Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

Інформаційна безпека в комп'ютерних мережах

Методичні рекомендації до виконання лабораторних робіт для студентів денної форми навчання галузі 12 Інформаційні технології

ЗАТВЕРДЖЕНО
на засіданні кафедри кібербезпеки та програмного забезпечення, протокол № 1 від 15.08.2022

Кропивницький
2023

Інформаційна безпека в комп’ютерних мережах: Методичні рекомендації до виконання лабораторних робіт для студентів денної форми навчання галузі 12 Інформаційні технології. – Кропивницький: ЦНТУ – 2023. – 34 с./М-во освіти і науки України, Центральноукр. нац. техн. ун-т; /уклад. Смірнова Т.В., Буравченко К.О., Смірнов О.А., Конопліцька-Слободенюк О.К., Смірнов С.А./ – Кропивницький: ЦНТУ – 2023. – 34 с.

Укладачі: Смірнова Т.В., Буравченко К.О., Смірнов О.А., Конопліцька-Слободенюк О.К., Смірнов С.А.

Рецензенти: Коваленко О.В., докт. техн. наук, доцент;
Улічев О.С., канд. техн. наук.

ЗМІСТ

ВСТУП

Метою освітньої компоненти «Інформаційна безпека в комп'ютерних мережах» є формування у здобувачів вищої освіти ґрунтовних теоретичних знань, практичних умінь та навичок, необхідних для застосування в професійній діяльності у сфері забезпечення інформаційної безпеки в комп'ютерних мережах.

Основними **завданнями** вивчення дисципліни є формування наступних **компетенцій магістра з комп'ютерних наук:**

– СК05. Здатність розробляти, описувати, аналізувати та оптимізувати архітектурні рішення інформаційних та комп'ютерних систем різного призначення.

– СК07. Здатність розробляти програмне забезпечення відповідно до сформульованих вимог з урахуванням наявних ресурсів та обмежень.

У результаті вивчення дисципліни студент повинен забезпечити наступні **програмні результати навчання**:

– РН9. Розробляти алгоритмічне та програмне забезпечення для аналізу даних (включно з великими).

– РН10. Проектувати архітектурні рішення інформаційних та комп'ютерних систем різного призначення

Основними **завданнями** вивчення дисципліни є формування наступних **компетенцій магістра з комп'ютерної інженерії:**

– СК2. Здатність розробляти алгоритмічне та програмне забезпечення, компоненти комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем з використанням сучасних методів і мов програмування, а також засобів і систем автоматизації проектування.

– СК3. Здатність проектувати комп'ютерні системи та мережі з урахуванням цілей, обмежень, технічних, економічних та правових аспектів.

– СК6. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп’ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

У результаті вивчення дисципліни студент повинен забезпечити наступні **програмні результати навчання:**

– РН4. Застосовувати спеціалізовані концептуальні знання, що включають сучасні наукові здобутки у сфері комп’ютерної інженерії, необхідні для професійної діяльності, оригінального мислення та проведення досліджень, критичного осмислення проблем інформаційних технологій та на межі галузей знань.

– РН6. Аналізувати проблематику, ідентифікувати та формулювати конкретні проблеми, що потребують вирішення, обирати ефективні методи їх вирішення.

– РН8. Застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп’ютерних систем та мереж для вирішення складних задач комп’ютерної інженерії та дотичних проблем.

– РН13. Зрозуміло і недвозначно доносити власні знання, висновки та аргументацію з питань інформаційних технологій і дотичних міжгалузевих питань до фахівців і нефахівців, зокрема до осіб, які навчаються.

У результаті вивчення навчальної дисципліни студент повинен:

– **знати:** загальні відомості про атаки на програмне забезпечення та дані у комп'ютерних системах та мережах; міжмережеві екрани (фаєрволи, брандмауери); віртуальні приватні мережі (VPN); технології тунелювання; архітектура безпеки для IP (IPSec); протокол SSL/TLS; безпека бездротових з’єднань; системи виявлення вторгнень (IDS-системи); системи протидії вторгненням (IPS-системи); реалізація мережевої безпеки у організаціях;

безпека банківських електронних платіжних систем; електронна комерція: вимоги до безпеки; резервування інформації та компонентів інформаційних та комп'ютерних систем різного призначення; відновлення функціонування комп'ютерних систем та мереж після реалізації загроз, здійснення кібератак, збоїв та відмов різних класів та походження; моніторинг процесів функціонування комп'ютерних систем та мереж; система візуалізації та управління подіями (SIEM); аналіз подій; проектування, створення, супровід КСЗІ комп'ютерних систем та мереж; оцінка захищеності інформації в комп'ютерних системах та мережах; управління інформаційною та / або кібербезпекою комп'ютерних систем та мереж;

– вміти: програмно реалізовувати наступні проекти: Реалізація мережевого антивірусу; Реалізація міжмережевого екрану; Реалізація сніффера; Реалізація протоколу IPSec; Реалізація протоколу TLS/SSL; Реалізація системи виявлення вторгнень.

Контроль знань

Критерії оцінки іспиту:

оцінку «відмінно» (90-100 балів, А) – заслуговує студент, який:

– всебічно, систематично і глибоко володіє навчально-програмовим матеріалом;

– вміє самостійно виконувати завдання, передбачені програмою, використовує набуті знання і вміння у нестандартних ситуаціях;

– засвоїв основну і ознайомлений з додатковою літературою, яка рекомендована програмою;

– засвоїв взаємозв'язок основних понять дисципліни та усвідомлює їх значення для професії, яку він набуває;

– вільно висловлює власні думки, самостійно оцінює різноманітні життєві явища і факти, виявляючи особистісну позицію;

– самостійно визначає окремі цілі власної навчальної діяльності, виявив творчі здібності і використовує їх при вивченні навчально-програмового матеріалу, проявив нахил до наукової роботи.

оцінку « добре» (82-89 балів, B) – заслуговує студент, який:– повністю опанував і вільно (самостійно) володіє навчально-програмовим матеріалом, в тому числі застосовує його на практиці, має системні знання достатньому обсязі відповідно до навчально-програмового матеріалу, аргументовано використовує їх у різних ситуаціях;

– має здатність до самостійного пошуку інформації, а також до аналізу, постановки і розв'язування проблем професійного спрямування;

– під час відповіді допустив деякі неточності, які самостійно виправляє, добирає переконливі аргументи на підтвердження вивченого матеріалу;

оцінку «добре» (74-81 бал, C) заслуговує студент, який:

– в загальному роботу виконав, але відповідає на екзамені з певною кількістю помилок;

– вміє порівнювати, узагальнювати, систематизувати інформацію під керівництвом викладача, в цілому самостійно застосовувати на практиці, контролювати власну діяльність;

– опанував навчально-програмовий матеріал, успішно виконав завдання, передбачені програмою, засвоїв основну літературу, яка рекомендована програмою;

оцінку «задовільно» (64-73 бали, D) – заслуговує студент, який:

– знає основний навчально-програмовий матеріал в обсязі, необхідному для подальшого навчання і використання його у майбутній професії;

– виконує завдання, але при рішенні допускає значну кількість помилок;

– ознайомлений з основною літературою, яка рекомендована програмою;

– допускає на заняттях чи екзамені помилки при виконанні завдань, але під керівництвом викладача знаходить шляхи їх усунення.

оцінку «задовільно» (60-63 бали, E) – заслуговує студент, який:

– володіє основним навчально-програмовим матеріалом в обсязі, необхідному для подальшого навчання і використання його у майбутній професії, а виконання завдань задовольняє мінімальні критерії. Знання мають репродуктивний характер.

оцінка «незадовільно» (35-59 балів, FX) – виставляється студенту, який:

– виявив суттєві прогалини в знаннях основного програмового матеріалу, допустив принципові помилки у виконанні передбачених програмою завдань.

оцінку «незадовільно» (35 балів, F) – виставляється студенту, який:

– володіє навчальним матеріалом тільки на рівні елементарного розпізнавання і відтворення окремих фактів або не володіє зовсім;

– допускає грубі помилки при виконанні завдань, передбачених програмою;

– не може продовжувати навчання і не готовий до професійної діяльності після закінчення університету без повторного вивчення даної дисципліни.

При виставленні оцінки враховуються результати навчальної роботи студента протягом семестру

Критерії оцінки заліку:

– «зараховано» – студент має стійкі знання про основні поняття дисципліни, може сформулювати взаємозв'язки між поняттями.

– «незараховано» – студент має значні пропуски в знаннях, не може сформулювати взаємозв'язку між поняттями, що вивчаються в курсі, не має уявлення про більшість основних понять дисципліни, що вивчається.

Шкала оцінювання: національна та ЄКТС

<table>
<tr><td rowspan="2">Сума балів за всі види навчальної діяльності</td><td rowspan="2">Оцінка ЄКТС</td><td colspan="2">Оцінка за національною шкалою</td></tr>
<tr><td>для екзамену, курсового проекту (роботи), практики</td><td>для заліку</td></tr>
<tr><td>90-100</td><td>A</td><td>відмінно</td><td rowspan="5">зараховано</td></tr>
<tr><td>82-89</td><td>B</td><td rowspan="2">добре</td></tr>
<tr><td>74-81</td><td>C</td></tr>
<tr><td>64-73</td><td>D</td><td rowspan="2">задовільно</td></tr>
<tr><td>60-63</td><td>E</td></tr>
<tr><td>35-59</td><td>FX</td><td>незадовільно з можливістю повторного складання</td><td>не зараховано з можливістю повторного складання</td></tr>
<tr><td>1-34</td><td>F</td><td>незадовільно з обов’язковим повторним вивченням дисципліни</td><td>не зараховано з обов’язковим повторним вивченням дисципліни</td></tr>
</table>

Лабораторна робота № 1
Реалізація мережевого антивірусу
Частина 1. Частина 2

Ціль роботи: розібратися з будовою сучасних антивірусів і реалізувати свій антивірус.

Теоретична частина

Антивірусна програма (антивірус) – спеціалізована програма для виявлення комп'ютерних вірусів, а також небажаних (які вважаються шкідливими) програм взагалі й відновлення заражених (модифікованих) такими програмами файлів, а також для профілактики – запобігання зараження (модифікації) файлів або операційної системи шкідливим кодом.

На даний момент антивірусне програмне забезпечення розробляється, в основному, для ОС сімейства Windows від компанії Microsoft. Це викликано великою кількістю шкідливих програм саме під цю платформу (а це, у свою чергу, викликано великою популярністю цієї ОС, так само, як і більшою кількістю засобів розробки, у тому числі безкоштовних і навіть «інструкцій з написання вірусів»). У даний момент на ринок виходять продукти й для інших операційних систем, таких, приміром, як Linux і Mac OS X. Це викликано початком поширення комп'ютерних вірусів і під ці платформи, хоча UNIX-подібні системи традиційно користуються репутацією більше стійких до впливу шкідливих програм.

Крім ОС для настільних комп'ютерів і ноутбуків, також існують платформи й для мобільних пристроїв, такі, як Apple iOS, BlackBerry, Android, Windows Phone і ін. Користувачі пристроїв на даних ОС також піддані ризику зараження шкідливим програмним забезпеченням, тому деякі розроблювачі антивірусних програм випускають продукти й для таких пристроїв.

Робота антивірусу

Говорячи про системи Майкрософт, варто знати, що звичайно антивірус діє за схемою:

– пошук у базі даних антивірусного ПЗ сигнатур вірусів;

– якщо знайдено інфікований код у пам'яті (оперативній й/або постійній), запускається процес «карантину», і процес блокується;

– зареєстрована програма звичайно видаляє вірус, незареєстрованна просить реєстрації й залишає систему уразливою.

Бази антивірусів

Для використання антивірусів необхідні постійні відновлення так званих баз антивірусів. Вони являють собою інформацію про віруси – як їх знайти й знешкодити. Оскільки віруси пишуть часто, то необхідний постійний моніторинг активності вірусів у мережі. Для цього існують спеціальні мережі, які збирають відповідну інформацію. Після збору цієї інформації виробляється аналіз шкідливості вірусу, аналізується його код, поводження, і після цього встановлюються способи боротьби з ним. Найчастіше віруси запускаються разом з операційною системою. У такому випадку можна просто видалити рядок запуску вірусу з реєстру, і на цьому в простому випадку процес може закінчитися. Більш складні віруси використовують можливість зараження файлів. Наприклад, відомі випадки, як якісь навіть антивірусні програми, будучи зараженими, самі ставали причиною зараження інших чистих програм і файлів. Тому більше сучасні антивіруси мають можливість захисту своїх файлів від зміни й перевіряють їх на цілісність по спеціальному алгоритмі. Таким чином, віруси ускладнилися, як і ускладнилися способи боротьби з ними. Зараз можна побачити віруси, які займають уже не десятки кілобайт, а сотні, а часом можуть бути й розміром у парі мегабайт. Звичайно такі віруси пишуть у мовах програмування більше високого рівня, тому їх легше зупинити. Але як і раніше існує погроза від вірусів, написаних на низькорівневих машинних кодах на зразок асемблера. Складні віруси заражають операційну систему,

після чого вона стає уразливою й неробочою. На жаль, за прогнозами, у найближчому майбутньому робота антивірусних компаній сильно ускладниться у зв'язку з тим, що будуть сильніше поширюватися віруси із захистом від копіювання.

Функції:

Базовий захист:

– Захист від вірусів, троянських програм і хробаків.

– Захист декількох пристроїв.

– Перевірка посилань.

– Регулярні перевірки.

– Захист від мережевих атак.

– Контроль активності програм і захист від блокерів.

– Захист від шпигунських і рекламних програм.

– Перевірка файлів в автоматичному режимі й на вимогу.

– Перевірка поштових повідомлень (для будь-яких поштових клієнтів).

– Перевірка інтернет-трафіку (для будь-яких інтернет-браузерів).

– Захист інтернет-пейджерів (ICQ, MSN).

– Проактивний захист від нових шкідливих програм.

– Перевірка Java- і Visual Basic-скриптів.

– Захист від схованих битих посилань.

– Постійна перевірка файлів в автономному режимі.

– Постійний захист від фішингових сайтів.

– Захист від злому веб-камери.

– Синхронізація паролів на всіх пристроях.

– Резервне копіювання й шифрування даних.

– Батьківський контроль.

Запобігання погроз:

– Пошук уразливостей в ОС і встановленому ПЗ.

– Аналіз і усунення уразливостей у браузері.

– Блокування посилань на заражені сайти.

– Розпізнавання вірусів за способом їхнього упакування.

– Глобальний моніторинг погроз.

Відновлення системи й даних:

– Можливість установки програми на заражений комп'ютер.

– Функція самозахисту програми від вимикання або зупинки.

– Відновлення коректних налаштувань системи після видалення шкідливого ПЗ.

– Наявність інструментів для створення диска аварійного відновлення.

Захист конфіденційних даних:

– Блокування посилань на фішингові сайти.

– Захист від всіх видів кейлоггерів.

Зручність використання:

– Автоматичне налаштування програми в процесі установки.

– Готові рішення (для типових проблем).

– Наочне відображення результатів роботи програми.

– Інформативні діалогові вікна для прийняття користувачем обґрунтованих рішень.

– Можливість вибору між простим (автоматичним) і інтерактивним режимами роботи.

– Цілодобова технічна підтримка.

– Автоматичне відновлення баз.

Завдання:

Реалізувати мережевий антивірус. Розроблене програмне забезпечення повинне включати базу даних сигнатур, пошук по сигнатурах і реалізовувати 5-6 функцій антивірусу з перерахованих у теоретичній частині.

Лабораторна робота № 2
Реалізація міжмережевого екрану

Ціль роботи: Розібратися з будовою міжмережевих екранів й реалізувати свій міжмережевий екран.

Теоретична частина

Міжмережевий екран являє собою комплекс завдань щодо запобігання несанкціонованого доступу, пошкодження або викрадення даних, або іншого негативного впливу, який може вплинути на працездатність мережі.

Міжмережевий екран, його також називають фаєрвол (Від англ. Firewall) або брандмауер на шлюзі дозволяє забезпечити безпечний доступ користувачів до мережі Інтернет, при цьому захищаючи віддалене підключення до внутрішніх ресурсів. Міжмережевий екран переглядає через себе весь трафік, що проходить між сегментами мережі, і для кожного пакета реалізує рішення – пропускати або не пропускати. Гнучка система правил брандмауера дозволяє забороняти або дозволяти з'єднання за численними параметрами: адрес, мереж, протоколам і портів.

Методи контролю трафіку між локальної та зовнішньою мережею

– Фільтрація пакетів. Залежно від того чи задовольняє вхідний пакет зазначеним у фільтрах умовами він пропускається в мережу або відкидається.

– Stateful inspection. У цьому випадку здійснюється інспектування вхідного трафіку – один з найбільш передових способів реалізації Firewall. Під інспекцією мається на увазі аналіз не всього пакету, а лише його спеціальної ключової частини і в порівнянні із заздалегідь відомими значеннями з бази даних дозволених ресурсів. Такий метод забезпечує найбільшу продуктивність роботи Firewall і найменші затримки.

– Proxy-сервер. У даному випадку між локальної та зовнішньої мережами встановлюється додатковий пристрій proxy-сервер, який служить «воротами», через який повинен проходити весь вхідний і вихідний трафік.

Міжмережевий екран дозволяє настроювати фільтри, які відповідають за пропуск трафіку по:

– IP-адресі. Задавши якусь адресу або певний діапазон можна заборонити отримувати з них пакети, або навпаки дозволити доступ тільки з даних IP адрес.

– Порту. Фаєрвол може налаштувати точки доступу додатків до послуг мережі. Приміром, ftp використовує порт 21, а додатки для перегляду web-сторінок порт 80.

– Протоколу. Брандмауер може бути налаштований на пропуск даних лише якогось одного протоколу, або заборонити доступ з його використанням. Найчастіше тип протоколу може говорити про виконуваних завданнях, використовуваного ним програми та про набір параметрів захисту. У зв'язку з цим, доступ може бути налаштований тільки для роботи якого-небудь одного специфічного додатки і запобігти потенційно небезпечний доступ з використанням всіх інших протоколів.

– Доменному імені. У даному випадку фільтр забороняє або дозволяє з'єднання конкретних ресурсів. Це дозволяє заборонити доступ з небажаних сервісів і додатків мережі, або навпаки дозволити доступ тільки до них.

Для установки можуть застосовуватися й інші параметри для фільтрів, характерні для даної конкретної мережі, залежно від виконуваних у ній завдань.

Найчастіше міжмережевий екран використовується в комплексі з іншими засобами захисту, наприклад, антивірусне програмне забезпечення.

Принцип дії брандмауера

Брандмауер може бути виконаний:

– Апаратно. У такому випадку в ролі апаратного фаєрвола виступає маршрутизатор, який розташовується між комп'ютером і мережею Інтернет.

До фаєрвол може бути підключено кілька ПК і при цьому всі вони будуть захищені фаєрволом, який виступає частиною маршрутизатора.

– Програмно. Найбільш поширений тип брандмауера, який представляють собою спеціалізоване програмне забезпечення, яке користувач встановлює на свій ПК.

Навіть якщо підключений маршрутизатор з вбудованим міжмережевим екраном, додатково може бути встановлений програмний фаєрвол на кожен комп'ютер окремо. У такому випадку зловмисникові буде складніше проникнути в систему.

Більш детально інформація про міжмережеві екрани наведені у лекції №2.

Завдання:

Реалізувати міжмережевий екран. Розроблене програмне забезпечення повинне включати функції перераховані у теоретичній частині.

Міжмережевий екран повинен настроювати фільтри, які відповідають за пропуск трафіку за:

– IP-адресою.

– Портом.

– Протоколом.

– Доменним іменем.

Лабораторна робота № 3

Реалізація сніффера

Частина 1. Частина 2

Ціль роботи: Розібратися з будовою сніффера й реалізувати свій сніффер.

Теоретична частина

Аналізатор трафіку, або сніффер (від англ. to sniff – нюхати) – мережевий аналізатор трафіку, програма або програмно-апаратний пристрій, призначений для перехоплення й наступного аналізу, або тільки аналізу мережевого трафіку, призначеного для інших вузлів.

Сніффер може аналізувати тільки те, що проходить через його мережеву карту. Усередині одного сегмента мережі Ethernet всі пакети розсилаються всім машинам, через цього можливо перехоплювати чужу інформацію. Використання комутаторів (switch, switch-hub) і їхня грамотна конфігурація вже є захистом від прослуховування. Між сегментами інформація передається через комутатори. Комутація пакетів – форма передачі, при якій дані, розбиті на окремі пакети, можуть пересилатися з вихідного пункту в пункт призначення різними маршрутами. Тому якщо хтось в іншому сегменті посилає всередині нього які-небудь пакети, то у ваш сегмент комутатор ці дані не відправить.

Перехоплення трафіку може здійснюватися:

– звичайним «прослуховуванням» мережевого інтерфейсу (метод ефективний при використанні в сегменті концентраторів (хабів) замість комутаторів (світчів), у противному випадку метод малоефективний, оскільки на сніффер попадають лише окремі фрейми);

– підключенням сніффера в розрив каналу;

– відгалуженням (програмним або апаратним) трафіку й напрямком його копії на сніффер (Network tap);

– через аналіз побічних електромагнітних випромінювань і відновлення в такий спосіб що прослуховується трафіку;

– через атаку на канальному (2) (MAC-spoofing) або мережевому (3) рівні (IP-spoofing), що приводить до перенапрямку трафіку жертви або всього трафіку сегмента на сніффер з наступним поверненням трафіку в належну адресу.

Сніффери застосовуються як у конструктивних, так і в деструктивних цілях. Аналіз трафіку, який пройшов через сніффер, дозволяє:

– Виявити паразитний, вірусний і закільцьований трафік, наявність якого збільшує завантаження мережевого встаткування й каналів зв'язку (сніффери тут малоефективні; як правило, для цих цілей використовують збір різноманітної статистики серверами й активним мережевим устаткуванням і її наступний аналіз).

– Виявити в мережі шкідливе й несанкціоноване ПЗ, наприклад, мережеві сканери, флудери, троянські програми, клієнти пірінгових мереж і інші (це звичайно роблять за допомогою спеціалізованих сніфферів – моніторів мережевої активності).

– Перехопити будь-який незашифрований (а часом і зашифрований) користувальницький трафік з метою одержання паролів і іншої інформації.

– Локалізувати несправність мережі або помилку конфігурації мережевих агентів (для цієї цілі сніффери часто застосовуються системними адміністраторами)

Оскільки в «класичному» сніффері аналіз трафіку відбувається вручну, із застосуванням лише найпростіших засобів автоматизації (аналіз протоколів, відновлення TCP-потоку), то він підходить для аналізу лише невеликих його обсягів.

Знизити погрозу сніффингу пакетів можна за допомогою таких засобів, як:

– Автентифікація.

– Криптографія.

– Антисніффери.

– Інфраструктура, що комутирується.

Приклад простого сніффера під Windows

Ціль: написати програму, що буде захоплювати мережевий трафік (Ethernet, WiFi), що передається за протоколом IP.

Засоби: Visual Studio 2005 або вище.

Теорія

У цей момент переважна більшість сучасних інформаційних мереж базуються на фундаменті стека протоколів TCP/IP. Стек протоколів TCP/IP (англ. Transmission Control Protocol/Internet Protocol) – збірна назва для мережевих протоколів різних рівнів, використовуваних у мережах. IP – маршрутизуємий мережевий протокол, використовуваний для негарантованої доставки даних, поділюваних на так звані пакети (більше вірний термін – дейтаграма) від одного вузла мережі до іншого.

Особливий інтерес для нас представляють IP-пакети, призначені для передачі інформації. Це досить високий рівень мережевий OSI-моделі даних, коли можна абстрагуватися від пристрою й середовища передачі даних, оперуючи лише логічним поданням.

Зовсім логічною є та обставина, що рано або пізно повинні були з'явиться інструменти для перехоплення, контролю, обліку й аналізу мережевого трафіку. Такі засоби звичайно називаються аналізаторами трафіку, пакетними аналізаторами або сніфферами (від англ. to sniff – нюхати). Це – мережевий аналізатор трафіку, програма або програмно-апаратний пристрій, призначений для перехоплення й наступного аналізу, або тільки аналізу мережевого трафіку, призначеного для інших вузлів.

Практика

На даний момент створено досить багато програмного забезпечення для прослуховування трафіку. Найбільш відомий з них: Wireshark. Нас цікавить завдання перехоплення трафіку методом звичайного «прослуховування» мережевого інтерфейсу. Важливо розуміти, що ми не

збираємося займатися зломом і перехоплювати чужий трафік. Потрібно всього лише переглядати й аналізувати трафік, що проходить через наш хост.

Для чого це може знадобитися:

– Дивитися поточний потік трафіку через мережеве з'єднання (вхідний/вихідний/усього).

– Перенаправляти трафік для наступного аналізу на інший хост.

– Теоретично, можна спробувати застосувати його для злому WiFi-мережі.

На відміну від Wireshark, що базується на бібліотеці libpcap/WinPcap, наш аналізатор не буде використовувати цей драйвер. У нас взагалі не буде драйвера, і свій NDIS ми писати не збираємося. Про це можна прочитати в цьому топіці. Він буде просто пасивним спостерігачем, що використовує тільки бібліотеку WinSock. Використання драйвера в цьому випадку непотрібно.

Ключовим кроком у перетворенні простого мережевого додатка в мережевий аналізатор є перемикання мережевого інтерфейсу в режим прослуховування (promiscuous mode), що й дозволить йому одержувати пакети, адресовані іншим інтерфейсам у мережі. Цей режим змушує мережеву плату приймати всі кадри, поза залежністю від того, кому вони адресовані в мережі.

Починаючи з Windows 2000 (NT 5.0) створити програму для прослуховування сегмента мережі стало дуже просто, тому що її мережевий драйвер дозволяє перевести сокет у режим прийому всіх пакетів.

Включення нерозбірливого режиму

```
long flag = 1;
SOCKET socket;
#define SIO_RCVALL 0x98000001

ioctlsocket(socket, SIO_RCVALL, &RS_Flag);
```

Наша програма оперує IP-пакетами, і використовує бібліотеку Windows Sockets версії 2.2 і «сирі» сокети (raw sockets). Для того щоб одержати прямий доступ до IP-пакета, сокет потрібно створювати в такий спосіб:

Створення сирого сокета:

```
s = socket(AF_INET, SOCK_RAW, IPPROTO_IP);
```

Тут замість константи SOCK_STREAM (протокол TCP) або SOCK_DGRAM (протокол UDP), ми використовуємо значення SOCK_RAW. Загалом кажучи, робота з raw sockets цікава не тільки з погляду захвата трафіку. Фактично, ми одержуємо повний контроль за формуванням пакета. Вірніше, формуємо його вручну, що дозволяє, наприклад, послати специфічний ICMP-пакет...

Ідемо далі. Відомо, що IP-пакет складається із заголовка, службової інформації й, властиво, даних. Раджу заглянути сюди, щоб освіжить знання. Опишемо у вигляді структури IP-заголовок (спасибі відмінній статті на RSDN [3]):

Опис структури IP-пакета

```
typedef struct _IPHeader
{
  unsigned char  ver_len;          //   версія   й   довжина
заголовку
  unsigned char  tos;              // тип сервісу
  unsigned short length;           // довжина всього пакета
  unsigned short id;               // Id
  unsigned short flgs_offset;      // прапори й зсув
  unsigned char  ttl;              // час життя
  unsigned char  protocol;         // протокол
  unsigned short xsum;             // контрольна сума
  unsigned long  src;              // IP-адреса відправника
  unsigned long  dest;             // IP-адреса призначення
```

```
      unsigned short *params;          // параметри (до 320 біт)
      unsigned char  *data;            //    дані    (до    65535
октетів)
    }IPHeader;
```

Головна функція алгоритму прослуховування буде виглядати в такий спосіб:

Функція захвату одного пакета

```
    IPHeader* RS_Sniff()
    {
        IPHeader *hdr;
        int count = 0;
        count    =    recv(RS_SSocket,    (char*)&RS_Buffer[0],
sizeof(RS_Buffer), 0);
        if (count >= sizeof(IPHeader))
        {
            hdr = (LPIPHeader)malloc(MAX_PACKET_SIZE);
            memcpy(hdr, RS_Buffer, MAX_PACKET_SIZE);
            RS_UpdateNetStat(count, hdr);
            return hdr;
        }
        else
            return 0;
    }
```

Тут все просто: одержуємо порцію даних за допомогою стандартної функції socket-функції recv, а потім копіюємо їх у структуру типу IPHeader.

І, нарешті, запускаємо нескінченний цикл захвата пакетів:

Захоплюємо всі пакети, які потраплять на наш мережевий інтерфейс

```
while (true)
{
    IPHeader* hdr = RS_Sniff();
    // обробка IP-пакета
    if (hdr)
    {
        // друкуємо заголовок у консолі
    }
}
```

Тут і далі в деяких важливих функцій і змінних автор зробив префкис RS_ (від Raw Sockets).

У принципі, можна піти далі, і описати заголовки всіх наступних протоколів, що перебувають вище. Для цього необхідно аналізувати поле protocol у структурі IPHeader.

Подивитесь на приклад коду (так, там повинен бути switch), де відбувається розфарбовування заголовка залежно від того, який протокол має пакет, інкапсульований в IP:

```
/*
*  Виділення пакета кольором
*/
void ColorPacket(const IPHeader *h, const u_long haddr,
const u_long whost = 0)
{
    if (h-h->xsum)
        SetConsoleTextColor(0x17); // якщо пакет не порожній
    else
        SetConsoleTextColor(0x07); // порожній пакет

    if (haddr == h-h->src)
```

```
        {
            SetConsoleTextColor(BACKGROUND_BLUE                |
/*BACKGROUND_INTENSITY |*/
                FOREGROUND_RED   |   FOREGROUND_INTENSITY);   //
"рідний" пакет на віддачу
        }
        else if (haddr == h-h->dest)
        {
            SetConsoleTextColor(BACKGROUND_BLUE                |
/*BACKGROUND_INTENSITY |*/
                FOREGROUND_GREEN   |   FOREGROUND_INTENSITY);   //
"рідний" пакет на прийом
        }

        if  (h-h->protocol  ==  PROT_ICMP  ||  h-h->protocol  ==
PROT_IGMP)
        {
            SetConsoleTextColor(0x70); // ICMP-пакет
        }
        else  if(h-h->protocol  ==  PROT_IP  ||  h-h->protocol  ==
115)
        {
            SetConsoleTextColor(0x4F); // in-IP-пакет, L2TP
        }
        else if(h-h->protocol == 53 || h-h->protocol == 56)
        {
            SetConsoleTextColor(0x4C);    //    TLS,    IP    with
Encryption
        }

        if(whost == h-h->dest || whost == h-h->src)
        {
            SetConsoleTextColor(0x0A);
        }
    }
```

Для нашого навчального приклада цілком достатньо буде подивитися ip-адреси хостів, з яких і на які йде трафік, і порахувати його кількість в одиницю часу.

Для того, щоб відобразити дані IP-заголовка, необхідно реалізувати функцію перетворення заголовка (але не даних) дейтаграми в рядок. Як приклад реалізації, можна запропонувати такий варіант:

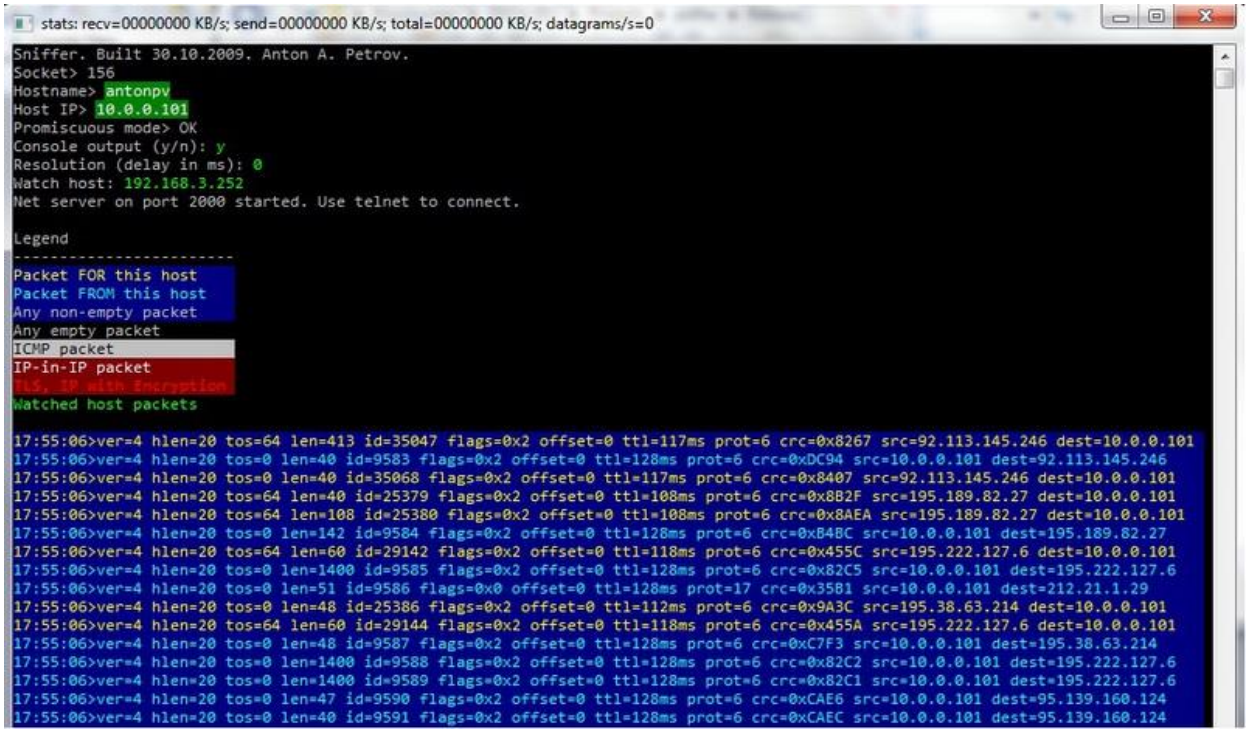
Перетворення IP-заголовка в рядок

```
inline char* iph2str(IPHeader *iph)
{
    const int BUF_SIZE = 1024;
    char *r = (char*)malloc(BUF_SIZE);
    memset((void*)r, 0, BUF_SIZE);

    sprintf(r, "ver=%d hlen=%d tos=%d len=%d id=%d
flags=0x%X offset=%d ttl=%dms prot=%d crc=0x%X src=%s dest=%s",

            BYTE_H(iph->ver_len),
            BYTE_L(iph->ver_len)*4,
            iph->tos,
            ntohs(iph->length),
            ntohs(iph->id),
            IP_FLAGS(ntohs(iph->flgs_offset)),
            IP_OFFSET(ntohs(iph->flgs_offset)),
            iph->ttl, iph->protocol,
            ntohs(iph->xsum), nethost2str(iph->src),
            nethost2str(iph->dest)
       );
    return r;
}
```

На підставі наведених вище базових відомостей, виходить от така невелика програма (ss, сокр. від англ. simple sniffer), що реалізує локальне прослуховування IP-трафіку. Інтерфейс її наведений нижче на рисунку.



Для компіляції буде досить навіть Visual Studio Express 2005.

Що в нас вийшло в підсумку:

– Сніффер працює в режимі користувача, однак вимагає привілею адміністратора.

– Пакети не фільтруються, відображаючись як є (можна додати налаштовуються фильтри).

– WiFi-трафік теж захоплюється (все залежить від конкретної моделі чипа), хоча є AirPcap, що чудово це вміє робити, але коштує грошей.

– Весь потік дейтаграм логується у файл.

– Програма працює як сервер на порту 2000. Можна підключитися за допомогою утиліти telnet до хосту й зробити моніторинг потоків трафіку. Кількість підключень обмежена двадцятьома.

Завдання лабораторної роботи: Реалізувати сніффер. Розроблене програмне забезпечення повинне включати функції перераховані у теоретичній частині.

Лабораторна робота № 4
Реалізація протоколу IPSec

Ціль роботи: Розібратися з будовою протоколу IPSec й реалізувати свій цей протокол.

Теоретична частина

Теоретичні відомості наведені у лекції №5.

Завдання лабораторної роботи: Реалізувати передачу даних за протоколом IPSec між хостами, **або** візуалізувати роботу протоколу IPSec.

Лабораторна робота № 5
Реалізація протоколу TLS/SSL

Ціль роботи: Розібратися з будовою протоколу TLS/SSL й реалізувати свій цей протокол.

Теоретична частина

Теоретичні відомості наведені у лекції №6.

Завдання лабораторної роботи: Реалізувати передачу даних за протоколом TLS/SSL між хостами, **або** візуалізувати роботу протоколу TLS/SSL.

Лабораторна робота № 6

Реалізація системи виявлення та протидії вторгненням

Частина 1. Частина 2

Ціль роботи: Розібратися з будовою системи виявлення вторгнень й реалізувати таку систему.

Теоретична частина

Теоретичні відомості наведені у лекції № 8, 9.

Завдання лабораторної роботи: Використовуючи програмне забезпечення, розроблене у ході виконання лабораторних робіт № 2 та № 3, реалізувати систему виявлення вторгнень у мережу або на хост, **або** візуалізувати роботу системи виявлення вторгнень.

Список використаної літератури

1. Смірнова Т.В., Смірнов О.А., Конопліцька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с. Режим доступу: http://dspace.kntu.kr.ua/jspui/handle/122456789/9799
2. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.
3. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.
4. Смірнов О.А., Стасєв Ю.В. Бараннік В.В. Захист інформації в автоматизованих системах управління. Навчальний посібник – Харків: ХУПС, 2015. – 264 с.
5. Смірнов О.А., Кавун С.В., Столбов В.Ф., Мелешко Є.В. Основи інформаційної безпеки. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. С.В. Кавуна. Гриф "Навчальний посібник" надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 26.04.2012 року № 1/11-5760. – Кіровоград: КНТУ 2012. – 442 с.
6. Смірнов О.А., Віхрова Л.Г., Осадчий С.І., Ковтун В.Ю., Мелешко Є.В. Основи захисту інформації. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія» та 8.050201 «Системна інженерія». За ред. О.А. Смірнова Гриф "Навчальний посібник" надано у відповідності з листом Міністерства освіти і науки України від 16.12.2010 року № 1/11-11486. – Кіровоград: КНТУ 2011. – 322 с.

7. Смірнов О.А., Кузнецов О.О., Євсеєв С.П., Мелешко Є.В., Король О.Г. Методи та алгоритми симетричної криптографії. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.О. Кузнецова. Гриф "Навчальний посібник" надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 26.04.2012 року № 1/11-5762. – Кіровоград: КНТУ 2012. – 315 с.

8. Захист інформації в автоматизованих системах управління : навчальний посібник / Уклад. І. А. Пількевич, Н. М. Лобанчикова, К. В. Молодецька. – Житомир : Вид-во ЖДУ ім. І. Франка, 2015. – 226 с.

9. Остапов С. Е. Технологія захисту інформації : навчальний посібник / С. Е. Остапов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2013. – 476 с.

10. Електронне урядування та електронна демократія: навч. посіб.: у 15 ч. / за заг. ред. А.І. Семенченка, В.М. Дрешпака. – К., 2017. Частина 13: Захист інформації в системах електронного урядування / [О.М. Хоша ба]. – К.: ФОП Москаленко О. М., 2017. – 72 с.

11. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 с.

12. Josh Armitage. Cloud Native Security Cookbook. O'Reilly Media. 2022. 516 с.

13. Alyssa Miller. Cybersecurity Career Guide. Manning Publications. 2022. 368 с.

14. Awais Rashid, Howard Chivers, George Danezis, Emil Lupu, Andrew Martin. CyBOK The Cyber Security Body of Knowledge. The National Cyber Security Centre. 2019. 854 с.

15. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 с.

16. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 с.

Допоміжна

17. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». CEUR Workshop Proceedings Volume 3156, 2022, Pages 390-399. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85133613188&origin=resultslist&sort=plf-f&featureToggles=FEATURE_NEW_DOC_DETAILS_EXPORT:1

18. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85118101973&origin=AuthorNamesList&txGid=9fba77a9424db54ff3b099e4400c22bb

19. Smirnova, T., Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., «Biometric authentication using convolutional neural networks». Lecture Notes in Networks and SystemsVolume 152, 2021, Pages 85-98. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85090914783&origin=resultslist

20. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., «Метод розрахунку критичності галузевих інформаційно-телекомунікаційних систем». Наукоємні технології № 2(54), 2022. С. 94-104. Режим доступу: https://jrnl.nau.edu.ua/index.php/SBT/article/view/16757 (Фахове видання. Категорія «Б»)

21. Смірнова Т.В., Гнатюк С.О., Юдін О.Ю., Сидоренко В.М., Жаксигулова Д.Д., «Експериментальне дослідження моделі розрахунку

кількісного критерію оцінювання захищеності інформаційно-телекомунікаційних систем критичної інфраструктури держави» Кібербезпека: освіта, наука, техніка. № 4(16). 2022. С. 6-18. Режим доступу: https://csecurity.kubg.edu.ua/index.php/journal/article/view/359/298 (Фахове видання. Категорія «Б»)

22. Смірнова Т.В., Якименко Н.М., Смірнов О.А., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022. Режим доступу: http://journals.khnu.km.ua/vestnik/?cat=65 (Фахове видання. Категорія «Б»)

23. Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., Смірнов О.А. «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89. Режим доступу: http://journals.nupp.edu.ua/sunz/article/view/2449/1918 (Фахове видання. Категорія «Б»)

24. Смірнова Т.В., Якименко Н.М., Улічев О.С., Конопліцька-Слободенюк О.К., Смірнов С.А., «Дослідження лінійних перетворень запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Кібербезпека: освіта, наука, техніка. № 3(15). С. 85-92. 2022. Режим доступу: https://csecurity.kubg.edu.ua/index.php/journal/article/view/337 (Фахове видання. Категорія «Б»)

25. Смірнова Т.В., Бурмак Ю.А., Улічев О.С., Усік П.С., Доренський О.П., «Стійка функція шифрування удосконаленого модуля криптографічного захисту інформації в інформаційно-комунікаційних

системах» Кібербезпека: освіта, наука, техніка. № 1(13). С. 183-201. 2021. Режим
доступу: https://csecurity.kubg.edu.ua/index.php/journal/article/view/346 (Фахове видання. Категорія «Б»)

26. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Бурмак Ю.А., Оспанова Д.М., «Удосконалений модуль криптографічного захисту інформації в сучасних інформаційно-комунікаційних системах та мережах». Кібербезпека: освіта, наука, техніка. № 2(14). С. 176-185. 2021. Режим
доступу: https://csecurity.kubg.edu.ua/index.php/journal/article/view/329 (Фахове видання. Категорія «Б»)

27. Смірнова Т.В., Смірнов О.А., Смірнов С.А., Поліщук Л.І., Конопліцька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87. https://doi.org/10.28925/2663-4023.2019.3.6387 Режим доступу: http://nbuv.gov.ua/UJRN/cest_2019_3_7 (Фахове видання).

28. Смірнова Т.В., Смірнов О.А., Смірнов С.А., Поліщук Л.І., Конопліцька-Слободенюк О.К.,. GERT-моделі технології хмарного антивірусного захисту. Кібербезпека: освіта, наука, техніка. – Том 2 № 2. – Київ: КУ ім.. Бориса Грінченка. – 2018. – С. 7-30. https://doi.org/10.28925/2663-4023.2018.2.730 Режим доступу: http://nbuv.gov.ua/UJRN/cest_2018_2_3 (Фахове видання).