Hacker Highschool

SECURITY AWARENESS FOR TEENS



Урок 3
Интернет изнутри

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM

WWW.ISECOM.ORG - WWW.OSSTMM.ORG - WWW.HACKERHIGHSCHOOL.ORG - WWW.BADPEOPLEPROJECT.ORG - WWW.OSSTMMTRAINING.ORG

Внимание

Проект Hacker Highschool является средством обучения и, как в любом обучающем средстве, существует опасность. Некоторые уроки, если ими злоупотреблять, могут привести к физической травме. Также дополнительные опасности могут быть там, где еще недостаточно исследований о возможных последствиях излучений от специфической техники. Студенты, использующие эти уроки находится под контролем преподавателя, и, в тоже время, должны быть мотивированы на изучение материалов и непрерывную практику. ISECOM не несет ответственности за применение информации, полученной из данных материалов и за дальнейшие последствия.

Все представленные здесь материалы являются открытыми и общедоступными в соответствии с положениями и условиями организации ISECOM:

Все материалы проекта Hacker Highschool предназначены для некоммерческого использования в работе с учениками средних государственных или частных школ, техникумов, студентами высших учебных заведений, слушателями младших курсов Hacker Highschool и учащимися на дому. Эти материалы в любой форме не могут быть использованы для продажи. Обучение по этим материалом в обучающей организации, техникумах, университетах, профессионально-технических заведениях, летних или компьютерных лагерях и других организациях, в которых взимается плата за обучение, категорически запрещено без приобретения лицензии. Для более подробного ознакомления с условиями использования либо приобретения лицензии для коммерческого использования материалов, посетите раздел сайта предназначенный для Лицензирования http://www.hackerhighschool.org/licensing.html.

Проект HHS является результатом труда открытого сообщества и, если вы находите наши труды ценными и полезными, мы просим Вас поддержать нас путем приобретения лицензии, пожертвований, либо спонсорства.

Содержание

Сотрудники журнала

Marta Barceló, ISECOM
Pete Herzog, ISECOM
Glenn Norman, ISECOM
Chuck Truett, ISECOM
Bob Monroe, ISECOM
Kim Truett, ISECOM
Gary Axten, ISECOM
Marco Ivaldi, ISECOM
Simone Onofri, ISECOM
Greg Playle, ISECOM
Tom Thomas, ISECOM
Mario Platt
Ryan Oberto, Johannesburg South Africa

Переводчики

Olena Boiko, Kharkiv National University of Radio Electronics
Vadim Chakryan, Kharkiv National University of Radio Electronics

ISECOM

Введение и цели

В далёком прошлом, до появления Интернета, электронная коммуникация была похожа на шаманство. У каждой фирмы по производству компьютеров было своё представление о том, как машины должны взаимодействовать. И никто даже не рассматривал возможность того, что компьютер Wang мог бы взаимодействовать с компьютером Burroughs.

Мир изменился, когда учёные и студенты осознали удобство использования терминалов для доступа к мэйнфреймам. Появился знаменитый ПК IBM, и пользователи захотели получать доступ к мэйнфрейму со своих персональных компьютеров. Вскоре с помощью модемов осуществлялись dial-up соединения, а пользователи смогли работать с программами эмуляции терминала. Работу в сети многие стали воспринимать как чёрную магию, а людей, которые разбирались в новых технологиях, называли не иначе как **гуру**.

Мир вновь значительно изменился, когда Интернет, начинавшийся как военный проект, стал общедоступным. Работа в сети всегда была локальной, то есть ограничена в пределах одного офиса или университета. Как же собирались взаимодействовать всё эти множества разных систем?

Ответом стало введение системы универсальной адресации для существующих сетей. В целом эта система называется **межсетевым протоколом (IP, Internet Protocol)**. Представьте, что ваш друг отправляет вам посылку из-за границы. Эта посылка может перевозиться на самолёте, поезде или автомобиле, но вам на самом деле не нужно знать расписание авиаперелётов или местоположение ближайшей железнодорожной станции. Ваша посылка в итоге прибудет по вашему адресу, что в конечном счёте является единственной значимой информацией. Ваш **IP-адрес** во многом выполняет ту же функцию: пакеты могут перемещаться как электроны, пучки света или радиоволны, но эти системы перемещения не важны. Важен ваш IP-адрес и IP-адрес системы, с которой вы взаимодействуете.

Одна проблема, которая усложняет эту идею в реальных условиях, — это то, что по одному адресу может жить несколько человек. В мире сетей аналогичная проблема возникает, когда один сервер поддерживает как обычный HTTP и безопасный HTTPS, так и FTP. Заметили букву «P» в конце этих аббревиатур? Это всегда является обозначением **протокола,** который можно охарактеризовать как «тип коммуникации».

Этот урок поможет вам разобраться в том, как работают протоколы и порты в Windows, Linux и OSX. Вы также ознакомитесь с некоторыми утилитами (некоторые из них уже были рассмотрены в предыдущем уроке), которые анализируют возможности вашей системы по организации сети.

К концу этого урока у вас будут базовые представления о следующих понятиях:

- принципы построения сетей и как происходит взаимодействие
- IP-адреса
- порты и протоколы

Базовые понятия сетевого взаимодействия

Начнём с рассмотрения локальных сетей (**LAN, Local Area Network**). LAN позволяет компьютерам, находящимся относительно близко друг от друга, совместно использовать ресурсы (например, принтеры, дисковое пространство), а **администраторы** управляют таким доступом. Далее описаны общие сетевые устройства и топологии.

Устройства

Продолжая карьеру хакера, вы встретите большое количество диаграмм сети. Полезно знать наиболее часто встречающиеся обозначения:

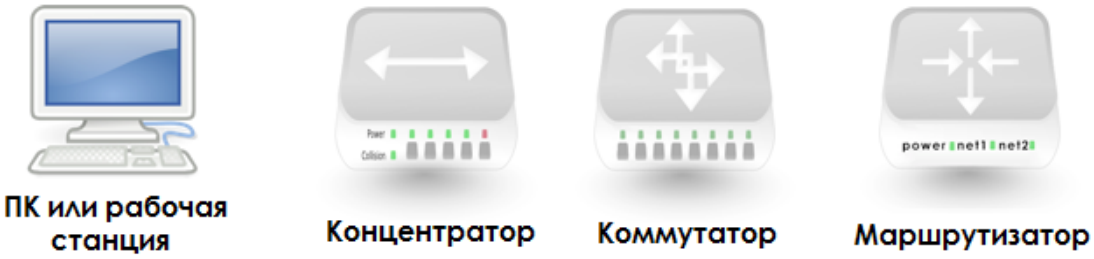

Рисунок 3.1: Часто встречающиеся обозначения на диаграммах сетей

Хаб (концентратор) похож на старомодную общую телефонную линию: все используют один и тот же провод и могут слушать разговоры всех остальных. Из-за этого LAN зашумляется.

Концентратор в этом плане лучше: он фильтрует трафик так, что только два взаимодействующих между собой компьютера могут «слышать разговор». Как и концентратор, он используется только в LAN.

Маршрутизатор размещают между несколькими LAN; он используется для доступа к другим сетям и к Интернету. Маршрутизатор использует IP-адреса. Он анализирует отправленные пакеты и определяет, какой сети они предназначены. Если пакет принадлежит «другой» сети, он переправляет пакет в место назначения.

Топологии

Топология представляет собой способ соединения компьютеров. Решения, принятые относительно топологии сети, в будущем могут принести как положительные, так и отрицательные результаты в зависимости от: используемых технологий, технологических и физических ограничений, производительности и требований к безопасности, размера и типа организации и т.п.

Физическая структура LAN может выглядеть как одна из следующих физических топологий:

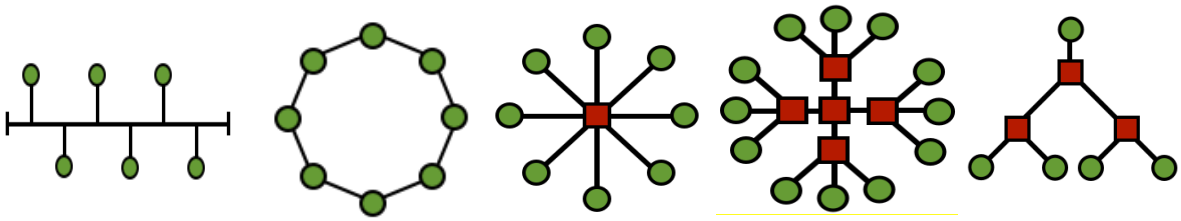

Рисунок 3.2 — Топологии

Шина: все компьютеры подключены к одному кабелю, и каждый может непосредственно взаимодействовать с любым другим компьютером. Но при повреждении какой-либо части шины работа сети будет прекращена.

Кольцо: каждый компьютер связан со следующим за ним, а последний — с первым. Каждый компьютер может непосредственно взаимодействовать только с двумя соседними компьютерами.

Топологии типа «шина» сейчас используются редко. Кольцо часто используется на межгосударственном уровне, обычно с двумя кольцами со встречным направлениям передачи данных, которые отправляют трафик в противоположных направлениях для обеспечения надёжности и отказоустойчивости.

Звезда: компьютеры непосредственно не соединены друг с другом. Они связаны через концентратор или коммутатор, который передает информацию от одного компьютера к другому.

При соединении друг с другом нескольких концентраторов или коммутаторов получается топология типа **расширенная звезда**.

В топологии типа «звезда» или «расширенная звезда» все центральные узлы являются **пирами**, то есть они по существу равноправны. На сегодняшний день это самая распространенная топология LAN.

Однако если соединить вместе две сети с топологией звезды или расширенной звезды через центральный узел, который управляет трафиком или ограничивает его между двумя сетями, то получится **иерархическая** топология сети. Она обычно используется на более крупных предприятиях.

Игра началась: оставляя лазейки открытыми

В летнюю жару Джейс была счастлива помочь местному полицейскому отделению, в котором был установлен кондиционер, провести небольшую сеть. Они платили ей печеньем, временем вдали от жары, разговорами и возможностями установить лазейки в системе. Ползая под стальными рабочими столами, которые десятилетиями не сдвигались с места, Джейс нашла самое грязное укромное место для того, чтобы спрятать Wi-Fi точку доступа. Джейс просто подключила её, засыпала сверху мусором и протянула Ethernet-кабель к стенным портам, которые она до этого установила.

Тяжелая рука хлопнула по столу над ней. Джейс ударилась о металлическую крышку и вскрикнула: «Ой! Моя голова!», — оправившись, она спросила, — «вы уверенны, что не хотите, чтобы я установила вам сервер?»

Полицейский откашлялся и попытался ответить голосом Тёмного Профессора:

«Что ж, возможно, и хотел бы, но я не уверен, что резистор лучевых потоков справится с взаимным влиянием микроканалов. Особенно когда полнолуние выпадает на последний вторник месяца.»

Джейс зашаркала ногами со свойственной подросткам насмешливой раздражительностью. «Видимо, у вас нет проблем с достижением квантовых уровней чепухи. А когда я получу своё печенье, офицер Кикэм?»

«Пожалуйста, Джейс, называй меня Хэнком. Я чувствую себя стариком, когда ты зовёшь меня офицер Кикэм». Он старался придать голосу страдальческий оттенок, но Джейс была знакома с приёмами социальной инженерии: на самом деле он пытался отвлечь её от печенья.

«Хэнк, не хотелось бы сообщать вам эту новость, но вы *на самом деле* старик.»

«Ох, звучит обидно. Я не старый, я авторитетный», — возразил он, рассматривая свои до блеска начищенные чёрные полицейские ботинки, а рваные кроссовки Джейс тем временем скрылись под тяжелым столом. Вскоре Хэнк вновь увидел её карие глаза и лицо, покрытое паутиной. У Джейс в руке всё ещё была катушка кабеля. Хэнк помог ей подняться и убрал паутину с её лица и плечей.

«Помогите, полицейский произвол», — поддразнила Джейс.

«Враждебный преступник», — ответил на это Хэнк. — «Так просветите меня по поводу Вашего адского плана», — Джейс показалось, что грубый и мускулистый представитель закона попросил её умоляющим тоном.

Она спросила: «Вы уверены, что хотите знать об устройстве всех этих сетевых штук?» Он нетерпеливо кивнул. Джейс подумала: *прямо как игрушка с качающейся головой*.

«Хорошо. Я сделала проект сетевой топологии. Это похоже на карту, на которой указано, где будет установлено всё оборудование, компьютеры, концентраторы, разъёмы, коммутаторы, маршрутизаторы и брандмауэры. Вы не можете начинать такой проект без карты», — сказала она, поглядывая на полицейского. «Это делается для гарантии того, что каждый узел сможет взаимодействовать с любым другим узлом, без единой точки отказа. К примеру, топология «шина» никуда не годится, потому что если один узел в шине будет повреждён, то вся сеть прекратит работу». Хэнк кивнул, и Джейс продолжила.

«Представьте, что работа в сети — это этот полицейский участок. К вам привели одного подозреваемого. Каждый коп ожидает своей очереди, чтобы допросить свидетеля, не отнимая при этом времени у своих коллег. Если жертву, в смысле подозреваемого, переводят в другую камеру, остальным копам, которые еще должны допросить его, нужно знать, куда его перевели.»

«Джейс, похоже, тебя тоже ждет пристрастный допрос, если ты будешь продолжать так говорить о нас, блюстителях порядка.» Хэнк подтянул свой оружейный ремень и втянул свой небольшой живот.

Джейс проглотила смешок. «Получается, что подозреваемый — это пакет данных, а вы, полицейские бандиты, — сетевые устройства. Каждому устройству, будь то коммутатор, маршрутизатор, брандмауэр, другой сервер или что-либо иное, необходимо знать, что пакет уже рассмотрен. Ну знаете, избит полицейскими дубинками. Я думаю, вы называете это помыть голову древесным шампунем.»

Хэнк закатил глаза и попытался нащупать дубинку, которой у него при себе не оказалось.

Хихикая, Джейс подняла катушку кабеля как щит. «Эй, у меня здесь катушка проводов и у меня хватит смелости использовать её. Поставьте на место чашку с

кофе и никто не пострадает.» Потеряв равновесие и засмеявшись, Джейс свалилась на Хэнка; он даже не пошевельнулся. *Ох, этот парень действительно крепкий*, подумала она. Его рука, которую он положил на её плечо, напомнила ей.. о чём-то.

Она поспешно встала, при этом покраснев. «Итак, есть два типа устройств. Умные и глупые. Как полицейские.» Четверо в полицейской форме появились в комнате в самый неподходящий момент, услышав «глупые, как полицейские». Джейс сбивчиво продолжила: «Умные устройства помнят всё, что они делают. Они ведут журнал своей деятельности.»

«А те, которые глупые, как полицейские?» — спросил начальник полиции.

Конец игры

Модель TCP/IP (DoD)

Модель TCP/IP была разработана Министерством обороны США **(DoD, Department of Defense)** и Агентством передовых оборонных исследовательских проектов **(DARPA, Defense Advanced Research Project Agency)** в 1970-х гг. Модель TCP/IP была спроектирована в качестве открытого стандарта, который каждый мог бы использовать для соединения компьютеров и обмена информацией между ними. В конечном счёте, она стала основой Интернета.

В целом, наиболее простая форма модели TCP/IP называется **моделью DoD.** С неё и начнем.

Рисунок 3.3 — Модель DoD

Уровни

Простая модель DoD определяет четыре полностью независимых уровня, на которые она разделят процесс коммуникации между двумя устройствами. Уровни, через которые проходит информация, следующие:

Прикладной уровень

Прикладной уровень — это именно то, о чем вы, возможно, подумали: это уровень, на котором работают такие приложения, как Firefox, Opera, почтовые клиенты, сайты социальных сетей, клиенты мгновенного обмена сообщениями и приложения для организации чатов. На самом деле, достаточно много приложений имеют доступ к Интернету: например, некоторые офисные приложения подключаются к онлайн коллекциям клипарта. Прикладной уровень создает полезную нагрузку (полезные пользовательский данные, без примеси управляющих заголовков), которую переносят все остальные уровни. Хорошей аналогией может послужить почтовая система. Приложение создаёт пакет, при этом «оборачивая» его в инструкции относительно того, как этот пакет должен быть использован. Затем пакет передается в отдел обработки корреспонденции — на транспортный уровень.

Транспортный уровень

Транспортный уровень устанавливает сетевые соединения, которые называются **сессиями.** В мире Интернет основным протоколом на транспортном уровне является **TCP (the Transmission Control Protocol, протокол управления передачей)**. TCP оборачивает сообщение в еще одну «оболочку» с указаниями о том, какое это сообщение по счету (например, первое из трех), какому приложению оно предназначено (на основе портов — каждый порт соответствует своему приложению) и как удостовериться в том, что сообщение доставлено без повреждений.

Допустим, вы хотите отправить письмо по электронной почте вашей маме. Письмо может быть маленьким или огромным, в любом случае, оно окажется слишком большим, чтобы отправить его по Интернету целиком. Вместо этого TCP разбивает это письмо на **сегменты** — последовательно пронумерованные маленькие порции данных с прикрепленным в конце кодом проверки ошибок. Если в процессе передачи пакет повреждается, TCP запрашивает повторную передачу. На принимающей стороне TCP собирает порции данных в правильном порядке, и ваша мама получает письмо на свой электронный почтовый ящик.

Но не забывайте, что TCP — не единственный протокол транспортного уровня: **UDP** также функционирует на этом уровне. Его особенностью является то, что он НЕ создаёт сессии. Он просто отправляет поток информации, при этом никогда не проверяет, была ли информация получена конечным узлом.

Уровень Интернет

Этот уровень добавляет информацию об адресах отправителя и получателя и о том, где начинается и заканчивается **пакет**. Аналогично работает служба доставки, которая отправляет посылки по правильному адресу. Этот уровень не заботится о том, все ли пакеты будут доставлены и не будут ли они повреждены; это работа транспортного уровня. Основным протоколом на этом уровне, соответственно, является протокол **IP (Internet Protocol, межсетевой протокол)**. Этот уровень, выбирает наиболее оптимальный маршрут для передачи информации на основе IP-адресов.

Уровень Сетевого доступа

Этот уровень представляет собой низкоуровневую физическую сеть, используемую для соединения с Интернетом. Если вы используете dial-up, то нам вас жаль, и вы

пользуетесь простым **PPP**-соединением. Если вы используете **DSL**, то вы, вероятно, пользуетесь **ATM** или **Metro Ethernet**. А если у вас кабельный Интернет, то вы используете физическую сеть **DOCSIS**. Вне зависимости от способа доступа в Интернет, TCP/IP успешно управляет соединением. Уровень сетевого доступа включает в себя Ethernet-кабель и **сетевую карту (network interface card, NIC)** или беспроводной адаптер и точку доступа. На этом уровне осуществляется обработка низкоуровневых данных (в виде бит — цифровых единиц и нулей), передаваемых от одного узла к другому.

Пища для ума: «Модель OSI»

Посмотрите раздел «Модель OSI» в конце этого урока.
Это альтернативный взгляд на моделирование сети.

Протоколы

Теперь у вас есть соединение с Интернетом. Кажется, что всё просто, но рассмотрим обычную ситуацию: вы что-то ищете в Интернете, в то время, как ваш любимый брат или сестра смотрят фильм онлайн. Почему же эти два потока трафика не перемешиваются? Как сети удается различать их?

Ответом на эти вопросы является использование **протоколов,** которые можно назвать языками, на которых разговаривают разные типы трафика. Веб-трафик использует один протокол, передача файлов — другой, а электронная почта — третий. Но как и всё в цифровом мире, протоколы на самом деле не используют имена на сетевом уровне; они используют IP-адреса и **номера портов.**

Протоколы прикладного уровня

FTP (*File Transfer Protocol, протокол передачи файлов)* используется для передачи файлов между двумя устройствами. Он использует один порт для доставки данных и еще один порт для отправки управляющих сигналов («Я получил файл! Спасибо!»). Обычно используются порты с номерами 20 и 21 (TCP).

HTTP (*Hyper-Text Transfer Protocol, протокол передачи гипертекста)* используется для передачи веб-страниц. Использует 80 TCP порт. **HTTPS** — это безопасный вариант HTTP, который шифрует сетевой трафик, обычно использует 443 TCP порт.

SMTP *(Simple Mail Transfer Protocol, простой протокол передачи почты)* — это протокол, по которому отправляется электронная почта. Использует 25 TCP порт.

DNS *(Domain Name Service, служба доменных имен)* — используется для преобразования доменных имен (например: ISECOM.org) в IP-адреса (например: 216.92.116.13). Использует 53 UDP порт.

Протоколы транспортного уровня

TCP и **UDP** — это два основных протокола, которые используются на транспортном уровне для передачи данных.

TCP (Transmission Control Protocol, протокол управления передачей) устанавливает логическое соединение (**сессию**) между двумя хостами в сети. Он устанавливает это соединение, используя процедуру трехэтапного рукопожатия.

1. Когда мой компьютер хочет соединиться с вашим, он отправляет пакет с флагом **SYN**, говоря тем самым: «Давай синхронизируем часы, чтобы можно было обмениваться трафиком с временными метками».

2. Ваш компьютер (если он собирается принять соединение) отвечает, отправляя пакет подтверждения синхронизации с флагами **SYN/ACK**.

3. Мой компьютер «скрепляет сделку», отправляя пакет с флагом **ACK**. Соединение установлено.

Но так происходит только в случае использования протокола TCP. В отличие от него, **UDP (User Datagram Protocol, протокол пользовательских дейтаграмм)** — это транспортный протокол, для которого даже не важно наличие соединения. Он просто передает данные даже не заботясь о нумерации фрагментов и о том, получил ли конечный узел передаваемую информацию или нет. Такая особенность делает UDP очень быстрым, поэтому его удобно использовать для голосового и видеотрафика, или для онлайн игр для которых потеря отдельных фрагментов информации несущественна.

Протоколы уровня Интернет

IP (Internet Protocol, межсетевой протокол) служит универсальным протоколом для коммуникации между любыми двумя компьютерами в любой сети в любое время. Можно провести аналогию с почтальоном, который доставляет почту; всё, что он должен сделать, — это донести пакеты по адресам их получателей.

Internet Control Message Protocol (ICMP, протокол межсетевых управляющих сообщений)

ICMP — это протокол, который используют сетевые устройства и администраторы сети для выявления неисправностей и поддержания работы сети. Он включает в себя утилиту **ping** (Packet InterNet Groper, пакетный «следопыт» Интернета) и похожие команды, которые тестируют сеть и сообщают об ошибках. Поскольку ping часто используется для проведения флуд атак на сетевые устройства, большинство систем ограничивают ICMP до одного отклика за секунду.

Порты и протоколы связаны следующим образом:

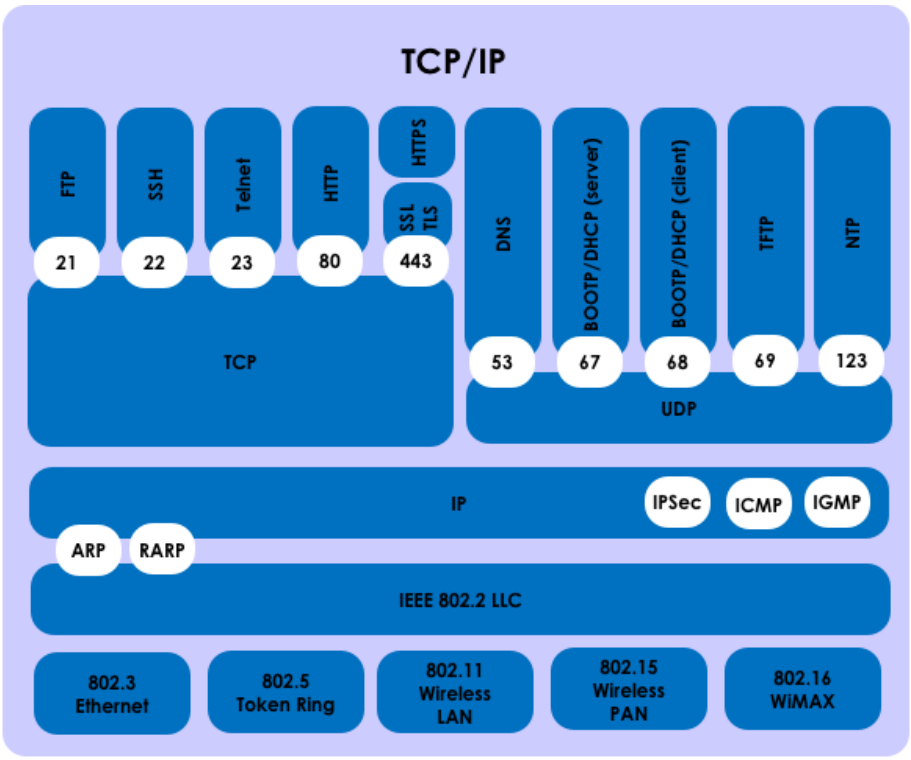

Рисунок 3.4 — Стек TCP/IP

IPv4-адреса

Доменные имена удобны для людей, поскольку мы легко запоминаем названия (к примеру, ISECOM.org). Но сети, в действительности, их не понимают; они понимают только числовые IP-адреса. Таким образом, когда вы запрашиваете ISECOM.org, ваш компьютер осуществляет быстрый поиск с помощью **DNS (Domain Name Service, служба доменных имён)**, чтобы найти соответствующий IP-адрес.

IP адреса похожи на почтовые адреса. Чтобы получать почту, вам нужен почтовый адрес. **IPv4**-адреса состоят из 32 бит, которые разбиты на четыре 8-битовые **октета,** разделённые точками. Это значит, что в Интернете существуют 2^{32} (или 4,294,967,296) уникальных IPv4-адресов. Одна часть IP-адреса идентифицирует сеть, а другая — идентифицирует отдельный компьютер в сети. Можно провести аналогию IP-адреса с почтовым адресом: страна/город — это адрес сети, название улицы — это хост.

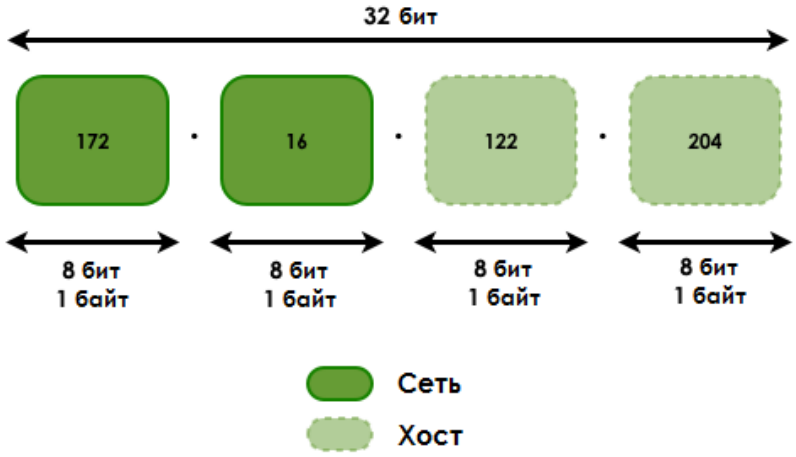

Рисунок 3.5 — Номер сети и ID хоста

Возвращаясь к аналогии с почтовой службой: IP — это грузовик, который доставляет пакет в нужное почтовое отделение. TCP — это внешний упаковщик, со списком того, сколько всего посылок в поставке и какая это именно посылка по счету. IP-адреса — это адрес конкретного дома (компьютера), для которого предназначен пакет.

> Различают **внешние** и **внутренние** IP-адреса. Внутренние IP-адреса используются частными сетями; подобные адреса не видны в Интернете, лишь внутри локальной сети.

IP-адреса в пределах одной внутренней сети не могут повторяться, но компьютеры в двух разных — но несвязанных — внутренних сетях могут иметь одинаковый IP-адрес. IP-адреса, определенные организацией IANA (the Internet Assigned Numbers Authority, Администрация адресного пространства Интернет) для внутренних сетей (в соответствии с RFC 1918), следующие:

с 10.0.0.0 по 10.255.255.255	(Класс A)
с 172.16.0.0 по 172.31.255.255	(Класс B)
с 192.168.0.0. по 192.168.255.255	(Класс C)

Классы

IP адреса разделены на классы в соответствии с тем, какая часть адреса используется для идентификации сети, а какая — для идентификации отдельных компьютеров.

В зависимости от размера каждой части либо в сети будет допустимо большее количество устройств, либо будет допустимо большее количество сетей. Существуют следующие классы:

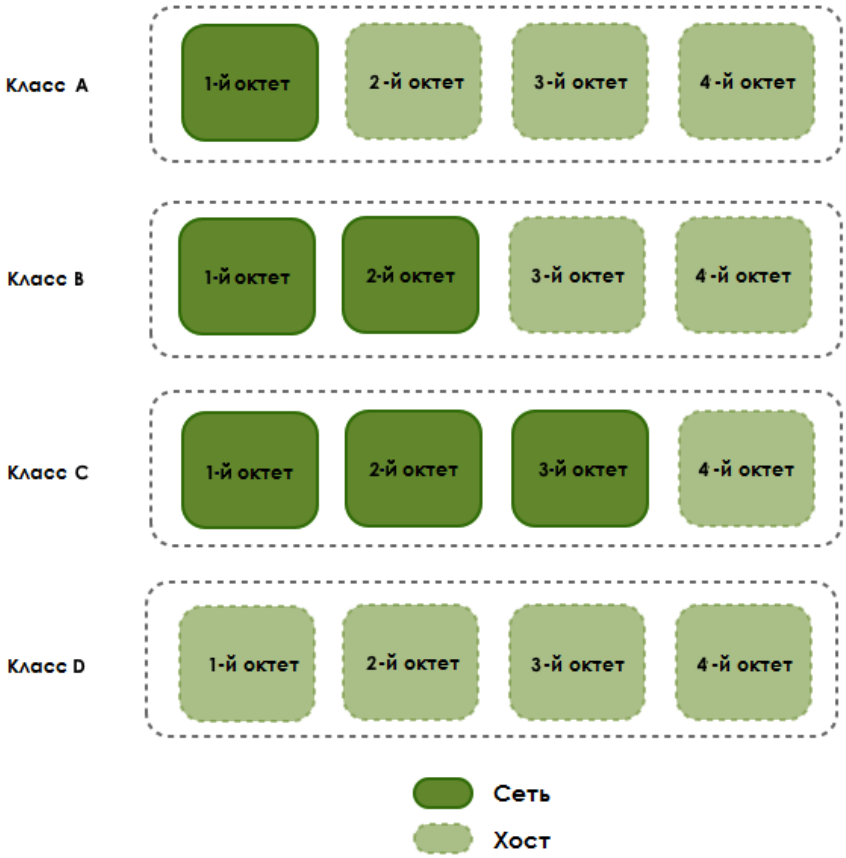

Рисунок 3.5: Разделения классов IP

Класс A: Первый бит всегда равен нулю, поэтому этот класс включает адреса от 0.0.0.0 (который, по соглашению, никогда не используется) до 126.255.255.255. Следует отметить, что адреса 127.x.x.x зарезервированы для loopback или localhost (см. далее).

Класс B: Первые два бита первого октета равны '10', поэтому этот класс включает в себя адреса с 128.0.0.0 по 191.255.255.255.

Класс C: Первые три бита первого октета равны '110', поэтому этот класс включает в себя диапазон адресов от 192.0.0.0 до 223.255.255.255.

Класс D: Первые четыре бита первого октета равны '1110', поэтому этот класс включает в себя диапазон адресов с 224.0.0.0 по 239.255.255.255. Эти адреса зарезервированы для реализации групповых рассылок (multicast addresses).

Оставшиеся адреса используются в экспериментальных целях либо зарезервированы для дальнейшего использования.

Маска *подсети* (или сетевая маска) используется для обозначения этих классовых разделений. В двоичном обозначении часть заполненная битом '1' – идентифицирует сеть, а часть заполненная битом '0' – идентифицирует отдельный хост. Стандартные сетевые маски для первых трёх классов следующие:

255.0.0.0 (Класс A)

255.255.0.0 (Класс B)

255.255.255.0 (Класс C)

На самом деле, такая система довольно хитрая, поскольку у сетей, которые используют стандартные классы, маска занимает один октет для класса A, два октета для класса B и три октета для класса C. Использовать стандартные классы удобно, но не все так поступают.

Это значит, что для идентификации хоста нужен и IP-адрес, и сетевая маска:

IP: 172.16.1.20
Маска: 255.255.255.0

Loopback-адреса

IP-адреса от 127.0.0.1 до 127.255.255.254 зарезервированы для использования в качестве **loopback** (досл. «обратная петля») или localhost-адресов, то есть они ссылаются непосредственно на локальный компьютер. Каждый компьютер имеет localhost-адрес со значением 127.0.0.1, поэтому этот адрес не может использоваться для идентификации других устройств. Он используется для проверки корректности работы YCP/IP стека на вашем устройстве.

Существует ряд других адресов, которые нельзя использовать. К ним относятся **адрес сети (network address)** и **широковещательный адрес (broadcast address)**.

Адреса сети

Адрес сети — состоит из сетевая части IP-адреса и **хост-части, заполненная нулями**. Этот адрес нельзя присвоить хосту, поскольку он идентифицирует всю сеть, а не просто один хост.

IP: 172.16.1.**0**
Маска: 255.255.255.**0**

Широковещательные адреса

Широковещательный адрес — состоит из сетевая части IP-адреса и **хост-части, заполненной единицами**. Этот адрес не может использоваться для идентификации хоста, поскольку это адрес, который прослушивают все хосты (в этом и состоит смысл широковещания: все слушают).

Порты

И TCP, и UDP использует **порты** для обмена информацией с приложениями. Порт — это расширение адреса, подобно добавлению номера квартиры к названию улицы. Письмо с названием улицы придет в правильное здание, но без номера квартиры оно не будет доставлено правильному адресату.

Порты работают аналогично. Пакет может быть доставлен по правильному IP адресу, но без соответствующего порта невозможно определить, какому приложению предназначен этот пакет. Номер порта — это также 16-битное число. Это значит, что он может принимать десятичные значения от 0 до 65535 (2 в 16-й степени).

Можно воспользоваться другой аналогией: каждый компьютер — это почтовое отделение. Каждое приложение имеет свой абонентский ящик; никакие два приложения не могут совместно использовать один и тот же абонентский ящик. Номер порта является тем самым номером абонентского ящика.

Использование номеров портов позволяет обрабатывать несколько потоков информации, которые поступают на один IP адрес; каждый из потоков отправляется соответствующему приложению. Номер порта позволяет сервису, который запущен на удаленном компьютере, узнать, какой тип информации запрашивает локальный клиент и какой протокол используется для отправки этой информации, при этом одновременно поддерживается связь с разными клиентами.

Например, если локальный компьютер пытается подключиться к веб-сайту www.osstmm.org, IP адрес которого — 62.80.122.203, причем веб-сервер использует порт с номером 80, то локальный компьютер подключится к удаленному компьютеру, используя **адрес сокета** (socket – сокет, означает указание IP-адреса с портом):

62.80.122.203:80

Для того чтобы поддерживать уровень стандартизации среди наиболее часто используемых портов, организация IANA утвердила, что порты с номерами от 0 до 1024 должны использоваться для общих, **привилегированных** или **общеизвестных приложений.** Оставшиеся порты – до 65535 – используются для динамических распределений или конкретных сервисов.

Наиболее часто используемые (общеизвестные) порты, назначенные **IANA,** представлены в таблице:

Назначения портов		
Номер	**Ключевые слова**	**Описание**
5	rje	Remote Job Entry, удаленный ввод заданий
0		Зарезервирован
1-4		Неназначенные
7	echo	Echo
9	discard	Discard, протокол отбрасывания

Назначения портов		
11	systat	Активные пользователи
13	daytime	Получение текущей даты и времени
15	netstat	Who is Up или NETSTAT
17	qotd	Quote of the Day, «цитата дня»
19	chargen	Character Generator, генератор символов
20	ftp-data	File Transfer (данные), протокол передачи файлов
21	ftp	File Transfer (управление), протокол передачи файлов
22	ssh	Протокол удаленного входа в систему SSH (Secure SHell – безопасная оболочка)
23	telnet	Telnet (TErminaL NETwork)
25	smtp	Simple Mail Transfer, простой протокол передачи почты
37	time	Time
39	rlp	Resource Location Protocol, протокол поиска ресурсов
42	nameserver	Host Name Server
43	nicname	Who Is (досл. «кто такой?»)
53	domain	Domain Name Server, протокол сервера имен
67	bootps	Bootstrap Protocol Server
68	bootpc	Bootstrap Protocol Client
69	tftp	Trivial File Transfer, простой протокол передачи файлов
70	gopher	Gopher
75		any private dial out service
77		any private RJE service
79	finger	Finger
80	www-http	World Wide Web HTTP
95	supdup	SUPDUP
101	hostname	NIC Host Name Server
102	iso-tsap	ISO-TSAP Class 0
110	pop3	Post Office Protocol - Version 3
113	auth	Authentication Service
117	uucp-path	UUCP Path Service
119	nntp	Network News Transfer Protocol
123	ntp	Network Time Protocol
137	netbios-ns	NETBIOS Name Service
138	netbios-dgm	NETBIOS Datagram Service
139	netbios-ssn	NETBIOS Session Service
140-159		Неназначенные

Назначения портов		
160-223		Зарезервированы

Инкапсуляция

Когда какая-то часть информации – электронное письмо, к примеру, – передается от одного компьютера к другому, то в процессе передачи над ней производятся определенные изменения. Уровень приложений генерирует данные для передачи, которые затем поступают на транспортный уровень. В свою очередь, транспортный уровень разбивает информацию на небольшие «сегменты» и добавляет к каждому из них заголовок, в котором хранится информация о том какому приложению адресованы данные и какому приложению их нужно вернуть при ответе (порты), а также множество других опций для контроля потока.

Затем сегмент передается на Сетевой уровень, где к ним прибавляются заголовки сетевого уровня – таким образом формируются «пакеты». В заголовке сетевого уровня содержатся IP-адреса назначения и отправителя для того, чтобы сетевые устройства понимали, куда им нужно передавать данные и как их вернуть обратно.

После сетевого уровня информация передается на Канальный уровень, где добавляется новый заголовок, в котором содержится информация, необходимая для адресации внутри локальной сети (MAC-адреса) и контроля целостности данных (контрольная сумма – CRC32). На данном уровне из пакетов формируются «кадры».

Процесс подобного добавления заголовков на различных уровнях и называется — **инкапсуляция**. Каждый нижележащий уровень добавляет свой кусочек информации в виде заголовка к информации, полученной от вышележащих уровней. Этот процесс продолжается до тех пор, пока информация не дойдет до самого нижнего уровня — физического, где данные уже непосредственно преобразовываются в сигналы и помещаются в линию связи.

Когда информация поступает на приемную сторону, то происходит обратный процесс «деинкапсуляции». В ходе данного процесса информация поднимается по уровням вверх и каждый из них анализирует и убирает заголовок, за который он отвечает.

Процесс инкапсуляции приведен на рис. 3.6, как видно процесс начинается сверху и идет вниз. На самом нижнем уровне мы получим кадр «облепленный» управляющими заголовками. Именно поэтому в сетях принято различать полезную пропускную способность и общую пропускную способность. Полезной считается пропускная способность, которая передает лишь полезную нагрузку, то есть лишь пользовательские данные, без примеси управляющих заголовков.

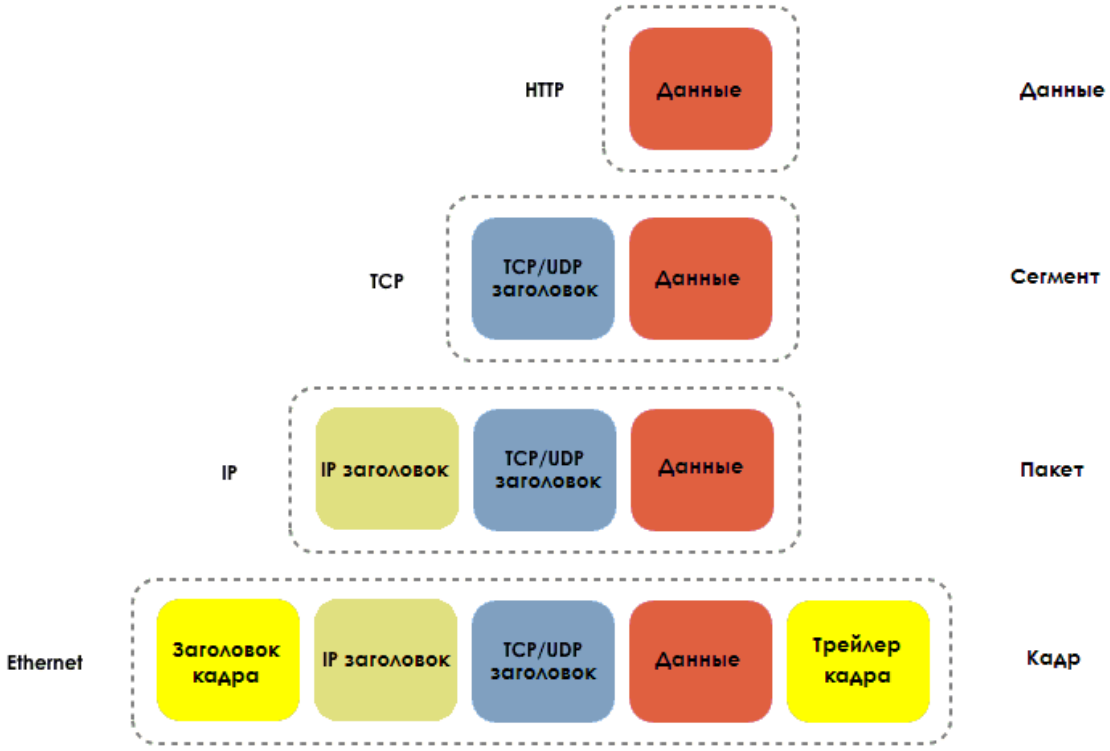

Рисунок 3.6 — Процесс инкапсуляции

Для адресации в локальной сети используются MAC-адреса (англ. **"MAC - Media Access Controller"**). Они добавляются в заголовке канального уровня, который находится ближе всего к физическому уровню (по этой же причине процесс обработки кадра намного проще процесса обработки пакета). При помощи MAC адресов, коммутаторы понимают в какой порт им нужно передать информацию.

Обычно MAC-адреса представляются в виде 6 пар шестнадцатеричных значений, отделенных друг-от-друга дефисами:

00-15-00-06-E6-BF

MAC-адреса являются физическими адресами сетевой карты и они не могут быть изменены (конечно, существуют способы, при помощи которых можно подменить MAC-адрес, однако про них нужно говорить отдельно).

Упражнения

3.1. Используя команды изученные в урока 1 и 2 узнайте IP-адрес вашего компьютера, сетевую маску, DNS-имя хоста и MAC-адрес. Сравните информацию полученную Вами, с информацией, полученной Вашим напарником за соседним компьютером. В чем схожесть и в чем разница? Какие IP-адреса используются в сети: публичные или приватные?

3.2. netstat

Команда **netstat** отображает сетевую статистику: с кем Вы соединены, как долго работает сеть и т.п. Чтобы запустить утилиту в Linux, Windows либо OSX нужно зайти в консоль операционной системы и набрать:

```
netstat
```

В консоли Вы увидите список установленных соединений. Если вы хотите, чтобы соединения отображали адреса и номера портов в числовом формате, наберите:

```
netstat -n
```

Для того, чтобы увидеть список всех соединения и ожидающих портов, наберите:

```
netstat -an
```

Чтобы вызвать справку по команду, наберите:

```
netstat -h
```

В списке netstat, обратите внимание на колонки, в которых приведены и локальный и удаленный IP-адреса и посмотрите какие порты используются данные подключения:

```
Proto Recv-Q Send-Q  Local Address        Foreign Address     (state)
tcp4     0      0    192.168.2.136:1043   66.220.149.94:443 ESTABLISHED
```

Порты это цифры, которые приведены после IP-адреса, отделенные двоеточием. Почему порты используемые удаленными адресами отличаются от портов, которые используют локальные адреса?

Откройте несколько окон в браузере и в каждом из них откройте разные сайты. Снова используйте утилиту netstat.

Когда открыты несколько вкладок в одном браузере, как он может понять какой из них передавать полученную информацию?

Почему происходит так, что когда используется браузер, то исчезают порты, находящиеся в процессе прослушивания?

Какие протоколы используются?

Что произойдет, если один протокол будет использоваться более одного раза одновременно?

3.3. Мой первый сервер

Для того, чтобы выполнить это упражнения, Вам понадобится программа **netcat (nc)**. В дистрибутиве BackTrack она есть по умолчанию, точно также как в OSX, однако ее легко можно скачать и установить и на другие операционные системы.

I. В консоли наберите:

```
nc -h
```

Эта команда отобразит опции, которые доступны в netcat.

Для создания простого сервера в Linux/Windows, наберите:

```
nc -l -p 1234
```

Если Вы используете OSX, наберите:

```
nc -l 1234
```

Вы только что запустили сервер, прослушивающий порт 1234.

II. Откройте второе окно консоли и наберите:

```
netstat -a
```

Так Вы проверите, что появился новый сервис, который прослушивает порт 1234.

Чтобы установить связь с сервером нужно использовать клиент! Во втором окне консоли наберите:

```
nc localhost 1234
```

Данная команда создаст соединение с сервером на порт 1234. Теперь, все, что печатается в в одном из открытых окон консоли будет отображаться и в другом.

Как можно использовать подобного рода сервис, чтобы взломать вашу систему?

Netcat пересылает весь трафик в открытом виде. Существует ли безопасная альтернатива?

III. Остановите сервер, вернувшись в первое окно консоли и нажав Ctrl+C.

IV. Теперь, создайте текстовый файл (.txt) и назовите его «test». Запишите в текстовый файл фразу: “Welcome to my server!”

Как только завершите, посмотрите на команду и разберитесь в ней и расскажите вашему преподавателю, что делает каждая из ее опций. Наберите:

```
nc -l -p 1234 < test
```

Из другого окна консоли подключитесь к серверу, набрав:

```
nc localhost 1234
```

Когда клиент подключится к серверу, Вы должны будете увидеть содержимое файла *test*.

Какой протокол используется для подключения к серверу? Позволяет ли netcat Вам это изменить? Если да, то каким образом?

Пища для ума: модель OSI

Модель OSI (семиуровневая модель) разработана в 1980-х (примерно через 10 лет после разработки модель TCP/IP) организацией **ISO** (International Standards Organization). OSI означает Взаимосвязь Открытых Систем (англ. **Open Systems Interconnection**), она занимается стандартизацией сетевой архитектуры как независимая компания, которая не вовлечена в процесс разработки и развития сетей.

Рисунок 3.7: Модель ISO/OSI

Модель OSI состоит из уровней с практичными и простыми правилами. Схожие функции группируются вместе на одном из уровней, и, пожалуйста, не забывайте, что каждый уровень обслуживается нижележащим уровнем и обслуживает вышележащие.

Каждый из уровней выполняет свою часть работы для осуществления коммуникации и нововведения на одном из них не влияют на работу всех остальных. Благодаря этой возможности мы наблюдаем глобальный Интернет бум во всем мире, где каждый день появляются новые услуги и приложения.

Каждый уровень включенный в процесс коммуникации на одном компьютере связывается с этим же уровнем на другом компьютере. Это означает, что когда Вы заходите на сайт www.google.com в браузере, то осуществляется прямая взаимосвязь между Уровне приложений (7 ур.) Вашего компьютера (Ваш веб браузер) и сервером Google (также на 7 ур.). То же самое относится ко всем остальным уровням модели OSI.

Давайте рассмотрим за что ответственен каждый из уровней модели OSI.

Уровень приложений	Отвечает за связь между приложением с пользовательским интерфейсом приложения, к примеру: использование веб-браузера.
Представительский уровень	Отвечает за то, чтобы обмен данными осуществлялся способом, который поддерживают обе стороны. К примеру: на данном уровне производится шифрование данных.
Сессионный уровень	Данный уровень управляет установлением, поддержанием и завершением сессий между устройствами.
Транспортный уровень	Осуществляет прозрачную передачу данных между устройствами. Он разбивает большой объем данных на маленькие сегменты, чтобы повысить надежность передачи данных по сети (множество маленьких кусочков информации намного надежнее передавать по сети, нежели один большой поток). Если какой-либо сегмент утерян в процессе передачи, то транспортный уровень ответственен за повторную передачу утраченных данных и за правильную последовательность приема всех фрагментов.
Сетевой уровень	Данный уровень отвечает за адресацию. И не только за то, чтобы каждый адрес был уникален, но также и доступность маршрута до точки назначения. Информация перемещается от одного устройства к другому, пока не достигнет конечной точки назначения, и каждое из устройств должно знать куда следует отправить информацию далее.
Канальный уровень	Канальный уровень позволяет убедиться в том, что на физическом уровне не возникнет ошибок при передачи информации через различные среды передачи. Инкапсуляция на данном уровне позволяет передать информацию в любой среде (радиоволны, оптоволокно, медный кабель). Также данный уровень ответственен за адресацию в локальной сети (на основе MAC-адресов) и за контроль ошибок (при помощи контрольной суммы CRC32).
Физический уровень	Этот уровень отвечает за физические спецификации устройств, которые должны быть осуществлены для того, чтобы иметь возможность передавать информацию в заданной среде передачи. Для Wi-Fi это радиоволны; для оптоволокна это световые волны; а для медного кабеля это электрические сигналы. Также физический уровень описывает принцип обработки сигналов и механизмы передачи полезной информации (модуляция).

Эти семь уровней включают в себя все, что нужно для надежной передачи данных по сети.

Вот как выглядят различные модели, которые мы обсуждали ранее:

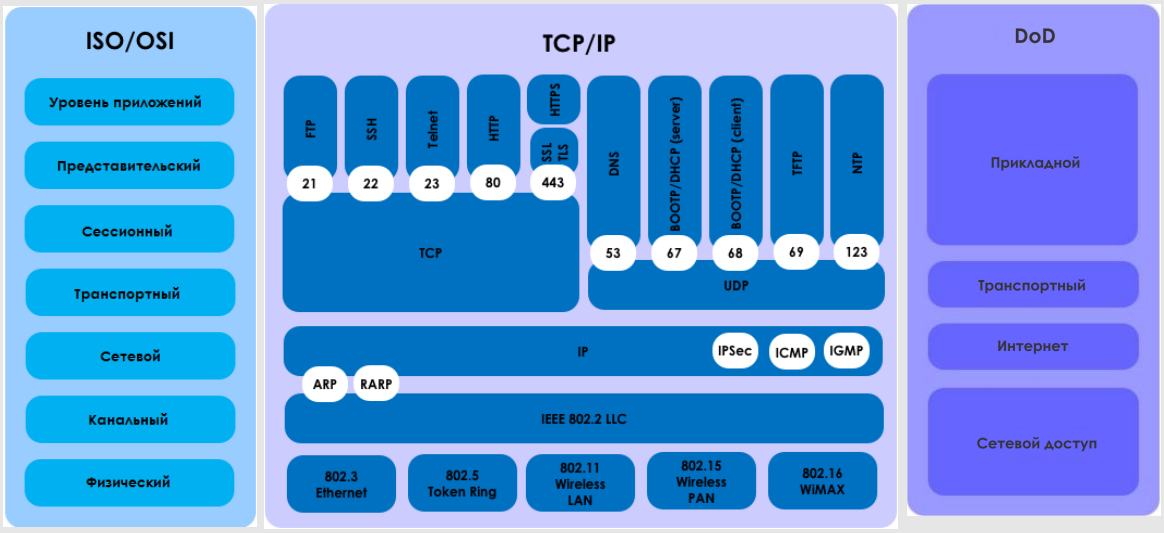

Рисунок 3.8 — Сравнение сетевых моделей

Today's teens are in a world with major communication and productivity channels open to them and they don't have the knowledge to defend themselves against the fraud, identity theft, privacy leaks and other attacks made against them just for using the Internet. This is the reason for Hacker Highschool.

The Hacker Highschool project is the development of security and privacy awareness learning materials for junior high and high school students.

Hacker Highschool is a set of lessons and a practical means of making hackers. Beyond just providing cybersecurity awareness and critical Internet skills, we need to teach the young people of today how to be resourceful, creative, and logical, traits synonymous with hackers. The program contains free security and privacy awareness teaching materials and back-end support for teachers of accredited junior high, high schools, and home schooling. There are multiple workbooks available in multiple languages. These are lessons that challenge teens to be as resourceful as hackers, including safe Internet use, web privacy, researching on the internet, avoiding viruses and Trojans, legalities and ethics, and more.

The HHS program is developed by ISECOM, a non-profit, open-source research group focused on security awareness and professional security development and accreditation.

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM