Hacker Highschool

SECURITY AWARENESS FOR TEENS



Урок 3
Інтернет зсередини

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM

Увага

Проект Hacker Highschool є засобом навчання і, як в будь-якому навчальному засобі, існує небезпека. Деякі уроки, якщо ними зловживати, можуть призвести до фізичної травми. Також додаткові небезпеки можуть бути там, де ще недостатньо досліджень про можливі наслідки випромінювань від специфічної техніки. Студенти, які використовують ці уроки, повинні перебувати під контролем, і, в той же час, заохочуватися на вивчення, практику і заняття. ISECOM не несе відповідальності за застосування інформації, отриманої з даних матеріалів, і за подальші наслідки.

Наступні уроки та книги є відкритими і загальнодоступними на наступних умовах ISECOM:

Всі роботи проекту Hacker Highschool призначені для некомерційного використання з учнями початкової школи, слухачами юнацьких курсів Highschool, і студентами вищих навчальних закладів, приватних організацій або частково для домашнього навчання. Ці матеріали в будь-якій формі не можуть бути використані для продажу. Надання цих матеріалів будь-якому класу, навчальній організації або табору, в яких стягується плата, категорично заборонено без ліцензії, в тому числі на уроки в коледжі, університеті, професійно-технічних заняттях, літніх або комп'ютерних таборах тощо.

Для придбання ліцензії відвідайте розділ сайту призначений для Ліцензування: http://www.hackerhighschool.org/licensing.html.

Проект HHS є результатом праці відкритого співтовариства і, якщо Ви знаходите наші труди цінними і корисними, ми просимо Вас підтримати нас шляхом придбання ліцензії, пожертвувань, або спонсорства.

ЗМІСТ

Співробітники

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

Перекладачі

Vadim Chakryan, Kharkiv National University of Radio Electronics

Olena Boiko, Kharkiv National University of Radio Electronics

Dmitriy Pichuev, Ukrainian Engineering Pedagogical Academy

Andrii Sezko, Kharkiv National University of Radio Electronics

ISECOM

Вступ та мета

У далекому минулому, до появи Інтернету, електронна комунікація була схожа на шаманство. У кожної фірми з виробництва комп'ютерів було своє уявлення про те, як машини повинні взаємодіяти. І ніхто навіть не розглядав можливість того, що комп'ютер Wang міг би взаємодіяти з комп'ютером Burroughs .

Світ змінився, коли вчені та студенти усвідомили зручність використання терміналів для доступу до мейнфреймів. З'явився знаменитий ПК IBM, і користувачі захотіли отримувати доступ до мейнфреймів зі своїх персональних комп'ютерів. Незабаром за допомогою модемів здійснювалися dial-up з'єднання, а користувачі змогли працювати з програмами емуляції терміналу. Роботу в мережі стали сприймати як чорну магію, а людей, які розбиралися в нових технологіях, називали не інакше як гуру.

Світ знову значно змінився, коли Інтернет, що починався як військовий проект, став загальнодоступним. Робота в мережі завжди була локальною, тобто обмежена в межах одного офісу або університету. Як же збиралися взаємодіяти всі ці різні системи?

Відповіддю стало введення системи універсальної адресації для існуючих мереж. У цілому ця система називається міжмережевим протоколом **(IP , Internet Protocol)**. Уявіть, що ваш друг відправляє вам посилку з-за кордону. Ця посилка може перевозитися на літаку, поїзді або автомобілі, але Вам, насправді, не потрібно знати розклад авіаперевізників або місце розташування найближчої залізничної станції. Ваша посилка в підсумку прибуде на Вашу адресу, що в кінцевому рахунку є єдиною значущою інформацією. Ваша IP-адреса виконує ту ж функцію: пакети можуть переміщатися як електрони, пучки світла або радіохвилі, але не важливо як вони переміщуються. Важлива Ваша IP-адреса і IP-адреса системи, з якою Ви взаємодієте.

Одна проблема, яка ускладнює цю ідею в реальних умовах, - це те, що за однією адресою може жити кілька людей. У світі мереж аналогічна проблема виникає, коли один сервер підтримує як звичайний HTTP і безпечний HTTPS, так і FTP. Помітили букву "P" в кінці цих абревіатур? Це завжди є позначенням протоколу, який можна охарактеризувати як «тип комунікації».

Цей урок допоможе вам розібратися в тому, як працюють протоколи і порти в Windows, Linux і OSX. Ви також ознайомитеся з деякими утилітами (деякі з них вже були розглянуті в попередньому уроці), які аналізують можливості вашої системи з організації мережі.

Після цього уроку у вас будуть базові уявлення про наступні поняття:

- принципи побудови мереж та їх взаємодія
- IP-адреси
- порти і протоколи

Базові поняття мережевої взаємодії

Почнемо з розгляду локальних мереж (**LAN, Local Area Network)**. LAN дозволяє комп'ютерам, що знаходяться відносно близько один від одного, спільно використовувати ресурси (наприклад, принтери, дисковий простір), а адміністратори керують таким доступом. Далі описані загальні мережеві пристрої і топології.

Пристрої

Продовжуючи кар'єру хакера, ви зустрінете велику кількість діаграм різних мереж. Корисно знати позначення, які зустрічаються на подібних діаграмах найбільш часто:

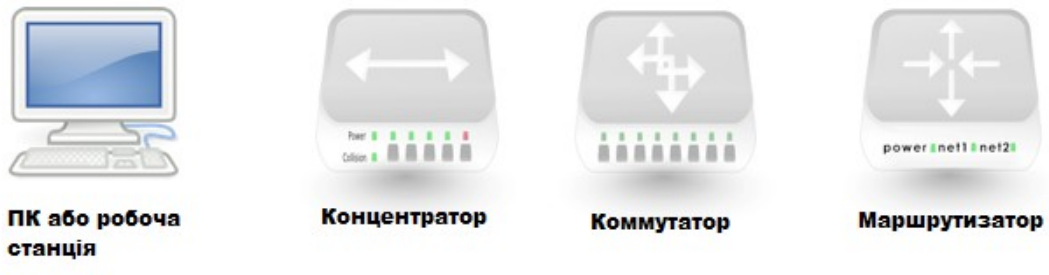


Рисунок **3.1:** поширені позначення на діаграмах мереж

Хаб **(**концентратор**)** схожий на старомодну загальну телефонну лінію: всі використовують один і той же провід і можуть слухати розмови всіх інших. Через це LAN перевантажується.

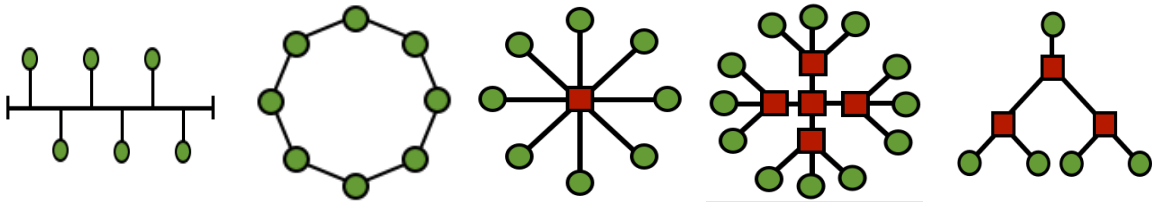
Концентратор в цьому плані краще: він фільтрує трафік так, що тільки два взаємодіючих між собою комп'ютера можуть «чути розмову». Як і концентратор, він використовується тільки в LAN.

Маршрутизатор розміщують між кількома LAN; він використовується для доступу до інших мереж і до Інтернету. Маршрутизатор використовує IP-адреси. Він аналізує відправлені пакети і визначає, якій мережі вони призначені. Якщо пакет належить «іншій» мережі, він переправляє пакет в місце призначення.

Топології

Топологія представляє собою спосіб з'єднання комп'ютерів. Рішення, прийняті щодо топології мережі, в майбутньому можуть принести як позитивні, так і негативні результати в залежності від: використовуваних технологій, технологічних і фізичних обмежень, продуктивності і вимог до безпеки, розміру і типу організації і т.п.

Фізична структура LAN може виглядати як одна з наступних фізичних топологій:



Шина Кільце Зірка Розширена зірка Ієрархічна

Рисунок **3.2** — Топології

Шина: всі комп'ютери підключені до одного кабелю, і кожен може безпосередньо взаємодіяти з будь-яким іншим комп'ютером. Але при пошкодженні якої частини шини робота мережі буде припинена.

Кільце: кожен комп'ютер пов'язаний з наступним за ним, а останній - з першим. Кожен комп'ютер може безпосередньо взаємодіяти тільки з двома сусідніми комп'ютерами.

Топології типу «шина» зараз використовуються рідко. Кільце часто використовується на міждержавному рівні, зазвичай з двома кільцями із зустрічним напрямках передачі даних, які відправляють трафік в протилежних напрямках для забезпечення надійності та відмовостійкості.

Зірка: комп'ютери безпосередньо не з'єднані один з одним. Вони пов'язані через концентратор або комутатор, який передає інформацію від одного комп'ютера до іншого.

При з'єднанні один з одним декількох концентраторів або комутаторів виходить топологія типу розширена зірка.

У топології типу «зірка» або «розширена зірка» всі центральні вузли є пірами, тобто вони по суті рівноправні. На сьогоднішній день це найпоширеніша топологія LAN.

Однак якщо з'єднати разом дві мережі з топологією зірки або розширеної зірки через центральний вузол, який управляє трафіком або обмежує його між двома мережами, то вийде ієрархічна топологія мережі. Вона зазвичай використовується на більш великих підприємствах.

Гра почалась: залишаючи лазівки відкритими

У літню спеку Джейс була щаслива допомогти місцевому поліцейському відділенню, в якому був встановлений кондиціонер, провести невелику мережу. Вони платили їй печивом, спасінням від спеки, розмовами і можливостями встановити лазівки в системі. Ковзаючи під сталевими робочими столами, які десятиліттями не зсувалися з місця, Джейс знайшла найбрудніше затишне місце для того, щоб заховати Wi -Fi точку доступу. Джейс просто підключила її, засипала зверху сміттям і простягнула Ethernet-кабель до настінних портів, які вона до цього встановила.

Важка рука ляснула по столу над нею. Джейс вдарилася об металеву кришку і скрикнула : «Ой! Моя голова!», - Оговтавшись, вона запитала, - «Ви впевнені, що не хочете, щоб я встановила вам сервер?»

Поліцейський відкашлявся і спробував відповісти голосом Темного Професора: «Що ж, можливо, і хотів би, але я не впевнений, що резистор променевих потоків впорається з взаємним впливом мікроканалів. Особливо коли повний місяць випадає на останній вівторок місяця. »

Джейс зашаркала ногами з властивою підліткам глузливою дратівливістю. «Мабуть, у вас немає проблем з досягненням квантових рівнів нісенітниці. А коли я отримаю своє печиво, офіцер Кікем?»

«Будь ласка, Джейс, називай мене Хенком. Я відчуваю себе старим, коли ти називаєш мене офіцер Кікем». Він намагався надати голосу страждальницький відтінок, але Джейс була знайома з прийомами соціальної інженерії: насправді він намагався відвернути її від печива.

«Хенк, не хотілося б повідомляти вам цю новину, але ви насправді старий. »

«Ох, звучить образливо. Я не старий, я авторитетний», - заперечив він, розглядаючи свої до блиску начищені чорні поліцейські черевики, а рвані кросівки Джейс тим часом зникли під важким столом. Незабаром Хенк знову побачив її карі очі та обличчя, покрите павутинням. У Джейс в руці все ще була котушка кабелю. Хенк допоміг їй піднятися і прибрав павутину з її обличчя і плечей .

«Допоможіть, міліцейське свавілля», - піддражнила Джейс.

«Злочинець», - відповів на це Хенк. - «Так, повідай мені Ваш пекельний план», - Джейс здалося, що грубий і м'язистий представник закону попросив її благальним тоном.

Вона запитала: «Ви впевнені, що хочете знати про устрій всіх цих мережевих штук?» Він нетерпляче кивнув. Джейс подумала: прямо як іграшка з головою що хитається.

«Добре. Я зробила проект мережевої топології. Це схоже на карту, на якій вказано, де буде встановлено все обладнання, комп'ютери, концентратори, роз'єми, комутатори, маршрутизатори і брандмауери. Ви не можете починати такий проект без карти», - сказала вона, поглядаючи на поліцейського. «Це робиться для гарантії того, що кожен вузол зможе взаємодіяти з будь-яким іншим вузлом, без єдиної точки відмови. Наприклад, топологія «шина» нікуди не годиться, тому що якщо один вузол в шині буде пошкоджений, то вся мережа припинить роботу». Хенк кивнув, і Джейс продовжила .

«Уявіть , що робота в мережі - це ця поліцейська ділянка. До вас привели одного підозрюваного. Кожен коп очікує своєї черги, щоб допитати свідка, не віднімаючи при цьому часу у своїх колег. Якщо жертву, тобто підозрюваного, переводять в іншу камеру, решті копів, які ще мають допитати його, потрібно знати, куди його перевели.»

«Джейс, схоже, тебе теж чекає допит, якщо ти будеш продовжувати так говорити про нас, блюстителів порядку.» Хенк підтягнув свій збройовий ремінь і втягнув свій невеликий живіт.

Джейс проковтнула смішок. «Виходить, що підозрюваний - це пакет даних, а ви, поліцейські бандити, - мережеві пристрої. Кожному пристрою, будь то комутатор, маршрутизатор, брандмауер, сервер або що-небудь інше, необхідно знати , що пакет вже розглянуто. Ну знаєте, побитий поліцейськими кийками. Я думаю, Ви називаєте це помити голову дерев'яним шампунем.»

Хенк закотив очі і спробував намацати кийок, якого у нього при собі не виявилося.

Хіхікаючи, Джейс підняла котушку кабелю як щит. «Гей, у мене тут котушка проводів і у мене вистачить сміливості використати її. Поставте на місце чашку з кавою і ніхто не постраждає.» Втративши рівновагу сміючись, Джейс звалилася на Хенка; він навіть не ворухнувся. Ох, цей хлопець дійсно міцний, подумала вона. Його рука, яку він поклав на її плече, нагадала їй про щось.

Вона поспішно встала та трохи почервоніла. «Отже, є два типи пристроїв. Розумні і дурні. Як поліцейські. » Четверо у поліцейській формі з'явилися в кімнаті в самий непідходящий момент, почувши «дурні, як поліцейські». Джейс плутано продовжила: «Розумні пристрої

пам'ятають все, що вони роблять. Вони ведуть журнал своєї діяльності.»

«А ті, які дурні ? Як поліцейські?", - запитав начальник поліції.

Кінець гри

Модель **TCP/IP (DoD)**

Модель TCP/IP була розроблена Міністерством оборони США **(DoD, Department of Defense)** і Агентством передових оборонних дослідницьких проектів **(DARPA, Defense Advanced Research Project Agency)** в 1970-х рр. Модель TCP/IP була спроектована в якості відкритого стандарту, який кожен міг би використовувати для з'єднання комп'ютерів і обміну інформацією між ними. У кінцевому рахунку, вона стала основою Інтернету.

У цілому, найбільш проста форма моделі TCP/IP називається моделлю DoD. З неї і почнемо.

Рисунок **3.3** — Модель DoD

Рівні

Проста модель DoD визначає чотири повністю незалежних рівня, на які вона розділяє процес комунікації між двома пристроями. Рівні, через які проходить інформація, наступні:

Прикладний рівень

Прикладний рівень - це саме те, про що ви, можливо, подумали: це рівень, на якому працюють такі застосунки, як Firefox, Opera, поштові клієнти, сайти соціальних мереж, клієнти миттєвого обміну повідомленнями і застосунки для організації чатів. Насправді,

досить багато застосунків мають доступ до Інтернету: наприклад, деякі офісні застосунки підключаються до онлайн колекцій кліпарту. Прикладний рівень створює корисне навантаження (корисні дані, без домішку керуючих заголовків), яку переносять всі інші рівні. Доброю аналогією може послужити поштова система. Застосунок створює пакет, при цьому «загортаючи» його в інструкції щодо того, як цей пакет повинен бути використаний. Потім пакет передається до відділу обробки кореспонденції - на транспортний рівень .

Транспортний рівень

Транспортний рівень встановлює мережеві з'єднання, які називаються сесіями. У світі Інтернет основним протоколом на транспортному рівні є **TCP (the Transmission Control Protocol** , протокол управління передачею**)** . TCP загортає повідомлення у ще одну «оболонку» з вказівками про те, яке це повідомлення за рахунком (наприклад, перше з трьох), якому застосунку воно призначене (на основі портів - кожен порт відповідає своєму застосунку) і як упевнитися в тому, що повідомлення доставлено без ушкоджень.

Припустимо, Ви хочете відправити лист електронною поштою Вашій мамі. Лист може бути маленьким або великим, в будь-якому випадку, воно виявиться занадто великим, щоб відправити його по Інтернету цілком. Замість цього TCP розбиває цей лист на сегменти - послідовно пронумеровані маленькі порції даних з прикріпленим в кінці кодом перевірки помилок. Якщо в процесі передачі пакет пошкоджується, TCP запитує повторну передачу. На приймальній стороні TCP збирає порції даних в правильному порядку, і ваша мама отримує лист на свою електронну поштову скриньку.

Але не забувайте, що TCP не єдиний протокол транспортного рівня: UDP також функціонує на цьому рівні. Його особливістю є те, що він НЕ створює сесії. Він просто відправляє потік інформації, при цьому ніколи не перевіряє, чи була інформація отримана кінцевим вузлом.

Рівень Інтернет

Цей рівень додає інформацію про адреси відправника і одержувача і про те, де починається і закінчується пакет**.** Аналогічно працює служба доставки, яка відправляє посилки за правильною адресою. Цей рівень не піклується про те, чи всі пакети будуть доставлені і чи не будуть вони пошкоджені; це робота транспортного рівня. Основним протоколом на цьому рівні, відповідно, є протокол **IP (Internet Protocol,** міжмережевий протокол**).** Цей рівень, вибирає найбільш оптимальний маршрут для передачі інформації на основі IP-адреси.

Рівень мережевого доступу

Цей рівень являє собою низькорівневу фізичну мережу, яку використовують для з'єднання з Інтернетом. Якщо ви використовуєте dial-up, то нам вас шкода, і ви користуєтеся простим **PPP-**з'єднанням. Якщо ви використовуєте DSL, то ви, мабуть, користуєтеся **ATM** або **Metro Ethernet**. А якщо у вас кабельний Інтернет, то ви використовуєте фізичну мережу **DOCSIS**. Незалежно від способу доступу в Інтернет, TCP/IP успішно керує з'єднанням. Рівень мережевого доступу включає в себе Ethernet-кабель і мережеву карту **(network interface card, NIC)** або безпроводовий адаптер і точку доступу. На цьому рівні здійснюється обробка низькорівневих даних (у вигляді біт - цифрових одиниць і нулів), переданих від одного вузла до іншого.

Пожива для розуму **«**Модель **OSI»**

Подивіться розділ «Модель OSI» наприкінці цього уроку. Це альтернативний погляд на моделювання мережі.

Протоколи

Тепер у вас є з'єднання з Інтернетом. Здається, що все просто, але розглянемо звичайну ситуацію: ви шукаєте щось в Інтернеті, у той час як ваш брат або сестра дивляться фільм онлайн. Чому ж ці два потоки трафіку не перемішуються? Як мережі вдається розрізняти їх?

Відповіддю на ці питання є використання протоколів, які можна назвати мовами, якими розмовляють різні типи трафіку. Веб-трафік використовує один протокол, передача файлів — інший, а електронна пошта — третій. Але як і все у цифровому світі, протоколи насправді не використовують імена на мережевому рівні; вони використовують IP-адреси та номера портів.

Протоколи прикладного рівня

FTP (*File Transfer Protocol,* протокол передачі файлів*)* використовується для передачі файлів між двома пристроями. Він використовує один порт для доставляння даних і ще один порт для відправлення керуючих сигналів («Я отримав файл! Дякую!»). Зазвичай використовуються порти за номерами 20 та 21 (TCP).

HTTP (*Hyper-Text Transfer Protocol,* протокол передачі гіпертексту*)* використовується для передачі веб-сторінок. Використовує 80 TCP порт. **HTTPS** — це безпечний варіант HTTP, який шифрує мережевий трафік, зазвичай використовує 443 TCP порт.

SMTP *(Simple Mail Transfer Protocol,* простий протокол пересилання пошти*)* — це протокол, за яким відправляється електронна пошта. Використовує 25 TCP порт.

DNS (Domain Name Service, служба доменних імен) — використовується для перетворення доменних імен (наприклад: ISECOM.org) в IP-адреси (наприклад: 216.92.116.13). Використовує 53 UDP порт.

Протоколи транспортного рівня

TCP та **UDP** — це два основних протоколи, які використовуються на транспортному рівні для передачі даних.

TCP (Transmission Control Protocol, протокол керування передачею**)** встановлює логічне з'єднання (сесію) між двома хостами в мережі. Він встановлює це з'єднання, використовуючи процедуру потрійного рукостискання.

1. Коли мій комп'ютер хоче під'єднатися до вашого, він відправляє пакет з прапорцем **SYN**, тим самим кажучи: «Давай синхронізуємо годинники, щоб можна було обмінюватися трафіком з часовими мітками».

2. Ваш комп'ютер (якщо він збирається прийняти з'єднання) відповідає, відправляючи пакет підтвердження синхронізації з прапорцями **SYN/ACK**.

3. Мій комп'ютер «скріплює угоду», відправляючи пакет з прапорцем **ACK**. З'єднання встановлено.

Але так виходить тільки у випадку використання протоколу TCP. На відміну від нього, **UDP (User Datagram Protocol,** протокол дейтаграм користувача**)** — це транспортний протокол, для якого навіть не є важливим наявність з'єднання. Він просто передає дані навіть не дбаючи про нумерацію фрагментів та про те, чи отримав кінцевий вузол інформацію, що передається, або ні. Така особливість робить UDP дуже швидким, тому його зручно використовувати для голосового та відеотрафіку, або для онлайн ігор, для яких втрата окремих фрагментів інформації несуттєва.

Протоколи рівня Інтернет

IP (Internet Protocol, міжмережевий протокол**)** виконує роль універсального протоколу для комунікації між будь-якими двома комп'ютерами у будь-якій мережі у будь-який час. Можна

провести аналогію з поштарем, який доставляє пошту; все, що він повинен зробити, — це донести пакети за адресами їх одержувачів.

Internet Control Message Protocol (ICMP, протокол міжмережевих керуючих повідомлень**)**

ICMP — це протокол, який використовують мережеві пристрої та адміністратори мережі для виявлення несправностей та підтримки роботи мережі. Він включає в себе утиліту **ping** (Packet InterNet Groper, пакетний «слідопит» Інтернету) та схожі команди, які тестують мережу та повідомляють про помилки. Оскільки ping часто використовується для проведення флуд (англ. flood) атак на мережеві пристрої, більшість систем обмежують ICMP до одного відгуку за секунду.

Порти та протоколи пов'язані наступним чином:

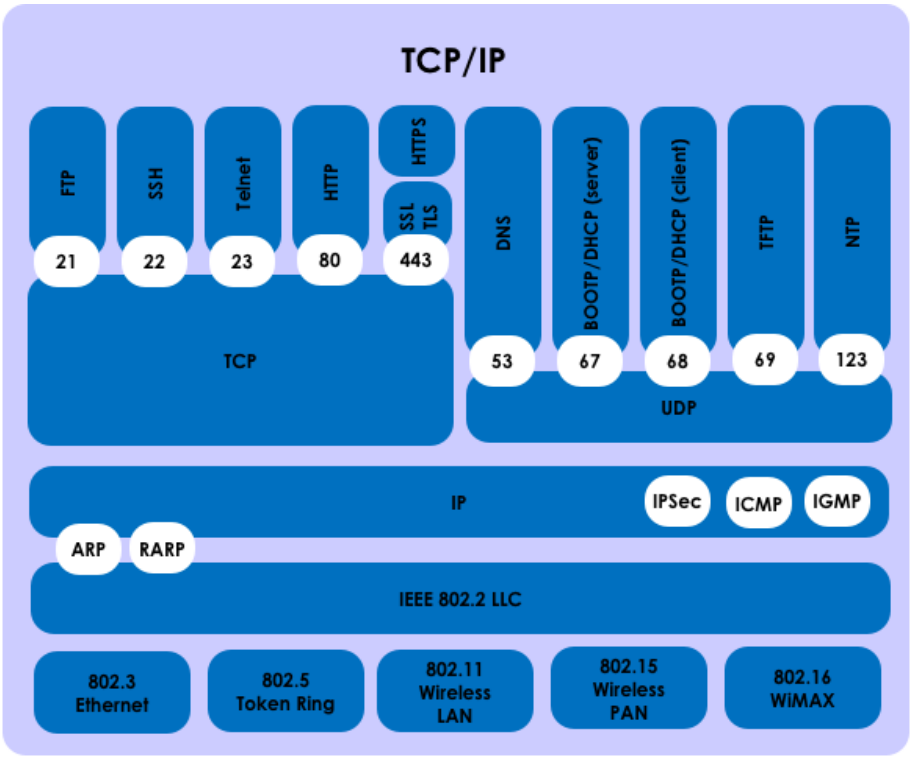


Рисунок **3.4** — Стек TCP/IP

IPv4-адреси

Доменні імена зручні для людей, оскільки ми легко запам'ятовуємо назви (наприклад, ISECOM.org). Але мережі, в дійсності, їх не розуміють; вони розуміють тільки числові IP-адреси. Таким чином, коли ви запитуєте ISECOM.org, ваш комп'ютер за допомогою **DNS (Domain Name Service**, служба доменних імен**)** здійснює швидкий пошук, щоб знайти відповідну IP-адресу.

IP-адреси схожі на поштові адреси. Щоб отримувати пошту, Вам потрібна поштова адреса. **IPv4**-адреси складаються з 32 біт, які розбиті на чотири 8-бітові октети, розділені крапками. Це означає, що в Інтернеті існують 2^{32} (або 4,294,967,296) унікальних IPv4-адрес. Одна частина IP-адреси ідентифікує мережу, а інша — ідентифікує окремий комп'ютер в мережі. Можна провести аналогію IP-адреси з поштовою адресою: країна/місто — це адреса мережі, назва вулиці — це хост.

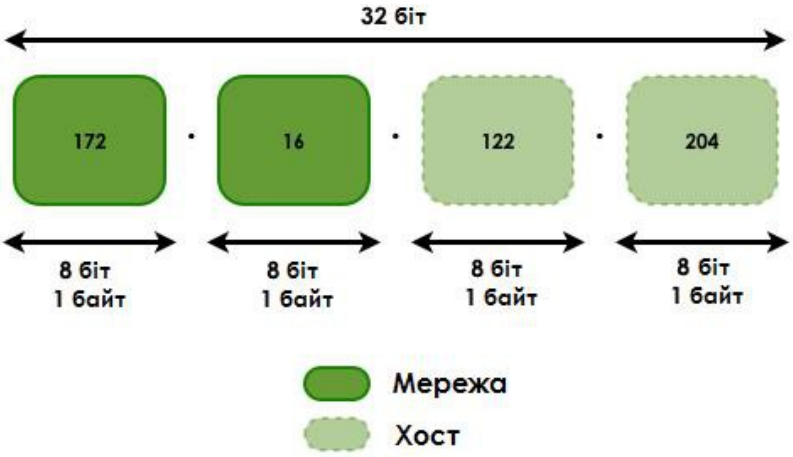


Рисунок 3.5 — Номер мережі та ID хоста

Повертаючись до аналогії з поштовою службою: IP — це вантажівка, яка доставляє пакети до потрібного поштового відділення. TCP — це зовнішній пакувальник, з переліком того, скільки всього пакунків та який це саме пакунок за рахунком. IP-адреса — це адреса конкретного будинку (комп'ютера), для якого призначений цей пакет.

Розрізняють зовнішні та внутрішні IP-адреси. Внутрішні IP-адреси використовуються приватними мережами; подібні адреси не можна побачити в Інтернеті, лише усередині локальної мережі.

IP-адреси в межах однією внутрішньої мережі не можуть повторюватися, але комп'ютери в двох різних — але незв'язаних — внутрішніх мережах можуть мати однакову IP-адресу. IP-адреси, які визначені організацією IANA (the Internet Assigned Numbers Authority, Адміністрація адресного простору Інтернет) для внутрішніх мереж (відповідно до RFC 1918), наступні:

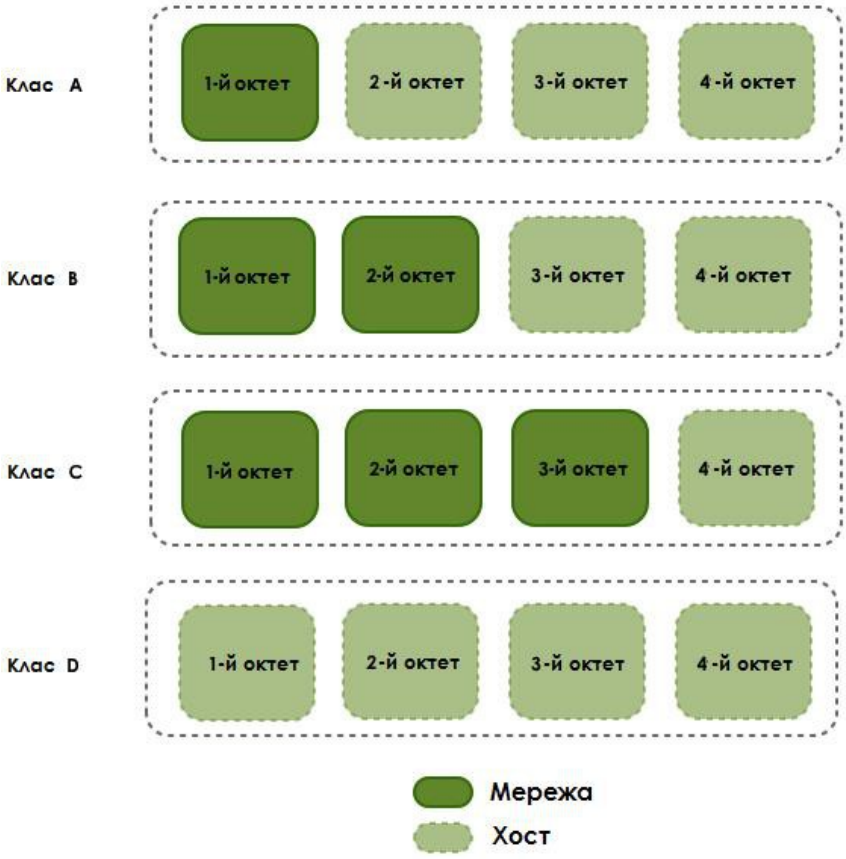
з 10.0.0.0 по 10.255.255.255	(Клас A)
з 172.16.0.0 по 172.31.255.255	(Клас B)
з 192.168.0.0. по 192.168.255.255	(Клас C)

Класи

IP адреси розділені на класи у відповідності до того, яка частина адреси використовується для ідентифікації мережі, а яка — для ідентифікації окремих комп'ютерів.

В залежності від розміру кожної частини або в мережі буде припустима більша кількість пристроїв, або буде припустима більша кількість мереж. Існують наступні класи:

Рисунок 3.5: Розділення класів IP



Клас **A:** Перший біт завжди дорівнює нулю, тому цей клас складають адреси від 0.0.0.0 (який, за згодою, ніколи не використовується) до 126.255.255.255. Варто зазначити, що адреси 127.x.x.x зарезервовані для loopback або localhost (див. далі).

Клас **B:** Перші два біти першого октету дорівнюють '10', тому цей клас складають адреси від 128.0.0.0 до 91.255.255.255.

Клас **C:** Перші три біти першого октету дорівнюють '110', тому цей клас складають адреси в діапазоні від 192.0.0.0 до 223.255.255.255.

Клас **D:** Перші чотири біти першого октету дорівнюють '1110', тому цей клас складають адреси в діапазоні від 224.0.0.0 до 239.255.255.255. Ці адреси зарезервовані для реалізації групових розсилок (multicast addresses).

Адреси, що залишились, використовуються з експериментальною метою або зарезервовані для подальшого використання.

Маска підмережі **(**або мережева маска**)** використовується для позначення цих класових розподілів. У двійковому позначенні частина, яка заповнена бітом '1' – ідентифікує мережу, а частина, яка заповнена бітом '0' – ідентифікує окремий хост. Стандартні маски мережі для перших трьох класів наступні:

255.0.0.0 (Клас A)

255.255.0.0 (Клас B)

255.255.255.0 (Клас C)

Насправді, така система є досить хитрою, оскільки у мереж, які використовують стандартні класи, маска займає один октет для класу A, два октети для класу B та три октети для класу C. Використовувати стандартні класи зручно, але не всі так роблять.

Це означає, що для ідентифікації хосту потрібна і IP-адреса, і маска мережі:

IP: 172.16.1.20
Маска: 255.255.255.0

Loopback-адреси

IP-адреси від 127.0.0.1 до 127.255.255.254 зарезервовані для використання у якості **loopback** (досл. «зворотня петля») або localhost-адрес, тобто вони посилаються безпосередньо на локальний комп'ютер. Кожний комп'ютер має localhost-адресу зі значенням 127.0.0.1, тому ця адреса не може використовуватися для ідентифікації інших пристроїв. Вона використовується для перевірки коректності роботи TCP/IP стеку на вашому пристрої.

Існує ряд інших адрес, які не можна використовувати. До них відносяться адреси мережі **(network address)** та широкомовні адреси **(broadcast address)**.

Адреси мережі

Адреса мережі — складається з мережевої частини IP-адреси та хост**-**частини, яка заповнена нулями. Цю адресу не можна надавати хосту, оскільки вона ідентифікує всю мережу, а не тільки один хост.

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

Широкомовні адреси

Широкомовна адреса — складається з мережевої частини IP-адреси та хост**-**частини, яка заповнена одиницями. Ця адреса не може використовуватися для ідентифікації хоста, оскільки це адреса, яку прослуховують всі хости (в цьому і є суть широкомовлення: всі слухають). Дані відправлені на широкомовний адрес будуть доставлені всім хостам в мережі.

Порти

І TCP, і UDP використовують порти для обміну інформацією із застосунками. Порт — це розширення адреси, подібно до додавання номеру квартири до назви вулиці. Лист з назвою вулиці прийде до правильного будинку, але без номеру квартири він не буде доставлений правильному адресату.

Порти працюють аналогічно. Пакет може бути доставлений за правильною IP адресою, але без відповідного порту неможливо визначити, якому застосунку призначений цей пакет. Номер порту — це також 16-бітове число. Це означає, що він може приймати десяткове значення від 0 до 65535 (2 в 16-му ступені).

Можна скористатися іншою аналогією: кожен комп'ютер — це поштове відділення. Кожний застосунок має свою абонентську скриньку; жодні два застосунки не можуть спільно використовувати одну і ту саму абонентську скриньку. Номер порту є тим самим номером абонентської скриньки.

Використання номерів портів дозволяє оброблювати декілька потоків інформації, які надходять на одну IP адресу; кожен з потоків відправляється до відповідного застосунку. Номер порту дозволяє службі, яка працює на віддаленому комп'ютері, дізнатися, який тип інформації запитує локальний клієнт та який протокол використовується для відправлення цієї інформації, при цьому одночасно підтримується зв'язок з різними клієнтами.

Наприклад, якщо локальний комп'ютер намагається підключитися до веб-сайту www.osstmm.org, IP адреса якого — 62.80.122.203, причому веб-сервер використовує порт за номером 80, то локальний комп'ютер підключиться до віддаленого комп'ютера, використовуючи адресу сокета (socket – сокет, зазначення IP-адреси з портом):

62.80.122.203:80

Для того, щоб підтримувати рівень стандартизації серед портів, які використовуються найчастіше, організація IANA затвердила, що порти за номерами від 0 до 1024 повинні використовуватися для загальних, привілейованих або загальновідомих застосунків. Порти, що залишилися – до 65535 – використовуються для динамічного розподілу або конкретних служб.

Порти, які використовуються найчастіше (загальновідомі), які призначені **IANA**, представлені в таблиці:

Призначення портів		
Номер	Ключові слова	Опис
5	rje	Remote Job Entry, віддалене введення завдань
0		Зарезервований
1-4		Не призначені
7	echo	Echo
9	discard	Discard, протокол відкидання
11	systat	Активні користувачі
13	daytime	Отримання поточної дати та часу
15	netstat	Who is Up або NETSTAT
17	qotd	Quote of the Day, «цитата дня»
19	chargen	Character Generator, генератор символів
20	ftp-data	File Transfer (дані), протокол передачі файлів
21	ftp	File Transfer (керування), протокол передачі файлів
22	ssh	Протокол віддаленого входу в систему SSH (Secure SHell – безпечна оболонка)
23	telnet	Telnet (TErminaL NETwork)

Призначення портів		
25	smtp	Simple Mail Transfer, простий протокол пересилання пошти
37	time	Time
39	rlp	Resource Location Protocol, протокол пошуку ресурсів
42	nameserver	Host Name Server
43	nicname	Who Is (досл. «хто такий?»)
53	domain	Domain Name Server, протокол сервера доменних імен
67	bootps	Bootstrap Protocol Server
68	bootpc	Bootstrap Protocol Client
69	tftp	Trivial File Transfer, простий протокол передачі файлів
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		Не призначені
160-223		Зарезервовані

Інкапсуляція

Коли якась частина інформації – електронний лист, наприклад, – передається від одного комп'ютеру до іншого, то у процесі передачі вона змінюється заздалегідь визначеним способом. Прикладний рівень генерує дані для передачі, які потім надходять до транспортного рівня. У свою чергу, транспортний рівень розбиває інформацію на невеликі «сегменти» та додає до кожного з них заголовок, у якому зберігається інформація про те, якому застосунку були адресовані дані та якому застосунку їх треба повернути при відповіді (порти), а також безліч інших опцій для контролю потоку.

Потім сегмент передається до мережевого рівня, де до них додаються заголовки мережевого рівня – таким чином формуються «пакети». У заголовку мережевого рівня містяться IP-адреси призначення і відправника для того, щоб мережеві пристрої розуміли, куди їм потрібно передавати дані та як їх повернути.

Після мережевого рівня інформація передається до канального рівня, де додається новий заголовок, у якому міститься інформація, що потрібна для адресації у локальній мережі (MAC-адреси) та контролю цілісності даних (контрольна сума – CRC32). На даному рівні з пакетів формуються «кадри».

Процес подібного додавання заголовків на різних рівнях і називають інкапсуляцією. Кожен нижче розташований рівень додає свій шматочок інформації у вигляді заголовку до інформації, отриманої від верхніх рівнів. Цей процес продовжується до тих пір, поки інформація не дійде до найнижчого рівня - фізичного, де дані вже безпосередньо перетворюються в сигнали і відправляються до лінії зв'язку.

Коли інформація надходить на приймальній стороні, то відбувається зворотній процес «деінкапсуляції». У ході даного процесу інформація піднімається по рівнях вгору і кожен з них аналізує і прибирає заголовок, за який він відповідає.

Процес інкапсуляції наведено на рис. 3.6, де видно, що процес починається зверху і йде вниз. На самому нижньому рівні ми отримаємо кадр «обліплений» керуючими заголовками. Саме тому в мережах прийнято розрізняти корисну пропускну здатність і загальну пропускну здатність. Корисною вважається пропускна здатність, яка передає лише корисне навантаження, тобто лише чисті дані, без домішки керуючих заголовків.

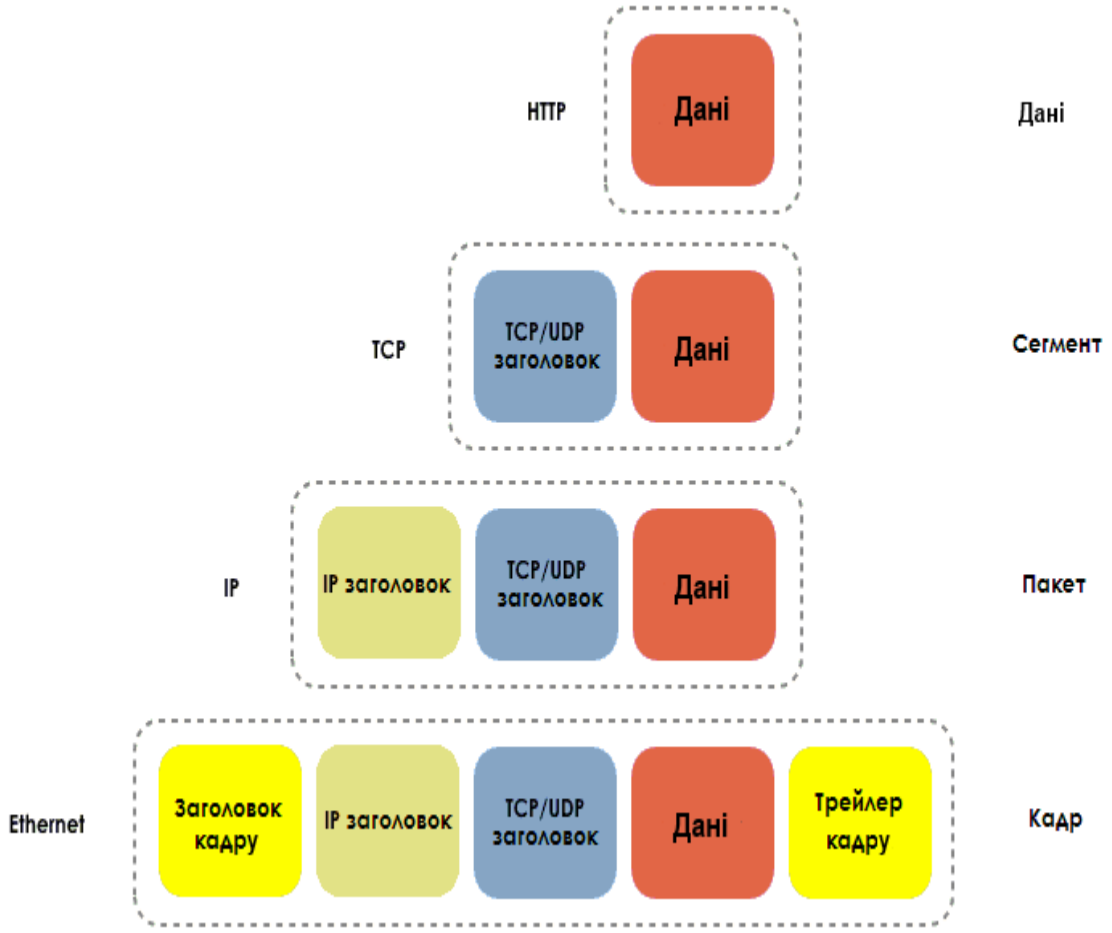


Рисунок **3.6** — Процес інкапсуляції

Для адресації в локальній мережі використовуються MAC-адреси (англ. "MAC - **Media Access Controller**"). Вони додаються в заголовку канального рівня, який знаходиться найближче до фізичного рівня (з цієї ж причини процес обробки кадру набагато простіше процесу обробки пакета). За допомогою MAC адрес, комутатори розуміють у який порт їм потрібно передати інформацію.

Зазвичай MAC-адреси представляються у вигляді 6 пар шістнадцятирічних значень, відокремлених один від одного дефісами:

00-15-00-06-E6-BF

MAC-адреси є фізичними адресами мережевої карти і вони не можуть бути змінені (звичайно, існують способи, за допомогою яких можна підмінити MAC-адресу, однак про них потрібно говорити окремо).

Вправи

3.1. Використовуючи команди вивчені в уроку 1 і 2 дізнайтеся IP-адресу вашого комп'ютера, мережеву маску, DNS-ім'я хоста і MAC-адресу. Порівняйте інформацію отриману Вами, з інформацією, отриманою Вашим напарником за сусіднім комп'ютером. У чому схожість і в чому різниця? Які IP-адреси використовуються в мережі: публічні або приватні?

3.2. netstat

Команда netstat відображає мережеву статистику: з ким Ви з'єднані, як довго працює мережа і т.п. Щоб запустити утиліту в Linux, Windows або OSX потрібно зайти в консоль операційної системи і набрати:

```
netstat
```

У консолі Ви побачите список встановлених з'єднань. Якщо ви хочете, щоб з'єднання відображали адреси і номери портів в числовому форматі, наберіть:

```
netstat -n
```

Для того, щоб побачити список усіх з'єднання та портів, що знаходяться в стані очікування, наберіть:

```
netstat -an
```

Щоб викликати довідку за команду, наберіть:

```
netstat -h
```

У списку netstat, зверніть увагу на колонки, в яких наведено і локальний і віддалений IP-адреси і подивіться які порти використовують дані підключення:

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

Порти це цифри, які наведені після IP-адреси, відокремлені двокрапкою. Чому порти використовуються віддаленими адресами відрізняються від портів, які використовують локальні адреси?

Відкрийте кілька вікон в браузері і в кожному з них відкрийте різні сайти. Знову використовуйте утиліту netstat.

Коли відкрито кілька вкладок в одному браузері, як він може зрозуміти який з них передавати отриману інформацію?

Чому відбувається так, що коли використовується браузер, то зникають порти, що знаходяться в процесі прослуховування?

Які протоколи використовуються?

Що станеться, якщо один протокол буде використовуватися більше одного разу одночасно?

3.3. Мій перший сервер

Для того, щоб виконати ці вправи, Вам знадобиться програма netcat (nc). У дистрибутиві (BackTrack) Kali Linux вона є за умовчанням, також як у OSX, проте її легко можна завантажити і встановити і на інші операційні системи.

I. У консолі наберіть:

```
nc -h
```

Ця команда відобразить опції, які доступні в netcat.
Для створення простого серверу в Linux / Windows, наберіть:

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

Якщо Ви використовуєте OSX, наберіть:

```
nc -l 1234
```

Ви тільки що запустили сервер, що прослуховує порт 1234.

II. Відкрийте друге вікно в консолі та наберіть:

```
netstat -a
```

Так Ви перевірите, що з'явився новий сервіс, що прослуховує порт 1234.

Щоб встановити зв'язок з сервером потрібно використовувати клієнт! У другому вікні консолі наберіть:

```
nc localhost 1234
```

Дана команда створить з'єднання з сервером на порт 1234. Тепер, все, що друкується в одному з відкритих вікон консолі буде відображатися і в іншому.

Як можна використати подібну службу, аби зламати вашу систему?

Netcat пересилає весь трафік у відкритому вигляді. Чи існує безпечна альтернатива?

III. Зупиніть сервер, повернувшись в перше вікно консолі і натиснувши Ctrl + C.

IV. Тепер, створіть текстовий файл (. Txt) і назвіть його «test». Запишіть у текстовий файл фразу: "Welcome to my server!"

Як тільки закінчите, подивіться на команду і розберіться в ній і розкажіть Вашому викладачеві, що робить кожна з її опцій. Наберіть:

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

З іншого вікна консолі підключіться до серверу, набравши:

```
nc localhost 1234
```

Коли клієнт підключиться до серверу, Ви повинні побачити вміст файлу *test*.

Який протокол використовується для підключення до серверу? Чи дозволяє netcat Вам це змінити? Якщо так, то яким чином?

Пожива для розуму: модель **OSI**

Модель **OSI (**семирівнева модель**)** розроблена у 1980-х (приблизно через 10 років після розробки моделі TCP/IP) організацією **ISO** (International Standards Organization). OSI значить Взаємозв'язок Відкритих Систем (англ. **Open Systems Interconnection**), що займається стандартизацією мережевої архітектури як незалежна компанія, яка не втягнута у процес розробки та розвитку мереж.

Рисунок **3.7:** Модель ISO/OSI

Модель OSI складається з рівнів з практичними й простими правилами. Схожі функції групуються разом на одному з рівнів, й, будь ласка, не забувайте, що кожен рівень обслуговується нижче розташованими рівнями та обслуговує вище розташовані.

Кожен з рівнів виконує свою частину роботи для здійснення комунікації і нововведення на одному з них, не впливають на роботу всіх інших. Завдяки цієї можливості ми спостерігаємо глобальний Інтернет-бум у всьому світі, де кожен день з'являються нові послуги та застосунки.

Кожен рівень, втягнутий у процес комунікації на одному комп'ютері, зв'язується з цим же рівнем на іншому комп'ютері. Це означає, що коли Ви заходите на сайт www.google.com

у браузері, то здійснюється прямий взаємозв'язок між рівнем додатків (7 рів.) Вашого комп'ютера (Ваш веб браузер) та сервером Google (також на 7 рів.). Те ж саме відноситься до всіх інших рівнів моделі OSI.

Давайте розглянемо за що відповідає кожен з рівнів моделі OSI.

Прикладний	Відповідає за зв'язок між застосунком і інтерфейсом користувача застосунку, наприклад: використання веб-браузера.
Рівень представлення	Відповідає за те, щоб обмін даними відбувався способом, який підтримують обидві сторони. Наприклад: на даному рівні здійснюється шифрування даних.
Сеансовий рівень	Даний рівень керує встановленням, підтримкою й завершенням сеансів між пристроями.
Транспортний рівень	Здійснює прозору передачу даних між пристроями. Він розбиває великий обсяг даних на маленькі сегменти, аби підвищити надійність передачі даних у мережі (безліч маленьких шматочків інформації набагато надійніше передавати по мережі, ніж один великий потік). Якщо якийсь сегмент втрачено у процесі передачі, то транспортний рівень відповідальний за повторну передачу втрачених даних і за правильну послідовність прийому всіх фрагментів.
Мережевий рівень	Даний рівень відповідає за адресацію. Не тільки за те, щоб кожен адрес був унікальним, але й також за доступність маршруту до точки призначення. Інформація переміщується від одного пристрою до іншого, поки не досягне кінцевої точки призначення, й кожен з пристроїв повинен знати, куди слід відправити інформацію далі.
Канальний рівень	Канальний рівень дозволяє переконатися у тому, що на фізичному рівні не виникнуть помилки при передачі інформації через різні середовища передачі. Інкапсуляція на даному рівні дозволяє передати інформацію у будь-якому середовищі (радіохвилі, оптоволокно, мідний кабель). Крім цього, даний рівень відповідальний за адресацію у локальній мережі (на основі MAC-адрес) та за контроль помилок (за допомогою контрольної суми CRC32).
Фізичний рівень	Цей рівень відповідає за фізичні специфікації пристроїв, які повинні бути здійснені для того, щоб мати змогу передавати інформацію у заданій середі передачі. Для Wi-Fi це радіохвилі; для оптоволокна це світлові хвилі; а для мідного кабелю це електричні сигнали. Також фізичний рівень описує принцип обробки сигналів та механізмів передачі корисної інформації (модуляція).

Ці 7 рівнів включають в себе все, що потрібно для надійної передачі даних у мережі.

Ось як выглядає різні моделі, які ми обговорювали раніше:

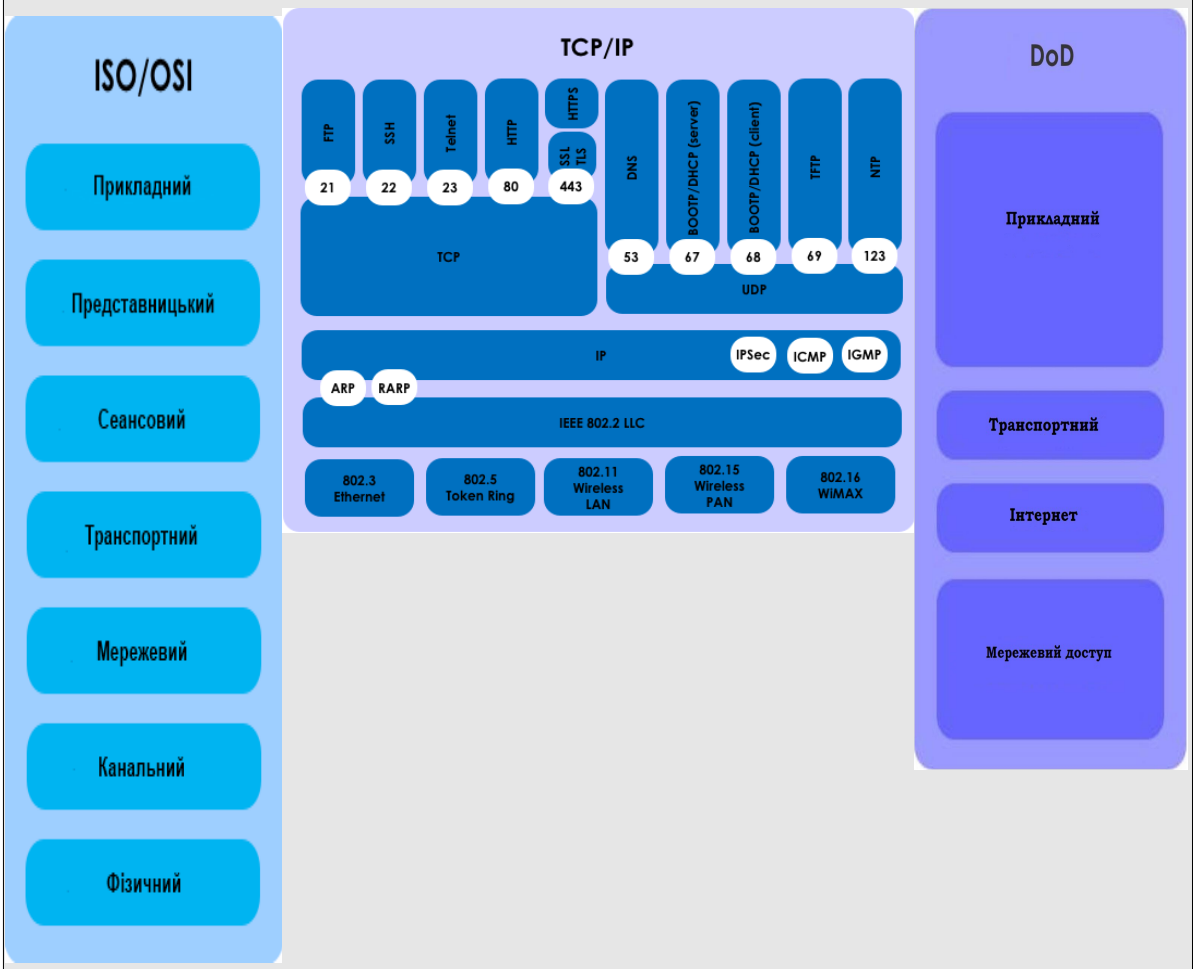


Рисунок **3.8** — Порівняння мережевих моделей

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