ЗАХИСТ ТА ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 004.42:004.7:004.056.53 (045)

Куклінський М.В., Лукаш М.О., Головня Г.В.
Національний авіаційний університет

ЗАСІБ ПОПЕРЕДЖЕННЯ НЕСАНКЦІОНОВАНОГО ПРОНИКНЕННЯ ДО МЕРЕЖІ ПЕРЕДАЧІ ДАНИХ

Питанням безпеки мереж передачі даних завжди приділяється доволі пильна увага. Насамперед це викликано значним відставанням темпів розвитку засобів мережевої безпеки від рівнів і темпів розвитку самих мережевих технологій. Великий ріст кількості мережевих користувачів та їх вільний доступ до інформації, в основному із мережі Інтернет, яка сприяє здійсненню деструктивних дій, призводить до комплексного підходу щодо питань безпеки мережі. Хоч ці деструктивні дії не завжди робляться із зловмисними цілями, навіть їх спроби часто можуть привести до великих збитків, витоків інформації, тощо. У роботі описано алгоритм засобу, який дає змогу виявити несанкціоноване втручання до мережі передачі даних, та у комплексі з іншими програмно-апаратними засобами здатний вирішувати загальні задачі системи мережевої безпеки. Крім цього розглянуто та проаналізовано переваги та недоліки аналогічних засобів. Приведено лістинг коду скануючого модуля засобу, а також його загальна практична програмна реалізація.

Вопросам безопасности сетей передачи данных всегда уделяется достаточно пристальное внимание. Прежде всего, это вызвано значительным отставанием темпов развития средств сетевой безопасности от уровней и темпов развития самих сетевых технологий. Большой рост количества сетевых пользователей и их свободный доступ к информации, в основном из сети Интернет, которая способствует осуществлению деструктивных действий, приводит к комплексному подходу в вопросах безопасности сети. Хотя эти деструктивные действия не всегда делаются с вредоносными целями, даже их попытки часто могут привести к большим убыткам, утечкам информации, и тому подобное. В работе описан алгоритм средства, который позволяет определить несанкционированное вмешательство в сеть передачи данных, и в комплексе с другими программно-аппаратными средствами способен решать общие задачи системы сетевой безопасности. Кроме этого рассмотрены и проанализированы преимущества и недостатки аналогичных средств. Приведен листинг кода сканирующего модуля средства, а также его общая практическая программная реализация.

Security of the data networks has always been an object of great attention. Its importance caused by sufficient lag in development rate of levels of network security comparing to rates of development of network technologies. A large increase in the number of network users and their access to information, mainly via the Internet, which facilitates the implementation of destructive actions, results in a necessity of comprehensive approach to network security. Although these destructive actions have not always aimed to malicious purposes, even such attempts may often lead to large losses, information leakages, etc. The paper describes the algorithm tool that allows identifying an unauthorized access in the data network. This tool is capable to solve the common tasks of network security in complex cooperation with other software and hardware approaches. In addition, the advantages and disadvantages of similar means were examined and analyzed. A code listing of scanning module tool and its overall practical software implementation are presented.

Ключові слова: *мережа передачі даних, мережевий сканер, програмний засіб, мережева безпека.*

Вступ

Для своєчасного виявлення та попередження несанкціонованого вторгнення до мережі необхідно весь час її сканувати або прослуховувати. На даний час, засоби сканування мережі займають важливу роль в ІТ сфері. Перш за все ці засоби використовуються при налаштуванні мережі та мережевого обладнання, а також для пошуку несправних вузлів. Але у них є і інша сторона використання. Завдяки тому, що за допомогою даних засобів можна зібрати всю первинну інформацію про архітектуру мережі, типи мережевого устаткування, відкриті порти на мережевих комп'ютерах, тощо, їх дуже широко використовують у зловмисних цілях, що негативно впливає на надійність і безпеку мережі.

Постановка проблеми та формування мети

В цілому засоби для автоматизованого сканування мереж умовно можна розділити на дві групи: одні призначені для сканування IP та MAC-адрес, другі для сканування портів. Проте, такий розподіл дуже умовний, оскільки в переважній більшості мережеві сканери поєднують в собі обидві можливості.

Методи, які використовуються в засобах сканування, засновані на частковому моделюванні дій як зовнішніх зловмисників, які не мають авторизованих засобів доступу до системи, так і внутрішніх зловмисників, які мають певний рівень санкціонованого доступу. Тобто аналіз проводиться з позиції потенційного нападника з активним використанням усіх потенційних уразливостей мережі, які можуть виникати внаслідок неправильної конфігурації системи, відомих і невідомих дефектів апаратних засобів та програмного забезпечення, а також при оперативному відставанні процедурних та технічних контрзаходів, тощо. Тому розробка засобів які дозволять захистити мережу є доволі актуальною.

Загалом питання безпеки є лише одним із аспектів загальної надійності мережі. Хоч у розподіленій системі забезпечити належний рівень безпеки набагато складніше, ніж в централізованій є декілька факторів, які впливають на безпеку однаково для усіх. По перше мережеві повідомлення передаються по лініях зв'язку, саме на них можуть бути встановлені засоби прослуховування, особливо, якщо ці лінії проходять через загальнодоступні приміщення. Іншим вразливим місцем можуть стати залишені без нагляду персональні комп'ютери. Крім того, завжди є потенційна загроза злому захисту мережі від неавторизованих користувачів, якщо мережа має виходи в глобальні загальнодоступні мережі, тощо [1].

Найнебезпечнішим з точки зору прослуховування є перший фактор, так як по причині складності виявлення прослуховуючого пристрою на лінії, він передбачає довготривале перехоплення даних. Метою статті є підвищення здатності мережі захистити дані, які нею передаються, шляхом розробки засобу, який дозволить ідентифікувати спробу несанкціонованого підключення до її лінії передачі даних.

Аналіз популярних мережевих сканерів

Загалом мережеві сканери діляться на декілька груп в залежності від їх функціональних можливостей, починаючи від сканерів, які просто прослуховують мережу закінчуючи сканерами уразливості [2], які служать для здійснення діагностики і моніторингу мережних комп'ютерів. Для здійснення задачі ідентифікації несанкціонованого підключення достатньо сканера, який просто сканує або прослуховує мережу. Серед найбільш популярного на сьогодні програмного забезпечення, яке виконує ці функції можна виділити наступні:

Free IP Scanner – сканер портів і IP-адрес. Призначений для моніторингу та управління мережами, як системними адміністраторами, так і звичайними користувачами. Завдяки багатопотоковій технології сканування, програма моніторить сотні комп'ютерів одночасно. Вона сканує кожну IP-адресу мережі на працездатність, а потім, при необхідності, дозволяє отримати ім'я комп'ютера, визначити його MAC-адресу, дані NetBIOS, порти тощо.

Особливості Free IP Scanner:

– швидке і стабільне багатопоточне сканування IP-адрес;
– сканування за рівнями пріоритету;
– встановлення максимальної кількості потоків для сканування портів;
– відображення NetBIOS-інформації: ім'я комп'ютера, робочої групи/домену, MAC-адреси;
– збереження отриманої інформації в окремий текстовий файл, тощо [3].

MyLanViewer Network/IP Scanner – програма для сканування і моніторингу комп'ютерів в мережі з можливістю пошуку їх загальнодоступних файлів. Вона показує знайдені комп'ютери в зручному для перегляду вигляді, який містить інформацію про ім'я, IP-адресу, MAC-адресу, загальні ресурси та інші деталі комп'ютера. За допомогою неї можна слідкувати за комп'ютерами в мережі та отримувати повідомлення, коли стан одного з них зміниться. Також можна управляти спільними ресурсами комп'ютерів. Можливості MyLanViewer:

– багатопотокове сканування, яке забезпечує високу швидкість сканування;
– пошук комп'ютерів в мережі за допомогою: ICMP, ARP, NetBIOS, DNS;
– сканування NetBIOS, FTP і HTTP ресурсів;
– пошук файлів в загальних ресурсах за заданими умовами;
– збереження списків всіх раніше знайдених комп'ютерів в мережі;

– відображення, які комп'ютери включені, а які ні;
– оповіщення у разі виключення/включення обраних комп'ютерів;
– отримання імені комп'ютера за IP-адресою і назад;
– збереження звітів в HTML, TXT файлі, тощо [4].

CommView – дозволяє бачити список мережевих з'єднань, IP-статистику, а також досліджувати окремі пакети. CommView здійснює повний аналіз понад 100 поширених протоколів, IP-пакети декодуються аж до найнижчого протокольного рівня. Перехоплені пакети можуть бути збережені у файл для подальшого аналізу. Гнучка система фільтрів дозволяє відкидати непотрібні пакети або перехоплювати тільки потрібні пакети. До можливостей CommView входить:
– перехоплення інтернет-трафіку і/або трафіку локальної мережі;
– перегляд перехоплених і декодованих пакетів в реальному часі або в offline-режимі;
– перегляд докладної статистики IP-з'єднань: IP-адреси, порти, сесії тощо;
– реконструювання TCP-сесій і UDP-потоків.
– перегляд графіка розподілу протоколів, завантаження мережі, списків активних мережевих вузлів і їх статистику;
– генерування звітів, експортування та імпортування архівів зі збереженими пакетами;
– налаштування попереджень, які повідомляють про підозрілі пакети, високе завантаження мережі, тощо [5].

Advanced IP Scanner – це швидкий, надійний і простий у використанні сканер локальних мереж. Дана утиліта дозволяє користувачеві збирати різну інформацію про комп'ютери в мережі за лічені секунди. Можливий доступ до багатьох корисних функцій. Технологія багатопотокового сканування дозволяє здійснювати сканування сотень комп'ютерів в мережах класу B і C за лічені секунди навіть з повільним з'єднанням. Можливості програми:
– швидке і надійне багатопотокове сканування IP-адрес з можливістю зазначення кількості потоків для балансу швидкості сканування і завантаження процесора;
– віддалене виключення або включення комп'ютера;
– групові операції функцій, які підтримуються Advanced IP Scanner;
– надання зручного доступу до знайдених загальних папок, HTTP і FTP серверів;
– збереження і завантаження списку знайдених та просканованих комп'ютерів, тощо [6].

Інші, менш популярні, мережеві сканери за функціональними можливостями практично нічим не відрізняються від розглянутих, за винятком додатків, які мають дуже вузьке призначення. Зазвичай вони підтримують одну, або декілька функцій, які напряму залежать від спектру та середовища їх використання. Наприклад додатки, які:
– сканують лише порти або лише IP-адреси заданого діапазону;
– здійснюють пошук конкретних мережевих ресурсів;
– прослуховують або перехоплюють певний трафік;
– працюють в мережах певної топології або архітектури, тощо.

Виявлення потенційного проникнення до мережі

Як видно з аналізу мережевих сканерів вони в переважній більшості лише сканують мережу. Несанкціоноване проникнення сприймається ними як доповнення мережі новим вузлом і відносяться вони до цього вузла, як до легального користувача мережі. Причому сканування зазвичай проходить за виділеним або прописаним наперед діапазоном адрес і лише одноразово за викликом додатку. Тобто сканування не проводиться весь час, а лише тоді, коли це потрібно адміністратору, або локальному користувачеві. При кожному такому скануванні список виявлених комп’ютерів доповнюється тими, які виявляються вперше, і при великій їх кількості задача виявлення нового несанкціонованого користувача стає важковирішуваною, або практично невирішуваною.

Лише невелика частка сканерів здатна прослуховувати мережу увесь час. Такі сканери здатні перехоплювати пакети, які передаються по мережі і збирати інформацію про усіх учасників мережі із цих пакетів. Проте майже усі вони платні та потребують встановлення додаткового програмного забезпечення, яке напряму залежить від роботи сканера та навпаки.

Зважаючи на виявлені недоліки існуючих мережевих сканерів в їх платному та повсякденному використанні, було розроблено засіб, який дозволяє виявити факт несанкціонованого підключення та

прослуховування мережі користувачами, які не являються легальними вузлами мережі. Додаток об'єднав у собі функції мережевих сканерів, які можуть сканувати мережу для виявлення всіх її учасників, та сканерів, які весь час прослуховують мережу перехоплюючи її пакети.

Алгоритм роботи засобу показаний на рис.1.

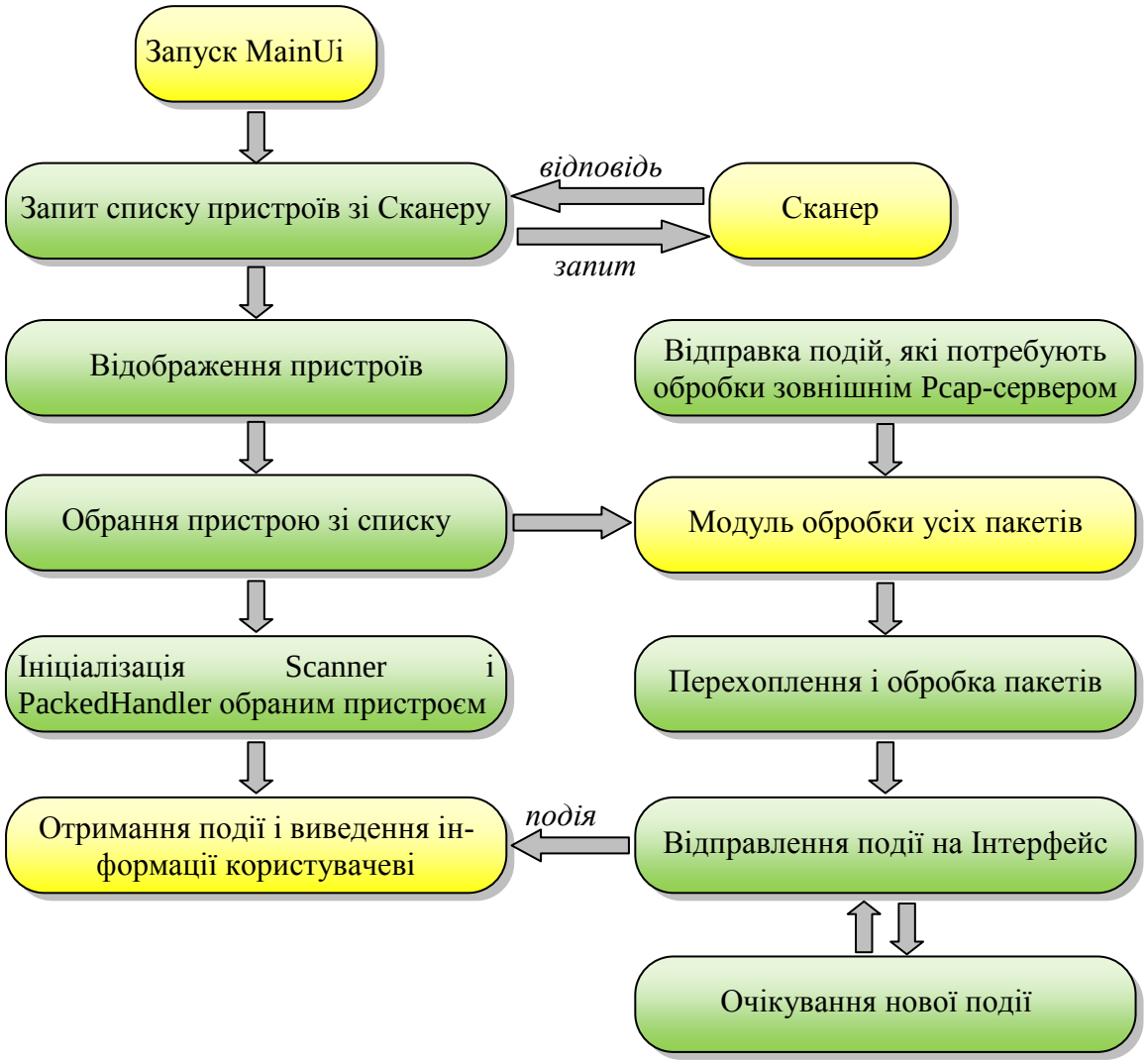

Рис.1. Схема алгоритму засобу

Також на відміну від аналогічних додатків код розробленого засобу є відкритим, а сам засіб є безкоштовним. Лістинг коду програми сканера (скануючого модуля) написаного на мові програмування Java представлений у додатку.

Програмна реалізація засобу

Слід зазначити, що для виявлення засобом нелегального користувача, йому необхідна інформація про усіх легальних користувачів та вузлів мережі. Для цих потреб у ньому передбачена база даних у якій містяться відомості про адреси мереж, підмереж, окремих комп'ютерів та вузлів, отриманих за допомогою спеціальних запитів перенесення зони.

Крім цього, так як засіб працює за принципом клієнт-серверної технології, для початку його роботи на стаціонарному комп'ютері необхідно встановити віртуальний сервер. Слід зазначити, що для кожної операційної системи існують свої віртуальні сервери, а їх встановлення не становить особливих труднощів. Якщо в наявності є мережевий, або будь-який сервер, то додаток можна встановити одразу на нього (рис.2).

Рис.2. Запуск та робота віртуального сервера

Після першого запуску засіб пропонує обрати одну з мережевих карт (пристроїв), які встановлені на комп'ютері чи сервері (рис. 3).

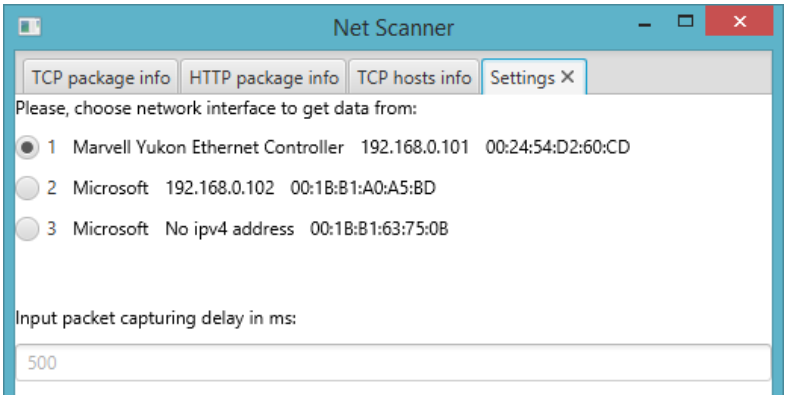

Рис.3. Вибір пристрою на комп'ютері

Після чого натиском кнопки «Start catching packets» починається процес сканування мережі та перехоплення пакетів (рис.4).

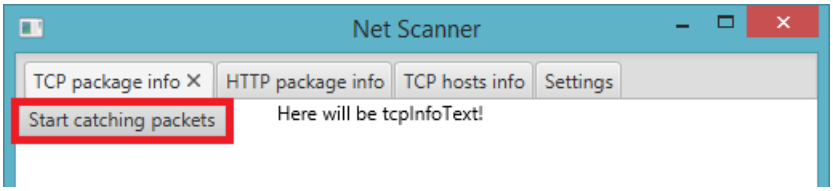

Рис.4. Старт засобу

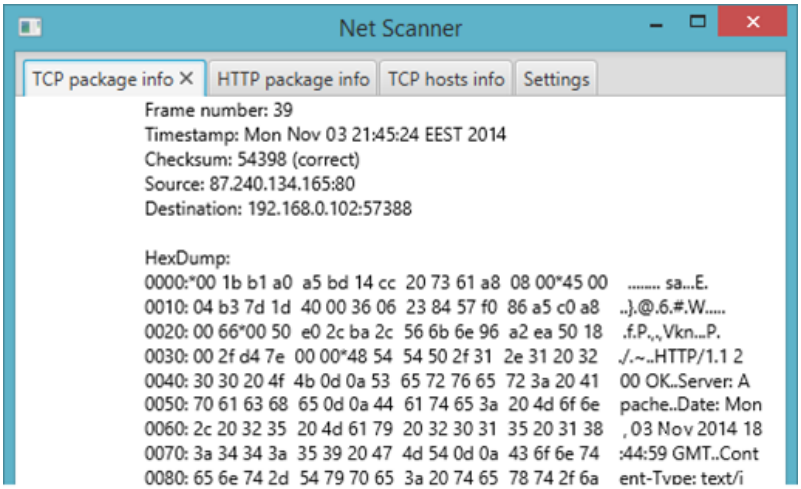

Рис.5. Результат роботи сканеру

Перехоплюючи пакети засіб розшифровує їх Header. Крім цього він слідкує за Timestamp, коли пакет вже відправлений, Cheksum, та повний Hex Dump пакету. Програма також виявляє до якого порту підключений той чи інший комп'ютер і яку кількість пакетів цей комп'ютер передав (рис.5.).

Крім цього засіб дозволяє бачити інформацію про пакети даних (рис.6.), а також кількість відправлених та прийнятих вузлами мережі пакетів (рис.7).

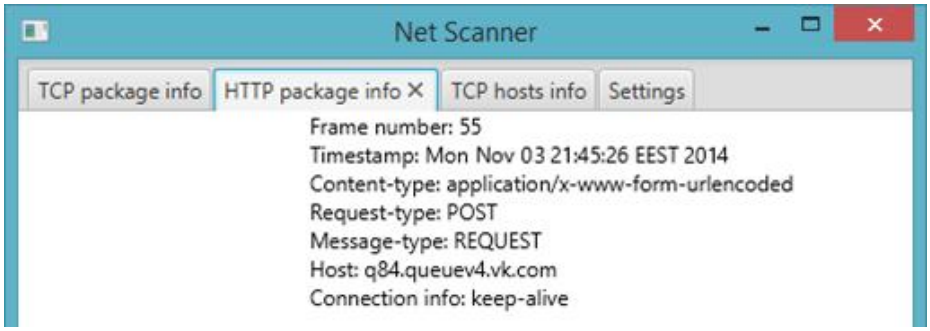

Рис.6. Інформація про пакети даних

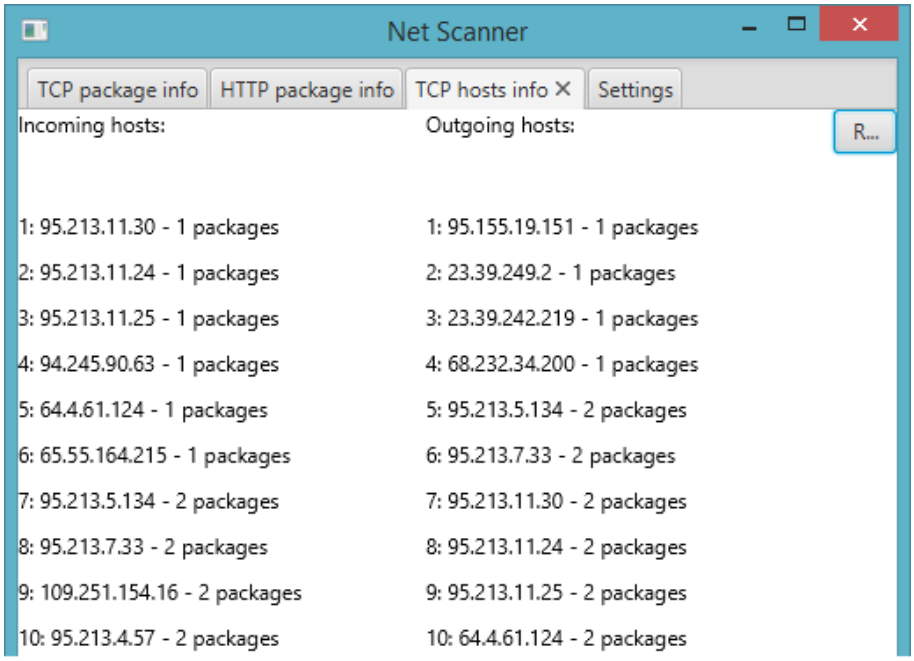

Рис.7. Хост інформація

Тому, коли в мережу, яка знаходиться під скануванням розробленого засобу, підключиться вузол, інформація про якого відсутня у базі, засобом одразу буде виявлено факт несанкціонованого втручання до мережі.

Висновки

Розроблений засіб направлений на підвищення загального рівня безпеки мережі передачі даних. Мінімальний інтерфейс та достатня продуктивність, дозволяє використовувати його, як у великих компаніях так і в малих. Крім цього його можна застосовувати і в локальних домашніх мережах.

Проте, розроблений засіб дозволяє лише виявити факт несанкціонованого втручання у мережу, тому його слід розглядати лише як частину загальної системи безпеки мережі. Дії щодо запобігання втручання та миттєвого блокування мережевого трафіку передбачаються лише за можливості інтегрування та взаємодії розробленого засобу з іншими програмно-апаратними засобами загальної мережевої системи безпеки такими, як мережеві екрани, сканери уразливості, тощо.

Список використаних джерел

1. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы : Учебник для вузов / В.Г. Олифер, Н.А. Олифер. – 4-е изд. – С-Пб.: Питер, 2010. – 944 с.
2. Wikipedia / Вільна онлайн-енциклопедія [Електронний ресурс] https://uk.wikipedia.org/wiki/Сканери_уразливості
3. Компанія Eusing Software / Розробник програмного забезпечення [Електронний ресурс] http://www.eusing.com/ipscan/free_ip_scanner.htm
4. Компанія S.K. Software / Розробник програмного забезпечення [Електронний ресурс] http://www.mylanviewer.com/network-ip-scanner.html
5. Компанія TamoSoft / Розробник програмного забезпечення [Електронний ресурс] http://www.tamos.com/products/commview/
6. Компанія Famatech / Розробник програмного забезпечення [Електронний ресурс] http://www.advanced-ip-scanner.com/

Додаток

```
package com.gmail.netscanner.scanner;

import com.gmail.netscanner.exceptions.DeviceAccessException;
import com.gmail.netscanner.exceptions.GetDeviceException;
import com.gmail.netscanner.utils.TcpSourceDestinationTuple;
import com.gmail.netscanner.utils.Utils;
import org.jnetpcap.Pcap;
import org.jnetpcap.PcapIf;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Objects;
import java.util.stream.Collectors;

/**
 * Created by Lukash Maksym.
 */
public class Scanner {

  private static int snaplen = 64 * 1024;           // Capture all packets, no trucation
  private static int flags = Pcap.MODE_PROMISCUOUS; // capture all packets
  private static int timeout = 10 * 1000;           // 10 seconds in millis

  private static volatile Pcap pcap;
  private static StringBuilder errorBuffer = new StringBuilder();

  //to diversify incoming and outgoing messages
  private static String ipv4Address;

  private static final Map<String, Integer> outgoingHosts = new HashMap<>();
  private static final Map<String, Integer> incomingHosts = new HashMap<>();

  private Scanner() {
  }
  //util method for finding devices
  public static List<PcapIf> findAllDevs() {
          List<PcapIf> alldevs = new ArrayList<>();
```

```
                int r = Pcap.findAllDevs(alldevs, errorBuffer);

                if (r != Pcap.OK || alldevs.isEmpty()) {
                        System.err.printf("Can't read list of devices, error is %s", errorBuffe-
r.toString());
                        throw new GetDeviceException(errorBuffer.toString());
                }
                return alldevs;
        }
        //double-checked locking in singleton done right, with volatile modifier
        // and local instance for speeding up
        public static Pcap initialize(PcapIf device) {
                Pcap localPcap = pcap;
                if (localPcap == null) {
                        synchronized (Scanner.class) {
                                if (localPcap == null) {

        /*************************************************************************
**
                                        * We open up the selected device

        *************************************************************************
        */
                                        pcap = localPcap = Pcap.openLive(device.getName(), snaplen,
flags, timeout, errorBuffer);
                                }
                        }
                }
                if (pcap == null) {
                        System.err.printf("Error while opening device for capture: " + errorBuffer);
                        throw new DeviceAccessException(errorBuffer.toString());
                }

                ipv4Address = Utils.getIpv4Address(device);
                return pcap;
        }
        //this method should be called after initializing
        public static Pcap getPcap() {
                return pcap;
        }

        /**
         * ************************************************************************
         * Last thing to do is close the pcap handle
         * <p>
         * This method should be called from UI before exiting application (or when we want to stop
capturing packets)
         * ************************************************************************
         */
        public static void closePcap() {
                pcap.close();
        }
        public static void addHost(TcpSourceDestinationTuple sourceDestinationTuple) {
                if (Objects.equals(sourceDestinationTuple.getSource(), ipv4Address)) {
```

```
                addOutgoingHost(sourceDestinationTuple.getDestination());
            } else {
                addIncomingHost(sourceDestinationTuple.getSource());
            }
        }
        private static void addIncomingHost(String host) {
            //if host occured first time - add to map with 1, other case - increase number of occurences
            incomingHosts.put(host, incomingHosts.getOrDefault(host, 0) + 1);
        }
        private static void addOutgoingHost(String host) {
            //if host occured first time - add to map with 1, other case - increase number of occurences
            outgoingHosts.put(host, incomingHosts.getOrDefault(host, 0) + 1);
        }
        // get outgoing sorted by amount of packets
        public static List<String> getOutgoingHosts() {
            return sortAndFormatMap(outgoingHosts);
        }
        // get incoming sorted by amount of packets
        public static List<String> getIncomingHosts() {
            return sortAndFormatMap(incomingHosts);
        }
        private static List<String> sortAndFormatMap(Map<String, Integer> hosts) {
            return hosts.entrySet().
                    stream().
                    sorted((e1, e2) -> e1.getValue().compareTo(e2.getValue())).
                    map(entry -> entry.getKey() + " - " + entry.getValue() + " packages").
                    collect(Collectors.toList());
        }
    }
```

Інформація про авторів:

Куклінський Максим Володимирович – к.т.н., доцент кафедри комп'ютерних інформаційних техпелогій Інституту комп'ютерних інформаційних технологій Національного авіаційного університету. Наукові інтереси: Internet та інженерія програмного забезпечення.
E-mail: maximum_inc@ua.fm

Лукаш Микола – студент 4-го курсу кафедри комп'ютерних інформаційних техпелогій Інституту комп'ютерних інформаційних технологій Національного авіаційного університету. Наукові інтереси: комп'ютерні інформаційні технології.
E-mail: masdiz@mail.ru

Головня Галина – студентка 4-го курсу кафедри комп'ютерних інформаційних техпелогій Інституту комп'ютерних інформаційних технологій Національного авіаційного університету. Наукові інтереси: комп'ютерні інформаційні технології.
E-mail: golovnia.galyna@gmail.com