ISSN 1681-7710

МІНІСТЕРСТВО ОБОРОНИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПОВІТРЯНИХ СИЛ ІМЕНІ ІВАНА КОЖЕДУБА

Системи обробки інформації

Щоквартальне наукове видання

Випуск 1 (151)

Заснований у березні 1996 року

У збірнику відображено результати досліджень з розробки нових інформаційних технологій як для рішення традиційних задач збору, обробки та відображення даних, так і для побудови систем обробки інформації у різних проблемних галузях. Збірник призначений для наукових працівників, викладачів, докторантів, ад'юнктів, аспірантів, а також курсантів та студентів старших курсів відповідних спеціальностей.

Засновник і видавець:
Харківський національний університет Повітряних Сил імені Івана Кожедуба

61023, м. Харків-23,
вул. Сумська, 77/79, НЦ ПС

Телефон:
+38 (057) 704-91-97
+38 (067) 998-02-70

E-mail редколегії:
red@hups.mil.gov.ua
red.hnups@gmail.com

Інформаційний сайт:
www.hups.mil.gov.ua

Харків • 2017

Затверджений до друку вченою радою Харківського національного університету Повітряних Сил імені Івана Кожедуба (протокол від 19 грудня 2017 року № 19)

Занесений до “Переліку наукових фахових видань України, в яких можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора і кандидата наук” (технічні та військові науки), затверджено наказом Міністерства освіти і науки України від 29.12.2014 № 1528 (із змінами від 22.12.2016 № 1604)

Свідоцтво про державну реєстрацію друкованого засобу масової інформації КВ № 22357 – 12257ПР від 30.09.2016 р.

Усі статті, що публікуються у журналі, проходять обов’язкове рецензування, яке здійснюється за відкритою та анонімною формою як для авторів, так і для рецензентів

За достовірність викладених фактів, цитат та інших відомостей відповідальність несе автор

Інформаційний сайт видання: *www.hups.mil.gov.ua.*
Реферативна інформація зберігається у загальнодержавній реферативній базі даних „Україніка наукова” та публікується у відповідних тематичних серіях УРЖ „Джерело”.

Видання індексується міжнародними бібліометричними та наукометричними базами даних:
Academic Resource Index (EC), Google Scholar (США), Scientific Indexed Service (США), Index Copernicus (Польща), Open Academic Journals Index (EC), General Impact Factor (EC).

Наукометричні показники:
ICV (Index Copernicus Value) = 60.92

З М І С Т

ОБРОБКА ІНФОРМАЦІЇ В СКЛАДНИХ ТЕХНІЧНИХ СИСТЕМАХ

ОБРОБКА ІНФОРМАЦІЇ В СКЛАДНИХ ОРГАНІЗАЦІЙНИХ СИСТЕМАХ

МАТЕМАТИЧНІ МОДЕЛІ ТА МЕТОДИ

ІНФОКОМУНІКАЦІЙНІ СИСТЕМИ

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ, ПІДПРИЄМСТВІ ТА ВИРОБНИЦТВІ

C O N T E N T S

INFORMATION PROCESSING IN COMPLEX ENGINEERING SYSTEMS

INFORMATION PROCESSING IN COMPLEX ORGANIZATIONAL SYSTEMS

MATHEMATICAL MODELS AND METHODS

INFOCOMMUNICATION SYSTEMS

INFORMATION TECHNOLOGIES IN ECONOMICS, ON AN ENTERPRISE AND A FACTORY

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В МЕДИЦИНІ ТА БІОЛОГІЇ

ЗАХИСТ ІНФОРМАЦІЇ ТА КІБЕРНЕТИЧНА БЕЗПЕКА

МЕТРОЛОГІЯ, ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

INFORMATION TECHNOLOGIES IN MEDICINE AND BIOLOGY

INFORMATION SECURITY AND CYBERSECURITY

METROLOGY, INFORMATION AND MEASUREMENT TECHNOLOGIES AND SYSTEMS

УДК 004.056.53

С.В. Шаров[1], Д.В. Лубко[2]

[1] *Мелітопольський державний педагогічний університет імені Богдана Хмельницького, Мелітополь*
[2] *Таврійський державний агротехнологічний університет, Мелітополь*

РОЗРОБКА ТА ВИКОРИСТАННЯ СНІФЕРУ ЯК ЗАСОБУ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ TCP З'ЄДНАНЬ

У статті повідомляється про розробку програмного засобу для перехоплення та аналізу вихідних TCP з'єднань (сніферу), описуються етапи його розробки та вимоги до використання, подається коротка інструкція користувача. Функціональні можливості розробленого програмного засобу дозволяють автоматизувати окремі функціональні обов'язки системного адміністратора. Також розроблений сніфер надає можливість контролювати весь процес передачі даних в довільному 32-розрядному програмному забезпеченні. Програмний засіб має зручний та простий користувальницький інтерфейс, який дозволяє використовувати його навіть початківцям.

Ключові слова: *кібербезпека, сніфер, захист мережевих даних, TCP з'єднання.*

Вступ

Постановка проблеми. Сьогодні кібербезпеку повною мірою можна вважати важливим аспектом діяльності будь-якого суспільства. В умовах існування інформаційного світового середовища, через складність та трудомісткість більшості процесів і методів захисту інформації та комп'ютерних систем від несанкціонованого доступу, а також вразливість інформаційних систем до певних дій становить значну проблему для різних користувачів (кінцевих користувачів, підприємств, держави тощо). Крім того, незважаючи на те, що більшість організацій вдосконалюють системи інформаційного захисту, кіберзлочинці продовжують знаходити шляхи їхнього обходу, здійснюючи руйнівну діяльність.

Дана проблема посилюється невизначеністю нормативно-законодавчої бази, яка б визначала розділення обов'язків щодо забезпечення кібербезпеки на різних державних рівнях. Як зазначають дослідники, в Україні відсутні нормативні документи, які б описували загрози для нашої держави саме у кіберпросторі; відсутні загальнонаціональні міжвідомчі координаційні структури, що могли б узгоджувати та координувати діяльність різних силових відомств під час розслідування злочинів у кіберпросторі [3, с. 77]. Крім того, Україна ще залишається уразливою через повсюдне впровадження закордонних програмних продуктів та використання матеріально-технічної бази іноземного виробництва. Захист інформації передбачає досягнення та збереження властивостей безпеки в ресурсах користувачів, що спрямовані на запобігання відповідним кіберзагрозам. Як наслідок, розробка якісних програмних продуктів та цифрового обладнання національного виробництва дозволить збільшити рівень кібербезпеки нашої держави.

Сьогодні на ринку програмного забезпечення існує багато системних та прикладних програмних засобів, призначених для забезпечення інформаційної безпеки. Із розвитком і розповсюдженням Інтернету достатньо часто стали використовуватися спеціальні програми, сніфери, які призначені для перехоплення та аналізу мережевого трафіку.

Аналіз останніх досліджень і публікацій. Питаннями комп'ютерних мереж, їх моніторингом та аналізом, а також розробкою відповідних спеціалізованих аналізаторів мережевого трафіку у свій час займалися такі вчені, дослідники та програмісти як Мічел Лукас (Michael W. Lucas), Олівер Хекман (Oliver M. Heckmann), Ед Вілсон (Ed Wilson), Н. Мендкович, Г. Конахович, В. Чуприн та інші. Серед вітчизняних науковців проблемою кібербезпеки займаються Д. Дубов, О. Баранов та ін.

Метою статті є повідомлення про розробку аналізатору мережевого трафіку для перехоплення та аналізу вихідних TCP з'єднань, опис його функціональних можливостей.

Виклад основного матеріалу

Сьогодні спостерігається тенденція на активне використання інформаційних ресурсів у всіх сферах економічної, соціальної, освітньої, політичної та інших видів діяльності. Поряд із стрімким зростанням апаратних потужностей поширюються випадки несанкціонованого збирання, використання, поширення персональних даних та важливої інформації, шахрайства у мережі Інтернет. Як наслідок, кіберзлочинність стала транснаціональною, здатною завдати значної шкоди інтересам особи, суспільства та держави [1, с. 30]. У таких умовах актуальності набувають розробка та використання ефективних ме-

ханізмів забезпечення кібербезпеки держави та вирішення супутніх проблем [5, с. 120].

На жаль, таких проблемних питань існує декілька, і в першу чергу це стосується уніфікації поняття кібербезпека. На сьогодні існують різні визначення цього терміну, а саме: це стан захищеності життєво важливих інтересів особистості, суспільства і держави в умовах використання комп'ютерних систем та/або телекомунікаційних мереж, за яким мінімізується завдання їм шкоди [2, с. 61]; це стан спроможності людини, суспільства й держави запобігати та уникати спрямованого, насамперед несвідомого, негативного впливу інформації [12, с. 30].

Внаслідок того, що на рівні національних та міжнародних документів визначення кібербезпеки значно різняться, відрізняються і підходи до забезпечення кібербезпеки [2, с. 57]. Як зазначає Д. Дубов, терміни із префіксом «кібер» фактично не зустрічаються в нормативно-правових документах України. Натомість, частіше використовується поняття «інформаційна безпека» та інші терміни, що тісно пов'язані з «інформаційним» складником [5, с. 120].

Інша проблема, пов'язана із забезпеченням інформаційної безпеки, пов'язана з розробкою відповідних механізмів, обладнання та програмних засобів для захисту від несанкціонованого доступу та використання інформації. Одним із таких засобів є сніфер (sniffers), або аналізатор трафіку, під яким розуміється програма або програмно-апаратний пристрій, призначений для перехоплення та подальшого аналізу (або тільки аналізу) мережевого трафіку.

Сніфери працюють на рівні мережевого адаптера NIC, тому вони аналізують тільки ту інформацію, що проходить через мережеву карту. В свою чергу, перехоплення трафіку здійснюється за допомогою декількох методів:

– звичайним «прослуховуванням» мережевого інтерфейсу. Даний метод ефективний при використанні в сегменті концентраторів замість комутаторів;

– підключенням сніфера до розриву каналу;

– відгалуженням (програмним або апаратним) трафіку та перенаправлення його копії на сніфер;

– аналізом побічних електромагнітних випромінювань та відновленням трафіку.

Аналіз мережевого трафіку, що пройшов через сніфер, дозволяє:

1) Виявити вірусний та/або закільцований трафік, який збільшує завантаження мережевого обладнання та каналів зв'язку.

2) Виявити шкідливе та несанкціоноване програмне забезпечення, наприклад, мережеві сканери, флудери, троянські програми, клієнти пірінгових мереж та інші.

3) Перехопити будь-який незашифрований (а деколи і зашифрований) мережевий трафік з метою отримання паролів або іншої інформації.

4) Локалізувати несправність мережі або помилку конфігурації мережевих агентів.

Як можна побачити, сніфери можна використовувати як для виявлення кібербезпеки, так і в руйнівних цілях.

Оскільки в «класичному» сніфері аналіз трафіку відбувається вручну, із застосуванням лише простих засобів автоматизації (аналіз протоколів, відновлення TCP-потоку), то він підходить для аналізу лише невеликих об'ємів даних.

Всі сніфери можна умовно розділити на дві категорії: сніфери, що підтримують завантаження та роботу з командного рядка; сніфери, що мають графічний інтерфейс. Також є гібридні варіанти, об'єднують обидва режими роботи [8, с. 94]. Прикладів існуючих аналізаторів мережевого трафіку для різних операційних систем доволі багато: Wireshark, Iris, WinDump, Sniffit, Ultra Network, Sniffer, Analyzer, Packetyzer, IPDump2, Ferret, LanGrabber Ethereal Network Analyzer Wireshark та інші. Всі ці програмні засоби (або пакети програм) схожі за функціоналом, але відрізняються протоколами, які вони підтримують, глибиною аналізу перехоплених пакетів, можливостями з налаштування фільтрів, сумісністю з іншими програмними продуктами [10], користувальницьким інтерфейсом та можливостями генерації статистичних звітів [9].

Як зазначає Г. Андрощук, українські користувачі значною мірою схильні до заражень через відмову від оновлення програмного забезпечення або використання піратських копій програм [1, с. 32]. Положення погіршується ще тим, що частина користувачів користуються застарілими операційними системами, наприклад Windows XP, у яких захист від сучасних кібератак майже відсутній.

З огляду на це, ми поставили перед собою завдання розробити програмний засіб для перехоплення та аналізу вихідних TCP з'єднань у 32-розрядних додатках. Цей програмний продукт повинен бути швидким, надійним, безкоштовним та стабільним, а також забезпечувати можливість зберігання отриманих даних. Розробка програмного засобу виконувалася за такими етапам:

1. Аналіз наочної області програмного засобу, а саме функціональні можливості існуючих аналізаторів мережевого трафіку, принцип роботи TCP з'єднань.

2. Вибір формату збереження даних, за допомогою якого буде зберігатися інформація, яка потрібна для роботи програмного засобу.

3. Вибір інструментального середовища для розробки сніферу.

4. Створення модулів програмного засобу.

5. Тестування, перевірка працездатності програмного продукту в реальних умовах, виправлення помилок.

Розглянемо основні вимоги до розробленого програмного засобу.

Вимоги до оформлення інтерфейсу. Графічний інтерфейс програмного засобу повинен бути легким в користуванні, гнучким та інтуїтивно зрозумілим, мати можливість масштабування, однаково виглядати на комп'ютерах з різною локалізацією. Мова інтерфейсу – англійська. Доступ до основних операцій має бути реалізований через головне меню. Операції над пакетами даних повинні бути реалізовані через контекстне меню. Головне вікно повинне мати мінімальну кількість елементів управління (кнопок, пунктів меню, полів введення).

У сніфері повинна бути реалізована можливість перегляду вмісту пакета як у текстовому, так і в 16-річному вигляді. Вікно завантаження нового процесу повинно мати можливість вибору виконуваного файлу за допомогою діалогового вікна або введення шляху до нього вручну. Також потрібна можливість завдання параметрів завантаження процесу та їх видалення. Вікно підключення до активного процесу повинно містити список процесів та блок керуючих кнопок. Список процесів потрібно реалізувати у вигляді таблиці, яка складається з двох колонок. В першій колонці міститься цифровий ідентифікатор програми та іконка. Якщо програма не має іконки – слід використати іконку за замовчуванням. У другій колонці міститься ім'я виконуваного файлу процесу.

Вікно ручної відправки пакету даних повинно містити дані про вміст пакету та блок керування відправкою даних. Повинна бути реалізована функціональність відправки пакету декілька разів підряд з заданим інтервалом. Вікно пошуку повинно мати інтуїтивно зрозумілий інтерфейс з можливістю вибору методу пошуку (текстовий, за допомогою регулярного виразу або 16-ричної послідовності).

Вимоги до програмного та технічного забезпечення.

Основною мовою розробки програмного засобу була обрана мова програмування високого рівня C++, так як вона є компільованою статично типізованою мовою загального призначення, яка підтримує об'єктно-орієнтоване програмування [7]. Для розробки графічного інтерфейсу була використана кросплатформна бібліотека розробки програмного забезпечення Qt [4]. У якості скриптової мови була використана мова програмування Lua [6, с. 11].

Для функціонування програмного засобу необхідне наступне програмне та мінімальне апаратне забезпечення: операційна система – Windows XP; наявність у ОС Microsoft Visual C++ 2010 Redistributable package (x86); ОС повинна бути тільки з 32-бітною архітектурою; процесор – Intel Pentium IV 1 Ghz або більш потужний; оперативна пам'ять – 2 Гб RAM або більше; жорсткий диск – 120 Гб або більше.

Програмні аспекти розробки сніферу.

Розробка програмного засобу для аналізу вихідних TCP з'єднань передбачала розробку dll-модуля, який перенаправляє усі нові TCP з'єднання програмного засобу, що аналізується, до основного модуля. Перед початком розробки цього модуля слід було вирішити такі питання: Як реалізувати метод перехоплення функцій ws2_32.connect та ws2_32.WSAConnect? Як реалізувати метод завантаження бібліотеки в цільовий процес? Як реалізувати протокол обміну інформацією з основним модулем програмного засобу?

Всього відомо чотири основних метода перехоплення виклику цільової функції, розташованої у DLL. Перший з них полягає у заміні вказівника на функцію у таблиці експорту бібліотеки. Другий передбачає заміну вказівника на функцію у таблиці імпорту головного модуля програми. Суть третього методу полягає у заміні декількох перших інструкції (або однієї, якщо її розмір >= 5 байт) на інструкцію безумовного переходу за адресою (jmp long addr), параметром якої є адреса функції обробника. Четвертий метод частіше всього використовують у тих випадках, коли структура коду функції не дозволяє замінити її перші інструкції на п'яти-байтову інструкцію jmp long [11]. При розробці програмного засобу був використаний третій варіант перехоплення функцій як найбільш універсальний та маючий найменшу кількість недоліків.

Для завантаження проксіфікуючої бібліотеки до адресного простору був використаний такий метод: цільовий процес відкривається за допомогою функції kernel32.OpenProcess. Після цього в ньому виділяється область пам'ять з правами доступу PAGE_EXECUTE_READ, в неї за допомогою функції kernel32.WriteProcessMemory записується код завантаження dll, та завантажується потік, який виконує цей програмний код.

Протоколом обміну інформацією з основним модулем програмного засобу був обраний SOCKS версії 5, причому була реалізована лише та частина протоколу, що необхідна для обробки вихідних TCP з'єднань. Це дало змогу перехоплювати трафік частини клієнтських мережевих програм (тих, які підтримують роботу через проміжний SOCKS сервер) без використання проксіфіцирующої бібліотеки, що дуже корисно в деяких випадках.

Основний модуль програмного засобу складається з п'яти основних частин. Загальна схема програмного засобу подана на рис. 1.

Розглянемо найбільш значущі моменти у розробці основного модуля програмного засобу.

Доступ до більшості функцій був реалізований за допомогою головного меню програми, але функції, які напряму працюють з конкретним пакетом (наприклад функція «Close connection» – яка реалі-

зує можливість примусового розриву з'єднання) було вирішено винести до контекстного меню дерева переданих даних.

На цьому етапі розробки програмного засобу було з'ясовано, що в програмному засобі «Qt Designer», який використовувався для розробки форм, не реалізована можливість створення та редагування контекстного меню, що, на нашу думку, є невеликим, але неприємним недоліком платформи Qt.

Рис. 1. Загальна схема розробленого сніферу

Враховуючи той факт, що структура та вигляд меню не змінюються під час виконання, було вирішено створювати об'єкти контекстного меню при ініціалізації основного вікна програмного засобу за допомогою процедури initContextMenu.

Далі була створена функція-обробник (в термінології Qt – слот) onPacketListContextMenu (QPointp) сигналу customContextMenuRequested, який генерується по кліку правої кнопки миші. Дана функція в процесі роботи виводить на екран побудоване при ініціалізації вікна контекстне меню. Зв'язування слоту із сигналом було виконано штатно – за допомогою редактора слотів/сигналів програмного засобу «Qt Designer».

Журнал подій реалізовано за допомогою віджету QTextBrowser. Його перевагами є дуже швидка робота, невеликі накладні витрати на збереження тексту та відсутність мерехтіння при додаванні великої кількості записів у малий проміжок часу. Інтерфейс функції додавання нових записів до журналу подій було вирішено реалізувати аналогічно до функції стандартної бібліотеки printf. Цей підхід дозволяє з легкістю форматувати записи журналу подій, без написання надмірного програмного коду.

Перегляд вмісту перехопленого пакету даних було вирішено реалізувати в текстовому та 16-ричному виглядах. Перший, реалізований за допомогою стандартного віджету QPlainText, дозволяє переглядати пакет даних у вигляді тексту, що дуже корисно при аналізі текстових протоколів передачі даних. Другий режим перегляду був реалізований за допомогою віджету QHexEdit2, який вільно поширюється автором за ліцензією GNU Lesser General Public License [13].

Вбудовування інтерпретатора мови програмування Lua можна розділити на три основні етапи:

Перший етап передбачає написання класу керування інтерпретатором, який виконується в окремому потоці. Завдяки тому, що платформа Qt надає можливість дуже просто реалізувати роботу з багатьма потоками та має вбудований функціонал передачі даних між ними за допомогою сигналів, цей етап зайняв відносно небагато часу. Для цього було реалізовано два класи – luaEngineWorker, який інкапсулює весь низькорівневий код керування інтерпретатором та керуючий клас luaEngine, який реалізує інтерфейс між потоками та керує виконанням потоку інтерпретатора.

Другий етап складався з написання класів, які реалізують інтерфейс між основним кодом та інтерпретатором мови програмування Lua. При виконанні цього етапу були написані класи, що надають повний доступ до сховища переданих даних та дозволяють керувати перехопленими з'єднаннями. Для можливості повноцінного керування процесом передачі даних була реалізована система подій, обробники якої можуть довільно керувати процесом передачі даних у перехоплених з'єднаннях.

Останній етап передбачав інтеграцію інтерпретатора мови Lua з програмним засобом за допомогою інструментарію SWIG [14]. Такий підхід звільняє програміста від написання величезної кількості інтерфейсного коду та дозволило значно скоротити час розробки та налагодження програмного засобу.

Для використання розробленого програмного засобу для аналізу мережевого трафіку наведемо коротку інструкцію користувача.

Робота з програмним засобом (виконуваний файл ReqPacketTool.exe) здійснюється за допомогою пунктів головного меню, контекстного меню дерева переданих даних та елементів керування (кнопок та опцій). Крім того, для автоматизації виконуваних дій даний програмний засіб може приймати параме-

три командного рядка, які можна комбінувати у довільному вигляді. На рис. 2 наведено проведений аналіз мережного трафіку головної сторінки сайту Вікіпедії за адресою https://uk.wikipedia.org.

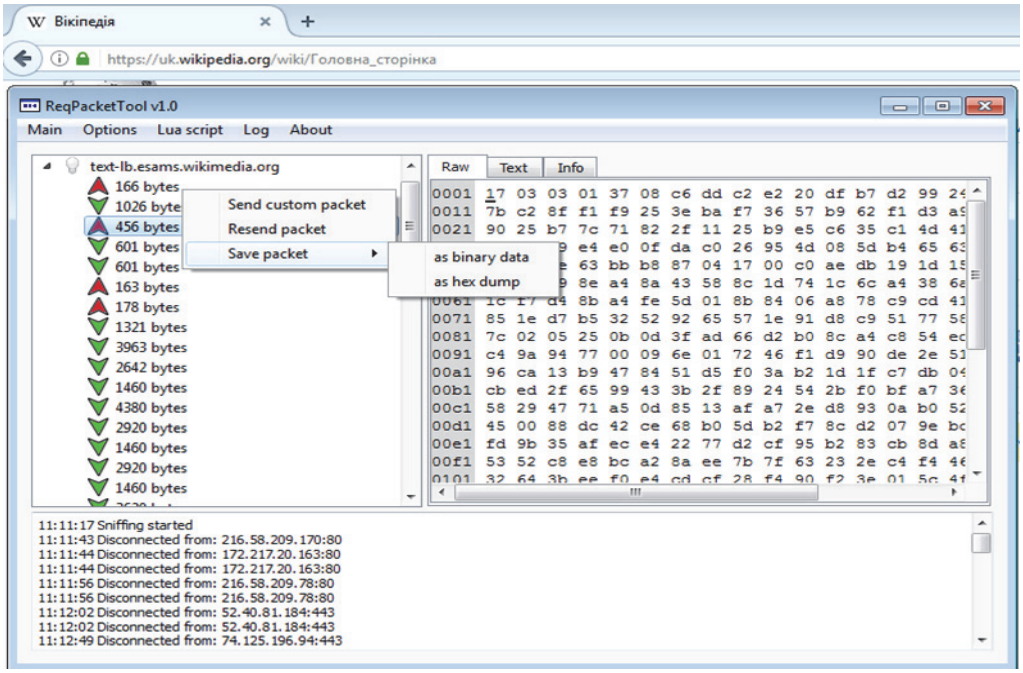

Рис. 2. Інтерфейс головного вікна програмного засобу

Головне меню містить декілька розділів, які, у свою чергу, складаються з підпунктів.

До розділу «Main» входять наступні пункти:

– пункт «Start process» дозволяє користувачу завантажити для аналізу довільний програмний засіб. Для цього він повинен вручну або за допомогою діалогового вікна (кнопка «. . .») вибрати основний виконуваний файл цільового програмного засобу, задати усі необхідні параметри завантаження (при їх наявності) та натиснути на кнопку «Start»;

– пункт «Inject to process» дозволяє користувачу перехопити всі нові з'єднання працюючого процесу. Після активізації цього пункту з'являється вікно вибору процесу, яке містить список усіх процесів операційної системи. Користувач повинен виділити потрібний процес та натиснути на кнопку «Inject». Слід зауважити, що за допомогою розробленого програмного засобу неможливо перехопити з'єднання процесів, які запущені користувачами з рівнем прав вищим, ніж у користувача, який запустив даний сніфер;

– пункт «Pause sniffing/Start sniffing» дозволяє користувачу при необхідності призупинити накопичування перехоплених даних та продовжити цей процес у будь-який момент;

– пункт «Clear captured packet list» дозволяє при необхідності очистити дерево перехоплених даних;

– пункт «Find» дозволяє користувачу вести пошук необхідної інформації у перехоплених даних. Вікно пошуку має три основні режими роботи, а саме: пошук за текстом (який є чутливим до регістру символів), пошук за допомогою регулярних виразів та пошук 16-ричних послідовностей. Для зручності користувача у режимі пошуку 16-річних послідовностей реалізована можливість пошуку за шаблоном. Для цього всі октети даних, які мають невизначене значення, помічаються символом '?'. На приклад, запит «FF ??» знайде усі двобайтові послідовності, які починаються з коду FF.

– пункт «Load packets queue» дозволяє завантажити раніше експортовані дані;

– пункт «Save packets queue» дозволяє експортувати перехоплені дані до файлу для збереження, або подальшої обробки.

Вікно ручної відправки пакету даних, яке з'являється після вибору пункту контекстного меню дерева переданих даних «Send custom packet», дозволяє користувачу відправити пакет даних, сформований вручну. У цьому вікні користувач може змінити вміст та розмір пакету даних, що відправляється, а також напрям його відправлення. При необхідності відправити пакет даних кілька разів підряд можна задати кількість пакетів даних та інтервал між їх відправкою.

Розділ «Options» містить пункт «Configure», який дозволяє змінювати конфігурацію програмного засобу. За допомогою цього вікна можливо виконання наступний опцій: відключити ведення журналу подій; отримувати доменні імена видалених серверів; змінювати кількість стовпців даних у 16-

ричному режимі перегляду даних або номер порта вбудованого SOCKS серверу (для зміни цього параметра потребується перезавантаження).

Розділ «Lua script» містить наступні пункти:

– підменю «Start script» дозволяє користувачу виконувати довільно обраний скрипт або один з п'яти скриптів, використаних останнім часом;

– пункт «Stop script» дозволяє користувачу примусово припинити роботу виконуваного скрипта.

За допомогою розділу головного меню «Log» можна зберегти до файлу вміст журналу подій та очистити вміст журналу подій.

Для прискорення виконання дій та для зручності користувача програмного засобу були призначені клавіші швидкого виклику для основних операцій головного меню.

Висновки

Отже, у процесі дослідження були проаналізовані функціональні властивості різних аналізаторів мережевих пакетів, на основі чого було сформульовано технічне завдання на розробку програмного засобу. На основі технічного завдання було розроблено програмний засіб для перехоплення та аналізу вихідних TCP з'єднань та написана інструкція користувача. Розроблений програмний засіб дає змогу виконати багато завдань, які часто доводиться вирішувати системним адміністраторам. Наприклад, виявити окремі причини збоїв клієнтського мережевого програмного забезпечення, здійснити перевірку програм на наявність підозрілої активності, перехоплювати передану інформацію тощо. Також він надає можливість контролювати за необхідністю весь процес передачі даних в довільному 32-розрядному програмному забезпеченні. Саме тому ми вважаємо розроблений програмний засіб ефективним при вирішення багатьох складних завдань, з якими зустрічаються системні адміністратори.

У подальших дослідженнях ми плануємо збільшити функціональні можливості програмного засобу, а саме модифікувати його для використання у мережевому програмному забезпеченні з 64-бітною архітектурою та додати україномовний інтерфейс.

Список літератури

1. Андрощук Г.О. Кібербезпека: тенденції в світі та Україні / Г.О. Андрощук // Кібербезпека та інтелектуальна власність: проблеми правового забезпечення: матеріали Міжнародної науково-практичної конференції (21 квітня 2017 р., Київ). – К.: Вид-во «Політехніка», 2017. – С. 30-36.

2. Баранов О.А. Про тлумачення та визначення поняття «кібербезпека» / О.А. Баранов // Правова інформатика. – 2014. – №. 2. – С. 54-62.

3. Бабич Є.Ю. Забезпечення кібербезпеки в Україні / Є.Ю. Бабич // Актуальні задачі та досягнення у галузі кібербезпеки: матеріали Всеукраїнської науково-практичної конференції (23-25 листопада 2016 р, Кропивницький). – Кропивницький: КНТУ, 2016. – С. 77-78.

4. Боровский А. Qt 4.7+. Практическое программирование на C++ / А. Боровский. – СПб.: БХВ–Петербург, 2011. – 488 с.

5. Дубов Д.В. Стратегічні аспекти кібербезпеки України / Д.В. Дубов // Стратегічні пріоритети. – 2013. – № 4. – С. 119-127.

6. Иерузалимски Р. Программирование на языке Lua / Р. Иерузалимски. – М.: ДМК Пресс, 2014. – 382 с.

7. Культин Н.Б. Microsoft Visual C++ в задачах и примерах / Н.Б. Культин. – СПб.: БХВ-Петербург, 2014. – 272 с.

8. Левчук А.С. Огляд програмних засобів для аналізу мережевого трафіку / А.С. Левчук // Інформаційні технології в освіті та науці: зб. наук. праць. – Мелітополь: Вид-во МДПУ ім. Б. Хмельницького, 2015. – Вип. 7. – С. 93-97.

9. Основы мониторинга. Самые популярные снифферы [Електронний ресурс]. – Режим доступу до ресурсу: www.askit.ru/custom/win2003_sec/m1/01_01_01_sniffering_basics.htm.

10. Пахомов С. Анализаторы сетевых пакетов [Електронний ресурс] / С. Пахомов. – Режим доступу до ресурсу: compress.ru/Article.aspx?id=16244.

11. Филимонов И. Методы перехвата API–вызовов в Win32 [Електронний ресурс] / И. Филимонов. – Режим доступу до ресурсу: rsdn.org/article/baseserv/apicallsintercepting.xml.

12. Фурашев В. Ключові аспекти проекту закону України «Про безпеку інформації» / В. Фурашев // Віче. – 2012. – №.6. – С. 29-30.

13. GNU Lesser General Public License [Електронний ресурс]. – Режим доступу до ресурсу: www.gnu.org/licenses/lgpl-3.0.en.html.

14. SWIG-3.0 Documentation [Електронний ресурс]. – Режим доступу до ресурсу: www.swig.org/Doc3.0/SWIGDocumentation.pdf.

References

1. Androshchuk, H.O. (2017), "Kiberbezpeka: tendentsii v sviti ta Ukraini" [Cybersecurity: trends in the world and Ukraine], *International Scientific Conference: Cybersecurity and intellectual property: problems of legal enforcement, April 21,* Politekhnika, Kiev, pp. 30-36.

2. Baranov, O.A. (2014), "Pro tlumachennia ta vyznachennia poniattia «kiberbezpeka»" [On the interpretation and definition of «cybersecurity»], *Legal informatics*, No. 2, pp. 54-62.

3. Babych, Ie.Iu. (2016), "Zabezpechennia kiberbezpeky v Ukraini" [Ensuring cybersecurity in Ukraine], *All-Ukrainian Scientific Conference: Current tasks and achievements in the field of cybersecurity*, November 23-25, KNTU, Kropivnitsky, pp. 77-78.

4. Borovskij, A. (2011), "*Qt 4.7+. Prakticheskoe programmirovanie na C++*" [*Qt 4.7+. Practical programming in C ++*], BHV–Peterburg, Saint Petersburg, 488 p.

5. Dubov, D.V. (2013), "Stratehichni aspekty kiberbezpeky Ukrainy" [Strategic aspects of cybersecurity of Ukraine], *Strategic priorities*, No. 4, pp. 119-127.

6. Ieruzalimski, R. (2014), "*Programmirovanie na yazyke Lua*" [*Programming in Lua language*], DMK Press, Moscow, 382 p.

7. Kul'tin, N.B. (2014), "*Microsoft Visual C++ v zadachah i primerah*" [*Microsoft Visual C ++ in tasks and examples*], BHV–Peterburg, Saint Petersburg, 272 p.

8. Levchuk, A.S. (2015), "Ohliad prohramnykh zasobiv dlia analizu merezhevoho trafiku" [Overview of software for analyzing network traffic], *Information technology in education and science*, Vol. 7, pp. 93-97.

9. "*Osnovy monitoringa. Samye populyarnye sniffery*" [*Basics of monitoring. Most popular sniffers*],: www.askit.ru/custom/win2003_sec/m1/01_01_01_sniffering_basics.htm (accessed 25 August 2016).

10. Pahomov, S. "*Analizatory setevyh paketov*" [*Network packet analyzers*], compress.ru/Article.aspx?id=16244 (accessed 25 August 2016).

11. Filimonov, I. "*Metody perekhvata API–vyzovov v Win32*" [*Methods of intercepting API-calls in Win32*], rsdn.org/article/baseserv/apicallsintercepting.xml (accessed 25 August 2016).

12. Furashev, V. (2012), "Kliuchovi aspekty proektu zakonu Ukrainy «Pro bezpeku informatsii»" [Key aspects of the draft law of Ukraine "On Information Security"], *The veche*, No. 6, pp. 29-30.

13. *GNU Lesser General Public License*, www.gnu.org/licenses/lgpl-3.0.en.html (accessed 25 August 2016).

14. *SWIG-3.0 Documentation*, available at: www.swig.org/Doc3.0/SWIGDocumentation.pdf (accessed 25 August 2016).

Надійшла до редколегії 11.08.2017
Схвалена до друку 16.11.2017

Відомості про авторів:

Шаров Сергій Володимирович
кандидат педагогічних наук доцент
доцент кафедри Мелітопольського державного педагогічного університету ім. Богдана Хмельницького, Мелітополь, Україна
https://orcid.org/0000-0001-5732-9980
e-mail:sharov@mdpu.org.ua

Лубко Дмитро Вікторович
кандидат технічних наук доцент
доцент кафедри Таврійського державного агротехнологічного університету,
Мелітополь, Україна
https://orcid.org/0000-0002-2506-4145
e-mail: di75ma@gmail.com

Information about the authors:

Sharov Sergiy
Candidate of Science Associate Professor
Senior Lecturer of the Department of Melitopol State pedagogical university named after Bogdan Khmelnitsky,
Melitopol, Ukraine
https://orcid.org/0000-0001-5732-9980
e-mail:sharov@mdpu.org.ua

Lubko Dmitro
Candidate of Sciences, Associate Professor
Senior Lecturer of the Department of
Tavria State Agrotechnological University,
Melitopol, Ukraine
https://orcid.org/0000-0002-2506-4145
e-mail: di75ma@gmail.com

РАЗРАБОТКА И ИСПОЛЬЗОВАНИЕ СНИФЕРА КАК СРЕДСТВА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ TCP СОЕДИНЕНИЙ

С.В. Шаров, Д.В. Лубко

В статье сообщается о разработке программного средства для перехвата и анализа исходных TCP соединений (снифера), описываются этапы его разработки и требования к использованию, дается краткая инструкция пользователя. Функциональные возможности разработанного программного средства позволяют автоматизировать отдельные функциональные обязанности системного администратора. Также разработанный снифер позволяет контролировать весь процесс передачи данных в произвольном 32-разрядном программном обеспечении. Программное средство имеет удобный и простой пользовательский интерфейс, который позволяет использовать его даже начинающим.

Ключевые слова: кибербезопасность, снифер, защита сетевых данных, TCP соединения.

THE DEVELOPMENT AND USAGE OF THE SNIFER AS A SAFETY METHOD OF TCP-CONNECTIVITY

S. Sharov, D. Lubko

The article focuses on the development of a software tool for intercepting and analyzing outbound TCP connections as one of the means of providing information security in the Internet. It is noted that cybersecurity is one of the main issues of the state level, as the loss of important information can lead to negative economic, political consequences, etc. There are some problems concerning the providing of cybersecurity in Ukraine that concern ambiguity in the definition of the term «cybersecurity», the use of unlicensed software, etc. The article describes the steps of software development, requirements for its use, a brief user's guide is provided.

There are also requirements for designing the interface of the software, the software aspects of the development of the sniffer; the functionalities of the developed software tool are described in the form of items and sub-points of the main and context menu. The functional capabilities of the development of the sniffer allow you to automate the individual functional responsibilities of the system administrator. It also provides the ability to control the whole process of data transfer in arbitrary software. It is noted that the software can intercept the input data of 32-bit applications running under the Windows XP operating system. The software has convenient and easy user interface that allows it to be used even by beginners.

***Keywords**: cybersecurity, sniffer, network data protection, TCP connection.*

Алфавітний покажчик

НАУКОВЕ ВИДАННЯ

СИСТЕМИ ОБРОБКИ ІНФОРМАЦІЇ

ЗБІРНИК НАУКОВИХ ПРАЦЬ
Випуск 5 (151)

Відповідальний за випуск *Г.М. Зубрицький*
Комп`ютерна верстка *В.В. Кірвас*
Комп'ютерний дизайн обкладинки *І.В. Ільїна*
Техн. редактор *В.В. Кірвас* Коректор *Н.К. Гур'єва*
Формат 60×84/8 Ум.-друк. арк. – 19,99
Підписано до друку 21.12.2017

Свідоцтво про державну реєстрацію друкованого засобу масової інформації
КВ № 22357 – 12257ПР від 30.09.2016 р.
Ціна договірна Тираж 150 пр. Зам. 1221-17

Адреса редакції: 61023, Харків-23, вул. Сумська, 77/79
тел. (057) 704-91-97, (067) 998-02-70 e-mail: red@hups.mil.gov.ua red.hnups@gmail.com

Видавництво Харківського національного університету Повітряних Сил імені Івана Кожедуба
Свідоцтво суб'єкта видавничої справи ДК № 5370 від 30.06.2017 р.
Адреса видавництва: 61023, Харків-23, вул. Сумська, 77/79

Віддруковано з готових оригінал-макетів у друкарні ФОП Петров В.В.
Єдиний державний реєстр юридичних осіб та фізичних осіб-підприємців.
Запис № 24800000000106167 від 08.01.2009 р.
61144, Харків, вул. Гв. Широнінців, 79в, к. 137
тел. (057) 778-60-34 e-mail: bookfabrik@mail.ua