, XR) ещё в ноябре 2015 (уже доступна поддержка двух моделей OpenConfig — BGP и Route Policy — прим. ред.).  
  
Добавят ли другие производители поддержку моделей OpenConfig? Время покажет. Но есть сильные подозрения, что так и будет. Вспомните, кто стоит за этой инициативой. Деньги в сетевой индустрии имеют право голоса. Потребитель, который может заплатить, как правило, получает, чего хочет.  
  
Как же IETF? OpenConfig идёт своей дорогой? И да, и нет. Всё несколько сложнее. Проект OpenConfig начинался отчасти как реакция на слабо упорядоченную и неспешную работу IETF. Тем не менее, OpenConfig сотрудничает с IETF, помогая им развивать различные стандарты. На странице годового отчета OpenConfig за 2015 внизу перечислены проекты, созданные рабочей группой OpenConfig для IETF. Так что, хоть OpenConfig и не желает ждать, пока IETF разберется со своими моделями, он признаёт важность IETF и даже привносит собственный вклад в их дело.  
  
Что может дать OpenConfig?

  
  
Основная идея в том, что OpenConfig ориентируется только на модели, уделяя меньшее внимание законченным API, включающим в себя как модель, так и транспорт (подробнее вопрос раскрывается в статье Джейсона Эделмана). Ведь наша главная цель — достичь целостности и предсказуемости модели на всех сетевых устройствах. Инструменты (транспорт), которыми мы загружаем модели на оборудование, важны, но в их случае проблема выбора и унификации стоит не так остро, как с представлением и моделированием данных. По большому счёту, нам всё равно, как данные были загружены. Но очень важно, чтобы они были представлены в нужном нам формате — чтобы мы могли их понять, то есть, чтобы наши инструменты легко могли их проанализировать и работать с ними.  
  
Когда стандартные сетевые модели станут встречаться повсеместно, мир откроется для разработки инструментария, платформ оркестровки, ПО для мониторинга и SDN контроллеров, и на рынке появятся прекрасные инструменты для конфигурирования и отчетности. Производителям ПО больше не нужно будет ориентироваться на самых распространенных вендоров, решая, чьё оборудование поддерживать, ведь рынком для них будет весь мир сетей.

Burp Suite

Burp Suite — это комплексное решение для тестирования безопасности веб-приложений, которое также поддерживает анализ мобильного трафика:

Перехват трафика: Burp Suite может перехватывать и анализировать HTTP/HTTPS запросы и ответы, что делает его полезным для поиска уязвимостей в мобильных приложениях.

Автоматизированные тесты безопасности: Инструмент предлагает ряд встроенных сканеров и автоматизированных тестов, которые позволяют выявлять уязвимости и ошибки в приложениях.

Интеграция с мобильными устройствами: Burp Suite можно настроить для работы с мобильными устройствами, что позволяет использовать его для анализа безопасности мобильных приложений.

Форма решения: Burp Suite доступен в виде десктопного приложения с возможностью настройки для работы с мобильными устройствами.

Платность: Burp Suite предлагает бесплатную версию (Community Edition) с ограниченными возможностями. Платная версия (Professional) предоставляет полный набор инструментов и автоматизацию, но доступна по подписке.



Packet Capture

Packet Capture — это мобильное приложение для Android, которое позволяет перехватывать и анализировать сетевой трафик непосредственно на устройстве:

Работа без root-доступа: Одним из ключевых преимуществ Packet Capture является то, что оно не требует root-доступа для работы, что делает его доступным для широкого круга пользователей.

Поддержка SSL: Приложение использует собственный сертификат для расшифровки SSL-трафика, что позволяет анализировать защищенные соединения.

Легкий интерфейс: Простой и удобный интерфейс делает приложение подходящим для быстрого анализа и диагностики сетевых проблем на Android-устройствах.

Экспорт пакетов: Возможность экспорта перехваченных данных в формате .pcap для дальнейшего анализа в других инструментах, таких как Wireshark.

Форма решения: Packet Capture доступен в виде мобильного приложения для Android.

Платность: Packet Capture бесплатно доступно в Google Play Store. Однако приложение поддерживается за счет рекламы, и некоторые функции могут быть ограничены без подписки.

Reqable

Reqable — это мощный инструмент для перехвата и анализа сетевого трафика, доступный как на Android, так и на iOS платформах. Он разработан специально для мобильных устройств, что делает процесс тестирования удобным и эффективным непосредственно на самом устройстве.

Основные возможности Reqable:

Перехват и анализ HTTP/HTTPS трафика: Reqable позволяет легко перехватывать и изучать сетевые запросы и ответы, что помогает в выявлении и устранении проблем с подключением и безопасностью.

Удобный пользовательский интерфейс: Интуитивно понятный интерфейс облегчает навигацию и работу с перехваченными данными, позволяя быстро находить и анализировать нужную информацию.

Редактирование запросов: Инструмент предоставляет возможность изменять и повторять сетевые запросы, что полезно при тестировании различных сценариев и обнаружении потенциальных уязвимостей.

Поддержка сертификатов SSL: Reqable способен работать с зашифрованным трафиком, обеспечивая полный доступ к данным для детального анализа.

Экспорт и импорт данных: Возможность экспорта перехваченных данных в различные форматы позволяет проводить дополнительный анализ с использованием других инструментов или делиться результатами с командой.

Форма решения: Reqable доступен в виде мобильного приложения для Android и iOS.

Платность: Reqable предлагает как бесплатную версию с базовыми функциями, так и платную подписку, которая открывает доступ к расширенным возможностям и снимает ограничения на объем перехватываемого трафика.

Proxyman

Proxyman — это современный и мощный инструмент для перехвата и анализа сетевого трафика, предназначенный для разработчиков и тестировщиков, работающих с мобильными приложениями. Этот инструмент особенно популярен среди тех, кто ищет простое, но эффективное решение для работы с HTTP/HTTPS трафиком на разных платформах.

Основные возможности Proxyman:

Перехват и анализ трафика: Proxyman позволяет перехватывать и анализировать HTTP/HTTPS запросы и ответы, проходящие через сеть. Он предоставляет подробную информацию о каждом запросе, включая заголовки, параметры и тело запроса.

Поддержка HTTPS: Proxyman легко справляется с перехватом и расшифровкой HTTPS трафика благодаря встроенной поддержке SSL-пиннинга, что позволяет тестировщикам анализировать зашифрованные данные.

Редактирование запросов и ответов: Пользователи могут изменять и повторять запросы, что полезно при тестировании различных сценариев и выявлении уязвимостей.

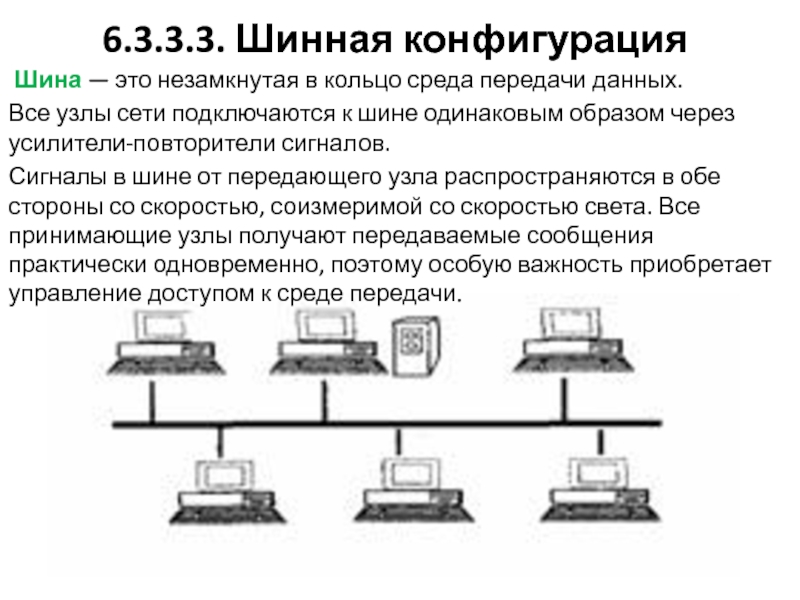
Кроссплатформенность: Proxyman поддерживает как мобильные устройства (iOS и Android), так и десктопные платформы (macOS и Windows). Это делает его универсальным инструментом для тестирования приложений на различных устройствах.

Простота интеграции: Proxyman легко интегрируется с мобильными устройствами через прокси-сервер, что позволяет быстро настроить перехват трафика без сложных конфигураций.

Форма решения: Proxyman доступен как десктопное приложение для macOS и Windows с поддержкой мобильных устройств (iOS и Android) через прокси-сервер.

Платность: Proxyman предлагает бесплатную пробную версию с ограниченным функционалом. Полная версия доступна по подписке или через разовую покупку, что открывает доступ ко всем функциям и снимает ограничения на использование.

Лекция 25.09.24



Туннели GRE

Обзор технологий VPN

Пакет PPTP

Пакет L2TP

Защита на третьем уровне сетевой модели OSI (сетевой уровень) обычно включает в себя меры, направленные на обеспечение безопасности передачи данных между устройствами в сети. Вот некоторые из основных методов защиты на этом уровне:

### 1. \*\*Сетевые брандмауэры (Firewalls)\*\*

- \*\*Функция:\*\* Фильтрация трафика на основе правил, определяющих, какой трафик разрешен или запрещен.

- \*\*Типы:\*\*

- \*\*Пакетный фильтр:\*\* Фильтрует трафик на основе IP-адресов, портов и протоколов.

- \*\*Состояниевое устройство (Stateful Firewall):\*\* Отслеживает состояние сессий и принимает решения о фильтрации на основе контекста.

- \*\*Следящий брандмауэр (Next-Generation Firewall, NGFW):\*\* Включает функции глубокой проверки пакетов, идентификации приложений и защиты от угроз.

### 2. \*\*Виртуальные частные сети (VPN)\*\*

- \*\*Функция:\*\* Создание зашифрованных туннелей для безопасной передачи данных через публичные сети (например, Интернет).

- \*\*Типы:\*\*

- \*\*IPsec VPN:\*\* Шифрование и аутентификация данных на основе протокола IPsec.

- \*\*SSL/TLS VPN:\*\* Использует SSL/TLS для создания безопасных соединений.

### 3. \*\*Маршрутизация с безопасностью (Secure Routing)\*\*

- \*\*Функция:\*\* Обеспечение безопасности маршрутизации путем проверки и фильтрации маршрутов, предотвращения атак типа "подмена маршрута" (Route Hijacking).

- \*\*Методы:\*\*

- \*\*Протоколы безопасности маршрутизации (например, BGPsec):\*\* Добавляют криптографическую подпись к обновлениям маршрутов.

- \*\*Фильтрация маршрутов:\*\* Использование ACL (Access Control Lists) для фильтрации нежелательных маршрутов.

### 4. \*\*Защита от DDoS-атак\*\*

- \*\*Функция:\*\* Обнаружение и защита от распределенных атак типа "отказ в обслуживании" (DDoS), которые направлены на перегрузку сетевых ресурсов.

- \*\*Методы:\*\*

- \*\*Анализаторы трафика:\*\* Мониторинг и анализ трафика для обнаружения аномальных паттернов.

- \*\*Блокировка трафика:\*\* Использование брандмауэров и других устройств для блокировки вредоносного трафика.

### 5. \*\*Сетевое шифрование (Network Encryption)\*\*

- \*\*Функция:\*\* Шифрование данных на сетевом уровне для защиты от перехвата и несанкционированного доступа.

- \*\*Методы:\*\*

- \*\*TLS/SSL:\*\* Шифрование трафика между клиентами и серверами.

- \*\*IPsec:\*\* Шифрование IP-пакетов для защиты данных на уровне сети.



### 6. \*\*Сетевая сегментация (Network Segmentation)\*\*

- \*\*Функция:\*\* Разделение сети на несколько сегментов для ограничения распространения атак и улучшения управления доступом.

- \*\*Методы:\*\*

- \*\*VLAN (Virtual LAN):\*\* Логическое разделение сети на виртуальные подсети.

- \*\*Зоны безопасности (Security Zones):\*\* Определение зон с разными уровнями доступа и безопасности.

### 7. \*\*Мониторинг и аудит сетевого трафика\*\*

- \*\*Функция:\*\* Непрерывный мониторинг и анализ сетевого трафика для обнаружения и реагирования на угрозы.

- \*\*Методы:\*\*

- \*\*Системы обнаружения вторжений (IDS):\*\* Обнаружение подозрительной активности в сети.

- \*\*Системы предотвращения вторжений (IPS):\*\* Не только обнаруживают, но и блокируют подозрительную активность.

### 8. \*\*Контроль доступа к сетевым ресурсам (Network Access Control, NAC)\*\*

- \*\*Функция:\*\* Обеспечение контроля доступа к сетевым ресурсам на основе политик безопасности.

- \*\*Методы:\*\*

- \*\*Аутентификация и авторизация:\*\* Проверка подлинности пользователей и устройств перед предоставлением доступа.

- \*\*Политики доступа:\*\* Определение правил, которые определяют, кто и как может получить доступ к сетевым ресурсам.

### 9. \*\*Защита от атак на протоколы маршрутизации\*\*

- \*\*Функция:\*\* Защита от атак, направленных на нарушение работы протоколов маршрутизации (например, BGP).

- \*\*Методы:\*\*

- \*\*Фильтрация маршрутов:\*\* Использование ACL для фильтрации нежелательных обновлений маршрутов.

- \*\*Протоколы безопасности маршрутизации (например, BGPsec):\*\* Добавление криптографической подписи к обновлениям маршрутов.

### 10. \*\*Резервное копирование и восстановление сетевых конфигураций\*\*

- \*\*Функция:\*\* Регулярное резервное копирование сетевых конфигураций для возможности быстрого восстановления в случае сбоя или атаки.

- \*\*Методы:\*\*

- \*\*Автоматизированное резервное копирование:\*\* Использование скриптов и инструментов для автоматизации процесса резервного копирования.

- \*\*Хранение резервных копий в безопасном месте:\*\* Обеспечение защиты резервных копий от несанкционированного доступа.

Эти методы и технологии помогают обеспечить безопасность на сетевом уровне, защищая данные и ресурсы от различных угроз.

Межсетевые экраны (брандмауэры) на третьем уровне сетевой модели OSI (сетевом уровне) работают на основе данных, которые могут быть извлечены из IP-пакетов. Вот основные данные, которые используются для принятия решений о фильтрации трафика:

### 1. \*\*IP-адреса\*\*

- \*\*Источник (Source IP):\*\* IP-адрес отправителя пакета.

- \*\*Назначение (Destination IP):\*\* IP-адрес получателя пакета.

- \*\*Применение:\*\* Брандмауэр может фильтровать трафик на основе IP-адресов, разрешая или запрещая трафик от/к определенным адресам.

### 2. \*\*Протоколы\*\*

- \*\*Тип протокола:\*\* Номер протокола, указанный в заголовке IP-пакета (например, TCP, UDP, ICMP).

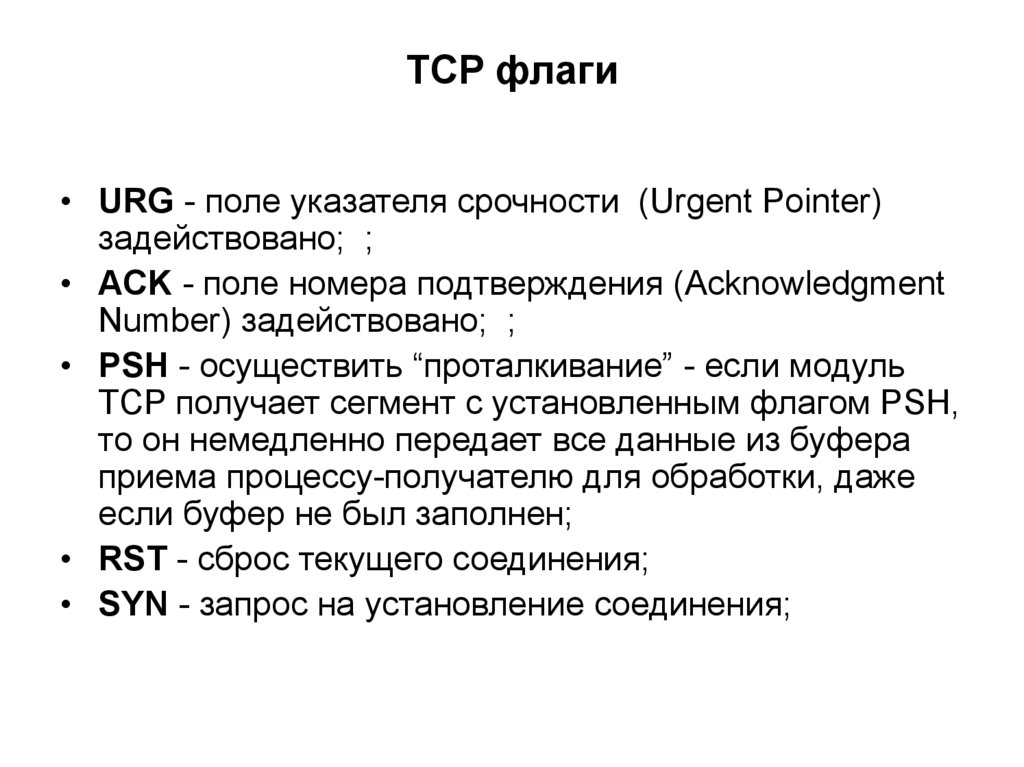
- \*\*Применение:\*\* Брандмауэр может фильтровать трафик на основе используемого протокола, разрешая или запрещая определенные типы протоколов.

### 3. \*\*Порты\*\*

- \*\*Источник (Source Port):\*\* Номер порта отправителя, используемый в заголовке транспортного уровня (например, TCP или UDP).

- \*\*Назначение (Destination Port):\*\* Номер порта получателя, используемый в заголовке транспортного уровня.

- \*\*Применение:\*\* Брандмауэр может фильтровать трафик на основе портов, разрешая или запрещая трафик к/от определенных портов.



### 4. \*\*Флаги TCP\*\*

- \*\*Флаги TCP:\*\* Флаги, используемые в заголовке TCP-пакета (например, SYN, ACK, FIN, RST).

- \*\*Применение:\*\* Брандмауэр может анализировать флаги TCP для обнаружения подозрительной активности, такой как сканирование портов или попытки установления соединения.

### 5. \*\*Типы ICMP\*\*

- \*\*Тип ICMP:\*\* Тип сообщения ICMP (например, Echo Request, Echo Reply, Destination Unreachable).

- \*\*Применение:\*\* Брандмауэр может фильтровать трафик на основе типов ICMP, разрешая или запрещая определенные типы сообщений ICMP.

### 6. \*\*Длина пакета\*\*

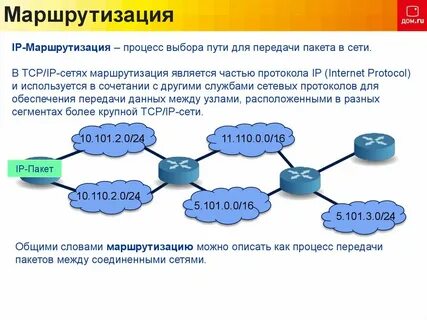
- \*\*Длина пакета:\*\* Общая длина IP-пакета.

- \*\*Применение:\*\* Брандмауэр может анализировать длину пакетов для обнаружения аномальной активности, такой как фрагментация пакетов или попытки переполнения буфера.

### 7. \*\*Маршрутизация\*\*

- \*\*Информация о маршрутизации:\*\* Данные о маршрутизации, такие как адреса следующего перехода (Next Hop) и метрики маршрута.

- \*\*Применение:\*\* Брандмауэр может анализировать маршруты для обнаружения нежелательных или вредоносных маршрутов.



### 8. \*\*Политики безопасности\*\*

- \*\*Правила брандмауэра:\*\* Набор правил, определяющих, какой трафик разрешен или запрещен на основе вышеуказанных данных.

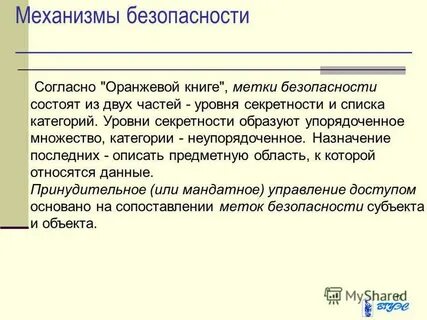
- \*\*Применение:\*\* Брандмауэр применяет эти правила для фильтрации трафика и принятия решений о доступе.

### 9. \*\*Состояние сессии\*\*

- \*\*Состояние сессии:\*\* Информация о текущих сессиях, отслеживаемая состояниевом брандмауэром.

- \*\*Применение:\*\* Состояниевое устройство (Stateful Firewall) отслеживает состояние сессий и может принимать решения о фильтрации на основе контекста, например, разрешая ответы на запросы, но блокируя незапрошенный трафик.

### 10. \*\*Метки безопасности\*\*

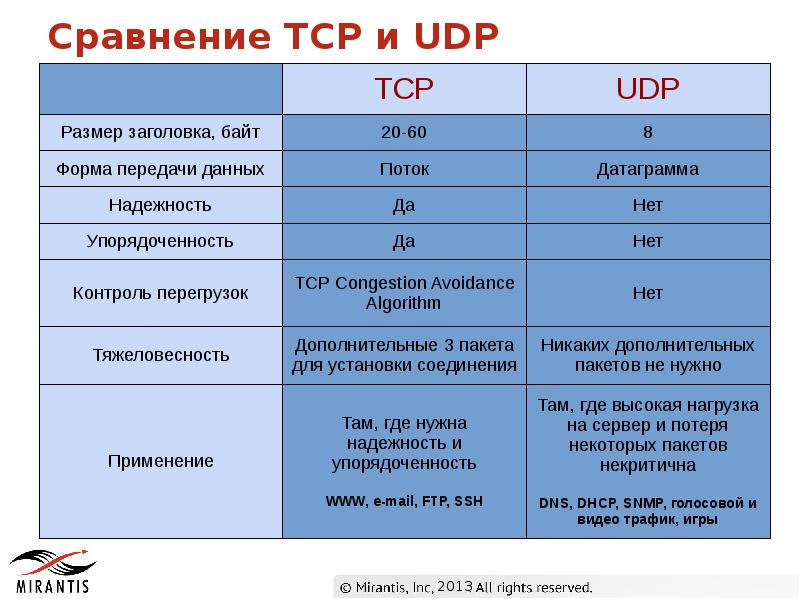


- \*\*Метки безопасности:\*\* Дополнительные метки, которые могут быть прикреплены к пакетам для указания уровня безопасности или категории.

- \*\*Применение:\*\* Брандмауэр может использовать эти метки для принятия решений о фильтрации трафика на основе политик безопасности.

Используя эти данные, межсетевые экраны на третьем уровне могут эффективно фильтровать трафик, обеспечивая безопасность сети и защиту от различных угроз.

На межсетевом уровне (третий уровень модели OSI) отслеживаются различные параметры и данные, которые помогают обеспечить безопасность и эффективность работы сети. Вот основные элементы, которые отслеживаются на этом уровне:



### 1. \*\*IP-адреса\*\*

- \*\*Источник (Source IP):\*\* IP-адрес отправителя пакета.

- \*\*Назначение (Destination IP):\*\* IP-адрес получателя пакета.

- \*\*Применение:\*\* Используются для маршрутизации пакетов и принятия решений о фильтрации трафика.

### 2. \*\*Протоколы\*\*

- \*\*Тип протокола:\*\* Номер протокола, указанный в заголовке IP-пакета (например, TCP, UDP, ICMP).

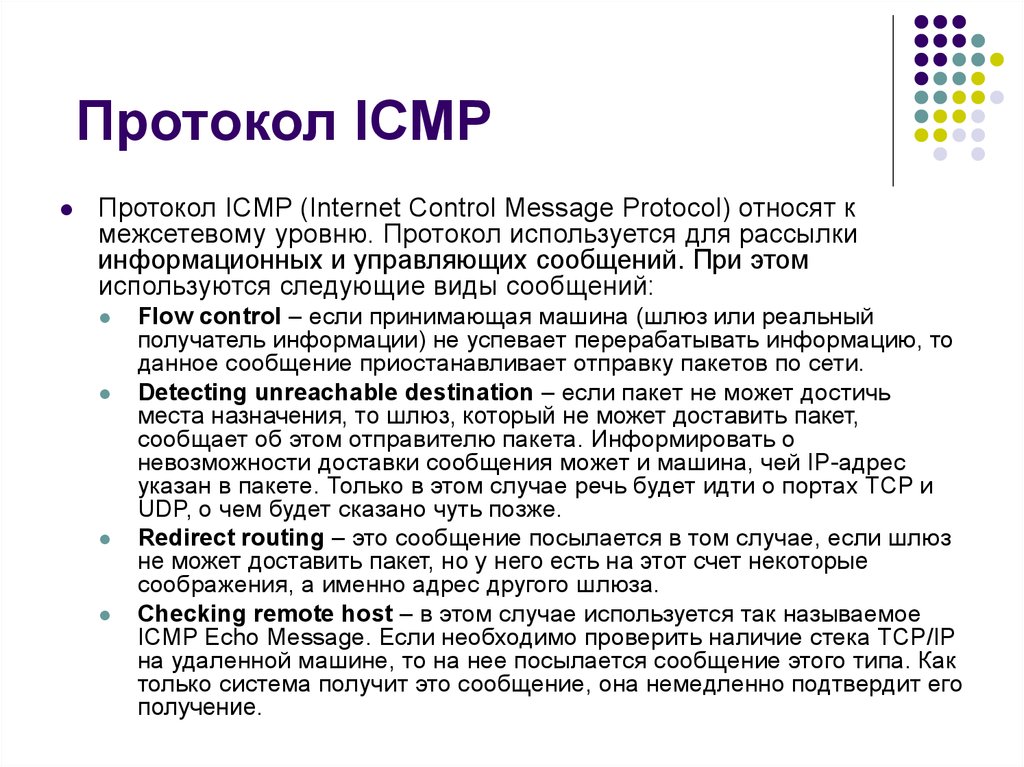
- \*\*Применение:\*\* Помогает определить, как обрабатывать пакеты и какие правила безопасности применять.

### 3. \*\*Порты\*\*

- \*\*Источник (Source Port):\*\* Номер порта отправителя, используемый в заголовке транспортного уровня (например, TCP или UDP).

- \*\*Назначение (Destination Port):\*\* Номер порта получателя, используемый в заголовке транспортного уровня.

- \*\*Применение:\*\* Помогает идентифицировать конкретные приложения и службы, к которым обращается трафик.



### 4. \*\*Флаги TCP\*\*

- \*\*Флаги TCP:\*\* Флаги, используемые в заголовке TCP-пакета (например, SYN, ACK, FIN, RST).

- \*\*Применение:\*\* Помогают отслеживать состояние соединений и обнаруживать подозрительную активность, такую как сканирование портов.

### 5. \*\*Типы ICMP\*\*

- \*\*Тип ICMP:\*\* Тип сообщения ICMP (например, Echo Request, Echo Reply, Destination Unreachable).

- \*\*Применение:\*\* Помогают отслеживать состояние сети и обнаруживать проблемы с доступом к ресурсам.

### 6. \*\*Длина пакета\*\*

- \*\*Длина пакета:\*\* Общая длина IP-пакета.

- \*\*Применение:\*\* Помогает обнаруживать аномальную активность, такую как фрагментация пакетов или попытки переполнения буфера.

### 7. \*\*Маршрутизация\*\*

- \*\*Информация о маршрутизации:\*\* Данные о маршрутизации, такие как адреса следующего перехода (Next Hop) и метрики маршрута.

- \*\*Применение:\*\* Помогает обеспечить правильную доставку пакетов и обнаружить нежелательные или вредоносные маршруты.

### 8. \*\*Политики безопасности\*\*

- \*\*Правила брандмауэра:\*\* Набор правил, определяющих, какой трафик разрешен или запрещен на основе вышеуказанных данных.

- \*\*Применение:\*\* Помогают фильтровать трафик и принимать решения о доступе.

### 9. \*\*Состояние сессии\*\*

- \*\*Состояние сессии:\*\* Информация о текущих сессиях, отслеживаемая состояниевом брандмауэром.

- \*\*Применение:\*\* Помогает отслеживать состояние соединений и принимать решения о фильтрации на основе контекста.

### 10. \*\*Метки безопасности\*\*

- \*\*Метки безопасности:\*\* Дополнительные метки, которые могут быть прикреплены к пакетам для указания уровня безопасности или категории.

- \*\*Применение:\*\* Помогают принимать решения о фильтрации трафика на основе политик безопасности.

### 11. \*\*Статистика трафика\*\*

- \*\*Статистика трафика:\*\* Данные о количестве пакетов, объеме трафика, частоте запросов и других метриках.

- \*\*Применение:\*\* Помогает анализировать поведение сети и обнаруживать аномальную активность.

### 12. \*\*Информация о фрагментации\*\*

- \*\*Информация о фрагментации:\*\* Данные о фрагментации пакетов, включая идентификатор пакета и смещение фрагмента.

- \*\*Применение:\*\* Помогает управлять фрагментированным трафиком и обнаруживать попытки фрагментации для обхода фильтров.

### 13. \*\*Информация о TTL (Time to Live)\*\*

- \*\*TTL:\*\* Значение, указывающее максимальное количество переходов, которые может пройти пакет до истечения срока его жизни.

- \*\*Применение:\*\* Помогает предотвратить бесконечный цикл маршрутизации и обнаружить подозрительную активность.

### 14. \*\*Информация о TOS (Type of Service)\*\*

- \*\*TOS:\*\* Поле, указывающее на тип обслуживания, который должен быть предоставлен пакету.

- \*\*Применение:\*\* Помогает управлять приоритетом трафика и обеспечивать качество обслуживания (QoS).

### 15. \*\*Информация о DSCP (Differentiated Services Code Point)\*\*

- \*\*DSCP:\*\* Значение, используемое для классификации трафика и управления приоритетом.

- \*\*Применение:\*\* Помогает управлять приоритетом трафика и обеспечивать качество обслуживания (QoS).

Используя эти данные, межсетевые экраны и другие устройства на третьем уровне могут эффективно управлять трафиком, обеспечивать безопасность и оптимизировать работу сети.

Алгоритм обмена ключами Диффи-Хеллмана (Diffie-Hellman Key Exchange) — это криптографический протокол, который позволяет двум сторонам безопасно обменяться секретным ключом по незащищенному каналу связи. Этот ключ затем может использоваться для шифрования дальнейшего обмена данными. Вот пошаговое описание алгоритма:

Веб-приложения, доступные из интернета, являются привлекательной мишенью для злоумышленников. Для атак на веб-приложения злоумышленники используют различные техники и векторы атак. Необходимым звеном защиты веб-приложения являются межсетевые экраны прикладного уровня (Web Application Firewall), т.к. только этот тип средств защиты позволяет обнаружить и предотвратить некоторые типы атак, например, атаки на уровне бизнес-логики.

Средства защиты web-приложений (WAF) реализуют комплекс мер защиты, направленных на обеспечение доступности работы web-приложений за счет защиты от различных классов атак.

К основным мерам и способам защиты web-приложений, которые используют WAF относятся:

проверка данных на соответствие стандартам протоколов;

самообучаемые механизмы проверки трафика, в т.ч. на основе нейронных сетей;

сигнатурный анализ;

защита от инъекций и XSS;

защита от DDOS-атак;

репутационный анализ;

пользовательские правила обработки трафика.

В результате проведения работ по проектированию и внедрению программно технических средств системы будут получены следующие результаты:

снижена вероятность возникновения инцидентов информационной безопасности, связанных с попытками несанкционированного доступа к информационным ресурсам, эксплуатацией уязвимостей системного и прикладного программного обеспечения Web приложений;

обеспечено выполнение требования PCI DSS 6.6 в части защиты Web-ориентированных приложений.

Использование программно-технических средств системы защиты Web приложений позволит своевременно обнаруживать и предотвращать попытки несанкционированных действий злоумышленников (как внешних, так и внутренних) в отношении информационных ресурсов.

### Шаги алгоритма Диффи-Хеллмана

#### 1. \*\*Генерация параметров\*\*

- \*\*Выбор большого простого числа \( p \)\*\* — модуль, по которому будут производиться вычисления.

- \*\*Выбор примитивного элемента \( g \)\*\* — число, которое является генератором мультипликативной группы по модулю \( p \).

Эти параметры \( p \) и \( g \) являются общими для обеих сторон и могут быть известны всем.

#### 2. \*\*Генерация секретных чисел\*\*

- \*\*Сторона А\*\* выбирает случайное целое число \( a \) (секретное число), где \( 1 < a < p-1 \).

- \*\*Сторона Б\*\* выбирает случайное целое число \( b \) (секретное число), где \( 1 < b < p-1 \).

#### 3. \*\*Вычисление открытых ключей\*\*

- \*\*Сторона А\*\* вычисляет открытый ключ \( A \):

\[

A = g^a \mod p

\]

- \*\*Сторона Б\*\* вычисляет открытый ключ \( B \):

\[

B = g^b \mod p

\]

#### 4. \*\*Обмен открытыми ключами\*\*

- \*\*Сторона А\*\* отправляет \( A \) стороне Б.

- \*\*Сторона Б\*\* отправляет \( B \) стороне А.

Открытые ключи \( A \) и \( B \) передаются по незащищенному каналу связи.

Таким образом, обе стороны получают один и тот же секретный ключ \( K \), который может использоваться для симметричного шифрования дальнейшего обмена данными.

Создание единой защищенной виртуальной частной сети (VPN), а также организация защищенного удаленного доступа (RA VPN) к информационным ресурсам является одним ключевых вопросов информационной безопасности организации.

«Информзащита» предлагает комплексный подход к решению данной проблемы. В рамках выполнения работ по проектированию системы защиты каналов связи экспертами «Информзащиты» разрабатывается техническое решение, которое позволит обспечить:

криптографическую защиту каналов связи, выходящих за пределы контролируемой зоны;

безопасность и целостность информации передаваемой по открытым и недоверенным каналам связи;

аутентификацию и идентификацию удаленных пользователей;

защищенный доступ удаленных пользователей к информационным ресурсам организации;

проверку соответствия удаленной рабочей станции требованиям информационной безопасности.

1. \*\*Генерация параметров:\*\*

- \( p = 23 \)

- \( g = 5 \)

2. \*\*Генерация секретных чисел:\*\*

- Сторона А выбирает \( a = 6 \)

- Сторона Б выбирает \( b = 15 \)

3. \*\*Вычисление открытых ключей:\*\*

- Сторона А: \( A = 5^6 \mod 23 = 8 \)

- Сторона Б: \( B = 5^{15} \mod 23 = 19 \)

4. \*\*Обмен открытыми ключами:\*\*

- Сторона А отправляет \( A = 8 \) стороне Б.

- Сторона Б отправляет \( B = 19 \) стороне А.

5. \*\*Вычисление общего секретного ключа:\*\*

- Сторона А: \( K = 19^6 \mod 23 = 2 \)

- Сторона Б: \( K = 8^{15} \mod 23 = 2 \)

Общий секретный ключ \( K = 2 \).

Тема защиты доступа к ресурсам сети Интернет является на сегодняшний день как никогда актуальной. Ресурсы сети Интернет представляют собой мишень для различного рода атак из-за своей доступности и ценной информации, которую они обрабатывают. Для предотвращения сложных и распространенных атак организации нуждаются в защищённом доступе к Интернет-ресурсам. Безопасность данного доступа реализуется за счет применения средств мониторинга и контроля доступа к сети Интернет, защиты от вредоносного кода, фишинга, внешних сетевых атак и т.д.

Эффективная web-защита включает в себя:

контроль доступа пользователей корпоративной сети к сети Интернет;

настройку политик и ограничение доступа пользователей корпоративной сети к сети Интернет;

мониторинг трафика в точках подключения сегмента корпоративной сети к сетям общего пользования на предмет обнаружения вредоносного программного обеспечения;

сбор статистических данных об обращении пользователей к ресурсам Интернет, а также о результатах сканирования трафика антивирусным ПО с возможностью создания отчетов для проведения расследования инцидентов информационной безопасности.

Если раньше выбор решений, позволяющих обеспечить качественную защиту доступа пользователей корпоративной сети к сети Интернет, был довольно небольшой, то сейчас на отечественном рынке представлено довольно много продуктов как западных производителей, так и российских. Для правильного выбора средств защиты эксперты «Информзащиты» проведут удаленное обследование сети вашей организации, спроектируют систему контроля доступа пользователей в сеть Интернет с учетом особенностей ИТ-инфраструктуры вашей компании, далее осуществят внедрение оптимальных в вашем случае решений.

Внедрение решения по защите доступа к ресурсам сети Интернет позволит:

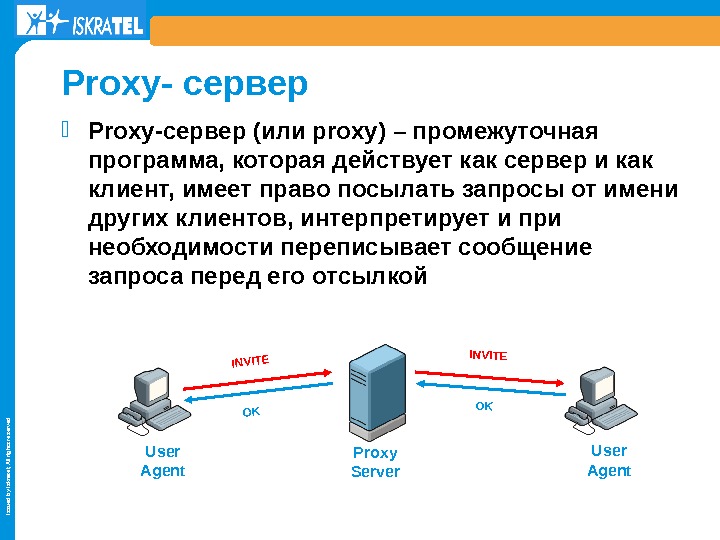
эффективно управлять информационными потоками при взаимодействии пользователей с ресурсами в сети Интернет;

повысить уровень защищенности корпоративной сети от акта из сети Интернет;

осуществлять мониторинг и анализ сетевого трафика при доступе пользователей к ресурсам Интернет.Прокси сервера



## Типы прокси-серверов и их особенности:



\*\*1. По назначению:\*\*

\* \*\*Web-прокси:\*\*

\* \*\*Особенности:\*\*

\* Обрабатывают только HTTP/HTTPS трафик.

\* Могут кэшировать веб-страницы, ускоряя загрузку.

\* Могут фильтровать контент (например, блокировать рекламу или нежелательные сайты).

\* \*\*Примеры:\*\* Squid, Privoxy, Nginx (с модулем проксирования).

\* \*\*Транспортные прокси:\*\*

\* \*\*Особенности:\*\*

\* Обрабатывают трафик различных протоколов (HTTP, FTP, SMTP и т.д.).

\* Могут использоваться для маршрутизации трафика, балансировки нагрузки, аутентификации пользователей.

\* \*\*Примеры:\*\* HAProxy, Nginx (с модулями проксирования различных протоколов).

\* \*\*SOCKS-прокси:\*\*

\* \*\*Особенности:\*\*

\* Универсальный тип прокси, работающий на транспортном уровне.

\* Поддерживает широкий спектр протоколов (HTTP, FTP, SMTP, P2P и т.д.).

\* Может использоваться для обхода блокировок, анонимизации трафика.

\* \*\*Примеры:\*\* Dante, 3proxy.

\*\*2. По способу работы:\*\*

\* \*\*Прямой прокси:\*\*

\* \*\*Особенности:\*\*

\* Клиент явно настроен на использование прокси-сервера.

\* Прокси-сервер знает IP-адрес клиента.

\* Может использоваться для фильтрации трафика, кэширования, анонимизации.

\* \*\*Примеры:\*\* Web-прокси, транспортные прокси.

\* \*\*Обратный прокси:\*\*

\* \*\*Особенности:\*\*

\* Клиент не знает о существовании прокси-сервера.

\* Прокси-сервер знает IP-адрес сервера, на который направляется запрос.

\* Может использоваться для балансировки нагрузки, защиты серверов от атак, кэширования.

\* \*\*Примеры:\*\* Nginx, HAProxy, Varnish.

\*\*3. По уровню анонимности:\*\*

\* \*\*Прозрачный прокси:\*\*

\* \*\*Особенности:\*\*

\* Клиент не знает о существовании прокси-сервера.

\* Прокси-сервер знает IP-адрес клиента и сервера.

\* Может использоваться для фильтрации трафика, кэширования.

\* \*\*Примеры:\*\* Web-прокси, установленные на уровне сети (например, в роутере).

\* \*\*Анонимный прокси:\*\*

\* \*\*Особенности:\*\*

\* Клиент не знает о существовании прокси-сервера.

\* Прокси-сервер знает IP-адрес клиента, но сервер не знает IP-адрес клиента.

\* Может использоваться для анонимизации трафика.

\* \*\*Примеры:\*\* Web-прокси, SOCKS-прокси.

\* \*\*Элитный прокси:\*\*

\* \*\*Особенности:\*\*

\* Клиент не знает о существовании прокси-сервера.

\* Прокси-сервер и сервер не знают IP-адрес клиента.

\* Обеспечивает максимальный уровень анонимности.

\* \*\*Примеры:\*\* Web-прокси, SOCKS-прокси.

\*\*4. По способу реализации:\*\*

\* \*\*Программные прокси:\*\*

\* \*\*Особенности:\*\*

\* Реализованы в виде программного обеспечения, устанавливаемого на сервере.

\* Могут настраиваться под конкретные задачи.

\* \*\*Примеры:\*\* Squid, Nginx, HAProxy, Dante.

\* \*\*Аппаратные прокси:\*\*

\* \*\*Особенности:\*\*

\* Реализованы в виде специализированного оборудования.

\* Обычно обеспечивают более высокую производительность и безопасность.

\* \*\*Примеры:\*\* Брандмауэры, UTM-системы.

\*\*Выбор типа прокси-сервера зависит от конкретных задач:\*\*

\* \*\*Для фильтрации трафика:\*\* Web-прокси, транспортные прокси, прозрачные прокси.

\* \*\*Для анонимизации трафика:\*\* SOCKS-прокси, анонимные прокси, элитные прокси.

\* \*\*Для балансировки нагрузки:\*\* Обратные прокси.

\* \*\*Для кэширования:\*\* Web-прокси, обратные прокси.

\*\*Важно учитывать:\*\*

\* \*\*Производительность:\*\* Прокси-серверы могут стать узким местом в сети.

\* \*\*Безопасность:\*\* Необходимо настроить правила фильтрации трафика и защиту от атак.

\* \*\*Масштабируемость:\*\* Прокси-серверы должны поддерживать возможность масштабирования.

1. \*\*Авторизация команд\*\*:

- Определяет, какие команды могут быть выполнены пользователем.

- Пример:

```

aaa authorization exec DEFAULT group tacacs+ local

```

2. \*\*Авторизация сессий\*\*:

- Определяет, какие типы сессий (например, Telnet, SSH) могут быть инициированы пользователем.

- Пример:

```

aaa authorization network DEFAULT group radius

```

---

#### \*\*6. Методы учета в AAA\*\*

Учет позволяет собирать информацию о действиях пользователей и отправлять ее на сервер.

1. \*\*Учет сессий\*\*:

- Собирает информацию о времени подключения, использованных ресурсах и т.д.

- Пример:

```

aaa accounting exec DEFAULT start-stop group tacacs+

```

2. \*\*Учет команд\*\*:

- Собирает информацию о выполненных командах.

- Пример:

```

aaa accounting commands 15 DEFAULT start-stop group radius

```

---

#### \*\*7. Пример полной настройки AAA\*\*

Пример настройки AAA с использованием локальной базы пользователей и RADIUS-сервера:

```

aaa new-model

! Настройка аутентификации

aaa authentication login DEFAULT group radius local

aaa authentication enable DEFAULT group radius local

! Настройка авторизации

aaa authorization exec DEFAULT group radius local

aaa authorization network DEFAULT group radius

! Настройка учета

aaa accounting exec DEFAULT start-stop group radius

aaa accounting network DEFAULT start-stop group radius

! Добавление RADIUS-сервера

radius-server host 192.168.1.100 key SECRET

radius-server timeout 5

radius-server retransmit 3

! Добавление локальных пользователей

username admin privilege 15 secret cisco123

```

---

#### \*\*8. Типичные ошибки при настройке AAA\*\*

1. \*\*Отсутствие резервного метода аутентификации\*\*:

- Если основной метод (например, RADIUS) недоступен, пользователи не смогут подключиться.

- Решение: всегда добавляйте локальную базу пользователей как резервный метод.

2. \*\*Неправильная настройка серверов RADIUS/TACACS+\*\*:

- Неверный IP-адрес или ключ может привести к сбоям аутентификации.

- Решение: проверьте настройки серверов и используйте команды для тестирования подключения.

3. \*\*Отсутствие учета\*\*:

- Без учета невозможно отследить действия пользователей.

- Решение: включите учет для всех типов сессий и команд.

---

#### \*\*9. Сравнение RADIUS и TACACS+\*\*

| \*\*Параметр\*\* | \*\*RADIUS\*\* | \*\*TACACS+\*\* |

|----------------------|-------------------------------------|-------------------------------------|

| \*\*Протокол\*\* | UDP (порт 1812 для аутентификации) | TCP (порт 49) |

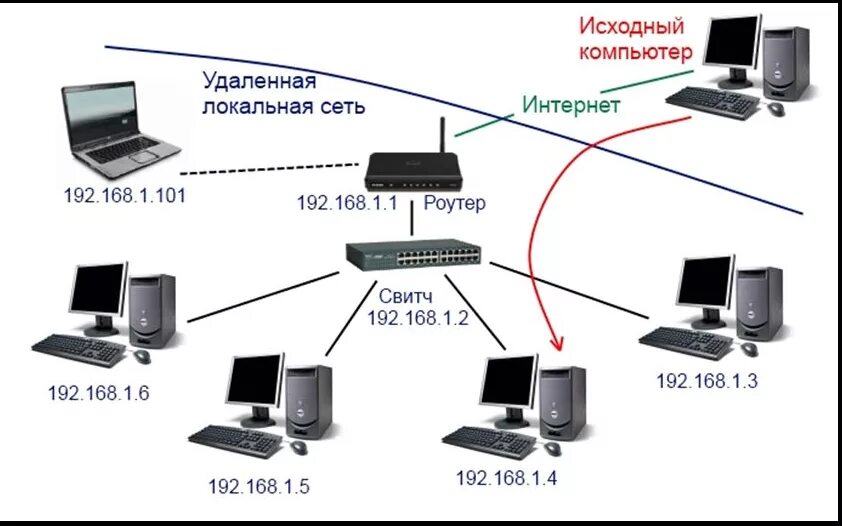
| \*\*Шифрование\*\* | Только пароли шифруются | Полный обмен данными шифруется |

| \*\*Функциональность\*\* | Аутентификация и учет | Аутентификация, авторизация, учет |

| \*\*Скорость\*\* | Быстрый | Медленнее из-за использования TCP |

| \*\*Применение\*\* | Сети общего назначения | Корпоративные сети, управление доступом к командам |

Техника IdleScan



Для понимания процесса скрытого сканирования вам не надо быть экспертом в области компьютерных технологий. Вам надо учесть следующее:

Большинство сетевых серверов прослушивают TCP порты, например, 80 - веб-сервер, 25 - почтовый. Порт считается открытым, если какой-либо процесс "висит" на нем и "слушает" этот порт, в противном случае порт считается закрытым.

Один из способов определить, открыт порт или нет, это послать "SYN" пакет (запрос на установление соединения) на этот порт. Целевая машина ответит на это пакетом "SYN|ACK" (уведомление о принятом запросе), если порт открыт, или пакетом "RST" (сброс), если порт закрыт.

Компьютер, получивший "ничей" пакет SYN|ACK ответит на него пакетом RST. "Ничьи" RST пакеты игнорируются.

Каждый IP пакет имеет свой "идентификатор фрагмента" IPID. Большинство операционных систем просто увеличивают это значение для каждого следующего пакета, посылаемого в сеть. Таким образом, проверка номеров пакетов может дать информацию, сколько пакетов было послано с предыдущего момента.

Осознав эти особенности становиться возможным сканировать сеть, прикрываясь чужим адресом. Эта техника описана на рисунке ниже. Атакующий A сканирует цель Target прикрываясь хостом-зомби Z. Прямоугольники представляют компьютеры, стрелки - пакеты.

Idlescan technique diagram

Как видно, "жертва" по разному отвечает хосту-зомби на якобы пришедшие от него пакеты. Если проверяемый порт открыт, "жертва" посылает пакет SYN|ACK хосту-зомби. Хост-зомби не ожидает никаких пакетов от "жертвы" и отвечает RST пакетом (сброс). Отвечая пакетом RST, хост-зомби увеличивает значение номера IPID (номер фрагмента). Настоящий атакующий определяет это в 3-м пункте рисунка. Если порт закрыт, "жертва" посылает хосту-зомби пакет RST, на который хост-зомби не отвечает (зачем отвечать, если вас послали? ;)), при этом IPID хоста-зомби не меняется. Атакующий проверяет на шаге 3 состояние счетчика IPID: если он увеличился на 2 (пакет RST и проверочный пакет) - порт открыт. Если на 1 - порт закрыт.

Преимущества IdleScan

Техника IdleScan предоставляет атакующему массу преимуществ по сравнению с классическими медодами сканирования, такими, как SYN или FIN-сканирование. Вот почему мы настоятельно рекомендуем обратить внимание на главу "Меры по защите от IdleScan". Ниже мы приводим причины, по которым хакеры могут воспользоваться этим методом:

Абсолютная невидимость сканирования. Речь идет не о незаметности сканирования для систем IDS, а о сокрытии атакующим своего реального IP адреса. Существуют некоторые методы, позволяющие атакующему "укрыться" от идентификации своего IP в процессе сканирования. Это метод "ложных хостов" (runmap -D) или полу-открытое сканирование (runmap -sS). Но даже эти методы требуют, чтобы атакующий отправил жертве несколько пакетов со своим реальным IP-адресов. IdleScan лишен этого недостатка: НИ ОДНОГО ПАКЕТА от атакующего не приходит жертве!

Обман файрволлов и пакетных фильтров. Фильтрация пакетов по признаку адреса источника - это широко распространенный метод защиты, используемый для ограничения числа машин, способных подключиться к защищаемому хосту. Например, сервер баз данных компании может разрешать подключения только от www-сервера. Домашний пользователь может разрешить только ssh-канал к своему рабочему месту. Более тяжелые последствия могут возникнуть, например, когда в компании bigwig руководство прикажет сетевым администраторам открыть дыру в файрволле для доступа к ресурсам внутренней сети с домашних IP-адресов (если они не хотят использовать VPN).

Выявление доверительных отношений на базе IP-адресов. Так, например, сканирование упомянутого выше сервера баз данных при помощи зомби-рабочего места может не дать результата. Однако если в качестве зомби выбран веб-сервер, мы получим открытый порт. Испльзуя эту информацию, атакующий может расставлять приоритеты в выборе целей для дальнейшей атаки.