МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Кафедра кібербезпеки та програмного забезпечення

На правах рукопису

Нетеса Владислав Юрійович

Програмне забезпечення системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK

Спеціальність: 125 «Кібербезпека»

Освітній ступінь: бакалавр

Науковий керівник:

Смірнов Сергій Анатолійович _____

(підпис) (дата)

кандидат технічних наук

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри

_____ О.А. Смірнов

(підпис) ПІБ

«____»_____2021 р.

Кропивницький – 2021

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Освітній ступінь *бакалавр*
Спеціальність *125 Кібербезпека*

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____О.А.Смірнов
« 11 » січня 2021 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Нетесі Владиславу Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK*

керівник роботи *Смірнов Сергій Анатолійович, канд. техн. наук*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 185-02 від 28.12.2020 року

2. Строк подання студентом роботи до захисту *22.05.2021 р.*

3. Мета та завдання кваліфікаційної бакалаврської роботи*: Метою розробки є програмне забезпечення системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.
2.Перегляд аналогічних існуючих систем.
3. Опис і обґрунтування проектних рішень.
4. Етапи програмування системи.
5. Впровадження системи в промислову експлуатацію.
6. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*

6. Дата видачі завдання « 11 » січня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів кваліфікаційної бакалаврської роботи	Примітка
1.	Аналіз існуючих систем	10.03.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2021 р.	
3.	Розробка моделі компонента	20.03.2021 р.	
4.	Розробка структур даних	25.03.2021 р.	
5.	Розробка алгоритмів зв’язку та відображення	30.03.2021 р.	
6.	Програмування алгоритмів	10.04.2021 р.	
7.	Оформлення ПЗ	17.04.2021 р.	
8.	Попередній захист роботи	14.05.2021 р.	

Студент _____ _____
(підпис) (прізвище та ініціали)

Керівник роботи _____ _____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Нетеса В.Ю. Програмне забезпечення системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK. 125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній кваліфікаційній бакалаврській розроблено програмне забезпечення, яке призначено для системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

Метою розробки є програмне забезпечення системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

Результат роботи – програмна реалізація системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10.

Програму розроблено в середовищі RAD Studio Delphi.

Ключові слова: кібербезпека, WPA2

ABSTRACT

Netesa V.Yu. Cybersecurity system software for simulating WPA2 hacking based on KRACK attack implementation. 125 Cybersecurity. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this bachelor's qualification, software has been developed that is designed for a cybersecurity system to simulate the hacking of the WPA2 protocol based on the implementation of the KRACK attack.

The purpose of the development is cybersecurity software to simulate the hacking of the WPA2 protocol based on the implementation of the KRACK attack.

The result is a software implementation of a cybersecurity system to simulate the hacking of the WPA2 protocol based on the implementation of the KRACK attack.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on an IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the environment of RAD Studio Delphi.

Keywords: cybersecurity, WPA2

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

AP	–	точка доступу
DSSS	–	технологія Direct Sequence Spread Spectrum
EAP	–	протокол розширеної автентифікації Extensible Authentication Protocol
KRACK	–	Key Reinstallation Attacks, (атака переустановки ключа
MIC	–	криптографічна контрольна сума
RADIUS	–	сервер доступу Remote Access Dial-in User Server
RC4	–	алгоритм шифрування
TKIP	–	протокол генерації генерація ключів WPA шифрування даних Temporal Key Integrity Protocol
TLS	–	протокол захисту транспортного рівня Transport Layer Security
SSID	–	ідентифікатор мережі який передає точка доступу
WEP	–	Wired Equivalent Privacy – протокол безпеки
WPA	–	стандарт безпеки Wi-Fi Protected Access

ВСТУП

Актуальність теми. Група дослідників повідомила про виявлення ряду критичних уразливостей в WPA2, технології, яка забезпечує безпеку з'єднання для всіх сучасних захищених Wi-Fi-мереж. Зловмисник, перебуваючи в зоні дії бездротової мережі жертви, може використовувати ці діри безпеки, щоб обійти захист і прослуховувати трафік між точкою доступу й бездротовим пристроєм. Таким чином, небезпеки піддаються будь-які дані, передані через будь-яку Wi-Fi-мережу у світі, у тому числі й конфіденційна інформація, яка раніше вважалася надійно зашифрованою. Це, зокрема, можуть бути номера кредитних карток, паролі, повідомлення в чатах, електронні листи, світлини і т.д. У ряді випадків, залежно від конфігурації мережі, можливо не тільки читання, але й зміна переданих даних.

Атаки на основі виявлених критичних уразливостей в WPA2, що одержали узагальнене назву KRACK (Key Reinstallation Attacks), використовують уразливості в самому стандарті Wi-Fi, а не в окремих продуктах або застосунках, тому погрозі піддана будь-яка реалізація WPA2. Інакше кажучи, будь-який сучасний пристрій, що підтримує Wi-Fi, є вразливим до атак KRACK, не залежно від виробника або того, якою операційною системою воно управляється: Android, iOS, macOS, Linux, Windows, OpenBSD і інші.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

– Дослідження системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

– Програмна реалізація системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Меті Ванхоф (Mathy Vanhoef), бельгійський дослідник з Левенського католицького університету, виявив дану проблему в конкретних застосунках ряду виробників ще у 2017 році. Близько року він займався її дослідженням, після чого в середині липня 2017 року повідомив про уразливість постачальників устаткування, яке тестував. Спілкування з останніми переконало його, що проблема носить не локальний характер, пов'язаний з конкретною помилкою в реалізації деяких застосунків, а глобальн, що ставиться до самого протоколу.

У відомість був поставлений координаційний центр CERT/CC по вирішеннях проблем безпеки в Інтернеті, чиї фахівці приєдналися до дослідження й побрали на себе керування по координації спільних дій широкого кола залучених у процес учасників. Зокрема, CERT/CC розіслав 28 серпня 2017 року повідомлення про уразливість великій кількості виробників по усьому світу, погодив дати розголошення інформації й виходу відновлень. Однак, не все пройшло гладко, як хотілося дослідникам. Зокрема, частину компаній і організацій, які про проблему довідалися ще в липні, поспішили випустити «мовчазні» відновлення раніше погодженого строку розкриття інформації. Широкого резонансу вони не викликали, але підвищили ризики виявлення проблеми сторонніми особами раніше наміченого строку.

Про виявлення проблеми широкої громадськості стало відомо ближче до вечора 16 жовтня 2017 року, коли група фахівців з безпеки мереж виступила зі скоординованою заявою. Більш детально про знайдені уразливості було розказано на конференції ACM по комп'ютерах і комунікаційної безпеки 1 листопада 2017 року, де була представлена доповідь «Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2» («Атака переустановки ключів: примусове

повторне використання Nonce в WPA2»). Ледве раніше Мети Ванхоф виклала своя дослідницьку доповідь у мережу (HTTP://papers.mathyvanhoef.com/ccs2017.pdf), присвятив проблемі сайт (HTTPs://www.krackattacks.com), опублікував відео з демонстрацією атаки, а також створив репозиторій на Github, де на даний момент доступний скрипт перевірки однієї з виявлених їм уразливостей (HTTPs://github.com/vanhoefm/krackattacks-test-ap-ft).

1.2 Область застосування

Областю застосування є Wi-Fi. Wi-Fi («бездротова точність») – стандарт на устаткування Wireless LAN. Розроблений консорціумом Wi-Fi Alliance на базі стандартів IEEE 802.11, «Wi-Fi» – торговельна марка «Wi-Fi Alliance». Технологію назвали Wireless-Fidelity (дослівно «бездротова точність») за аналогією з Hi-Fi.

Установка Wireless LAN рекомендувалася там, де розгортання кабельної системи було неможливе або економічно недоцільно. У нинішній час у багатьох організаціях використовується Wi-Fi, тому що за певних умов швидкість роботи мережі вже перевищує 100 Мбіт/с. Користувачі можуть переміщатися між точками доступу по території покриття мережі Wi-Fi. При цьому, при зміні точок доступу відбувається короткочасний розрив зв'язку, за винятком використання устаткування Cisco.

Мобільні пристрої (кишенькові комп'ютери, смартфони й ноутбуки), оснащені клієнтськими приймально-передавальними пристроями, можуть підключатися до локальної мережі й одержувати доступ в Інтернет через точки доступу.

Невелика ширина використовуваного спектра частот, відсутність можливостей роумінгу й авторизації не дозволяють Wi-Fi пристроям потіснити на ринку стільниковий мобільний зв'язок. Проте, такі компанії, як Zyxel

Communications, Socketip і Symbol Technologies, пропонують застосунку по організації Wi-Fi телефонії.

Стандарти

IEEE 802.16: WiMax

WiMax – це міжнародна технологія функціональної сумісності для мікрохвильового доступу. Вона дозволяє передавати дані зі швидкістю від 30 до 40 мегабіт у секунду. Термін конкретно ставиться до можливості взаємодії й реалізації усередині стандарту IEEE 802.16.

Ця технологія для бездротової передачі даних своїм клієнтам колись використовувалася декількома операторами мобільного зв'язку, зокрема Sprint. Потім ці оператори відмовилися від WiMax і перейшли на використання більш швидких LTE 4G-мереж для передачі даних.

WiMax Forum сертифікує пристрою перед поставками часткам і корпоративним користувачам. Технологія працює краще, якщо зв'язок організований поза приміщеннями.

IEEE 802.15.4: ZigBee

ZigBee, бездротова технологія й група LPWAN-технологій, представляє відкритий глобальний стандарт, розроблений спеціально для використання в M2M-мережах.

Технологія недорога для запуску й не вимагає більших потужностей. Це робить ZigBee ідеальним застосунком для багатьох промислових застосунків. ZigBee має низьку затримку й мале енергоспоживання, що дозволяє продуктам працювати від однієї батареї кілька років без підзарядки. Протокол ZigBee пропонує 128-бітне AES-шифрування. Ця технологія також використовується в Mesh-мережах, які дозволяють вузлам з'єднуватися один з одним за допомогою декількох маршрутів.

Очікується, що ZigBee буде реалізована в пристроях для розумного будинку. Можливості одночасно з'єднувати декілька різних « речей» робить її ідеальної для підключеного домашнього середовища. Користувачі можуть

підключити такі предмети, як розумні замки, джерела світла, термостати. Ці «речі» зможуть взаємодіяти один з одним.

ZigBee Alliance стандартизував технологію ZigBee PRO 2017, сподіваючись розширити можливості connectivity. Однак поки пристрою ZigBee не здатні повноцінно взаємодіяти між собою. Надалі планується, що стандартизація виправить цю проблему, а девайси зможуть працювати в єдиному просторі. Діапазон частот також обмежує можливості цієї технології [1].

EEE 802.15.1: Bluetooth і BLE

Bluetooth і Bluetooth Low Energy (BLE, Bluetooth Smart) – це бездротові технології, використовувані для передачі даних на короткі відстані. Вони часто задіються в невеликих пристроях, які підключаються до телефонів і планшетам користувачів. Наприклад, технологія використовується в багатьох акустичних системах.

Bluetooth Low Energy використовує менше енергії, чому стандартний Bluetooth. Підтримкою BLE оснащують фітнес-трекери, смарт-годинник і інші підключені пристрої для того, щоб заряд акумулятора витрачався більш ощадливо.

Застосування BLE масово тільки почалося. Спочатку ця технологія була представлена компанією Nokia в 2006 році. Однак до 2010 року вона була невід'ємною частиною стандарту Bluetooth. Сьогодні BLE підтримується більшістю виробників смартфонів і комп'ютерів, а також основними операційними системами (Windows 8, OS X, Linux, Windows Phone, Android і iOS).

Bluetooth використовує радіохвилі високочастотного електромагнітного поля для передачі даних. Спочатку ця технологія була стандартизована як 802.15.1, але IEEE більше не підтримує цей окремий стандарт.

Компанії, що працюють із Bluetooth, часто пов'язані із групою Bluetooth Special Interest (SIG). У цей час група нараховує більш 20 тис. учасників. SIG повинна сертифікувати продукт із Bluetooth, перш ніж його можна буде

поставляти на ринок. Ця процедура допомагає гарантувати, що всі пристрої Bluetooth проходять стадію нормативного регулювання й забезпечують однаковий рівень захищеності й безпеки.

IEEE 802.11ax – Wi-Fi 6

Технологія Wi-Fi 6 підвищить швидкість передачі даних і пропускну здатність як нов, що так і вже існують бездротових мереж. Орієнтований на корпоративних користувачів стандарт забезпечить гнучкість і масштабованість інфраструктури для швидкого запуску цифрових сервісів.

Стандарт 802.11ax припускає, що підтримуючі його пристрої зможуть у чотири рази швидше завантажувати дані й у шість раз швидше передавати їх на вилучений сервер у порівнянні з застосунками, що мають підтримку Wi-Fi попереднього покоління (802.11ac). Також заявлені в чотири рази більш широка зона приймання сигналу й 7-кратний виграш у частині енергоефективності, говориться в повідомленні компанії. Перші чипи з підтримкою 802.11ax представила компанія Broadcom.

На початку вересня 2018 року Wireless Broadband Alliance (WBA) повідомила про те, що 802.11ax одержить цілий ряд нових функцій. Серед них, зокрема, багатокористувацькі висхідні й спадний канали зв'язку MIMO (multiple input, multiple output) для розширення пропускної здатності каналів при одночасній підтримці безлічі пристроїв. Ця функція призначена для підприємств і мережних провайдерів, а також для великих громадських місць і будинків, у яких перебуває безліч користувачів.

Підтримка діапазонів частот 2,4 і 5 ГГц розширить використовуваний спектр і забезпечить сумісність із наявними пристроями. Крім того, буде підтримуватися також діапазон 6 ГГц. Змінювана пропускна здатність каналів і можливість використання різних пристроїв дозволять операторам запропонувати більш ефективний IoT, у тому числі підключення з використанням низької пропускної здатності для вузьких виділених каналів, які будуть заощаджувати електроенергію. Завдяки цій функції оператори зможуть в одній мережі

підтримувати широкосмугові сервіси й IoT. Функція Target Wake Time дозволить пристроям IoT переходити в режим сну, щоб зменшити конкуренцію за доступ до мережі, і активізуватися при необхідності, заощаджуючи заряд батарей.

Згідно з документом WBA «Enhanced Wi-Fi – 802.11ax Decoded», в 802.11ax будуть представлені функції не тільки для підтримки десятків мільйонів смартфонів, які використовують Wi-Fi, але й для задоволення вимог таких сегментів, як пристрою інтернету речей (IoT), доповненої й віртуальної реальності. Застосовувати можливості 802.11ax, за інформацією WBA, можна буде в мережах високої щільності, на транспорті, у роздрібній торгівлі й індустрії розваг, на підприємствах, у промисловості й розумних містах.

По даним WBA, в Wi-Fi наступного покоління, крім оптимізованих продуктивності, пропускної здатності й ефективності, будуть підтримуватися сценарії раннього застосування 5G. Оновлена технологія Wi-Fi буде відповідати вимогам, розробленим МСЕ стосовно до 5G для стандарту IMT-2020.

По очікуваннях WBA, стандарт 802.11ax буде ратифіковано в IV кварталі 2019 року. За прогнозами об'єднання операторів мобільних мереж GSM Association, до 2022 року більш ніж в 70% продуктів Wi-Fi корпоративного класу буде застосовуватися 802.11ax. [2]

IEEE 802.11ad (Wigig)

Компанія Samsung Electronics повідомила в жовтні 2014 року про успішну розробку технології, здатної забезпечити в 5 раз більш високу швидкість передачі даних по стандарту Wi-Fi у смартфонах, планшетах і роутерах. Розроблена Samsung технологія відповідає специфікації IEEE 802.11ad.

IEEE 802.11ac – Wi-Fi 5

В 2015 році прийнятий новий стандарт для бездротових мереж передачі даних IEEE 802.11ac, який дозволить передавати інформацію до трьох раз швидше, чим останній сьогоднішній стандарт IEEE 802.11n. У цей час він обмежує передачу даних швидкістю до 300 Мбіт/с.

Очікується, що підвищення швидкодії буде досягнуто в першу чергу завдяки тому, що пристрою зможуть працювати не тільки з каналами шириною 20-40 МГц, але й 80-160 МГц, особливо в частотному діапазоні 5 ГГц. З великою часткою ймовірності в стандарту збережеться сумісність із попередніми версіями Wi-Fi-стандартів. Крім росту швидкості Wi-Fi очікується значне збільшення кількості пристроїв, які будуть використовувати його для передачі даних. Аналітичне агентство In-Stat припускає, що до 2015 року їх загальне число перевищить 1 млрд.

Продукти для бездротових локальних мереж на базі стандарту IEEE 802.11ac, який зараз перебуває в стадії розробки, почнуть поставлятися в 2012 році. У компанії Broadcom уважаються, що їх поява ознаменує нову еру розвитку мереж Wi-Fi, що володіють високою продуктивністю й значно більшим радіусом дії.

У новій технології будуть використовуватися вузьконаправленне випромінювання антен, більш широкі канали, кілька антен для передачі й приймання даних. Усе це дозволить довести швидкодію до 1,3 Гбіт/с і побільшати відстань зв'язку. Новий стандарт забезпечить також краще проходження сигналів через стіни будинків, тому мережа на базі технології 11ac буде надійно працювати в межах цілого будинку, уважається Рауль Пател, віце-президент групи мобільному й бездротовому зв'язку Broadcom. Цей стандарт стане кроком уперед у порівнянні з IEEE 802.11n, найбільш зробленим сьогодні стандартом бездротового зв'язку, що забезпечують, як правило, швидкодія до 300 Мбіт/с.

В індустрії продуктів для домашніх бездротових мереж розглядається можливість застосування декількох їх модифікацій для передачі відео високого дозволу, що вимагає високої швидкодії й стабільності роботи. Пател не бачить погрози для 11ac з боку конкуруючих технологій. Wireless HDMI, наприклад, забезпечує більшу швидкість, але значно дорожче, тому чи навряд одержить масове поширення. Wigig передбачає передачу даних зі швидкодією, що

досягають 6 Гбіт/с, але на менші відстані. Можливо, Wigig і 11ac зможуть вдало доповнювати один одного.

В Broadcom розраховують, що продукти на базі 802.11ac почнуть поставлятися в другій половині 2012 року. Пател уважає, що до кінця майбутнього року в Wi-Fi Alliance може бути готова програма їх сертифікації, хоча роботи над стандартом IEEE ще, можливо, не будуть завершені. Втім, коли стандарт IEEE 802.11n перебував ще в стадії розробки, уже почалися поставки продуктів на його основі. Їхню сертифікацію на відповідність попереднім редакціям стандарту виконували в Wi-Fi Alliance.

У листопаді аналітики In-Stat опублікували прогноз, у якому вказувалося, що продажу продуктів на основі 11ac почнуть швидко рости після випуску цього стандарту. Якщо в 2012 році можна чекати поставок близько 1 млн маршрутизаторів і модемів різних типів, то в 2015 році цей показник складе вже 350 млн. Але навіть тоді 11n буде цілком успішно конкурувати з 11ac, і очікується, що в 2015 році буде продано майже 1,5 млрд пристроїв на базі 11n.

Так само як в 802.11n, у новому стандарті регламентується застосування декількох антен для формування декількох потоків трафіку. Однак радіомодуль 11ac, що використовує тільки один потік, практично не буде уступати по продуктивності модулю 11n із трьома потоками, затверджує Пател. Один з методів, що дозволяє майже втроє побільшати його швидкодію, – застосування більш широкосмугових каналів – від 80 до 160 МГц, у той час як максимум в 11n – 40 МГц. Стандарт 802.11ac визначає також, яким образом можуть спільно працювати кілька пристроїв, що використовують цей розширений частотний діапазон каналів.

З появою стандарту 11ac удасться вийти за рамки приватного діапазону 2,4 ГГц, єдино дозволеного старим стандартом 802.11b і використовуваного також у якості одного з діапазонів стандарту 11n. Новий стандарт 11ac розрахований тільки на діапазон 5 ГГц, який, як відзначив Пател, не є настільки «перенаселеним». Він звернув увагу на те, що 90-95% пристроїв працюють

сьогодні в мережах Wi-Fi у діапазоні 2,4 ГГц, де є тільки три «непересічні» канали, тоді як у діапазоні 5 ГГц таких каналів 20.

IEEE 802.11n – Wi-Fi 4

Вимоги до устаткування стандарту IEEE 802.11n. утримуються в частині I наказу № 124 «Про твердження Правил застосування устаткування радіодоступа», датованого 14 вересня 2010 року. Одночасно із прийняттям нового наказу, старий, від 13 лютого 2007 року, втратив свою силу. Ця новина напевно порадує мобільних людей – устаткування стандарту Wi-Fi 802.11n, що працює в діапазоні 2,4-2,5 і 5,0 ГГц, забезпечує швидкість передачі даних до 600 Мбіт/с і більший радіус покриття.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми кваліфікаційної бакалаврської роботи

Серйозною проблемою для всіх бездротових локальних мереж (і, якщо вже на те пішло, те й усіх провідних локальних мереж) є безпека. Безпека тут так само важлива, як і для будь-якого користувача мережі Інтернет. Безпека є складним питанням і вимагає постійної уваги. Величезна шкода може бути нанесений користувачеві через те, що він використовує випадкові хот-споти (hot-spot) або відкриті точки доступу WI-FI удома або в офісі й не використовує шифрування або VPN (Virtual Private Network – віртуальна приватна мережа). Небезпечно це тим, що користувач уводить свої особисті або професійні дані, а мережа при цьому не захищена від стороннього вторгнення.

WEP

Споконвічно було складно забезпечити належну безпеку для бездротових локальних мереж.

Хакери легко здійснювали підключення практично до будь-якої Wifi мережі зламуючи такі первісні версії систем безпеки, як Wired Equivalent Privacy (WEP). Ці події залишили свій слід, і довгий час деякі компанії неохоче впроваджували або зовсім не впроваджували в себе бездротові мережі, побоюючись, що дані, передані між бездротовими Wifi пристроями й Wi-Fi точками доступу можуть бути перехоплені й розшифровані. Таким чином, ця модель безпеки сповільнювала процес інтеграції бездротових мереж у бізнес і змушувала нервувати користувачів, що використовують Wifi мережі будинку. Тоді інститут IEEE, створив робочу групу 802.11i, яка працювала над створенням всеосяжної моделі безпеки для забезпечення 128-бітного AES шифрування й автентифікації для захисту даних. Wi-Fi Альянс представив свій власний

проміжний варіант цього специфікації безпеки 802.11i: Wi-Fi захищений доступ (WPA – Wi-Fi Protected Access). Модуль WPA поєднує кілька технологій для вирішення проблем уразливості 802.11 WEP системи. Таким чином, WPA забезпечує надійну автентифікацію користувачів з використанням стандарту 802.1x (взаємна автентифікація й інкапсуляція даних переданих між бездротовими клієнтськими пристроями, точками доступу й сервером) і розширюваний протокол автентифікації (EAP).

Також, WPA оснащений тимчасовим модулем для шифрування WEP-движка за допомогою 128 – бітного шифрування ключів і використовує часовий протокол цілісності ключів (TKIP). А за допомогою контрольної суми повідомлення (MIC) запобігає зміні або форматування пакетів даних. Така комбінація технологій захищає конфіденційність і цілісність передачі даних і гарантує забезпечення безпеки шляхом контролю доступу, так щоб тільки авторизовані користувачі одержали доступ до мережі.

WPA

Подальше підвищення безпеки й контролю доступу WPA полягає в створенні нового унікального майстра ключів для взаємодії між кожним користувацьким бездротовим устаткуванням і точками доступу й забезпеченні сесії автентифікації. А також, у створенні генератора випадкових ключів і в процесі формування ключа для кожного пакета.

В IEEE стандарт 802.11i, ратифікували в червні 2004 року, значно розширивши багато можливостей завдяки технології WPA. Wi-Fi Альянс зміцнив свій модуль безпеки в програмі WPA2. Таким чином, рівень безпеки передачі даних Wifi стандарту 802.11 вийшов на необхідний рівень для впровадження бездротових застосунків і технологій на підприємствах. Одне з істотних змін 802.11i (WPA2) відносно WPA це використання 128-бітного розширеного стандарту шифрування (AES). WPA2 AES використовує в боротьбі з CBC-MAC режимом (режим роботи для блоку шифру, який дозволяє один ключ використовувати як для шифрування, так і для автентифікації) для забезпечення

конфіденційності даних, автентифікації, цілісності й захисту відтворення. У стандарті 802.11i пропонується також кешування ключів і попередньої автентифікації для упорядочивания користувачів по точках доступу.

WPA2

Зі стандартом 802.11i, увесь ланцюжок модуля безпеки (вхід у систему, обмін повноваженнями, автентифікація й шифрування даних) стає більш надійним і ефективним захистом від ненаправлених і цілеспрямованих атак. Система WPA2 дозволяє адміністраторові Wi-Fi мережі перемкнутися з питання безпеки на керування операціями й пристроями.

Стандарт 802.11r є модифікацією стандарту 802.11i. Даний стандарт був ратифіковано в липні 2008 року. Технологія стандарту більш швидко й надійно передає ключові ієрархії, засновані на технології Handoff (передача керування) під час переміщення користувача між точками доступу. Стандарт 802.11r є повністю сумісною з Wifi стандартами 802.11a/b/g/n.

Також існує стандарт 802.11w, призначений для вдосконалення механізму безпеки на основі стандарту 802.11i. Цей стандарт розроблений для захисту керуючих пакетів.

Стандарти 802.11i і 802.11w – механізми захисту мереж Wifi стандарту 802.11n.

Шифрування файлів і папок в Windows 7

Функція шифрування дозволяє вам зашифрувати файли й папки, які буде в наслідку неможливо прочитати на іншому пристрої без спеціального ключа. Така можливість присутня в таких версіях пакета Windows 7 як Professional, Enterprise або Ultimate. Далі будуть освітлені способи включення шифрування файлів і папок.

Включення шифрування файлів:

Пуск -> Комп'ютер(виберіть файл для шифрування)-> права кнопка миші по файлу->Властивості->Розширений(Генеральна вкладка)->Додаткові атрибути->Поставити маркер у пункті шифрувати вміст для захисту даних->Ок-

>Застосувати->Ok(Виберіть застосувати тільки до файлу)->Закрити діалог Властивості (Нажати Ok або Закрити).

Включення шифрування папок:

Пуск -> Комп'ютер(виберіть папку для шифрування)-> права кнопка миші по папку-> Властивості->Розширений(Генеральна вкладка)->Додаткові атрибути-> Поставити маркер у пункті шифрувати вміст для захисту даних->Ок->Застосувати->Ok(Виберіть застосувати тільки до файлу)->Закрити діалог Властивості (Нажати Ok або Закрити).

В обоє випадку в області повідомлення з'явиться пропозиція зробити резервну копію ключа шифруємих файлів або папок. У діалоговому вікні ви зможете вибрати місце збереження копії ключа (рекомендується знімний носій). Файл або папка змінить свій колір при вдалому шифруванні.

Шифрування вкрай корисне якщо ви ділите свій робочий комп'ютер з кількома людьми.

Сфера застосування

У більшості випадків бездротові мережі (використовуючи точки доступу й маршрутизатори) будуються в комерційних цілях для залучення прибутки з боку клієнтів і орендарів для підготовки й реалізації наступних проектів по впровадженню мережної інфраструктури на основі бездротових застосунків:

- Офісні центри.
- Торгові центри.
- Готельні комплекси.
- Складські приміщення.
- Створення бездротової локальної мережі.
- Проектування бездротових мереж.
- Місця проведення комерційних і суспільних заходів.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

− Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

− Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовучи класичну реалізацію керування пам'яттю об'єктів.

− Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

− Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.

− Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

− Відладник Win 64 (на LLDB) і збирач для C++.

− Поліпшення для C++: включена велика кількість поліпшень STL з Dinkumware.

− Підтримка Metal Driver GPU для macOS і iOS.

− Вбудований Fmxlinux.

− Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API. Реалізація компонента Media Player для macOS тепер використовує Avfoundation. Реалізований заново стилізуємий FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.

− Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

− Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

− Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

− У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуться, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom managed records. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуться щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовучи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існучим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

Розширена підтримка бібліотек C++

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

Підвищення якості й швидкодії інструментів

– Велика кількість поліпшень STL від Dinkumware.

– Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.

– Поліпшена підтримка Cmake.

– Велика кількість виправлень для підвищення стабільності і якості.

– Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.

– Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуться під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати декілька VCL стилів на різних формах в одному застосунку або в різних компонентів на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовуть стиль.

Поліпшена кроссплатформеність

– Додана підтримка Metal Driver GPU для macOS і iOS.

– Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.

– Реалізований заново стилізуємий FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.

– Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

– Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну бакалаврську роботу, реалізації підлягає програмне забезпечення, яке призначено для системи

кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

В процесі розробки кваліфікаційної бакалаврської роботи необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи кібербезпеки, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Розглянемо існуючі технології злому Wi-Fi.

Android-застосунок для пошуку точок доступу Wi-Fi розкрив паролі 2 млн мереж

Застосунок Wifi Finder, завантажений з Google Play Store тисячами користувачів, дозволяє знаходити поблизу доступні мережі Wi-Fi. Користувачі також можуть завантажувати паролі для доступу до бездротових мереж зі своїх пристроїв у базу даних додатки, щоб ними могли користуватися інші. Однак ця база даних перебувала у відкритому доступі, і покопатися в ній міг будь-який бажаючий [11].

Дослідник безпеки Саньям Джайн (Sanyam Jain) виявив БД і повідомив про неї журналістам видання Techcrunch. Протягом більш двох тижнів вони спільними зусиллями намагалися зв'язатися з розроблювачем застосунку, що приблизно перебувають у Китаї, але безуспішно. У підсумку журналісти звернулися до хостинг-провайдеру Digitalocean, який у той же день відключив незахищену БД.

Кожний запис у БД містила ім'я мережі Wi -Fi, її точне місце розташування, ідентифікатор BSSID і пароль у незашифрованому виді. Хоча, за словами розроблювача, застосунок надає паролі тільки для доступу до суспільних бездротових мереж, у БД утримуються дані безлічі домашніх мереж.

Застосунок не вимагає, щоб користувачі одержували дозвіл від власника мережі Wi-Fi, тим самим піддаючи її погрозі несанкціонованого доступу. Маючи доступ до мережі, зловмисник може змінити налаштування маршрутизатора таким чином, щоб нічого користувачі, що не підозрюють, попадали на шкідливі

сайти. Перебуваючи в мережі, злочинець також може переглядати минаючий через бездротову мережу незашифрований трафік і викрадати конфіденційні дані.

Уразливості в протоколі WPA3

Не пройшло й роки з моменту запуску стандарту WPA3 (Wi-Fi Protected Access III), покликаного усунути технічні недоліки протоколу WPA2, який довгий час уважався небезпечним і вразливим до атаки реінсталяції ключів (Key Reinstallation Attack, KRACK), як дослідники виявили ряд серйозних уразливостей у стандарті, що дозволяють витягти пароль Wi-Fi і проникнути в мережу [12].

Хоча протокол WPA3 покладається на більш безпечне рукостискання SAE (Simultaneous Authentication of Equals), відоме як Dragonfly, яке спрямовано на захист мереж Wi-Fi від автономних атак по словникові, фахівці Меті Венхоф (Mathy Vanhoef) і Ийал Ронен (Eyal Ronen) [13] виявили кілька недоліків у дизайні ранньої реалізації WPA3-personal, що надають можливість відновити паролі від мережі Wi-Fi за допомогою тайминг-атак або атак на кеш.

«Зокрема, що атакують можуть прочитати інформацію, яка вважається надійно зашифрованою. Вони можуть скористатися цим для крадіжки конфіденційних даних, наприклад, номерів платіжних карт, паролів, повідомлень чата, електронних листів і так далі», – пояснили експерти.

Усього фахівці виявили п'ять проблем, що одержали загальну назву Dragonblood. У своїй доповіді Венхоф і Ронен описали два типи недоліків у дизайні – один веде до атак зниження рівня (downgrade attacks), другий – до витоків кешу.

Оскільки WPA3 ще не так широко розповсюджений, для підтримки старих пристроїв сертифіковані WPA3 пристрої пропонують «перехідний режим роботи», у якому можна налаштувати підключення з використанням як WPA3-SAE, так і WPA2. Виявилося, що даний режим уразливий до атак зниження рівня, чим можуть скористатися зловмисники для створення шкідливої точки доступу, що підтримує тільки WPA2, змушуючи пристрою з підтримкою WPA3

підключатися за допомогою небезпечного чотирибічного рукостискання WPA2. Крім того, до атак зниження вразливе й саме рукостискання Dragonfly. Цей недолік зловмисники можуть використовувати для того, щоб змусити пристрій використовувати більш слабку еліптичну криву, чому звичайно.

Як відзначається, для здійснення атак зниження рівня зловмисникові досить знати SSID мережі WPA3- SAE.

Дослідники також описали ряд атак на основі синхронізації (CVE-2019-9494) і атак на кеш (CVE-2019-9494), що дозволяють одержати пароль Wi-Fi, а також Dos-атаку, яка може бути здійснена шляхом ініціювання великої кількості рукостискань із точкою доступу WPA3.

Протокол захисту даних WPA3 (Wi-Fi Protected Access III)

Об'єднання найбільших виробників комп'ютерної техніки й бездротових Wi-Fi пристроїв Wi-Fi Alliance опублікувало [14] на початку 2018 року перші подробиці про протокол захисту даних WPA3, який повинен прийти на зміну WPA2 [15].

Перший офіційний проект протоколу автентифікації WPA3 доступний в 2018 році, однак Wi-Fi Alliance оприлюднило дані про чотири основних функції, що присутні у новому стандарті безпеки.

Першою особливістю є захист від брутфорс-атак шляхом блокування процесу автентифікації після декількох невдалих спроб авторизації.

Друга функція являє собою можливість використовувати Wi-Fi пристрою, що перебувають недалеко друг від друга, у якості панелі конфігурації для інших пристроїв. Наприклад, користувач зможе використовувати свій телефон або планшет для налаштування параметрів Wi-Fi WPA3 на іншому пристрої, у якого немає екрана, такого як «розумні» лампочки, дверні замки та ін.

Третя функція, описувана як «індивідуальне шифрування даних», дозволяє шифрувати з'єднання між кожним пристроєм і маршрутизатором або точкою доступу. Четверта особливість – поліпшений криптографічний стандарт, призначений для мереж Wi-Fi з більш високими вимогами безпеки, такими як

урядові й оборонні установи, а також промислове виробництво. Більш докладна інформація про протокол очікується пізніше в 2021році.

Уразливість року – WPA2: дані у всіх Wi-Fi мережах світу виявилися доступні для перехоплення

Дослідник по безпеці по імені Маті Ванхуф виявив набір дуже серйозних уразливостей у протоколі WPA2, використовуваному для захисту бездротових мереж стандарту Wifi. Уразливість дозволяє зчитувати дані, які раніше вважалися надійно зашифрованими. Порушеними виявилися всі існуючі реалізації WPA2, що означає, що вразливі всі Wifi-мережі у світі. Користувачам рекомендується встановити патчи для своїх бездротових пристроїв, як тільки вони стануть доступні. [16]

Атака, описана Ванхуфом, одержала назву KRACK – від "key reinstallation attack" (атака переустановки ключа). При її успішному результаті потенційний зловмисник може перехопити й розшифрувати будь-які дані, які жертва відправляє, а в деяких випадках – і одержує. Найбільше «руйнівними» можуть бути наслідку атаки на Linux або Android версій 6.0 і вище, оскільки, за словами дослідників, «обидві операційні системи можна хитрістю змусити (пері)установити нульові ключі шифрування». З іншими системами розшифрувати всі пакети буде складніше, однак більша їхня частини як і раніше може бути розкрита.

WPA2 має на увазі чотирибічне представлення при встановленні з'єднання. Коли клієнт намагається підключитися до захищеної мережі Wifi, процедура такого представлення запускається для підтвердження, що й клієнт, і точка доступу використовують коректні дані для доступу. При цьому запитується новий ключ шифрування, який буде використовуватися для захисту всього наступного трафіку.

Атака KRACK за допомогою досить простих маніпуляцій із криптографічними повідомленнями дозволяє змусити систему, що атакується, видати не новий, а ключ, що вже використовувався. При переустановці ключа

пов'язані з ним параметри – такі як номера переданого пакета (Nonce) і номер прийнятого пакета (лічильник повторів) – скидаються до споконвічного значення. Гарантувати безпека може тільки одноразовий ключ, однак протокол WPA2 цього, як виявилося, не гарантує.

Справа в тому, що при чотирибічній виставі точка доступу може повторно випускати своє власне ідентифікаційне повідомлення («повідомлення номер 3»), якщо не одержує правильної відповіді. Клієнт, таким чином, може одержати це повідомлення кілька раз. І щораз він буде переустановлювати той самий ключ шифрування, скидаючи вищезгадані параметри до колишніх значень. Зловмисник може багаторазово спровокувати подібні скидання й через це атакувати сам протокол шифрування. Пакети можна пересилати повторно, розшифровувати або навіть підробляти, якщо використовується шифрування WPA-TKIP або GCMP, а не AES-CCMP.

Виявленим уразливостям привласнені наступні індекси:

- реінсталяція групового ключа GTK при чотирибічній виставі
- реінсталяція ключа перевірки цілісності групи IGTK при чотирибічній виставі
- реінсталяція ключа GTK при виставі групового ключа.
- реінсталяція ключа IGTK при чотирибічному при виставі групового ключа.
- приймання повторного запиту швидкої передачі базового набору служб (FT) і переустановка парного головного ключа (PTK-TK) при його обробці.
- реінсталяція ключа STK при автентифікації за допомогою Peerkey.
- реінсталяція ключа Peerkey при авторизації по протоколу TDLS.
- реінсталяція групового ключа (GTK) при обробці кадра відповіді у випадку переходу в «сплячий режим» у рамках Wireless Network Management (WNM).

– реінсталяція ключа перевірки цілісності (IGTK) при обробці кадра відповіді у випадках переходу в «сплячий режим» у рамках Wireless Network Management (WNM).

Корпоративні Wi-Fi-мережі вразливі для атак хакерів в 100% випадків

В усіх без винятку проектах по аналізі захищеності дослідники Positive Technologies виявили проблеми безпеки, що відкривають можливість проведення атак через бездротові мережі компаній. Вирішити ці проблеми можна тільки за допомогою комплексного підходу до забезпечення безпеки корпоративної інфраструктури, відзначають експерти [26].

Одна з найпоширеніших проблем безпеки корпоративних Wi-Fi-мереж – використання словникових паролів, які легко підібрати. З ними дослідники Positive Technologies зустрічалися практично у всіх проектах по аналізі захищеності IT-інфраструктури.

Основні недоліки захисту Wi-Fi

Крім того, часто зустрічаються й помилки конфігурування мереж Wi-Fi, що розширюють можливості порушника для проведення атак. До таких недоліків безпеки ставиться відсутність обмеження потужності сигналу бездротових маршрутизаторів, у результаті якого підключення до мережі компанії можна здійснювати поза межами контрольованої зони – із сусіднього будинку або з паркування. Це, наприклад, дозволяє хакерам проводити атаки на пристрої співробітників компанії за межами контрольованої зони й перехоплювати автентифікаційні дані для доступу до корпоративних ресурсів.

Крім цього, як показує досвід робіт з аналізу захищеності, у багатьох випадках після підключення до гостьової мережі може бути отриманий доступ до інших мережних сегментів, у тому числі до ресурсів ЛОМ.

Часто буває, що в компанії обмежений доступ до інтернету для співробітників, заблоковані окремі веб-ресурси. Щоб обійти ці обмеження, найчастіше працівники підключаються до потрібних їх сайтам зі смартфонів. А

для більшої зручності вони можуть розвертати на смартфоні бездротову точку доступу, до якої підключають робочу станцію й користуються інтернет-ресурсами через таке несанкціоноване з'єднання, яке ніяк не захищене. У середньому три несанкціоновані точки доступу виявлялися в хо ді робіт з аналізу захищеності бездротових мереж на кожному об'єкті в 2016 році. В одній з компаній виявлене відразу 7 таких точок.

Як визначити факт злому Wi-Fi мережі

Перша ознака злому домашньої мережі Wi-Fi, з високою ймовірністю, буде падіння швидкості доступу в Інтернет.

Перше, що необхідно – перевірити роутер Wi-Fi: підключитися до роутеру, використовуючи ім'я користувача й пароль (якщо їх не перемінили після покупки – треба терміново поміняти) – найчастіше вони зазначені на наклейці, на задній панелі пристрою). Кожний роутер має особливості й потрібно знайти сторінку зі статусом Wi-Fi, де представлений список підключених до мережі пристроїв.

Ця інформація може перебувати в розділі "Підключені пристрої", "Список пристроїв" або "Домашня мережа".

У такому або подібному списку треба перевірити присутність невідомих пристроїв. Будь-які незнайомі пристрої вказують на те, що безпека мережі порушена: хакер це або власна необачність – потрібно з'ясувати.

При цьому можливо, деякі пристрої мають «незрозумілі» імена, які можуть нічого не сказати, але при цьому вони будуть повністю легітимні. Спочатку треба перевірити всі підключені до Wi-Fi пристрою.

Для підвищення рівня безпеки Wi-Fi мережі діють технології, зокрема – надання доступу до мережі тільки конкретним пристроям на основі їх Mac-адреси. Досить ефективний доступ за рахунок більш надійного з погляду безпеки протоколу, наприклад, WPA2.

Після зміни налаштувань протоколу безпеки, імовірно, буде потрібно заново підключити всі Wi-Fi пристрою.

Імовірно, слід додати ще один рівень безпеки – комплексний застосунок, необхідне для установки на ПК і мобільних пристроях з Android, воно допоможе запобігти випадкам розкрадання персональних даних, блокувати віруси й захистити дітей від впливу небажаної інформації.

Британська страхова компанія CPP провела восени 2020 року експеримент, перевіривши безпеку 40 тис. Wi-Fi мереж у шести найбільших містах країни. Він показав, що 20 тис. з них взагалі не мали пароля для доступу або базового шифрування. Однак навіть багато мереж, захищені паролем, фахівці здатні зламати протягом декількох секунд.

У рамках експерименту по «етичному злому» експерти CPP провели в кожному з міст по півгодини, використовуючи вільно доступне всім ПО для одержання доступу до як можна більшому числу бездротових мереж.

Майже чверть Wi-Fi мереж (9,249) не були закриті паролем для доступу, незважаючи на те, що 82% британців певен у їхній повній безпеці. Втім, навіть для захищених мереж логіни й паролі вдавалося підбирати дуже швидко. Так, у годину експериментатори підбирали по 350 логінів і паролів, сидячи в центрі одного з міст у кафе або магазинах.

Опитування, проведений на замовлення організації Wi-Fi Alliance улітку 2020 року, показав ріст уваги користувачів Wi-Fi до теми безпеки, але разом з тим виявив як і раніше невисокий рівень практичної реалізації заходів захисту.

86% опитаних почали хоча б деякі кроки по забезпеченню захисту приналежних їм точок доступу Wi-Fi і маршрутизаторів. І, хоча 97% уважається, що дані на їхніх пристроях і в мережах у цілковитій безпеці, при відповідях на запитання про рекомендовані заходи безпеки вони набирали в середньому тільки 66%. Приблизно 59% використовує паролі, що не відповідають елементарним критеріям безпеки, тільки 62% відключили автоматичне надання доступу до файлів пристрою в бездротовій мережі й тільки 18% при підключенні до незнайомих точок доступу Wi-Fi використовує VPN.

Wi-Fi Alliance рекомендує включати в мережах шифрування WPA2, використовувати надійні паролі й регулярно їх міняти, а також відключити функцію автоматичного підключення пристроїв до виявлених бездротових мереж.

Результати нагадують про опитування водіїв, проведеному американською асоціацією AAA Foundation for Traffic Safety. 95% з них знають, що за кермом небезпечно читати й відправляти повідомлення з телефону, але 35% надходили так принаймні протягом останнього місяця.

Лише 5% IT-адміністраторів, що приймали участь в опитуванні, проведеному у вересні 2020 року на сайті американського журналу PC World, працює в компаніях, чиї корпоративні мережі Wi-Fi відкриті для доступу всім бажаючим і не вимагають для використання навіть уведення пароля. Інші застосовують ті або інші заходи безпеки. Так, 36% створюють у компанії дві мережі Wi-Fi: одну для співробітників, а іншу, відкриту, але без доступу до корпоративних ресурсів – для відвідувачів. 24% опитаних закривають доступ до мережі Wi-Fi паролем, який повідомляється відвідувачам у міру потреби. 12% використовує засоби контролю доступу до мережі, що перевіряють наявність антивірусних програм, установлених відновлень і налаштувань безпеки. Нарешті, в 17% компаній взагалі не дозволяють відвідувачам користуватися корпоративною мережею Wi-Fi.

Результати паралельного опитування для керівників бізнес-підрозділів показали, що 47% з них задоволені покриттям і режимом роботи мережі Wi-Fi у їхній компанії. 29% відзначає скарги співробітників і необхідність модернізації мережі, а цілих 10% повідомляє, що в їхній компанії взагалі немає корпоративної мережі Wi-Fi.

3.2 Розробка структурної схеми

Дослідниками була виявлена можливість при встановленні з'єднання між точкою доступу й клієнтом проводити маніпуляції із трафіком узгодження (також часто називаним «рукостисканням» від англ. handshake) для стандарту WPA2 (Wi-Fi Protected Access II), а також більш старої версії стандарту WPA. Вони змогли добитися повторного використання параметра nonce і сесійного (сеансового) ключа в результаті ініціації процедури переустановки ключа шифрування з боку, що атакується клієнта або точки доступу (у деяких випадках).

Жертва Кіберзлочинець

Рисунок 3.1 – Структурна схема системи

Уразливість KRACK при реалізації атаки посередника полегшує зловмисникам дешифрування й інжект пакетів, перехоплення TCP-з'єднання й додавання шкідливого коду в HTTP-контент.

Таким чином, зловмисник при реалізації атаки посередника (Man in the middle) між точкою доступу й клієнтом, порушивши порядок приймання або повторного відправлення повідомлень, може одержати можливість частково маніпулювати синхронізацією й передачею повідомлень у протоколах WPA2 Four-way, Group Key, Fast Basic Service Set (BSS) Transition, Peerkey, Tunneled

Direct-Link Setup (TDLS) Peerkey (TPK), а також Wireless Network Management (WNM) Sleep Mode. Залежно від використовуваного протоколу шифрування даних (WPA-TKIP, AES-CCMP або GCMP) і деяких ситуаційних факторів, ефектом від цих маніпуляцій буде переустановка раніше вже використовуваних сесійних ключів, а також перевантаження лічильників nonces і replay. Як результат, повторне використання ключів полегшує зловмисникам дешифрування й інжект (ін'єкцію) пакетів, перехоплення TCP-з'єднання (TCP connection hijacking), додавання шкідливого коду в HTTP-контент або повторне віщання unicast-, broadcast- і multicast-кадрів.

Для документування цих уразливостей у протоколі WPA2 координаційним центром CERT/CC були призначені наступні ідентифікатори CVE:

- CVE-2017-13077: reinstallation of the pairwise key in the Four-way handshake;
- CVE-2017-13078: reinstallation of the group key in the Four-way handshake;
- CVE-2017-13079: reinstallation of the integrity group key in the Four-way handshake;
- CVE-2017-13080: reinstallation of the group key in the Group Key handshake;
- CVE-2017-13081: reinstallation of the integrity group key in the Group Key handshake;
- CVE-2017-13082: accepting a retransmitted Fast BSS Transition Reassociation Request and reinstalling the pairwise key while processing it;
- CVE-2017-13084: reinstallation of the STK key in the Peerkey handshake;
- CVE-2017-13086: reinstallation of the Tunneled Direct-Link Setup (TDLS) Peerkey (TPK) key in the TDLS handshake;
- CVE-2017-13087: reinstallation of the group key (GTK) when processing a Wireless Network Management (WNM) Sleep Mode Response frame;
- CVE-2017-13088: reinstallation of the integrity group key (IGTK) when processing a Wireless Network Management (WNM) Sleep Mode Response frame.

Суть атаки KRACK

Основна атака KRACK спрямована проти чотирьохетапного рукостискання протоколу WPA2. Воно виконується тоді, коли клієнт прагне приєднатися до захищеної мережі Wi-Fi, і використовується для підтвердження того, що й клієнт, і точка доступу мають правильні облікові дані. Також чотирьохетапне рукостискання служить для твердження нового згенерованого ключа шифрування, який буде використовуватися для шифрування всього наступного трафіку.

У зловмисників з'явилася можливість за допомогою повторного транслювання криптографічних повідомлень рукостискання обдурити пристрій-жертву й спровокувати переустановку вже раніше використаного ключа шифрування.

Коли переустановлюється ключ шифрування, пов'язані з ним параметри, такі як інкрементний номер переданого пакета (nonce) і номер прийнятого пакета (replay counter) скидаються до своїх споконвічних значень. Виявлена уразливість дозволяє активному зловмисникові, який затримує або блокує обмін пакетами між клієнтом і точкою доступу, втручатися в спілкування між точкою доступу й клієнтом. Він може за допомогою повторного транслювання криптографічних повідомлень рукостискання обдурити пристрій-жертву й спровокувати переустановку вже раніше використаного ключа. Таким чином ключовий потік, що випливає, буде ідентичний попередньому ключовому потоку, тому що ті ж самі значення параметра nonce (тобто значення лічильника) використовуються в парі з тим же самим ключем шифрування, який уже раніше використовувався. Як тільки це відбудеться, зловмисник з невеликим зусиллям зможе розшифрувати трафік (приміром, це завдання стає тривіальною, якщо повторно використовуваний ключовий потік здійснює передачу відомого зловмисникові контенту), і в такий спосіб одержати доступ до персональної інформації, яка передається через Wi-Fi-мережу. Властиво, що атакує зможе розшифрувати далеко не всі передані пакети, але, тому що відтепер така можливість стала

реальністю, краще виходити із припущення, що будь-який переданий пакет може бути дешифрований зловмисником.

Подібний принцип здійснення захищеного з'єднання (чотирьохетапне рукостискання) визначається поточною версією набору стандартів бездротовому зв'язку IEEE 802.11, є обов'язковим при сертифікації Wi-Fi-пристроїв і застосунків і, відповідно, використовується всіма сучасними захищеними Wi-Fi-мережами. Це означає, що всі захищені Wifi-мережі у світі вразливі (з певною варіативністю) до атак KRACK. Так, наприклад, атака працює проти персональних і корпоративних мереж Wi-Fi, проти старого стандарту WPA і сучасного WPA2, і навіть проти мереж, які побудовані на використанні тільки захищеного стандарту шифрування ключів AES.

Ключ шифрування встановлюється після одержання клієнтом повідомлення 3 чотирьохетапного рукостискання. По суті, щоб гарантувати безпеку з'єднання, ключ необхідно встановлювати й використовувати тільки один раз. На жаль, протоколом WPA2 це не гарантується. Тому що при передачі по бездротовій мережі повідомлення може бути загублене або перекручене, точка доступу може повторно кілька раз передавати повідомлення 3, якщо вона не одержала відповідного підтвердження від клієнта. Клієнт, у свою чергу, може кілька раз одержувати повідомлення 3 чотирьохетапного рукостискання, щораз переустановлюючи той самий ключ шифрування, а також обнуляючи параметри nonce і replay counter, використовувані протоколом шифрування. Збираючи й ретранслюючи повідомлення чотирьохетапного рукостискання, що атакує може добитися використання того ж ключа шифрування кілька раз. Використовуючи подібну техніку, можна також маніпулювати рукостисканнями протоколів Group Key, Fast Basic Service Set (BSS) Transition, Peerkey, Tunneled Direct-Link Setup (TDLS) Peerkey (TPK), а також кадрами Wireless Network Management (WNM) Sleep Mode.

Наслідку злому WPA2

Можливість дешифрування пакетів може використовуватися для дешифрування пакетів TCP SYN. Це дозволить атакуючому одержати порядкові номери TCP-з'єднання й захопити TCP-сеанс. У результаті в зловмисників з'явилася можливість здійснювати на Wi-Fi-мережі, захищені за допомогою WPA2, найпоширеніші атаки проти відкритих бездротових мереж: додавання шкідливих модулів в HTTP-з'єднання. Приміром, це може бути вставка шкідливого програмного забезпечення в HTTP-дані, які одержує жертва з переглянутих нею веб-сайтів.

Атаки KRACK ефективні не залежно від виробника й операційної системи, якою управляється пристрій.

Вплив атаки може мати особливо катастрофічні наслідки, якщо жертва використовує протоколи шифрування WPA-TKIP або GCMP, а не AES-CCMP. У цьому випадку повторне використання nonce дозволяє атакуючому не тільки розшифрувати, але й змінювати передані пакети. Більше того, GCMP використовує той самий ключ автентифікації в обох напрямках, і цей ключ може бути відтворений завдяки даній атаці. Варто також відзначити, що протокол шифрування GCMP лежить в основі стандарту IEEE 802.11ad (більш відомий як Wireless Gigabit або Wigig), який, як очікується, одержить широке поширення в найближчі парі років.

Напрямок, у якому пакети можуть бути скомпрометовані, залежить від рукостискання, яке атаковано. При звичайній атаці чотирьохетапного рукостискання зловмисник одержує можливість розшифровувати (і в деяких випадках підробляти) пакети, відправлені клієнтом. Але при атаці рукостискання стандарту IEEE 802.11r (Fast BSS Transition, відомого також як швидкий роумінг), можна розшифровувати (і в деяких випадках підробляти) пакети, відправлені від точки доступу клієнтові.

Атака KRACK має важкі наслідки, якщо спрямована проти утиліти «wpa_supplicant» версії 2.4 і вище, яку звичайно використовують Wifi-клієнти під

керуванням різних дистрибутивів Linux, а також Android версії 6.0 і вище. Додаткова проблема полягає в тому, що ключ шифрування обнуляється, тобто очищається з пам'яті після установки в перший раз. Це полегшує атакуючому перехоплення й керування трафіком, переданий пристроями під керуванням цими операційними системами. Тому, як очікується, процес відновлення смартфонів і інших Android-пристроїв, який дозволить ефективно протидіяти даної уразливості, затягнеться. Таким чином, близько 50% сучасних Android-пристроїв в усьому світі вкрай уразливі до виявленої проблеми.

Варто також відзначити, що атака KRACK не здатна визначити пароль, яким захищена Wi-Fi-мережа. Також ця уразливість не зможе або допомогти визначити (повністю або навіть частково) новий ключ шифрування (у тому числі й видати себе за точку доступу або клієнта для одержання нового сесійного ключа), який буде встановлений під час наступного незаблокованого сеансу чотирьохетапного рукостискання. Таким чином, мова йде не про злом протоколу безпеки WPA2, чий процес чотирьохетапного рукостискання залишається надійно захищеним за умови обмеження на встановлення ключа шифрування одним разом, а про уразливість, яка може бути нівельована для конкретного пристрою або застосунку за допомогою назад сумісного відновлення.

Як захиститься від уразливості KRACK?

Властиво, тому що проблема може бути вирішена за допомогою назад сумісного відновлення (тобто оновлений, а значить уже захищений клієнт буде коректно, повноцінно й безпечно для себе взаємодіяти із точкою доступу без установленого на ній відновлення, як і навпаки), те мова про необхідність забути про безпечний Wi-Fi до появи нового стандарту WPA3, на щастя, не йде.

Для повної захищеності від атаки KRACK досить установити відновлення на всі пристрої, що входять в Wi-Fi-мережу (як точки доступу, так і клієнтів).

Для повної захищеності від атаки KRACK досить установити відновлення на всі пристрої, що входять в Wi-Fi-мережу (як точки доступу, так і клієнтів) у міру появи відповідних відновлень у виробників для конкретних застосунків.

Зверніть увагу, що відновлення прошивання точки доступу не завжди може гарантувати безпека підключених до неї клієнтських пристроїв без відповідного патча, якщо про цей прямо не сказане в описі до відновлення прошивання точки доступу. Інакше кажучи, уважно читайте, від яких атак вас захистить свіже відновлення прошивання, а від яких немає. І, звичайно ж, намагайтеся встановлювати відповідне відновлення відразу, як тільки воно з'явиться у вашого виробника.

Заміна пароля ніяк не вплине на ефективність атаки KRACK. Використання VPN і HTTPS, відмова від використання стандарту IEEE 802.11r ускладнить завдання атакуючому, але повністю убезпечити вас не зможе. Тому подібні кроки в жодному разі не повинні сприйматися як застосунок проблеми, а можуть бути лише тимчасовим заходом, поки ви повністю не забезпечите безпеку своєї Wi-Fi-мережі.

Уже очевидно, що обновити операційні системи пристроїв і вбудоване програмне забезпечення точок доступу у вашій бездротовій мережі вдасться далеко не завжди. У цьому, швидше за все, полягає основна проблема від виявленої уразливості. І під погрозою не тільки застарілі пристрої й застосунку, які вже не підтримуються виробниками, але, що й стали раптово вразливими мільйони IoT-пристроїв, чиє спілкування по захищеної Wi-Fi-мережі часто відбувається без якого-небудь додаткового шифрування, і які можуть так і ніколи не одержати свого відновлення безпеки.

3.3 Розробка функціональної схеми

WPA2_KRACK – це повний набір інструментів для оцінки безпеки мережі Wi-Fi.

Він фокусується на різних областях безпеки Wi-Fi:

– Моніторинг: захват пакетів і експорт даних у текстові файли для подальшої обробки сторонніми інструментами.

– Атака: повторні атаки, деавтентифікація, підроблені точки доступу й інші за допомогою ін'єкції пакетів.

– Тестування: перевірка карт Wifi і можливостей драйвера (захват і впровадження).

– Злом WEP і WPA PSK (WPA 1 і 2)

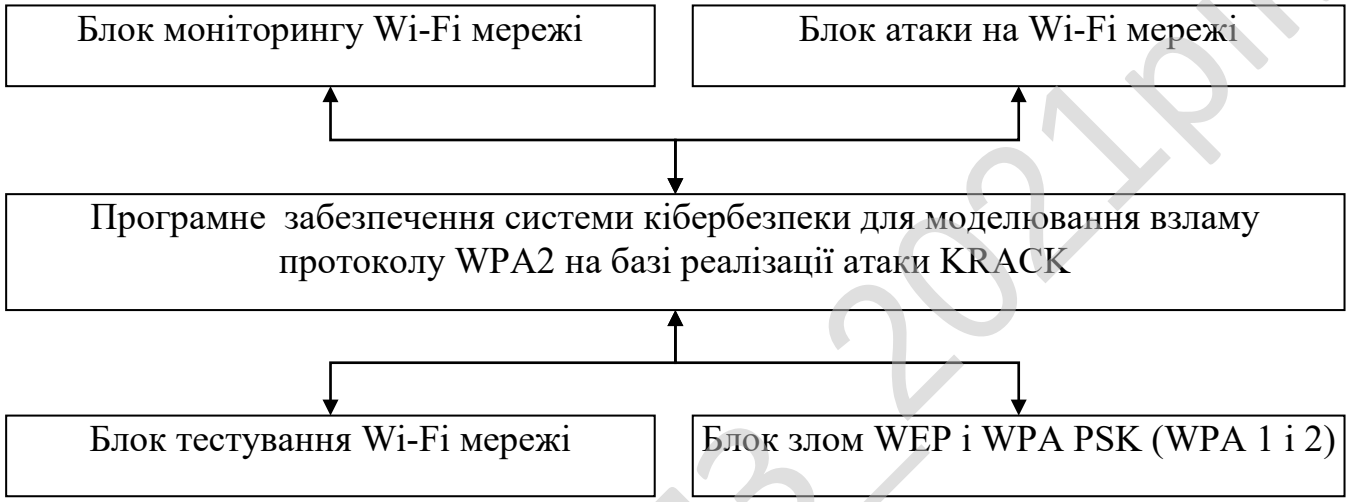

Рисунок 3.2 – Функціональна схема системи

Розглянемо більш детально перераховані вище функції:

– WPA2_KRACK: додана підтримка MidnightBSD.

– WPA2_KRACK: виправлене відображення процесорів ARM за допомогою -u.

– WPA2_KRACK: виправлена підтримка AVX-512F.

– WPA2_KRACK: виправлені розрахунки швидкості злому.

– WPA2_KRACK: виправлений злом WEP за межами 10k IVS.

– WPA2_KRACK: виправлене створення нового сеансу й доданий тестовий приклад.

– WPA2_KRACK: виправлене відображення шифрування в деяких випадках при запиті мережі на злом.

– WPA2_KRACK: виправлений вихід з WPA2_KRACK у деяких випадках.

– WPA2_KRACK: виправлене визначення кількості логічних і фізичних процесорів.

– WPA2_KRACK: виправлена перевірка довжини PMKID.

– WPA2_KRACK: різні виправлення й поліпшення механізму злому WPA і його продуктивності.

– WPA2_KRACK: розшифровувати обоє напрямку при використанні WDS.

– WPA2_KRACK: виправлене дешифрування WPA PCAP при зміні BSSID.

– WPA2_KRACK: додана підтримка WPA3.

– WPA2_KRACK: перемкнутися на argparse.

– WPA2_KRACK: додане виявлення wicd, Intel Wireless Daemon (iwd), net_applet.

– WPA2_KRACK: обробляти випадок, коли avahi продовжує перезапускатися.

– WPA2_KRACK: указує, що інтерфейс не існує.

– WPA2_KRACK: доданий інтерактивний ключ автоколоризації.

– WPA2_KRACK: додана опція для читання PCAP у реальному часі (-T).

– WPA2_KRACK: додане виявлення PMKID.

– WPA2_KRACK: додана підтримка GMAC.

– WPA2_KRACK: додана підтримка WPA3 і OWE (Enhanced Open).

– WPA2_KRACK: базова підтримка UTF-8.

– WPA2_KRACK: перевірені кадри керування завершені перед обробкою IE, щоб уникнути перемикання з WEP на WPA.

– WPA2_KRACK: відображення сигналу при читанні з PCAP.

– WPA2_KRACK: виправлений вивід netxml зі схованим SSID.

– WPA2_KRACK: поліпшений розрахунки швидкості для 802.11n / ac.

– WPA2_KRACK: виправлене з використанням -p з -e.

– WPA2_KRACK: фіксований порядок бібліотек ssl і crypto.

– WPA2_KRACK: виправлений еталон звітів клієнтів.

– WPA2_KRACK: тепер обробляє фрагментоване кодування при спілкуванні (за замовчуванням в Python3).

– WPA2_KRACK: оновлений патч для v3.0.20.

– Загальні: додана підтримка NetBSD endianness.

– Загальні: у скрипти додана підтримка python3.

– Загальні: доданий скрипт для відновлення автоинструментов на Centos 7.

– Загальні: додана політика безпеки для повідомлення про проблеми безпеки.

– Загальні: реорганізація структури файлової системи (див. PR 2032) і перемикання на automake 1.14+.

– Загальні: перетворення в нерекурсивний make (частина PR 2032).

– Загальні: дедупликація функцій і очищення коду.

– Загальні: виправлене впакування на cygwin через зміну імені бібліотеки openssl.

– Загальні: виправлене складання SPARC в Solaris 11.

– Загальні: вилучений coshopss.io.

– Загальні: оновлені залежності в README.md/INSTALLING.

– Загальні: використовувати вихідну бібліотеку радіопередач як піддерево.

– Загальні: різні виправлення й поліпшення (код, CI, інтеграційні тести, покриття).

– Тести: додані інтеграційні тести для aireplay-ng, WPA2_KRACK, WPA2_KRACK, airbase-ng і інших.

– Тести: додані тести на WPA2_KRACK, WPA2_KRACK.

– WPA2_KRACK: виправлена помилка вибору AP в інтерактивному режимі.

– WPA2_KRACK: оновлена функція реєстрації GPS і доданий новий формат реєстрації (logcsv).

– WPA2_KRACK: завантажуйте тільки максимально підтримуваний і доступний крипто-движок.

– WPA2_KRACK: перероблена черга виробників / споживачів списку слів.

– WPA2_KRACK: виправлений зв'язок між платформами з різним розміром int.

– WPA2_KRACK: поліпшене визначення Raspberry Pis.

– Загальні: виправлення порівняння підписаних і непідписаних.

– Пакет: доданий пакет для Ubuntu 18.10 (Cosmic).

– Загальні: очищення коду.

– Загальні: додані додаткові тести.

– Загальні: поліпшення / виправлення компіляції в autotools.

– Загальні: виправлення із прямим порядком байтів.

– Загальні: виправлене складання на FreeBSD і OpenBSD.

– Загальні: додані інструкції з компіляції на DragonflyBSD і OpenBSD.

– Загальні: виправлені орфографічні помилки.

– WPA2_KRACK: доданий злом PMKID.

– WPA2_KRACK: серйозне прискорення й зменшення використання пам'яті при завантаженні більших файлів (декілька гігабайт) з використанням дерев AVL.

– WPA2_KRACK: доданий hwloc (розташування устаткування) для підвищення продуктивності.

– WPA2_KRACK: підтримка злому PCAP за допомогою захищених фреймів керування (802.11w).

– WPA2_KRACK: об'єднані check_thread () і read_thread ().

– WPA2_KRACK: дозволити статичне зв'язування з обраної SIMD.

– WPA2_KRACK: відображення AVX512F при його наявності.

– WPA2_KRACK: додані координати GPS клієнтів у файл Netxml.

– WPA2_KRACK: підвищити швидкість дешифрування / парсинга.

– WPA2_KRACK: оновлене / виправлене використання з nexmon.

– WPA2_KRACK: краща перевірка вимог lspci у системах, у яких немає пристроїв PCI / Pcie.

– WPA2_KRACK: додана підтримка драйвера rtl8812au / 8814au / rtl88xxau.

– Складання: фіксований будинок з декількома різними архітектурами.

– Складання: перехід на новий інструмент CI / CD, Pydeployer.

– Складання: поліпшене складання в Windows (і складання / тестування за допомогою Appveyor).

– Складання: поліпшені й настроєні системи CI (buildbots, Travis, Appveyor).

– Складання: підтримка статичного зв'язування бібліотек / двійкових файлів.

– Складання: пакети автоматичної розробки збираються для декількох дистрибутивів Linux і завантажуються в Packagecloud.io.

– Тести: додані нові тести для WPA2_KRACK.

– Тести: додані нові файли захвата.

– WPA2_KRACK: виправлене відкриття файлу журналу двічі.

– Загальні: виправлене завантаження PCAP у системі з іншим порядком байтів.

– Загальні: виправлені витоки пам'яті й проблеми, про яких повідомляють інструменти статичного аналізу.

– Загальні: виправлена «помилка при завантаженні загальних бібліотек».

– Загальні: різні інші невеликі поліпшення в інструментах, системі складання, тестах і документації.

– Загальні: обновити адресу FSF.

– Загальні: форматування коду.

– WPA2_KRACK: дозволити використання файлів Hashcat HCCAPX у якості вхідних файлів.

– WPA2_KRACK: виправлене виключення із плаваючої коми через розподіл на 0 при відображенні статистики.

– WPA2_KRACK: оновлена довідкова сторінка, що стосується використання шістнадцяткового списку слів.

– WPA2_KRACK: доданий сеанс збереження / відновлення при зломі з використанням списків слів (-N і -R).

– WPA2_KRACK: криптовалюта перенесена у власну бібліотеку aircrack-crypto / (також відому як криптодвижок).

– WPA2_KRACK: Тепер повернемося до єдиного двійкового файлу завдяки крипто-движку.

– WPA2_KRACK: підвищення продуктивності криптографічного движка на різних архітектурах ЦП.

– WPA2_KRACK: додана підтримка AVX512 у крипто-движку (і складанню).

– WPA2_KRACK: поліпшена побудова крипто-движка для архітектур ЦП і компіляторів (gcc, clang і ICC).

– WPA2_KRACK: дозволяє виводити список доступних оптимізацій SIMD.

– WPA2_KRACK: поліпшений інструмент тестування для незвичайної кількості процесорів / ядер.

– WPA2_KRACK: виправлена робота у фоновому режимі.

– WPA2_KRACK: не показувати повідомлення про перехоплення підтвердження WPA для ESSID поза областю дії.

– WPA2_KRACK: додайте –background 0/1, щоб примусово відключити / включити налаштування тла й перевизначити автоматичне визначення тла.

– WPA2_KRACK: додана підтримка GCMP, GCMP-256 і CCMP-256.

– WPA2_KRACK: виправлений імпорт даних при перериванні процесу за допомогою Ctrl-C.

– WPA2_KRACK: перевірте, чи існують словник і файл захвата, перш ніж завантажувати їх.

– WPA2_KRACK: очистите словник перед завантаженням на сервер.

– WPA2_KRACK: по можливості акуратно зупиняти й відображати помилки замість виводу трасування стека.

– WPA2_KRACK: перемістите тимчасові користувацькі файли в / tmp і очистите, коли закінчите.

– WPA2_KRACK: виправлене відображення статусу, коли до сервера не підключені клієнти.

– WPA2_KRACK: перевірити завантажений PCAP і відобразити успіх / невдачу.

– WPA2_KRACK: Поліпшене видалення BSSID.

– Wpaclean: виправлений збій з неприпустимим PCAP заголовка prism2 і додані тести.

– Wpaclean: не створювати вихідний файл, якщо немає підтвердження або якщо вхідний файл поганої.

– Wpaclean: виправлений витік пам'яті.

– WPA2_KRACK: виправлене відображення імені інтерфейсу, якщо ім'я занадто довге.

– Osdep: виправлений витік пам'яті й розіменування нульового покажчика.

– Osdep: виправлене падіння макросів byteorder і інших інструментів.

– Osdep: додана підтримка складання як поділюваної бібліотеки.

– WPA2_KRACK: оновлені інструкції для v3.0.17.

– WPA2_KRACK: додане ведення журналу Response-Identity і відображення хеша NETNTLM у форматі hashcat.

– WPA2_KRACK: dump-join перейменований в airodump-join.

– Загальні: доданий значок coshopss.io.

– Загальні: доданий параметр –without-opt для відключення захисту стека при використанні GCC> = 4.9.

– Загальні: різні поліпшення й виправлення, у тому числі від Coverity Scan і Valgrind.

– Загальні: виправлені друкарські помилки, виявлені при написанні коду.

– Загальні: витягнуті функції консолі й перенесені в aircrack-util /.

– Загальні: osdep / перейменований в aircrack-osdep /.

– Загальні: виправлені й додані прототипи функцій і закомментовані невикористовувані функції.

– Загальні: переформатувати вихідний код з використанням формату clang і додати файл формату .clang для IDE.

– Загальні: Поліпшення в складаннях Appveyor і Travisci.

– Будівництво: додана внутрішня підтримка NEON.

– Будівництво: Підтримка шляхів, що містять пробіли, під час автоконфигурації.

– Складання: виправлена компіляція без getauxval у двійковому файлі trampoline.

– Складання: виправлені попередження компілятора в Windows, FreeBSD.

– Складання: виправлення й документація для OSX.

– Складання: додана підтримка tcmalloc і jemalloc.

– Складання: додана інструкція зі складання двійкових файлів Windows за допомогою Airpcap.

– Модульний тест: використання Cmocka для деяких тестів.

– Документація: оновлені пояснення по створенню деяких експериментальних інструментів.

3.4 Розробка діаграми процесів

Відповідно до методичних рекомендацій розроблення графічної частини кваліфікаційної бакалаврської роботи розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3.

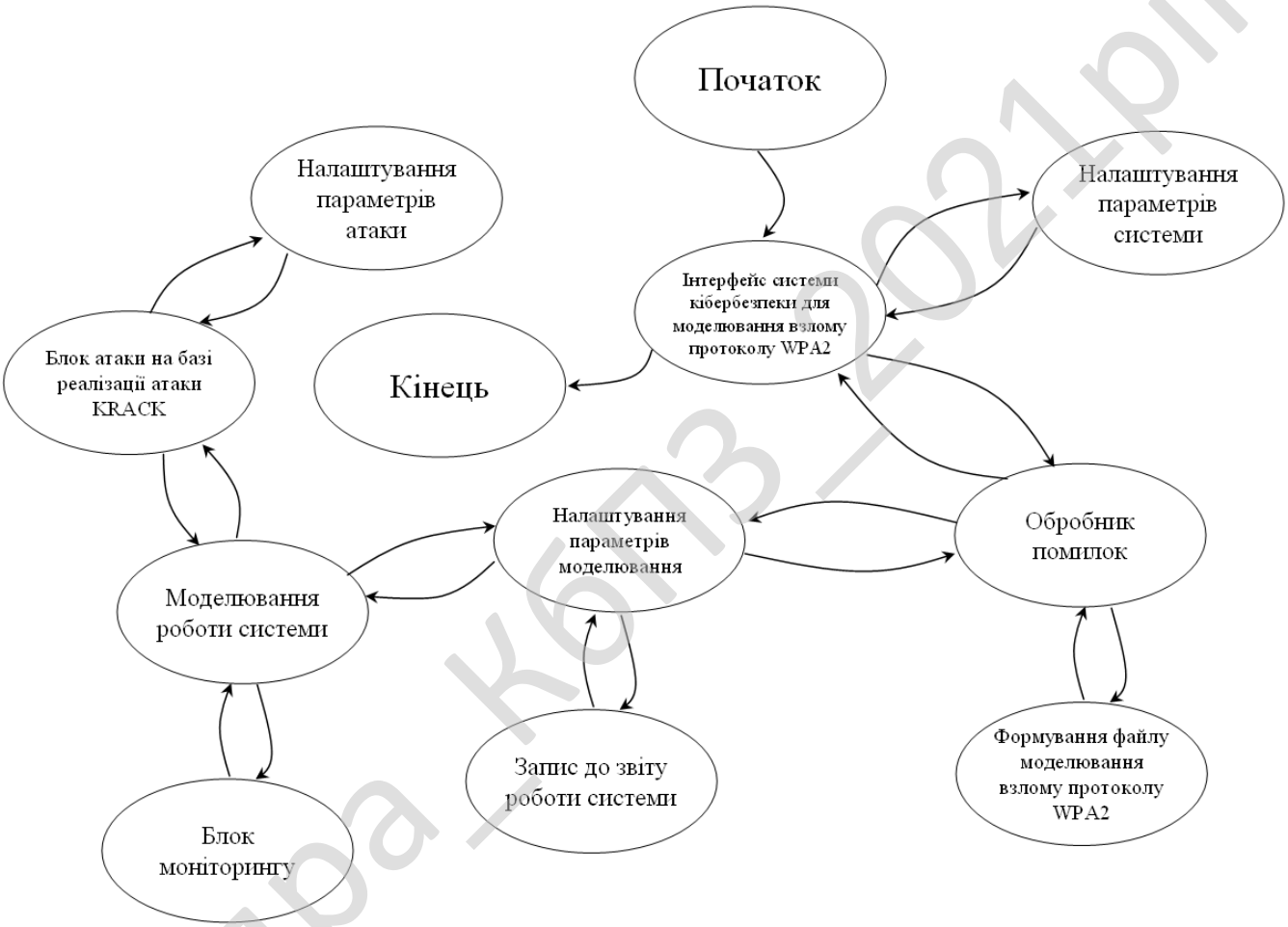

Рисунок 3.3 – Діаграма взаємодії процесів

Розроблена діаграма взаємодії процесів використовується для представлення та візуалізації процесів обробки даних тобто структурного проектування бакалаврської роботи.

Основні складові елементи діаграми взаємодії процесів це потоки даних:

– Репозиторії, потік сховища даних.

– Потоки зовнішні по відношенню до системи сутності.

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Потоки даних гібридні між елементами трьох попередніх типів.

Відповідно до документації основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Для схематичного представлення системи що розробляється необхідно спочатку представити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи в цілому у подальшому. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі. Розроблена діаграма взаємодії процесів системи в подальшому уточнюється шляхом деталізації процесів та потоків даних з метою показати систему що розробляється. Таким чином у результаті після розгляду, вищеописаної системи, схеми структурної, функціональної, діаграми взаємодії процесів перейдемо до опису та розгляду блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо послідовність дій та викликів підпрограм в загальному алгоритмі роботи основної програми що зображено на рисунку 4.1 у вигляді блок-схеми:

– Виведення вікна системи кібербезпеки для моделювання взлому протоколу WPA2.

– Налаштування параметрів моделювання взлому протоколу WPA2.

– Завантажити файл WPA2.

– Завантаження файлу WPA2 та системи реалізації.

– Встановлено параметри моделювання.

– Моделювання роботи системи.

– Виведення даних на екран.

– Реалізація атаки KRACK.

– Підпрограма системи реалізації атаки KRACK рисунок 4.2.

– Файл проаналізовано успішно.

– Виведення на екран результату аналізу.

– Обчислення показників стійкості застосованого алгоритму.

– Виведення на екран значень показників стійкості.

– Виведення рекомендацій по забезпеченню стійкості системи.

– Запит завершення роботи системи.

На рисунку 4.2 зображено роботу підпрограми з реалізацією наступних дій:

– Є файл реалізації атаки KRACK.

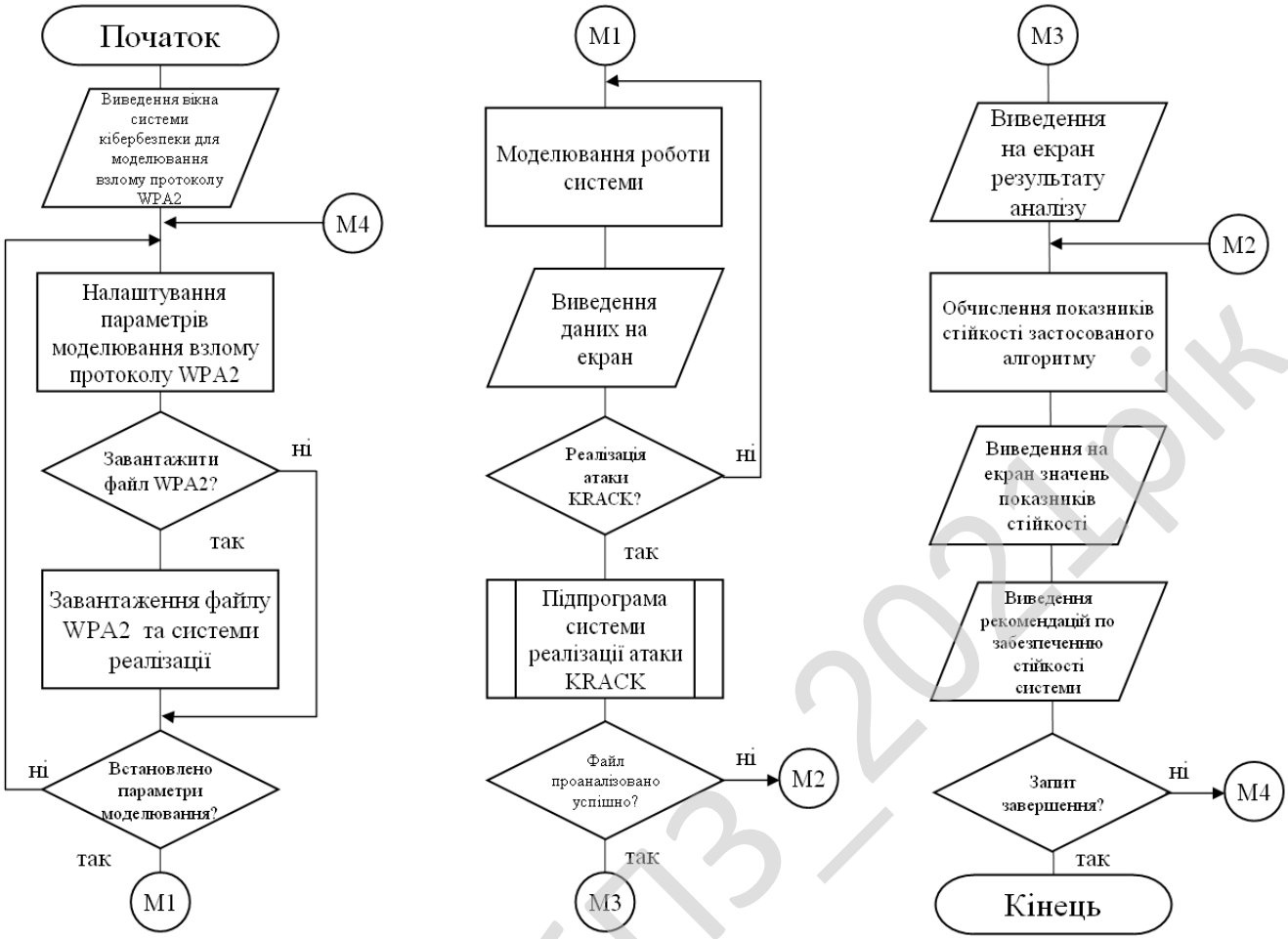

Рисунок 4.1 – Блок-схема основної програми

– Створення файлу поводження нормального трафіку протоколу WPA2.

– Формування базової інформації трафіку.

– Визначення інтенсивності пакетів для кожного типу пакетів.

– Визначення співвідношення пакетів даних.

– Визначення кількості одночасних TCP-з'єднань, відкритих одним джерелом.

– Запис до журналу роботи ПЗ.

– Моделювання взлому протоколу WPA2 на базі реалізації атаки KRACK.

– Сформовано файл атаки.

– Запис поточного файлу атаки на диск.

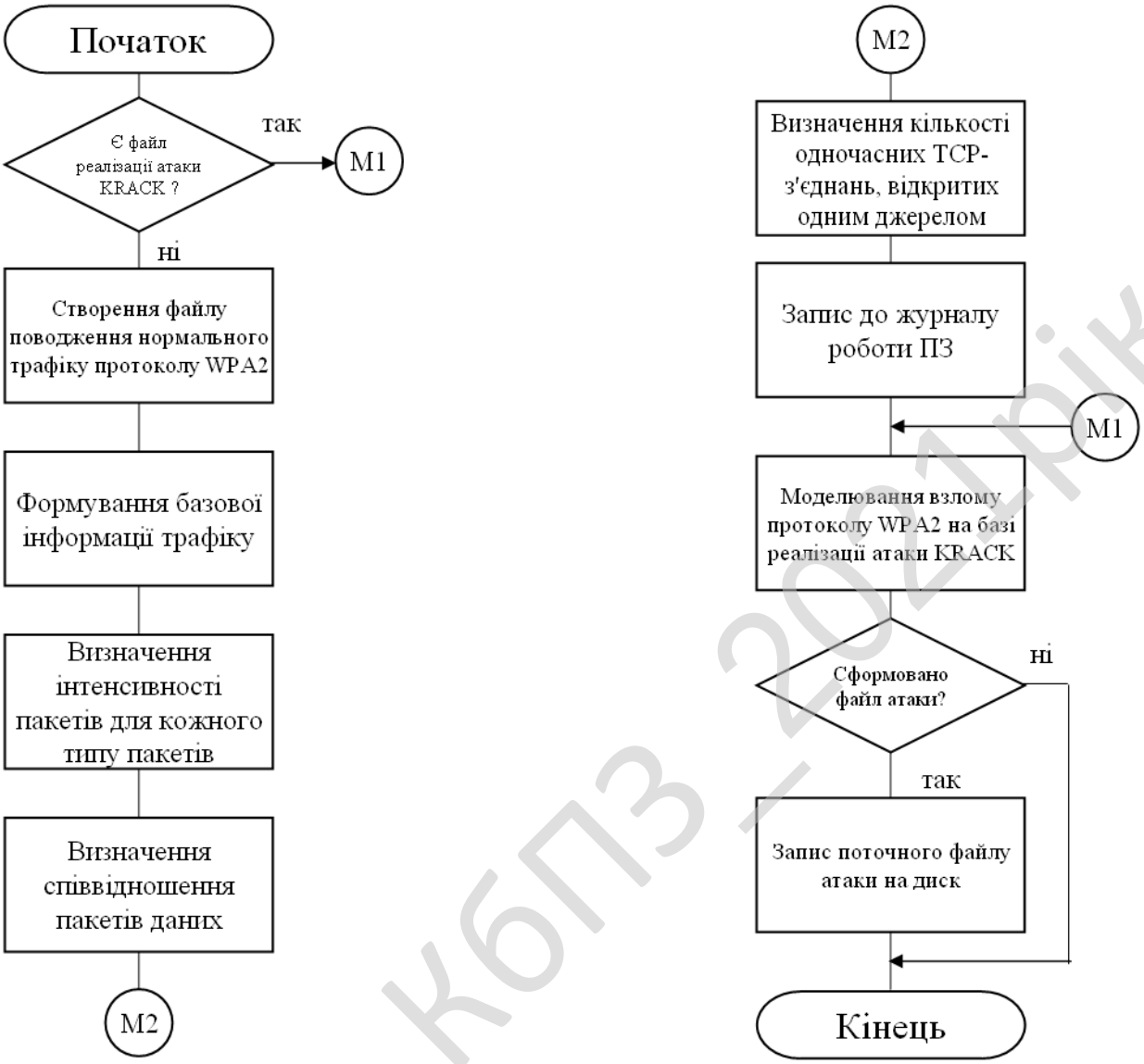

Рисунок 4.2 – Блок-схема роботи підпрограми

Розглянемо модуль реалізації процесу модулювання роботи з взламу протоколу WPA2 на базі реалізації атаки KRACK у вигляді вихідного коду:

```
unit Model_WPA2; // назва

interface // інтерфейс

uses // підключення
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, Buttons, Spin;
```

```
Type / типи
  TWPA2_1 = class(TForm) // об'ява класу
    StringGrid1: TStringGrid;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    ComboBox1: TComboBox;
    BitBtn1: TBitBtn;
    GroupBox2: TGroupBox;
    BitBtn3: TBitBtn;
    BitBtn2: TBitBtn;
    Button1: TButton;
    BitBtn4: TBitBtn;
    SpinEdit1: TSpinEdit;
    Label6: TLabel;
    procedure FormShow(Sender: TObject);
    procedure stringgrid1DrawCell(sender:TObject;
           ACol,ARow:integer; Rect:TRect; state:TGridDrawState);
    procedure Button1Click(Sender: TObject);
    procedure step1_KRACK(Sender: TObject);
    function absx(counter:integer):integer;
    function absy(counter:integer):integer;
    procedure SpinEdit1Change(Sender: TObject);
    procedure step2(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure StringGrid1SelectCell(Sender: TObject; ACol, ARow: Integer;
      var CanSelect: Boolean);
    procedure StringGrid1MouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
  end;

var
  Form1: TWPA2_1; // екземпляр

Implementation // реалізація процедур та функцій

{$R *.dfm} // ресурс

function TWPA2_1.absx(counter:integer):integer;
```

```
begin
result:=counter mod stringgrid1.rowCount;
end;

function TWPA2_1.absy(counter:integer):integer;
begin
result:=(counter div stringgrid1.colCount);
end;

// робота з канвою
procedure TWPA2_1.stringgrid1DrawCell(sender:TObject;
       ACol,ARow:integer; Rect:TRect; state:TGridDrawState);
begin
case strtoint(stringgrid1.Cells[ACol,Arow]) of
0:stringgrid1.Canvas.Brush.Color:=clgreen;
1:stringgrid1.Canvas.Brush.Color:=clred;
2:stringgrid1.Canvas.Brush.Color:=clwhite;
3:stringgrid1.Canvas.Brush.Color:=clblack;
end;
StringGrid1.canvas.fillRect(Rect);
end;

// реалізація частину при оновлення/показу форми
procedure TWPA2_1.FormShow(Sender: TObject);
var i,j,k:byte;
begin
stringgrid1.OnDrawCell:=stringgrid1drawcell;
randomize;
for i:=0 to stringgrid1.ColCount-1 do
for j:=0 to stringgrid1.RowCount-1 do
  begin
  stringgrid1.Cells[i,j]:=inttostr(random(4));
  for k:=1 to combobox1.ItemIndex do
  if stringgrid1.Cells[i,j]='3' then
stringgrid1.Cells[i,j]:=inttostr(random(4));
  end;
bitbtn2.Enabled:=true;
bitbtn2.SetFocus;
end;
// редагування
procedure TWPA2_1.SpinEdit1Change(Sender: TObject);
begin
```

```
if spinedit1.Text<>'' then
begin
Stringgrid1.ColCount:=spinedit1.Value;
Stringgrid1.rowCount:=spinedit1.Value;
end;
end;

procedure TWPA2_1.Button1Click(Sender: TObject);
begin
close;
end;
// перший крок алгоритму
procedure TWPA2_1.step1_KRACK(Sender: TObject);
var counter,step:integer;
    x,y:byte;
    tmp:string;
begin
for counter:=0 to (stringgrid1.rowCount*stringgrid1.ColCount)-2 do
begin
x:=absx(counter);
y:=absy(counter);

if (stringgrid1.Cells[x,y]='1') or (stringgrid1.Cells[x,y]='2') then
  begin
  step:=counter;
  repeat
  inc(step);
  if (stringgrid1.Cells[absx(step),absy(step)]='0') then
    begin
    tmp:=stringgrid1.Cells[x,y];
    stringgrid1.Cells[x,y]:=stringgrid1.Cells[absx(step),absy(step)];
    stringgrid1.Cells[absx(step),absy(step)]:=tmp;
    end;
  until (stringgrid1.Cells[absx(step),absy(step)]='0')or
 (step=(stringgrid1.rowCount*stringgrid1.ColCount)-1);
// кінцеві умови циклу
  end;
end;

if bitbtn2.Enabled then
begin
bitbtn2.Enabled:=false;
```

```
bitbtn3.Enabled:=true;
end
else
begin
bitbtn4.Enabled:=false;
bitbtn1.SetFocus;
end;
end;
// другий крок алгоритму
procedure TWPA2_1.step2(Sender: TObject);
var counter,step:integer;
    x,y:byte;
    tmp:string;
begin
for counter:=0 to (stringgrid1.rowCount*stringgrid1.ColCount)-2 do
begin
x:=absx(counter);
y:=absy(counter);
if (stringgrid1.Cells[x,y]='1') or (stringgrid1.Cells[x,y]='2') then
  begin
  step:=counter;
  repeat
  inc(step);
// перевірка початкова
  if (stringgrid1.Cells[absx(step),absy(step)]='1')or
(stringgrid1.Cells[absx(step),absy(step)]='2') then
    if (stringgrid1.Cells[x,y]='2') then
    begin
    tmp:=stringgrid1.Cells[x,y];
    stringgrid1.Cells[x,y]:=stringgrid1.Cells[absx(step),absy(step)];
    stringgrid1.Cells[absx(step),absy(step)]:=tmp;
    end
    else if (stringgrid1.Cells[x,y]='1') then
    begin
    stringgrid1.Cells[x,y]:='2';
    stringgrid1.Cells[absx(step),absy(step)]:='0';
    end;
  until (stringgrid1.Cells[absx(step),absy(step)]='0')or
(step=(stringgrid1.rowCount*stringgrid1.ColCount)-1);
// кінцеві умови циклу
  end;
end;
```

```
bitbtn3.Enabled:=false;
bitbtn4.Enabled:=true;
end;

procedure TWPA2_1.BitBtn4Click(Sender: TObject);
begin
step1_KRACK(self);
end;
// натиснення кнопки миші
procedure TWPA2_1.StringGrid1MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
if ssCtrl in shift then
stringgrid1.Cells[stringgrid1.col,stringgrid1.Row]:='3'
else
stringgrid1.Cells[stringgrid1.col,stringgrid1.Row]:='1'
end;

end.
```

NetBIOS (Network Basic Input/Output System) – протокол для роботи в локальних мережах на персональних ЕОМ типу IBM/PC, розроблений у вигляді інтерфейсу, який не залежить від фірми–виробника. Він включає в себе інтерфейс сеансового рівня (англ. NetBIOS interface), в якості транспортних протоколів використовує TCP і UDP.

Особливістю NetBIOS є можливість його роботи поверх різних протоколів, найпоширенішими/відомими з яких є NetBEUI, IPX і стек протоколів TCP/IP; причому якщо старі версії Windows орієнтувалися на більш легкі в реалізації і менш ресурсомісткі NetBEUI і IPX, то сучасні Windows орієнтуються на TCP/IP.

При використанні NetBEUI і IPX NetBIOS сам забезпечує надійність доставки даних (функціональність SPX не використовувати), а при використанні TCP/IP надійність доставки забезпечує TCP, за що удостоївся окремого імені «NBT».

Інтерфейс NetBIOS являє собою типовий інтерфейс взаємодії програм (API) для забезпечення мережевих операцій введення–виведення і управління транспортним протоколом.

Програми, що використовують NetBIOS API інтерфейс, можуть працювати тільки при наявності протоколу, що допускає використання такого інтерфейсу.

NetBIOS також визначає протокол, що функціонує на сеансовому/транспортному рівнях моделі OSI. Цей протокол використовується протоколами нижчих рівнів, такими як NBFP (NetBEUI) і NetBT для виконання мережевих запитів вводу–виводу і операцій, описаних в стандартному интерфейсном наборі команд NetBIOS.

Тобто, NetBIOS сам не підтримує виконання файлових операцій. Ця функція покладається на протоколи нижчих рівнів, а сам NetBIOS забезпечує тільки зв'язок з цими протоколами і NetBIOS API–інтерфейс.

Пакет NETBIOS створений для використання групою ЕОМ. Підтримує як режим сесій (робота через з'єднання), так і режим дейтаграм (без встановлення з'єднання). 16–символьні імена об'єктів в NETBIOS розподіляються динамічно.

NETBIOS має власну DNS, яка може взаємодіяти з інтернетівський. Ім'я об'єкта при роботі з NETBIOS не може починатися з символу *.

Додатки можуть знайти через NETBIOS потрібні їм ресурси, встановити зв'язок і послати або отримати інформацію. NETBIOS використовує для служби імен порт 137, для служби дейтаграм – порт 138, а для сесій – порт 139. Будь–яка сесія починається з NETBIOS–запиту, завдання IP–адреси і визначення TCP–порту віддаленого об'єкта, далі йде обмін NETBIOS–повідомленнями, після чого сесія закривається.

Сесія здійснює обмін інформацією між двома NETBIOS – додатками. Довжина повідомлення лежить в межах від 0 до 131 071 байт. Припустимо одночасне встановлення декількох сесій між двома об'єктами.

При організації IP–транспорту через NETBIOS IP–дейтаграмма вкладається в NETBIOS–пакет. Інформаційний обмін відбувається в цьому випадку без встановлення зв'язку між об'єктами. Імена NETBIOS повинні містити в собі IP–адреси. Так, частина NETBIOS–адреси може мати вигляд IP.XX.XX. XX. XX, де IP вказує на тип операції (IP через Netbios), а XX. XX. XX.XX – IP– адреса.

Система NETBIOS має власну систему команд (call, listen, hang up, send, receive, session status, reset, cancel, adapter status, unlink, remote program load) і примітивів для роботи з дейтаграммами (send datagram, send broadcast datagram, receive datagram , receive broadcast datagram).

Всі кінцеві вузли NETBIOS діляться на три типи:

– широкомовні («b») вузли;

– вузли точка–точка («p»);

– вузли змішаного типу («m»).

IP–адреса може асоціюватися з одним із зазначених типів. B–вузли встановлюють зв'язок зі своїм партнером за допомогою широкомовних запитів. P– і M–вузли для цієї мети використовують netbios сервер імен (NBNS) і сервер розподілу дейтаграм (NBDD).

NetBIOS забезпечує:

– реєстрацію і перевірку мережевих імен;

– встановлення і розрив з'єднань;

– зв'язок з підтвердженням доставки інформації;

– зв'язок без підтвердження доставки інформації;

– підтримку управління і моніторингу драйвера і мережевої карти.

Розглянемо протокол SNMP (Simple Network Management Protocol, простий протокол керування мережею) – це протокол керування мережами зв'язку на основі архітектури TCP/IP.

На основі концепції TMN в 1980–1990 р. різними органами стандартизації був вироблений ряд протоколів керування мережами передачі даних з різним спектром реалізації функцій TMN. До одного з типів таких протоколів керування належить Simple Network Management Protocol (SNMP).

SNMP – це технологія, покликана забезпечити керування й контроль за пристроями й програмами в мережі зв'язку шляхом обміну керуючою інформацією між агентами, що розташовуються на мережних пристроях, і менеджерами, розташованими на станціях керування. SNMP визначає мережу як

сукупність мережних керуючих станцій й елементів мережі (головні машини, шлюзи й маршрутизатори, термінальні сервери), які спільно забезпечують адміністративні зв'язки між мережними керуючими станціями й мережними агентами. SNMP різних версій присвячений цілий ряд рекомендацій IETF (RFC).

Зазвичай при використанні SNMP присутні керовані та керуючі системи. До складу керованої системи входить компонент, який називається агентом, який відправляє звіти керуючій системі.

По суті SNMP а генти передають управлінську інформацію на керуючі системи як змінні (такі як «вільна пам'ять», «ім'я системи», «кількість працюючих процесів» тощо).

Керуюча система може отримати достовірну інформацію через операції протоколу GET, GETNEXT і GETBULK. Агент може самостійно без запиту надсилати дані, використовуючи операцію протоколу TRAP або INFORM.

Управляючі системи можуть також відправляти конфігураційні оновлення або контролюючі запити, використовуючи операцію SET для безпосереднього управління системою. Операції конфігурування та управління використовуються тільки тоді, коли потрібні зміни у мережній інфраструктурі. Операції моніторингу зазвичай виконуються на регулярній основі.

Змінні, доступні через SNMP, організовані в ієрархії. Ці ієрархії та інші метадані (такі як тип і опис змінної) описуються Базами Керуючої Інформації' (Management Information Bases (MIBs)).

SNMP не визначає, яку інформацію (які змінні) керована система повинна надавати. Навпаки, SNMP використовує розширювану модель, в якій доступна інформація визначається Базами Керуючої Інформації (MIB – Management Information Base).

Бази Керуючої Інформації описують структуру керуючої інформації пристроїв. Вони використовують ієрархічний адресний простір імен, що містить унікальний ідентифікатор об'єкта (object identifier (OID)).

Грубо кажучи, кожен унікальний ідентифікатор об'єкта ідентифікує змінну, яка може бути прочитана чи встановлена через SNMP. MIB'и використовують нотацію, визначену в ASN.1.

Ієрархія MIB може бути зображена як дерево з безіменним коренем, рівні якого приписані різними організаціями. На найвищому рівні MIB OID'и належать різним організаціям, що займаються стандартизацією, в той час як на нижчих рівнях OID'и виділяються асоційованими організаціями. Ця модель забезпечує управління на всіх шарах мережної моделі OSI, адже MIB'и можуть бути визначені для будь-яких типів даних і операцій.

Керований об'єкт – це одна з будь-якого числа характеристик, специфічних для керованого пристрою. Керований об'єкт включає в себе один або більше екземплярів об'єкта (що ідентіфікуються за OID), які насправді є змінними.

Існує два типи керованих об'єктів: Скалярні об'єкти визначають єдиний екземпляр об'єкта; Табличні об'єкти визначають множинні, пов'язані екземпляри об'єктів які групуються в таблицях MIB.

Прикладом керованого об'єкта може бути atInput, який є скалярним об'єктом що містить єдиний екземпляр об'єкта, ціле число, яке показує загальну кількість вхідних пакетів AppleTalk на мережний інтерфейс маршрутизатора.

Ідентифікатор об'єкта (OID) унікально ідентифікує керований об'єкт в ієрархії MIB.

В телекомунікаціях і комп'ютерних мережах, ASN.1 є стандартною гнучкою нотацією для опису структур даних, що служать для кодування, передачі і декодування даних. ASN.1 являє собою набір правил для опису структури об'єктів, незалежних від специфічних для обладнання методик кодування, і формальну нотацію, яка дозволяє уникнути неоднозначностей.

ASN.1 це єдиний ISO та ITU-T стандарт, спочатку визначений у 1984 році як частина стандарту CCITT X.409: 1984. Пізніше, в 1988 році, завдяки його широкому застосуванню, він був виділений в окремий стандарт X.208. Значно переглянута версія 1995року описана в X.680.

Адаптована підмножина ASN.1 Структура Керуючої Інформації (SMI) описана в протоколі SNMP для визначення наборів пов'язаних MIB об'єктів, також званих MIB модулями.

SNMP працює на прикладному рівні TCP/IP (сьомий рівень моделі OSI). Агент SNMP отримує запити по UDP-порту 161. Менеджер може посилати запити з будь-якого доступного порту джерела на порт агента. Відповідь агента буде відправлений назад на порт джерела на менеджері. Менеджер отримує повідомлення (Traps і InformRequests) по порту 162. Агент може генерувати повідомлення з будь-якого доступного порту. При використанні TLS або DTLS запити виходять по порту 10161, а пастки відправляються на порт 10162.

У SNMPv1 зазначено п'ять основних протокольних одиниць обміну (protocol data units - PDU). Ще дві PDU, GetBulkRequest і InformRequest, були введені в SNMPv2 і перенесені в SNMPv3.

Нижче перераховані сім протокольних одиниць обміну SNMP:

1. GetRequest. Запит від менеджера до об'єкту для отримання значення змінної або списку змінних. Необхідні змінні вказуються в полі variable bindings (розділ поля values при цьому не використовується). Отримання значень зазначеної змінної повинно бути виконано агентом як Атомарна операція. Менеджеру буде повернений Response (відповідь) з поточними значеннями.

2. SetRequest. Запит від менеджера до об'єкту для зміни змінної або списку змінних. Зв'язані змінні вказуються в тілі запиту. Зміни всіх зазначених змінних повинні бути виконані агентом як атомарна операція. Менеджеру буде повернений Response з (поточними) новими значеннями змінних.

3. GetNextRequest. Запит від менеджера до об'єкту для виявлення доступних змінних і їх значень. Менеджеру буде повернений Response зі зв'язаними змінними для змінної, яка є наступною в базі MIB в лексіграфіческом порядку. Обхід всієї бази MIB агента може бути проведений ітераційним використанням GetNextRequest, починаючи з OID 0. Рядки таблиці можуть бути прочитані, якщо вказати в запиті OID-и колонок в пов'язаних змінних.

4. GetBulkRequest. Покращена версія GetNextRequest. Запит від менеджера до об'єкту для численних ітерацій GetNextRequest. Менеджеру буде повернений Response з декількома пов'язаними змінними, обійденими починаючи зі пов'язаної змінної (змінних) в запиті. Специфічні для PDU поля non-repeaters і max-repetitions використовуються для контролю за поведінкою відповіді. GetBulkRequest був введений в SNMPv2.

5. Response. Повертає зв'язані змінні і значення від агента менеджеру для GetRequest, SetRequest, GetNextRequest, GetBulkRequest і InformRequest. Повідомлення про помилки забезпечуються полями статусу помилки і індексу помилки.

Ця одиниця використовується як відповідь і на Get-, і на Set-запити, в SNMPv1 називається GetResponse.

6. Trap. Асинхронне повідомлення від агента - менеджеру. Включає в себе поточне значення sysUpTime, OID, що визначає тип trap (пастки), і необов'язкові зв'язані змінні. Адресація одержувача для пасток визначається за допомогою змінних trap-конфігурації в базі MIB. Формат trap-повідомлення був змінений в SNMPv2 і PDU перейменували в SNMPv2-Trap.

7. InformRequest. Асинхронне повідомлення від менеджера менеджеру або від агента менеджеру. Повідомлення від менеджера менеджеру були можливі вже в SNMPv1 (за допомогою Trap), але SNMP зазвичай працює на протоколі UDP, в якому доставка повідомлень не гарантована, і не повідомляється про втрачені пакетах. InformRequest виправляє це зворотним відправленням підтвердження про отримання. Одержувач відповідає Response-му, що повторює всю інформацію з InformRequest. Цей PDU був введений в SNMPv2.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм Blowfish, який є симетричним алгоритмом

шифрування, тобто таким, у якому ключ шифрування дорівнює ключу дешифрування. Він є мережею Фейштеля, у якій кількість ітерацій дорівнює 16. Довжина блоку дорівнює 64 бітам, ключ може мати будь-яку довжину в межах 448 біт. Хоча перед початком будь-якого шифрування виконується складна фаза ініціалізації, саме шифрування даних виконується досить швидко.

Алгоритм призначений в основному для застосунків, у яких ключ міняється нечасто, до того ж існує фаза початкового рукостискання, під час якої відбувається автентифікація сторін і узгодження загальних параметрів і секретів. При реалізації на 32-бітних мікропроцесорах з більшим кешем даних Blowfish значно швидше DES.

Алгоритм складається із двох частин: розширення ключа й шифрування даних. Розширення ключа перетворює ключ довжиною, принаймні, 448 біт у кілька масивів підключів загальною довжиною 4168 байт.

В основі алгоритму лежить мережа Фейштеля з 16 ітераціями. Кожна ітерація складається з перестановки, що залежить від ключа, і підстановки, що залежить від ключа й даних. Операціями є XOR і додавання 32-бітних слів.

Blowfish використовує велику кількість підключів. Ці ключі повинні бути обчислені заздалегідь, до початку будь-якого шифрування або дешифрування даних. Елементи алгоритму:

1. P – масив, що складається з вісімнадцяти 32-бітних підключів:

$$P_1, P_2, ..., P_{18}.$$

2. Чотири 32-бітних S-boxes с 256 входами кожний. Перший індекс означає номер S-box, другий індекс – номер входу.

$$S_{1,0}, S_{1,1}, \ldots S_{1,255};$$

$$S_{2,0}, S_{2,1}, \ldots S_{2,255};$$

$$S_{3,0}, S_{3,1}, \ldots S_{3,255};$$

$$S_{4,0}, S_{4,1}, \ldots S_{4,255};$$

Шифрування

Входом є 64-бітний елемент даних X, що ділиться на дві 32-бітні половини, X_l і X_r.

$$X_l = X_l \text{ XOR } P_i$$

$$X_r = F(X_l) \text{ XOR } X_r$$

$$\text{Swap } X_l \text{ and } X_r$$

Функція F

Розділити X_l на чотири 8-бітних елементи A, B, C, D.

$$F(X_l) = ((S_{1,A} + S_{2,B} \bmod 2^{32}) \text{ XOR } S_{3,C}) + S_{4,D} \bmod 2^{32}$$

Дешифрування відрізняється від шифрування тим, що P_i використовуються у зворотному порядку.

Генерація підключів

Підключи обчислюються з використанням самого алгоритму Blowfish.

1. Ініціалізувати перший P-масив і чотири S-boxes фіксовані рядки.

2. Виконати операцію XOR P_1 з першими 32 бітами ключа, операцію XOR P_2 із другими 32 бітами ключа й т.д. Повторювати цикл доти, поки весь P-масив не буде побітово складний з усіма бітами ключа. Для коротких ключів виконується конкатенація ключа із самим собою.

3. Зашифрувати нульовий рядок алгоритмом Blowfish, використовуючи підключи, описані в пунктах (1) і (2).

4. Замінити P_1 і P_2 виходом, отриманим на кроці (3).

5. Зашифрувати вихід кроку (3), використовуючи алгоритм Blowfish з модифікованими підключами.

6. Замінити P_3 і P_4 виходом, отриманим на кроці (5).

7. Продовжити процес, заміняючи всі елементи P-масиву, а потім всі чотири S-boxes, виходами відповідним чином модифікованого алгоритму Blowfish.

Для створення всіх підключів потрібна 521 ітерація.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програма має простий та інтуїтивно зрозумілий інтерфейс, на рисунку 51. зображено під вікно введення даних. З рисунку вікна можна побачити що інтерфейс вікна розподілено на наступні функціональні розділи:

– Функціональних кнопок ПЗ: Запуск; Вихід.

– Верхнього меню: Файл; Вид; Робота з ключами; Параметри Довідка.

– Розділу введення мережних даних: IP адреси; Порт; Алгоритм шифрування; Початковий спільний ключ; Часовий параметр зміни ключа.

Рисунок 5.1 – Вікно введення даних ПЗ

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення. Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких

помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

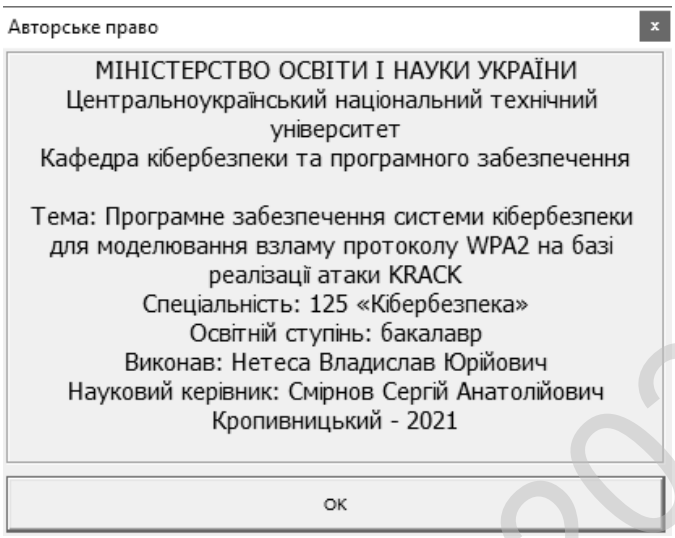

Рисунок 5.2 – Авторське право

Процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Таким чином у результаті вищерозглянутого можна стверджувати що розроблено інтерфейс системи у відповідності з вибраною метою роботи. Система містить максимальний необхідний набір функцій придатних для виконання будь-яких дій для забезпечення повноцінної роботи програми. Далі розглянемо висновки та використані літературні джерела.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання кваліфікаційної бакалаврської роботи, призначено для системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

– Досліджена система кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

– На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

Розроблені під час виконання кваліфікаційної бакалаврської роботи алгоритми дозволяють успішно вирішувати завдання для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створені програмного забезпечення було використано об’єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня RAD Studio Delphi. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK. Це дозволило мінімізувати строк розробки програмного

забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Blowfish.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гласс Р. Креативное программирование: Пер. с англ. / Роберт Гласс. – СПб.: Символ, 2009. – 352с.

2. Гленфорд М. Искусство тестирования программ / Гленфорд Майерс, Том Баджетт, Кори Сандлер. – М.: Диалектика-Вильямс, 2007. – 272 с.

3. Глухова Л.А. Технологии разработки программного обеспечения: Учебн. пос. / Л.А. Глухова. – Минск: БГУИР, 2007. – 178 с.

4. Глухова Л.А. Основы алгоритмизации и структурного проектирования программ / Л.А. Глухова, В.В. Бахтизин. – Минск.: БГУИР, 2003. – 72 с.

5. Глушков В.М. Синтез цифровых автоматов / В.М. Глушков. – М.: Физматгиз, 1962. – 476 с.

6. Громов Ю.Ю. Надёжность информационных систем: Учебное пособие / Ю.Ю. Громов, О.Г. Иванова, Н.Г. Мосягина, К.А. Набатов. – Тамбов: Изд-во ГОУ ВПО ТГТУ, 2010. – 160 с.

7. Громов Ю.Ю. Управление данными: Учебное пособие / Громов Ю.Ю., Иванова О.Г., Точка В.Н. – Тамбов: Изд-во ТГТУ, 2009. – 80 с.

8. Данылив И.В. Движущие силы развития Case-средств / Данылив И.В., Лукашин С.О., Назаренко С.Г. // Корпоративные системы. – 2002. – № 1. – С. 31 - 35.

9. Дастин Э. Автоматизированное тестирование программного обеспечения / Элфрид Дастин, Джефф Рэшка, Джон Пол. – М.: Издательство “ЛОРИ”, 2003. – 567с.

10. Добрынин В.Ю. Технологии компонентного программирования: Учеб. пособие / В.Ю. Добрынин. – СПб.: Изд-во СПбГУ, 2004. – 121 с.

11. Розробка методів підвищення оперативності передачі та захисту інформації у телекомунікаційних системах : Звіт з держбюджетної теми / Кіровогр. нац. техн. ун-т. – № 36Б113; № ДР 0113U003086; керівн. Смірнов

О.А. – Кіровоград, 2014. – 240 с.

12. Інформаційна технологія автоматизації проектування та тестування об'єктно-орієнтованого програмного забезпечення : Звіт з НДР / Кіровогр. нац. техн. ун-т. – № ДР 0114U003831. – Кіровоград: 2014. —58 с.

13. Інформаційна технологія компресії цифрових зображень на основі ортогональних перетворень : Звіт з НДР / Кіровогр. нац. техн. ун-т. —№ ДР 0114U003375. – Кіровоград, 2013. – 51 с.

14. Методи підвищення оперативності передачі даних та захисту інформації у телекомунікаційній мережі : Звіт з НДР / Кіровогр. нац. техн. ун-т. —№ ДР 0112U006631. – Кіровоград, 2012. – 40 с.

15. Смирнов А.А. Математическая формализация процесса проектирования объектно-ориентированного программного обеспечения информационных систем / А.А. Смирнов, А.П. Доренський // Информационные технологии и системы в управлении, образовании, науке: монография под ред. проф. В.С. Пономаренко. – Х.: Цифрова друкарня № 1, 2014. – С. 22-36. – ISBN 978-617-7188-50-5.

16. Смірнов О.А. Дослідження впливу ступеня стиснення зображень на оперативність їх доставки у телекомунікаційній системі / О.А. Смірнов, О.М. Дрєєв, О.П. Доренський // Системи обробки інформації. – 2013. – Вип. 8(115). – С. 234-239.

17. Смірнов О.А. Аналіз процесів стиснення та відновлення зображень на основі цифрових методів // О.А. Смірнов, О.П. Доренський, О.М. Дрєєв // Наука і техніка Повітряних сил Збройних Сил України. – 2013. – № 3(12). – С.122-127.

18. Доренський О.П. Формалізація процесу зміни станів програмних об'єктів складних систем на основі формального апарату скінченних автоматів Мура / О.П. Доренський, О.А. Смірнов // Зв'язок : Науково-виробничий журнал. – 2014. – № 3 (109) – С. 27-31.

19. Доренський О.П. Синтез структури інтегрованої моделі об'єктно-орієнтованого програмного забезпечення / О.П. Доренський // Системи обробки інформації. – 2014. – Т. 2, Вип. 2(118). – С. 68-72.

20. Dorensky O. Method of the Models' Synthesis for Software Automated System Objects' States in Digital Images Processing / Oleksandr Dorensky // Збірник наукових праць Кіровоградського національного технічного університету: Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – 2014. – Вип. 27. – С. 283-292.

21. Доренський О.П. Метод синтезу тестових структур взаємодії програмних об'єктів під час проектування програмного забезпечення на основі об'єктно-орієнтовної технології / О.П. Доренський // Системи управління, навігації та зв'язку. – Полтава: ПолтНТУ, 2014. – Вип. 3 (31). – С. 107–114.

22. Доренський О.П. Метод синтезу тестових моделей поведінки програмних об'єктів інформаційно-телекомунікаційної системи спеціального призначення / О.П. Доренський // Збірник наукових праць Харківського університету Повітряних Сил. – 2014. – Вип. 3(40). – С. 109-112.

23. Dorensky O. Development of the theoretical bases of logical domain modeling of a complex software system / Oleksandr Dorensky, Alexey Smirnov // International Journal of Computational Engineering Research (IJCER). – India, Delhi, 2014. – Vol. 4, Issue 4. – P. 19-23.

24. Доренський О.П. Дослідження помилок програмного забезпечення // О.П. Доренський, О.М. Змеул // Актуальні задачі сучасних технологій : Міжнар. наук.-техн. конф., 19-20 груд, 2012 р. : збірн. тез доп. – Тернопіль, 2012. – С. 187-188.

25. Доренський О.П. Особливості процесу розробки програмного забезпечення компресії цифрових зображень / О.П. Доренський // Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку : наук.-практ. конф., 12-13 бер, 2014 р. : збірн. тез доп. – Х., 2014. – С. 10-12.

26. Dorensky O.P. Comparative research of the color brightness distortion of the image compressed by DHT / O.P. Dorensky // Computer Science and Engineering: 6th Internat. Academ. Conf. CSE-2013, November 21-23, 2013. : Proceedings. – Lviv,

2013. – P. 154-155.

27. Доренський О.П. Особливості представлення структури компонент інтегрованої моделі об’єктно-орієнтованого програмного забезпечення інформаційних систем / О.П. Доренський // Системи обробки інформації : Проблеми та перспективи розвитку ІТ-індустрії : VI Міжнар. наук.-практ. конф., 17-18 квіт, 2014 р. : тези доп. – Х., 2014 – Т. 2, Вип. 2 (118). – С. 239.

28. Доренський О.П. Синтез структури динамічної компоненти комплексної моделі об’єктно-орієнтованого програмного забезпечення / О.П. Доренський // Комп’ютерні системи та мережні технології (CSNT-2014) : VII Міжнар. наук.-техн. конф., 17-19 квіт, 2014 р., : збірн. тез. – К., 2014. – С. 54.

29. Dorensky O. The structure of a dynamic component of software systems’ comprehensive model within the individual behavior of conceptual units / Oleksandr Dorensky // Advanced Information Systems and Technologies (AIST 2014): 3d Intern. Conf., May 14-16, 2014. : Proceedings. – Sumy, 2014. – P. 99-100.

30. Доренський О.П. Порівняльний аналіз технологій проектування й розроблення складних програмних систем / О.П. Доренський // Інформаційні технології в освіті, науці і техніці (ІТОНТ-2014) : ІІ Міжнар. наук.-практ. конфер., 24-26 квіт, 2014 р. : тези доп., у 2-х томах. – Черкаси, 2014. – Т. 1. – С. 13-14.

31. Dorensky O.P. Research results on the dependence of infocomm system compressed images delivery efficiency on compression ratio / O.P. Dorensky // Computer Modelling in High Tech (CMHT-2014) : Scient. and Techn. Intern. Conf., May 28-31, 2014. : Proceedings. – Kh., 2014. – P. 118-119.

32. Доренський О.П. Автоматизація розробки моделей станів об’єктів об’єктно-орієнтованого програмного забезпечення систем обробки інформації / О.П. Доренський // Інформаційні технології та комп’ютерна інженерія: Четверта Міжнар. наук.-практ. конф., 28-30 трав, 2014 р. : тези доп. – Вінниця, 2014. – С. 292-295.

33. Доренський О.П. Алгоритм синтезу тестової моделі взаємодії екземплярів класу на стадії проектування об’єктно-орієнтованого програмного

забезпечення / О.П. Доренський // Проблеми інформатики та комп'ютерної техніки (ПІКТ-2014) : Міжнар. наук.-практ. конф., 27-30 трав, 2014 р. : Праці. – Чернівці, 2014. – С. 168-170.

34. Доренський О.П. Перетворення моделі станів екземплярів класу об'єктів програмного забезпечення до автомата Мура / О.П. Доренський // Інженерія програмного забезпечення 2014: міжнар. наук.-практ. конф. аспір. та студ., 10-14 чер, 2014 р. : тези доп. – К., 2014. – С. 25-26.

35. Доренський О.П. Синтез тестової моделі станів об'єкта програмного забезпечення системи компресії цифрових зображень / О.П. Доренський // Електроніка та інформаційні технології (ЕлІТ-2014): VI Українсько-польська наук.-практ. конф., 27-31 серп, 2014 р. : матер. конф. – Львів - Чинадієво, 2014. – С. 47-49.

36. Доренський О.П. Алгоритм побудови тестової моделі поведінки програмних об'єктів системи компресії зображень / О.П. Доренський // Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій : VII міжнар. наук.-практ. конф., 17-19 вер, 2014 р. : тези доп. – Запоріжжя, 2014. – С. 140-141.

37. Dorensky O. Design and development of software for digital image compression based on discrete Hartley transform / Dorensky Oleksandr, Meleshko Elisaveta, Smirnov Sergii // Internet-Education-Science (IES-2014): Ninth International Scientific-Practical Conference, Oct. 14-17, 2014 : Proceedings. – Vinnytsia, 2014 – P. 124-126.

38. Доренський О.П. Оцінювання структури трансформант перетворення Хартлі щодо стиснення цифрових зображень / О.П. Доренський // Компьютерные, программные и интернет-технологии, программирование компьютерных мобильных систем : Междунар. конф., 14-16 апр, 2014 г. : сборн. материалов – Т. 5. – Х., 2014. – С. 213-214.

39. Коваленко О.В. Оцінка якості стиснення зображень на основі дискретного перетворення Хартлі / О.В. Коваленко, О.П. Доренський, О.М. Дрєєв

// Системи озброєння і військова техніка. – 2013. – Вип. 2(34). – С. 99-102.

40. Доренський О.П. Порівняльне дослідження ортогональних перетворень для компресії цифрових зображень / О.П. Доренський // Наукоємні технології. – 2013. – № 4 (20). – С. 416-420.

41. Доренський О.П. Аналіз цифрових методів кодування з перетворенням для стиснення зображень / О.П. Доренський, О.А. Смірнов // Проблемы инфокоммуникаций. Наука и технологии : Первая Междунар. науч.-практ. конф., 9-11 окт, 2013 г. : Сборн. науч. трудов – Х., 2013. – С. 238-240.

42. Доренський О.П. Дослідження методів стиснення статичних цифрових зображень з частковою втратою якості / О.П. Доренський // Матиматичне та програмне забезпечення інтелектуальних систем (MPZIS-2013) : XI Міжнар. наук.-практ. конф., 20-22 листоп, 2013 р. : тези доп. – Дніпропетровськ, 2013. – С. 73-74.

43. Доренський О.П. Особливості застосування ортогональних перетворень для розробки метода компресії цифрових зображень об’єктів / О.П. Доренський // Актуальні задачі сучасних технологій: Міжнар. наук.-техн. конф., 19–20 груд, 2013 р. : збірн. тез доп. – Тернопіль, 2013. – С. 186-187.

44. Доренський О.П. Оцінювання структури трансформант перетворення Хартлі щодо стиснення цифрових зображень / О.П. Доренський // Компьютерные, программные и интернет-технологии, программирование компьютерных мобильных систем : Междунар. конф., 14-16 апр, 2014 г. : сборн. материалов – Т. 5. – Х., 2014. – С. 213-214.

45. ДСТУ ISO/IEC TR 15271:2010 “Інформаційні технології. Настанови щодо застосування ISO/IEC 12207 (Процеси життєвого циклу програмного забезпечення)”. – К.: Держспоживстандарт України, 2010. – 8 с.

46. Дудзяний І.М. Об’єктно-орієнтоване моделювання програмних систем: Навчальний посібник. / І.М. Дудзяний. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. – 108 с.

47. Дудзяний І.М. Програмування мовою Object Pascal / І.М. Дудзяний. –

Львів: Вид-во ЛНУ ім. Івана Франка, 2009. – 329 с.

48. Ємельянова Н.З. Основы построения автоматизированных информационных систем: Учебное пособие / Ємельянова Н.З., Партика Т.Л., Попов И.И. – М.: ФОРУМ: ИНФРА-М, 2007. – 416 с.

49. Еркин С.Н. Методы проектирования ИУС в Rational Rose : Учебное пособие, Ч. 1 / С.Н. Еркин – Таганрог: Изд-во ТТИ ЮФУ, 2007 – 58с.

50. Еркин С.Н. Методы проектирования в Rational Rose : Учебное пособие, Ч. 2 / С.Н. Еркин – Таганрог: Изд-во ТТИ ЮФУ, 2007 – 74с.

51. Ємельянов В.О. Об'єктна модель програмного забезпечення захисту конфіденційних даних користувача / В.О. Ємельянов // Системи обробки інформації. – 2012. – Вип. 3 (101), Т. 2. – С. 156-159.

52. Жоголев Е.А. Технология программирования / Е.А. Жоголев. – М.: Научный мир, 2004. – 216 с.

53. Жураковський Б.Ю. Об'єктно-орієнтована технологія проектування систем управління // Жураковський Б.Ю., Варфоломієва О.Г., Гладких О.В., Хахлюк О.А. / Вісник ДУІКТ. – 2013. – №1. – С. 49-53.

54. Зайцев Д.А. Композиционный анализ сетей Петри / Д.А. Зайцев // Кибернетика и системный анализ. – 2006. – № 1. – С. 143-154.

55. Зайцев Д.А. Универсальная сеть Петри / Д.А. Зайцев // Кибернетика и системный анализ. – № 4. – 2012. – С. 24-39.

56. Захаров Н.Г. Синтез цифровых автоматов: Учебное пособие / Н.Г. Захаров, В.Н. Рогов. – Ульяновск: УлГТУ, 2003. – 135 с.

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

2 Підстава для розробки

Підставою для розробки служить завдання на кваліфікаційну бакалаврську роботу, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 185-02 від 28.12.2020 року).

3 Мета та призначення розробки

Метою кваліфікаційної бакалаврської роботи є розробка програмного забезпечення системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK.

4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської роботи є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

– вибір і обґрунтування методів реалізації проекту;

– розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

– системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK;

– цілісність даних у процесі роботи та при зберіганні;

– простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

– температура повітря: 19-20 град. по Цельсію;

– відносна вологість повітря до 80%;

– атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10 і з сумісними з цією платформою пристроями і прикладним програмним забезпечення.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows XP/Vista/7/8/10.

5.8.1 Обладнання

Комп’ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14” 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище RAD Studio Delphi.

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

– Структурна схема системи – 1 аркуш.
– Функціональна схема системи – 1 аркуш.
– Діаграма процесів – 1 аркуш.
– Блок-схема алгоритму роботи програми – 2 аркуша.
– Пояснювальна записка – 77 аркушів.

8 Етапи розробки

8.1 Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи. Постановка задачі на виконання кваліфікаційної бакалаврської роботи (складання ТЗ).

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень кваліфікаційної бакалаврської роботи.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання кваліфікаційної бакалаврської роботи на попередній захист 22.05.2021 р.

11.2 Подання кваліфікаційної бакалаврської роботи на захист 7.06.2021 р.

Додаток Б

(обов’язковий)

Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник кваліфікаційної бакалаврської роботи

_____ Смірнов С.А.

Програмне забезпечення системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 42

Літера: РП

Кропивницький – 2021 року

Клієнтська частина програмного забезпечення
Файл WPA2_KRACK.dpr

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
program WPA2_KRACK;
uses
Forms,SysUtils,
frmMAIN in 'frmMAIN.pas' {Form1},
frmSettings in 'frmSettings.pas' {Form2},
frm_WPA_KEYGEN in 'frm_WPA_KEYGEN.pas' {Form3},
frmLOG in 'frmLOG.pas' {Form4},
frm_WPA_KEYSTAT in 'frm_WPA_KEYSTAT.pas' {Form5},
frmSPLASH in 'frmSPLASH.pas' {U_Form_Splash},
frmAbout in 'frmAbout.pas' {AboutBox};
frmUnit1 in 'frmUnit.pas' {Unit6};

{$R *.res}
begin
try
U_Form_Splash:=TU_Form_Splash.Create(Application);
U_Form_Splash.Show;
U_Form_Splash.Update;
U_Form_Splash.Label2.Caption:='Підключення модулів';
U_Form_Splash.Update;
U_Form_Splash.Label2.Caption:='Налагодження інтерфейсів';
U_Form_Splash.Update;
Application.HintPause:=200;
Application.HintHidePause:=7000;
Application.HintShortPause:=25;
Application.Initialize;
Application.CreateForm(TForm1, Form1);
Application.CreateForm(TForm2, Form2);
Application.CreateForm(TForm3, Form3);
Application.CreateForm(TForm4, Form4);
Application.CreateForm(TForm5, Form5);
Application.CreateForm(Tform6, Form6);
Application.CreateForm(TAboutBox, AboutBox);
finally
U_Form_Splash.free;
end;
Application.Run;
end.
```

Файл frm_WPA_KEYGEN.pas – файл форми генерації ключів WPA2

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
unit frm_WPA_KEYGEN;

interface

uses


Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls;

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;
    Panel6: TPanel;
    Panel7: TPanel;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

{$R *.dfm}

procedure TForm5.Button2Click(Sender: TObject);
begin

if SaveDialog1.Execute then
begin
ListBox1.Items.SaveToFile(SaveDialog1.FileName);
end;

end;

end.
```

Файл frm_WPA_KEYSTAT.pas – файл форми тестування ключів WPA2

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
unit frm_WPA_KEYSTAT;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TForm5 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Button1: TButton;
    Button2: TButton;
    ListBox1: TListBox;
    SaveDialog1: TSaveDialog;
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;

implementation

{$R *.dfm}

procedure TForm5.Button2Click(Sender: TObject);
begin

if SaveDialog1.Execute then
begin
ListBox1.Items.SaveToFile(SaveDialog1.FileName);
end;

end;

end.
```

Файл Unit1.pas

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Menus, RxMenus, RXShell, ComCtrls, ExtCtrls, StdCtrls, ScktComp,
ToolWin,WinSock;
const
PORT_NUM=5555;
type
TForm1 = class(TForm)
RxTrayIcon1: TRxTrayIcon;
RxPopupMenu1: TRxPopupMenu;
Showform1: TMenuItem;
N1: TMenuItem;
About1: TMenuItem;
N2: TMenuItem;
Exit1: TMenuItem;
Panel1: TPanel;
StatusBar1: TStatusBar;
Panel3: TPanel;
Panel2: TPanel;
Memo1: TMemo;
Panel4: TPanel;
GroupBox1: TGroupBox;
TrackBar1: TTrackBar;
StaticText1: TStaticText;
GroupBox2: TGroupBox;
Button4: TButton;
Button5: TButton;
GroupBox3: TGroupBox;
Edit1: TEdit;
GroupBox4: TGroupBox;
Edit2: TEdit;
Timer1: TTimer;
GroupBox5: TGroupBox;
Button1: TButton;
Button2: TButton;
Button3: TButton;
GroupBox6: TGroupBox;
Edit3: TEdit;
Edit4: TEdit;
StaticText2: TStaticText;
StaticText3: TStaticText;
procedure RxTrayIcon1Click(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure Showform1Click(Sender: TObject);
procedure TrackBar1Change(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button2Click(Sender: TObject);
private A:TClientSocket;
procedure Add_Memo(s:string);
procedure ApplicationMinimize(Sender : TObject);
procedure ApplicationRestore(Sender : TObject);
function GetIP:string;
function Get_User: string;
procedure Accept(Sender: TObject; Socket: TCustomWinSocket);
procedure ClientConnect(Sender: TObject; Socket: TCustomWinSocket);
procedure ClientDisconnect(Sender: TObject; Socket: TCustomWinSocket);
procedure ClientError(Sender: TObject; Socket: TCustomWinSocket;
ErrorEvent: TErrorEvent; var ErrorCode: Integer);
procedure ClientRead(Sender: TObject; Socket: TCustomWinSocket);
procedure ClientWrite(Sender: TObject; Socket: TCustomWinSocket);
procedure GetThread(Sender: TObject;
ClientSocket: TServerClientWinSocket;
```

```
var SocketThread: TServerClientThread);
procedure Listen(Sender: TObject; Socket: TCustomWinSocket);
procedure ThreadEnd(Sender: TObject; Thread: TServerClientThread);
procedure ThreadStart(Sender: TObject; Thread: TServerClientThread);
public
{-----------}
DT:TDateTime;
procedure Connect(Sender: TObject; Socket: TCustomWinSocket);
procedure Read(Sender: TObject; Socket: TCustomWinSocket);
procedure Write(Sender: TObject; Socket: TCustomWinSocket);
{-----------}
function
PACKED_SBOR(LOGIN:string;PASS:string;exclusion:string;RASMER:integer):string;
procedure PACKED_RASBOR(s:string);
end;

var
Form1: TForm1;

implementation
var
Login,Pass,Excl:string;
COL:integer;
{$R *.dfm}
{---------------Процедура складання вікна у трей----------------}
procedure TForm1.RxTrayIcon1Click(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
if (IsWindowVisible(Form1.Handle)=false) then
begin
Application.Restore;
Form1.show;
Form1.BringToFront;
end
else
begin
Form1.hide;
end;end;
{---------------Процедура відображення вікна-----------------}
procedure TForm1.Showform1Click(Sender: TObject);
begin
Application.ProcessMessages;
Form1.Show; Form1.BringToFront;

Application.Restore;
Application.BringToFront;
end;

procedure TForm1.TrackBar1Change(Sender: TObject);
begin
StaticText1.caption:=inttostr(TrackBar1.position)+' сек.';
Timer1.Interval:=TrackBar1.position*1000;
end;
{---------------Процедура відпрацювання натисканя на кнопку-----}
procedure TForm1.Button4Click(Sender: TObject);
begin
A:=TClientSocket.Create(self);
A.OnRead:=Read; A.OnWrite:=Write;
A.OnConnect:=Connect;
A.ClientType:=ctNonBlocking;
A.Host:=Edit1.text;
A.Port:=strtoint(Edit2.text);
A.Open; Timer1.Enabled:=true;
end;

procedure TForm1.Connect(Sender: TObject; Socket: TCustomWinSocket);
begin

end;
```

```
{---------------Процедура читання сокета-----------------}
procedure TForm1.Read(Sender: TObject; Socket: TCustomWinSocket);
begin


Add_memo('Read '+Socket.ReceiveText);


end;
{---------------Процедура запису сокету-----------------}
procedure TForm1.Write(Sender: TObject; Socket: TCustomWinSocket);
begin
Add_memo('Write');
end;
{---------------Процедура створення вікна-----------------}
procedure TForm1.FormCreate(Sender: TObject);
begin
Application.OnMinimize := ApplicationMinimize;
Application.OnRestore := ApplicationRestore;
Add_Memo('Запуск клиента');

DT:=now;
end;


{---------------Процедура визначення часу-----------------}
procedure TForm1.Add_Memo(s: string);
begin
Memo1.lines.Add(DateTimeToStr(now)+' : '+s);
end;
{---------------Процедура звертання додатку-----------------}
procedure TForm1.ApplicationMinimize(Sender: TObject);
begin
Application.ProcessMessages;
ShowWindow(Application.Handle, SW_HIDE);
end;

procedure TForm1.ApplicationRestore(Sender: TObject);
begin
Application.ProcessMessages;
ShowWindow(Application.Handle, SW_HIDE);
end;
{---------------Процедура визначення IP-адреси---------------}
function TForm1.GetIP: string;
var
WSAData : TWSAData;
p : PHostEnt;
Name : array [0..$FF] of Char;
s:string;
begin
WSAStartup($0101, WSAData);
GetHostName(name, $FF);
p := GetHostByName(Name);
s:=inet_ntoa(PInAddr(p.h_addr_list^)^);
WSACleanup;
result:=s;
end;
{---------------Процедура визначення користувача-----------------}
function TForm1.Get_User: string;
var
Buffer: array[0..MAX_PATH] of Char;
sz:DWord;
begin
sz:=MAX_PATH-1;
if windows.GetUserName(Buffer,sz)
then begin
if sz>0 then dec(sz);
SetString(Result,Buffer,sz);
end else begin
```

```
Result:='Error '+inttostr(GetLastError);
end;
end;

{---------------Процедура з'єднання з клієнтом-----------------}
procedure TForm1.ClientConnect(Sender: TObject; Socket: TCustomWinSocket);
begin
Add_memo('ХОСТ:'+Socket.RemoteHost+' АДРЕС:'+Socket.RemoteAddress+'
ПОРТ:'+inttostr(Socket.RemotePort));
end;

{---------------Процедура роз'єднання з клієнтом-----------------}
procedure TForm1.ClientDisconnect(Sender: TObject;
Socket: TCustomWinSocket);
begin
Add_memo('Сервер');
end;
{---------------Процедура помилки з'єднання з клієнтом-----------}
procedure TForm1.ClientError(Sender: TObject; Socket: TCustomWinSocket;
ErrorEvent: TErrorEvent; var ErrorCode: Integer);
begin
Add_memo('Помилка з'єднання з клієнтом. Код= = '+IntToStr(ErrorCode));
MessageDlg('Помилка з'єднання з клієнтом. Код= =
'+IntToStr(ErrorCode),mtInformation,[mbOK],0);
SysErrorMessage(GetLastError);
end;
{---------------Процедура читання даних від клієнта----------}

procedure TForm1.ClientRead(Sender: TObject; Socket: TCustomWinSocket);
begin
Add_memo('COMAND:'+Socket.ReceiveText);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
if (Edit3.Text<>'')and(Edit4.Text<>'') then
begin
A.Socket.SendText(PACKED_SBOR(Edit3.Text,Edit4.Text,'324-02',10));
 end;
end;
{---------------Процедура розшифрування даних-----------}
procedure TForm1.PACKED_RASBOR(s: string);
var
A:string;
i:integer;
Point:integer;
begin
A:=s;
MessageDlg(A,mtInformation,[mbOK],0);
if (s<>'') then
begin
Point:=pos('*',A);
Login:=copy(A,0,point-1);
MessageDlg(Login,mtInformation,[mbOK],0);
A:=copy(A,point+1,Length(A)-point);

Point:=pos('*',A);
Pass:=copy(A,0,point-1);;
MessageDlg(Pass,mtInformation,[mbOK],0);

A:=copy(A,point+1,Length(A)-point);
Point:=pos('*',A);
Excl:=copy(A,0,point-1);;
MessageDlg(Excl,mtInformation,[mbOK],0);

A:=copy(A,point+1,Length(A)-point);
COL:=strtoint(A);
MessageDlg(A,mtInformation,[mbOK],0);
end
```

```
else MessageDlg('Немає даних!!!',mtInformation,[mbOK],0);
end;


function TForm1.PACKED_SBOR(LOGIN, PASS, exclusion: string;
RASMER: integer): string;
begin
result:=LOGIN+'*'+PASS+'*'+exclusion+'*'+inttostr(RASMER);
end;

end.
```

Файл frmLOG.pas – вікно авторизації

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
unit frmLOG;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TForm4 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Memo1: TMemo;
    Button1: TButton;
    Button2: TButton;
    SaveDialog1: TSaveDialog;
    procedure FormShow(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form4: TForm4;

implementation

uses frmMAIN;
{$R *.dfm}

procedure TForm4.FormShow(Sender: TObject);
begin
      Memo1.Clear;
      Memo1.Lines.Assign(Form1.Memo1.Lines);
end;
procedure TForm4.Button1Click(Sender: TObject);
begin
      form4.Hide;
      Form1.Show;
end;
procedure TForm4.Button2Click(Sender: TObject);
begin
     if SaveDialog1.Execute then
     begin
          Memo1.Lines.SaveToFile(SaveDialog1.FileName);
     end;
end;
end.
```

Файл frmMAIN.pas – головне вікно програми

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
unit frmMAIN;

interface

uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Menus, StdCtrls, ExtCtrls,ScktComp, Buttons;
const
PORT_NUM=5555;
 type
TForm1 = class(TForm)
Panel1: TPanel;
MainMenu1: TMainMenu;
Fil1: TMenuItem;
N1: TMenuItem;
N2: TMenuItem;
N3: TMenuItem;
N4: TMenuItem;
Panel2: TPanel;
Memo1: TMemo;
Panel3: TPanel;
Panel5: TPanel;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
BitBtn3: TBitBtn;
BitBtn4: TBitBtn;
BitBtn5: TBitBtn;
BitBtn6: TBitBtn;
BitBtn7: TBitBtn;
BitBtn8: TBitBtn;
BitBtn9: TBitBtn;
BitBtn10: TBitBtn;
BitBtn11: TBitBtn;
BitBtn12: TBitBtn;
BitBtn13: TBitBtn;
BitBtn14: TBitBtn;
ListBox2: TListBox;
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure BitBtn9Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn13Click(Sender: TObject);
procedure BitBtn12Click(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);
procedure N4Click(Sender: TObject);

private
CLIENT_CONNECT:integer;
A:TServerSocket;
public
procedure Add_Memo(s:string);
procedure Listen(Sender : TObject;Socket: TCustomWinSocket);
procedure Accept(Sender: TObject;Socket: TCustomWinSocket);
procedure ClientConnect(Sender: TObject; Socket: TCustomWinSocket);
procedure ClientDisconnect(Sender: TObject;Socket: TCustomWinSocket);
procedure ClientError(Sender: TObject;Socket: TCustomWinSocket; ErrorEvent:
TErrorEvent; var ErrorCode: Integer);
procedure ClientRead(Sender: TObject; Socket: TCustomWinSocket);
procedure ClientWrite(Sender: TObject;Socket: TCustomWinSocket);
procedure GetThread(Sender: TObject;ClientSocket: TServerClientWinSocket;var
SocketThread: TServerClientThread);
procedure ThreadEnd(Sender: TObject;Thread: TServerClientThread);

procedure ThreadStart(Sender: TObject; Thread: TServerClientThread);
procedure ADD_CLIENT_DATA(A:string);
```

```
procedure PACKED_RASBOR(s:string);
procedure PACKED_processing;
end;

var
Form1: TForm1;
implementation

uses frmLOG, frm_WPA_KEYSTAT;
var
Login,Pass,Excl:string;
COL:integer;
{$R *.dfm}


{---------------Процедура з'єднання з клієнтом-----------------}
procedure TForm1.ClientConnect(Sender: TObject; Socket: TCustomWinSocket);
begin
Memo1.lines.add('Клієнти:'+Socket.RemoteAddress+'Хост:'+Socket.RemoteHost+' ->
CONNECT');
ListBox2.Items.Add('ADD:');
ListBox2.Items.Add(' Klient:'+Socket.RemoteAddress);
ListBox2.Items.Add(' Хост:'+Socket.RemoteHost);
end;
{---------------Процедура роз'єднання з клієнтом----------------}
procedure TForm1.ClientDisconnect(Sender: TObject;
Socket: TCustomWinSocket);
begin
Memo1.lines.add('Klient:'+Socket.RemoteAddress+'Хост:'+Socket.RemoteHost+' ->
DISCONNECT');
ListBox2.Items.Add('Diskonect:');
ListBox2.Items.Add('Klient:'+Socket.RemoteAddress);
ListBox2.Items.Add('Хост:'+Socket.RemoteHost);
end;

{---------------Процедура читання даних від клієнта-----------}
procedure TForm1.ClientRead(Sender: TObject; Socket: TCustomWinSocket);
begin
PACKED_RASBOR(Socket.ReceiveText);
PACKED_processing;
Add_Memo('Name='+Login);
Add_Memo('Pass='+Pass);
Add_Memo('Excl='+Excl);
Add_Memo('Dlinna='+inttostr(COL));
end;
{-----Процедура створення вікна серверу системи кібербезпеки для моделювання
взламу протоколу WPA2 на базі реалізації атаки KRACK--}
procedure TForm1.FormCreate(Sender: TObject);
begin
A:=TServerSocket.Create(self);
A.ServerType:=stNonBlocking;
A.OnListen:=Listen;
A.OnAccept:=Accept;
A.OnClientConnect:=ClientConnect;
A.OnClientDisconnect:=ClientDisconnect;
A.OnClientError:=ClientError;
A.OnClientRead:=ClientRead;
A.OnClientWrite:=ClientWrite;
A.OnGetThread:=GetThread;
A.OnThreadEnd:=ThreadEnd;
A.OnThreadStart:=ThreadStart;
A.Port:=PORT_NUM;
A.Open;
CLIENT_CONNECT:=0;
Form1.Caption:='ADD DATA(ПОРТ'+inttostr(PORT_NUM)+')';
Add_Memo('Сервер запущен');
end;
{---------------Процедура видалення вікна-----------}
procedure TForm1.FormDestroy(Sender: TObject);
```

```
begin
A.Close;
A.Destroy;
end;
procedure TForm1.Add_Memo(s: string);
begin
Memo1.lines.Add(DateTimeToStr(now)+' : '+s);
end;
procedure TForm1.PACKED_RASBOR(s: string);
var
A:string;
Point:integer;
begin
A:=s;
MessageDlg(A,mtInformation,[mbOK],0);
if (s<>'') then
begin
Point:=pos('*',A);
Login:=copy(A,0,point-1);
MessageDlg(Login,mtInformation,[mbOK],0);
A:=copy(A,point+1,Length(A)-point);
Point:=pos('*',A);
Pass:=copy(A,0,point-1);;
MessageDlg(Pass,mtInformation,[mbOK],0);
A:=copy(A,point+1,Length(A)-point);
Point:=pos('*',A);
Excl:=copy(A,0,point-1);;
MessageDlg(Excl,mtInformation,[mbOK],0);
A:=copy(A,point+1,Length(A)-point);
COL:=strtoint(A);
MessageDlg(A,mtInformation,[mbOK],0);
end;
end;
procedure TForm1.BitBtn9Click(Sender: TObject);
begin
      form1.Close;
end;
procedure TForm1.PACKED_processing;
begin
      A.Socket.Connections[0].SendText('JHJKTT');
      A.Socket.SendText('OK');
      A.Socket.SendText('OK');
      A.Socket.SendText('OK');}
end;
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
      Form4.show;
      Form1.Hide;
end;
procedure TForm1.BitBtn13Click(Sender: TObject);
begin
      Memo1.Lines.Clear;
      ListBox2.Items.Clear;
end;
procedure TForm1.BitBtn12Click(Sender: TObject);
begin
      MessageDlg('Clear',mtInformation,[mbOK],0);
      Form5.ListBox1.Items.Clear;
end;
procedure TForm1.BitBtn10Click(Sender: TObject);
begin
      MessageDlg('Ok',mtInformation,[mbOK],0);
end;
procedure TForm1.N4Click(Sender: TObject);
begin
      MessageDlg('',mtInformation,[mbOK],0);
end;
end.
```

Файл frmSettings.pas – Вікно встановлення параметрів системи кібербезпеки для моделювання взламу протоколу WPA2 на базі реалізації атаки KRACK

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
unit frmSettings;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, ExtCtrls, StdCtrls;

type
  TForm2 = class(TForm)
    Panel2: TPanel;
    Panel3: TPanel;
    Button1: TButton;
    Button2: TButton;
    Panel4: TPanel;
    PageControl2: TPageControl;
    TabSheet6: TTabSheet;
    GroupBox4: TGroupBox;
    TabSheet7: TTabSheet;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    TabSheet8: TTabSheet;
    GroupBox3: TGroupBox;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    CheckBox4: TCheckBox;
    CheckBox5: TCheckBox;
    TabSheet9: TTabSheet;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation
{$R *.dfm}
end.
```

Файл frmSPLASH.pas

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
unit frmSPLASH;

interface

uses


Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ComCtrls, StdCtrls, ExtCtrls;

type
  TU_Form_Splash = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Animate1: TAnimate;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  U_Form_Splash: TU_Form_Splash;

implementation

{$R *.dfm}

end.
```

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
unit Logics;

interface

uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
glLogics,
glSBox, StdCtrls, glGrBox, ExtCtrls, ComCtrls, dsgnintf, ToolWin, ImgList,
richEdit,
glPage, Tabs;

type
TchGroupBoxPlus = class;

TchLogicsComponentEditor = class(TComponentEditor)
procedure ExecuteVerb(Index: Integer); override;
function GetVerb(Index: Integer): string; override;
function GetVerbCount: Integer; override;
private
procedure ShowEditor(LogicProducer: TLogicProducer);
end;

TchLogicsEditor = class(TForm)
SB: TchScrollBox;
Panel1: TPanel;
iP_WPA_key: TImage;
iF_WPA_key: TImage;
Label2: TLabel;
cbMode: TComboBox;
SBar: TStatusBar;
iLink: TImage;
cbNext: TComboBox;
Label4: TLabel;
eStepName: TEdit;
Label1: TLabel;
ImageList1: TImageList;
ToolBar1: TToolBar;
ToolButton1: TToolButton;
tbNew: TToolButton;
ToolButton3: TToolButton;
ToolButton4: TToolButton;
cbNextFalse: TComboBox;
Label5: TLabel;
Image3: TImage;
Label7: TLabel;
Label8: TLabel;
Shape1: TShape;
ImageList: TImageList;
ToolButton2: TToolButton;
ToolButton5: TToolButton;
ToolButton6: TToolButton;
ToolButton7: TToolButton;
cbIgnoreSpaces: TCheckBox;
ToolButton8: TToolButton;
pLeft: TPanel;
pLog: TPanel;
Splitter1: TSplitter;
tbStop: TToolButton;
Panel2: TPanel;
Splitter2: TSplitter;
reReslt: TRichEdit;
PC: TchPageControl;
tsLog: TTabSheet;
mLog: TMemo;
```

```
tsDictionary: TTabSheet;
mDictionary: TMemo;
Shape2: TShape;
Shape3: TShape;
Shape4: TShape;
Shape5: TShape;
Shape6: TShape;
Shape7: TShape;
Shape8: TShape;
Shape9: TShape;
Shape10: TShape;
Shape11: TShape;
Shape12: TShape;
Shape13: TShape;
TabSet1: TTabSet;
procedure SBEraseBkgndEvent(Sender: TObject; DC: HDC);
procedure cbNextChange(Sender: TObject);
procedure cbModeChange(Sender: TObject);
procedure tbNewClick(Sender: TObject);
procedure cbNextFalseChange(Sender: TObject);
procedure eStepNameChange(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure ToolButton5Click(Sender: TObject);
procedure ToolButton7Click(Sender: TObject);
procedure cbIgnoreSpacesClick(Sender: TObject);
procedure pLeftDockOver(Sender: TObject; Source: TDragDockObject; X,
Y: Integer; State: TDragState; var Accept: Boolean);
procedure pLeftUnDock(Sender: TObject; Client: TControl;
NewTarget: TWinControl; var Allow: Boolean);
procedure pLeftDockDrop(Sender: TObject; Source: TDragDockObject; X,
Y: Integer);
procedure ToolButton8Click(Sender: TObject);
procedure tbStopClick(Sender: TObject);
procedure TabSet1Change(Sender: TObject; NewTab: Integer;
var AllowChange: Boolean);
private
FActiveBox: TchGroupBoxPlus;
LogicProducer: TLogicProducer;
Logics: TLogics;

procedure MouseDown_(Sender: TObject; Button: TMouseButton; Shift: TShiftState;
X, Y: Integer);
procedure MouseMove_(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure MouseUp_(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X,
Y: Integer);
procedure DblClick_(Sender: TObject);
procedure SetActiveBox(const Value: TchGroupBoxPlus);
procedure UpdateView;
procedure AddBox(LogicElement_: TLogicElement);
procedure AddShape(CommentArea: TCommentArea);

procedure OnTraceMessage(Sender: TLogics; fStepResult: boolean; const
StepResult, ParsedResult, Msg: string);
public
function Execute(LogicProducer: TLogicProducer): boolean;
property ActiveBox: TchGroupBoxPlus read FActiveBox write SetActiveBox;
end;

TchGroupBoxPlus = class(TchGroupBox)
public
pt: TPoint;
Selected: boolean;
LogicElement: TLogicElement;
fAsLogical: boolean;
constructor Create(AOwner: TComponent); override;
destructor Destroy; override;
procedure Paint; override;
end;
```

```
TchShapePlus = class(TShape)
public
pt: TPoint;
Selected: boolean;
CommentArea: TCommentArea;
procedure Paint; override;
end;

var
glLogicsEditor: TchLogicsEditor;

implementation
uses glTypes, glUtils, geLogicItemEditor, clipbrd;
{$R *.DFM}

function TchLogicsEditor.Execute(LogicProducer: TLogicProducer): boolean;
var
i: integer;
begin
fLogicItemEditor := TfLogicItemEditor.Create(nil);

mDictionary.Lines.Assign(LogicProducer.Dictionary);
try

self.LogicProducer := LogicProducer;
self.Logics := LogicProducer.Logics;

Logics.OnTraceMessage := OnTraceMessage;

cbNext.Items.Clear;
cbNextFalse.Items.Clear;
cbNext.Items.Add('');
cbNextFalse.Items.Add('');

for i := 0 to Logics.Count-1 do AddBox(Logics[i]);

for i := 0 to LogicProducer.CommentAreas.Count-1 do
AddShape(LogicProducer.CommentAreas[i]);

Result := ShowModal = mrOK;
finally
fLogicItemEditor.Free;
end;

pLog.Dock(pLeft, rect(1,1,10,10));
pLog.Dock(pLeft, rect(1,1,10,10));
pLeft.Dock(pLog, rect(1,1,1,1));
end;

procedure TchLogicsEditor.AddBox(LogicElement_: TLogicElement);
var
Box: TchGroupBoxPlus;
begin
Box := TchGroupBoxPlus.Create(self);
with Box do
begin
Parent := SB;
Left := LogicElement_.Left;
Top := LogicElement_.Top;
Width := 100;
Height := 50;
Caption := LogicElement_.Caption;
Options := Options - [fgoCanCollapse];
CaptionAlignment := fcaWidth;
CaptionBorder.Inner := bvRaised;
CaptionBorder.Outer := bvLowered;
Colors.Caption := clBtnShadow;
Colors.TextActive := clBtnHighlight;
OnMouseDown := MouseDown_;
```

```
OnMouseMove := MouseMove_;
OnMouseUp := MouseUp_;
OnDblClick := DblClick_;
Border.Inner := bvRaised;
Border.Outer := bvNone;
Colors.Client := clWhite;
Colors.ClientActive := clWhite;
Box.LogicElement := LogicElement_;

cbNext.Items.AddObject(LogicElement_.Caption, LogicElement);
cbNextFalse.Items.AddObject(LogicElement_.Caption, LogicElement);

end;
end;

procedure TchLogicsEditor.AddShape(CommentArea: TCommentArea);
var
Shape: TchShapePlus;
begin
Shape := TchShapePlus.Create(self);
Shape.CommentArea := CommentArea;
with Shape do
begin
Parent := SB;
Left := CommentArea.Left;
Top := CommentArea.Top;
Width := CommentArea.Width;
Height := CommentArea.Height;
Pen.Style := psDashDot;
Brush.Style := bsClear;

OnMouseDown := MouseDown_;
OnMouseMove := MouseMove_;
OnMouseUp := MouseUp_;
OnDblClick := DblClick_;
end;

end;

procedure TchLogicsEditor.MouseDown_(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var i: integer;
begin
if Sender is TchShapePlus then with Sender as TchShapePlus do
begin
pt.X := X; pt.Y := Y;
pt := ClientToScreen(pt);
Selected := true;
Tag := 1;
if (X>=Width-5)and(X<Width)and(Y>=Height-5)and(Y<Height) then Tag := 2;
exit;
end;

with TchGroupBoxPlus(Sender) do
begin
pt.X := X; pt.Y := Y;
pt := ClientToScreen(pt);
pt.Y := pt.Y+ SB.VertScrollBar.ScrollPos;
Tag := 1;
Options := Options + [fgoDelineatedText];
Colors.Caption := clBtnHighlight;
Colors.TextActive := clBlack;
Font.Style := [fsBold];
Selected := true;
ActiveBox := TchGroupBoxPlus(Sender);
end;

for i:=0 to SB.ControlCount-1 do
```

```
if (SB.Controls[i] is TchGroupBoxPlus) then with TchGroupBoxPlus(SB.Controls[i])
do
begin
if ActiveBox = TchGroupBoxPlus(SB.Controls[i]) then continue;
Options := Options - [fgoDelineatedText];
Colors.Caption := clBtnShadow;
Colors.TextActive := clBtnHighlight;
Font.Style := [];
Selected := false;
Repaint;
end;
end;

procedure TchLogicsEditor.MouseMove_(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
var
pt_new: TPoint;
begin
if Sender is TchShapePlus then with Sender as TchShapePlus do
begin
case Tag of
1:
begin
pt_new.X := X; pt_new.Y := Y;
pt_new := ClientToScreen(pt_new);
Left := Left + pt_new.X - pt.X;
Top := Top + pt_new.Y - pt.Y;
CommentArea.Left := Left + SB.HorzScrollBar.ScrollPos;
CommentArea.Top := Top + SB.VertScrollBar.ScrollPos;
end;
2:
begin
pt_new.X := X; pt_new.Y := Y;
pt_new := ClientToScreen(pt_new);
Width := Width + pt_new.X - pt.X;
Height := Height + pt_new.Y - pt.Y;
if Width < 50 then Width := 50; if Height < 50 then Height := 50;
CommentArea.Width := Width;
CommentArea.Height := Height;
end;
end;
pt.X := pt_new.X; pt.Y := pt_new.Y;
exit;
end;

with TchGroupBoxPlus(Sender) do
begin
if bool(Tag) then
begin
pt_new.X := X; pt_new.Y := Y;
pt_new := ClientToScreen(pt_new);
pt_new.Y := pt_new.Y+ SB.VertScrollBar.ScrollPos;
Left := Left + pt_new.X - pt.X;
Top := Top + pt_new.Y - pt.Y;

LogicElement.Left := Left + SB.HorzScrollBar.ScrollPos;
LogicElement.Top := Top + SB.VertScrollBar.ScrollPos;

SBar.SimpleText := IntToStr(Left) + ':' + IntToStr(Top);

UpdateView;
end;
pt.X := pt_new.X; pt.Y := pt_new.Y;
end;
end;

procedure TchLogicsEditor.UpdateView;
var
DC: HDC;
```

```
begin
DC := GetDC(SB.Handle);
SendMessage(SB.Handle, WM_EraseBkgnd, WPARAM(DC),0);
ReleaseDC(SB.Handle, DC);
end;

procedure TchLogicsEditor.MouseUp_(Sender: TObject; Button: TMouseButton; Shift:
TShiftState; X, Y: Integer);
var i: integer;
begin
TControl(Sender).Tag := 0;
for i := 0 to SB.ControlCount-1 do
if (SB.Controls[i] is TchShapePlus) then (SB.Controls[i] as TchShapePlus).Paint;
end;

procedure TchLogicsEditor.DblClick_(Sender: TObject);
var str: string;
begin
TControl(Sender).Tag := 0;
if Sender is TchShapePlus then with Sender as TchShapePlus do
begin
str := CommentArea.Text;
if InputQuery('Caption', 'Comments', str) then CommentArea.Text := str;
PostMessage(TWinControl(Parent).Handle, WM_LBUTTONUP, 1, 1);
exit;
end;

fLogicItemEditor.Execute(Logics, TchGroupBoxPlus(Sender).LogicElement);
PostMessage(TWinControl(Sender).Handle, WM_LBUTTONUP, 1, 1);
end;


procedure TchLogicsEditor.SetActiveBox(const Value: TchGroupBoxPlus);
var
i, Index: integer;
Box: TchGroupBoxPlus;
begin
FActiveBox := Value;

Index := cbNext.Items.IndexOfObject(Value.LogicElement.NextElement);
if Index <> -1 then cbNext.ItemIndex := Index else cbNext.ItemIndex := 0;

Index := cbNextFalse.Items.IndexOfObject(Value.LogicElement.NextFalseElement);
if Index <> -1 then cbNextFalse.ItemIndex := Index else cbNextFalse.ItemIndex :=
0;

cbMode.ItemIndex := integer(FActiveBox.LogicElement.IsFirst);
eStepName.Text := FActiveBox.LogicElement.Caption;
end;

procedure TchLogicsEditor.OnTraceMessage(Sender: TLogics; fStepResult: boolean;
const StepResult, ParsedResult, Msg: string);
begin
mLog.Lines.Add(Msg);
if reReslt.Text = '' then reReslt.Tag := 0;

if length(ParsedResult) = 0 then exit;
tag := 1 - tag;
reReslt.Lines.BeginUpdate;
reReslt.Text := reReslt.Text + ParsedResult;

reReslt.SelStart := length(reReslt.Text) - length(ParsedResult);
reReslt.SelLength := length(ParsedResult);


if tag = 0 then reReslt.SelAttributes.Color:=clRed else
reReslt.SelAttributes.Color:=clGreen;
reReslt.SelAttributes.Color := RGB(100+Random(100), 100+Random(100),
100+Random(100));
```

```
reReslt.SelLength := 0;
reReslt.Lines.EndUpdate;
end;

procedure TchGroupBoxPlus.Paint;
var
i: integer;
R: TRect;
str: string;
begin
inherited;
ChangeBitmapColor((Owner as TchLogicsEditor).iLink.Picture.Bitmap,
GetPixel((Owner as TchLogicsEditor).iLink.Picture.Bitmap.Canvas.Handle, 0,0),
IIF(LogicElement.NextElement<>nil, clGreen, clRed));
BitBlt(Canvas.handle, 100-14-3, 3, 14, 13, (Owner as
TchLogicsEditor).iLink.Picture.Bitmap.Canvas.Handle, 0, 0, SRCCOPY);

Canvas.Font.Color := clTeal;
 Canvas.Font.Style := [];
str := LogicElement.Expression + ' ' + LogicRuleLabels[LogicElement.Rule] + ' '
+ LogicElement.Value;
R := Bounds(3, 20, Width-6, Height-22);
DrawText(Canvas.handle, PChar(str), length(str), R, DT_WORDBREAK or
DT_END_ELLIPSIS or DT_MODIFYSTRING);
end;

procedure TchLogicsEditor.SBEraseBkgndEvent(Sender: TObject; DC: HDC);
var
Canvas: TCanvas;
LogicElement: TLogicElement;
i: integer;
PenFalse, Pen, PenTrue, OldPen, PenGrid: HPen;
Brush, OldBrush: HBrush;
NextBox, PrevBox, PrevFalseBox: TchGroupBoxPlus;
bmp: TBitmap;

function FindBox(LogicElement: TLogicElement): TchGroupBoxPlus;
var
i: integer;
begin
Result := nil;
if LogicElement = nil then exit;
for i := 0 to SB.ControlCount-1 do
if SB.Controls[i] is TchGroupBoxPlus then
if TchGroupBoxPlus(SB.Controls[i]).LogicElement = LogicElement then Result :=
TchGroupBoxPlus(SB.Controls[i]);
end;
procedure Line(X, Y, X2, Y2: integer; isTrueLine: boolean);
const R = 5;
begin
if isTrueLine then SelectObject(DC, PenTrue) else SelectObject(DC, PenFalse);
MoveToEx(DC, X, Y, nil); LineTo(DC, X2, Y2);
SelectObject(DC, Pen);
if (abs(X2-X) < 500)and(abs(Y2-Y) < 500) then
MoveToEx(DC, X, Y+1, nil); LineTo(DC, X2, Y2+1);
Ellipse(DC, X-R-2, Y-R-2, X+R*2-2, Y+R*2-2);
Ellipse(DC, X2-R-2, Y2-R-2, X2+R*2-2, Y2+R*2-2);
end;
procedure DrawGrid;
var i, j: integer;
const step = 14;
begin
FillRect(DC, SB.ClientRect, Brush);
for i:=1 to SB.Width div step do
begin
j := i*14;
MoveToEx(DC, j, 0, nil); LineTo(DC, j, SB.Height);
end;
```

```
for i:=1 to SB.Height div step do
begin
j := i*14;
MoveToEx(DC, 0, j, nil); LineTo(DC, SB.Width, j);
end;
end;
begin
try
Brush := CreateSolidBrush(clWhite);
Pen := CreatePen( PS_SOLID, 1, clBlack );
PenGrid := CreatePen( PS_SOLID, 1, $E0E0E0 );
PenLong := CreatePen( PS_DASHDOT, 1, $E0E0E0 );
PenTrue := CreatePen( PS_SOLID, 1, $FF9090 );
PenFalse := CreatePen( PS_SOLID, 1, $009090 );
OldPen := SelectObject( DC, PenGrid );
OldBrush := SelectObject( DC, Brush );
DrawGrid;
for i := 0 to SB.ControlCount-1 do
begin
if not(SB.Controls[i] is TchGroupBoxPlus) then continue;
LogicElement := TchGroupBoxPlus(SB.Controls[i]).LogicElement;
if LogicElement = nil then exit;
if LogicElement.IsFirst then
begin
MoveToEx(DC, 0, 0, nil);
LineTo(DC, SB.Controls[i].Left, SB.Controls[i].Top);
end;
PrevBox := FindBox(LogicElement.NextElement);
if Assigned(PrevBox) then
begin
Line(SB.Controls[i].Left + SB.Controls[i].Width, SB.Controls[i].Top,
PrevBox.Left, PrevBox.Top, true);
DeleteObject( SelectObject( DC, OldPen ) );
end;
PrevFalseBox := FindBox(LogicElement.NextFalseElement);
if Assigned(PrevFalseBox) then
begin
Line(SB.Controls[i].Left + SB.Controls[i].Width, SB.Controls[i].Top +
SB.Controls[i].Height, PrevFalseBox.Left, PrevFalseBox.Top, false);
DeleteObject( SelectObject( DC, OldPen ) );
end;
bmp := TBitmap.Create;
if LogicElement.NextElement <> nil then
begin
ImageList.GetBitmap(0, bmp);
BitBlt(DC, SB.Controls[i].Left + SB.Controls[i].Width-3, SB.Controls[i].Top,
bmp.width, bmp.height, bmp.Canvas.Handle, 0, 0, SRCCOPY);
end;
if LogicElement.NextFalseElement <> nil then
begin
ImageList.GetBitmap(1, bmp);
BitBlt(DC, SB.Controls[i].Left + SB.Controls[i].Width-3, SB.Controls[i].Top +
SB.Controls[i].Height-17, bmp.width, bmp.height, bmp.Canvas.Handle, 0, 0,
SRCCOPY);
end;
if LogicElement.IsFirst then
begin
ImageList.GetBitmap(2, bmp);
BitBlt(DC, SB.Controls[i].Left - 20, SB.Controls[i].Top, bmp.width, bmp.height,
bmp.Canvas.Handle, 0, 0, SRCCOPY);
end;
bmp.Free;
end;

finally
SelectObject(DC, Pen);
DeleteObject(SelectObject(DC, OldPen));
DeleteObject(PenTrue);
DeleteObject(PenFalse);
```

```
DeleteObject(PenGrid);
DeleteObject(Brush);
end;

for i := 0 to SB.ControlCount-1 do

if (SB.Controls[i] is TchShapePlus) then (SB.Controls[i] as TchShapePlus).Paint;
end;

procedure TchLogicsEditor.cbNextChange(Sender: TObject);
begin
if FActiveBox = nil then exit;
if FActiveBox.LogicElement <> cbNext.items.Objects[cbNext.ItemIndex] then
begin
FActiveBox.LogicElement.NextElement :=
TlogicElement(cbNext.items.Objects[cbNext.ItemIndex]);
end;
UpdateView;
end;

procedure TchLogicsEditor.cbModeChange(Sender: TObject);
begin
if FActiveBox = nil then exit;
FActiveBox.LogicElement.IsFirst := cbMode.ItemIndex = 1;
end;

procedure TchLogicsComponentEditor.ExecuteVerb(Index: Integer);
begin
inherited;
ShowEditor(TLogicProducer(Component));
end;

function TchLogicsComponentEditor.GetVerb(Index: Integer): string;
begin
case Index of
0: Result := 'Edit component...';
end;
end;

function TchLogicsComponentEditor.GetVerbCount: Integer;
begin
Result := 1;
end;

procedure TchLogicsComponentEditor.ShowEditor(LogicProducer: TLogicProducer);
var
glLogicsEditor: TchLogicsEditor;
Logics: TLogics;
begin
Logics := LogicProducer.Logics;
with Logics.Add do
begin
Left := 10; Top := 10;
IsFirst := true;
end;
with Logics.Add do
begin
Left := 200; Top := 30;
PrevElementID := Logics[0].ID;
end;
Logics[0].NextElementID := Logics[1].ID;
with Logics.Add do
begin
Left := 200; Top := 100;
end;

try
glLogicsEditor := TchLogicsEditor.Create(nil);
glLogicsEditor.Execute(LogicProducer);
```

```
finally
FreeAndNil(glLogicsEditor);
end;
end;


procedure TchLogicsEditor.tbNewClick(Sender: TObject);
var
LogicElement: TLogicElement;
begin
LogicElement := Logics.Add;
with LogicElement do
begin
Left := SB.Width div 2; Top := SB.Height div 2;
AddBox(LogicElement);
end;
end;

procedure TchLogicsEditor.cbNextFalseChange(Sender: TObject);
begin
if FActiveBox = nil then exit;
if FActiveBox.LogicElement <> cbNextFalse.items.Objects[cbNextFalse.ItemIndex]
then
begin
FActiveBox.LogicElement.NextFalseElement :=
TlogicElement(cbNextFalse.items.Objects[cbNextFalse.ItemIndex]);
end;
UpdateView;
end;

procedure TchLogicsEditor.eStepNameChange(Sender: TObject);
begin
if not Assigned(ActiveBox) then exit;
ActiveBox.LogicElement.Caption := eStepName.Text;
ActiveBox.Caption := eStepName.Text;
end;

procedure TchLogicsEditor.FormShow(Sender: TObject);
begin
SB.BufferedDraw := true;
end;

procedure TchLogicsEditor.ToolButton5Click(Sender: TObject);
var i: integer;
begin
pLog.Visible := true;

mLog.Lines.Clear;
reReslt.Text := '';

Logics.Analyze;

for i := 0 to SB.ControlCount-1 do
if (SB.Controls[i] is TchGroupBoxPlus) then
if TchGroupBoxPlus(SB.Controls[i]).LogicElement.IsTrue then
TchGroupBoxPlus(SB.Controls[i]).Colors.Caption := clGreen;

end;

procedure TchLogicsEditor.ToolButton7Click(Sender: TObject);
var
CommentArea: TCommentArea;
begin
CommentArea := LogicProducer.CommentAreas.Add;
with CommentArea do
begin
Left := SB.Width div 2; Top := SB.Height div 2;
Width := 100; Height := 100;
AddShape(CommentArea);
```

```
end;
end;

procedure TchShapePlus.Paint;
var
i: integer;
R: TRect;
str: string;
begin
inherited;

Canvas.Font.Color := clBlue;
Canvas.Font.Style := [fsBold];
str := CommentArea.Text;
R := Bounds(3, 2, Width, Height);
DrawText(Canvas.handle, PChar(str),length(str), R, DT_WORDBREAK);
end;

procedure TchLogicsEditor.cbIgnoreSpacesClick(Sender: TObject);
begin
LogicProducer.IgnoreSpaces := cbIgnoreSpaces.Checked;
end;

procedure TchLogicsEditor.pLeftDockOver(Sender: TObject; Source:
TDragDockObject; X, Y: Integer; State: TDragState; var Accept: Boolean);
begin
Accept := true;
end;

procedure TchLogicsEditor.pLeftUnDock(Sender: TObject; Client: TControl;
NewTarget: TWinControl; var Allow: Boolean);
begin
(Sender as TWinControl).Height := 6;
end;

procedure TchLogicsEditor.pLeftDockDrop(Sender: TObject; Source:
TDragDockObject; X, Y: Integer);
begin
(Sender as TWinControl).Height:=100;
SBar.Top:=1500;
end;

procedure TchLogicsEditor.ToolButton8Click(Sender: TObject);
var i: integer;
begin
if Logics.TraceItem = nil then
begin
Logics.StartAnalyze;
reReslt.Text := '';
end;
tbStop.Enabled :=true;
Logics.AnalyzeStep;
for i := 0 to SB.ControlCount-1 do
if (SB.Controls[i] is TchGroupBoxPlus) then
if TchGroupBoxPlus(SB.Controls[i]).LogicElement.IsTrue then
TchGroupBoxPlus(SB.Controls[i]).Colors.Caption := clGreen;
ShowMessage(Logics.Result);
end;
procedure TchLogicsEditor.tbStopClick(Sender: TObject);
var i: integer;
begin
      Logics.TraceItem := nil;
      tbStop.Enabled :=false;
      for i := 0 to SB.ControlCount-1 do
      if (SB.Controls[i] is TchGroupBoxPlus) then
      if TchGroupBoxPlus(SB.Controls[i]).LogicElement.IsTrue then
      TchGroupBoxPlus(SB.Controls[i]).Colors.Caption := clBtnShadow;
end;
```

```
procedure TchLogicsEditor.TabSet1Change(Sender: TObject; NewTab: Integer; var
AllowChange: Boolean);
begin
      PC.ActivePageIndex := NewTab;
end;
end.
```

Розроблена бібліотека файл HShape.pas

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
unit HShape;

interface
{$I DEF.INC}
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
ExtCtrls, glTypes, glUtils, glCommCl;

type
TRGNCombineMode = ( cmAND, cmCOPY, cmDIFF, cmOR, cmXOR );
THoleShapeType = (stRectangle, stSquare, stRoundRect, stRoundSquare, stEllipse,
stCircle);
TchHoleShape = class(TGraphicControl)
private
FShape            : THoleShapeType;
FShapeBitmap      : TBitmap;
FBevelInner       : TPanelBevel;
FBevelOuter       : TPanelBevel;
FBoldInner        : boolean;
FBoldOuter        : boolean;
FRectEllipse      : TPointClass;
FBevelOffset      : integer;
fNeedUpdateRGN    : boolean;
fDestroyed        : boolean;
fRunOnce          : boolean;
fNeedRebuildBitmapShape : boolean;
OldX,OldY,OldW,OldH : integer;
procedure SetEnabled( Value: boolean );
procedure SetEnabledDT( Value: boolean );
procedure SetShape( Value: THoleShapeType );
procedure SetShapeBitmap( Value: TBitmap );
procedure SetBevelInner( Value: TPanelBevel );
procedure SetBevelOuter( Value: TPanelBevel );
procedure SetBoldInner( Value: boolean );
procedure SetBoldOuter( Value: boolean );
procedure SetCombineMode( Value: TRGNCombineMode );
procedure SetBevelOffset( Value: integer );

procedure Update_;
procedure CalcRGNs;
procedure SmthChanged(Sender: TObject);
procedure SayAllDTEnabledState( EnabledDT: boolean );
protected
procedure Paint; override;
public
RGNOuter, RGNInner      : HRGN;
FCombineMode      : TRGNCombineMode;
FEnabledDT        : boolean;
FEnabled          : boolean;
constructor Create( AOwner : TComponent ); override;
destructor Destroy; override;
procedure UpdateRGN;
procedure Loaded; override;
published
property Align;
property ShowHint;
property ParentShowHint;
property PopupMenu;
property Visible;
property Enabled: boolean read FEnabled write SetEnabled
default true;
property EnabledAllInDesignTime: boolean read FEnabledDT write SetEnabledDT
default true;
property Shape: THoleShapeType read FShape write SetShape
```

```
default stEllipse;
property BevelInner: TPanelBevel read FBevelInner write SetBevelInner
default bvNone;
property BevelOuter: TPanelBevel read FBevelOuter write SetBevelOuter
default bvLowered;

property BevelInnerBold: boolean read FBoldInner write SetBoldInner
default true;
property BevelOuterBold: boolean read FBoldOuter write SetBoldOuter
default true;
property CombineMode: TRGNCombineMode read FCombineMode write SetCombineMode
default cmDIFF;
property BevelOffset: integer read FBevelOffset write SetBevelOffset
default 0;
property RectEllipse: TPointClass read FRectEllipse write FRectEllipse;
property ShapeBitmap: TBitmap read FShapeBitmap write SetShapeBitmap;
end;

procedure Register;

implementation
const
aCombMode : array[0..4] of integer = (RGN_AND, RGN_COPY, RGN_DIFF, RGN_OR,
RGN_XOR );
procedure Register;
begin
RegisterComponents('Proba', [TchHoleShape]);
end;
constructor TchHoleShape.Create( AOwner : TComponent );
begin
inherited;
FShapeBitmap:=TBitmap.Create;
FEnabled := (Owner is TWinControl);

ControlStyle := ControlStyle - [csOpaque];
FEnabledDT:=FEnabled;
fDestroyed:=false;
FRectEllipse:=TPointClass.Create;
FRectEllipse.x:=30; FRectEllipse.y:=30;
FRectEllipse.OnChanged:=SmthChanged;
FShape:=stEllipse;
FBevelOuter:=bvLowered;
FBevelInner:=bvNone;
FCombineMode:=cmDIFF;
FBoldInner:=true; FBoldOuter:=true;
FRectEllipse.y:=45; FRectEllipse.x:=45;
FBevelOffset:=0;
Width:=112; Height:=112;
fNeedUpdateRGN:=false;
fRunOnce:=true;
end;

destructor TchHoleShape.Destroy;
begin
FShapeBitmap.Free;
FRectEllipse.Free;
if not (csDestroying in Owner.ComponentState) then
begin FEnabledDT:=false; FEnabled:=false; UpdateRGN(); end;
inherited;
end;

procedure TchHoleShape.Paint;
var
r             :TRect;
H,W,EH,EW,i   :integer;

procedure DrawShape( Bevel: TPanelBevel; fBold, fRect: boolean );
procedure SetPenAndBrush( c: Tcolor );
begin
```

```
Canvas.Pen.Color:=c;
if fRect and ((EW and EH)=0) then
begin Canvas.Brush.Style:=bsClear; end
else begin Canvas.Brush.Color:=c; end
end;
begin
Canvas.Brush.Style:=bsClear;//bsSolid bsClear
i := integer( fBold );
with Canvas do
case Bevel of
bvLowered:
begin
SetPenAndBrush( clBtnHighlight );
if fRect then RoundRect( R.Left, R.Top, R.Right, R.Bottom, EW, EH)
else Ellipse( R.Left, R.Top, R.Right, R.Bottom );
SetPenAndBrush( clBtnShadow );
if fRect then RoundRect( R.Left, R.Top, R.Right-1, R.Bottom-1, EW, EH)
else Ellipse( R.Left, R.Top, R.Right-1, R.Bottom-1 );
if FBold then begin
SetPenAndBrush( cl3DDkShadow );
if fRect then RoundRect( R.Left+1, R.Top+1, R.Right-1, R.Bottom-1, EW, EH)
else Ellipse( R.Left+1, R.Top+1, R.Right-1, R.Bottom-1 );
end;
InflateRect( R, -1, -1 ); inc(R.Left,i); inc(R.Top,i);
end;
bvRaised:
begin
SetPenAndBrush( clBtnHighlight );
if fRect then RoundRect( R.Left, R.Top, R.Right, R.Bottom, EW, EH)
else Ellipse( R.Left, R.Top, R.Right, R.Bottom );
if FBold then begin
SetPenAndBrush( cl3DDkShadow );
if fRect then RoundRect( R.Left+1, R.Top+1, R.Right, R.Bottom, EW, EH)
else Ellipse( R.Left+1, R.Top+1, R.Right, R.Bottom );
end;
SetPenAndBrush( clBtnShadow );
if fRect then RoundRect( R.Left+1, R.Top+1, R.Right-i, R.Bottom-i, EW, EH)
else Ellipse( R.Left+1, R.Top+1, R.Right-i, R.Bottom-i );
InflateRect( R, -1, -1 ); dec(R.Right,i); dec(R.Bottom,i);
end;
else
begin
Brush.Color:=clBlack;
FrameRect( Rect(Left, Top, Left+W, Top+H) );
end;
end;
SetPenAndBrush( clBtnFace );

end;

begin
fNeedUpdateRGN := fNeedUpdateRGN
or(OldX<>Left)or(OldY<>Top)or(OldW<>Width)or(OldH<>Height);

if fNeedUpdateRGN then UpdateRGN();
OldX:=Left; OldY:=Top; OldW:=Width; OldH:=Height;

if IsItAFilledBitmap( FShapeBitmap ) then
begin
BitBlt( Canvas.handle, -1,-1, Width, Height, FShapeBitmap.Canvas.handle, 0,0,
SRCCopy);
exit;
end;

case FShape of
stRectangle, stRoundRect, stEllipse:
begin H:=Height; W:=Width; end
else
begin H:=min(Height,Width); W:=H; end;
```

```
end;
R := Bounds( 0, 0, W, H );
with Canvas do
case FShape of
stRectangle, stSquare, stRoundRect, stRoundSquare:
begin
if (FShape = stRectangle)or(FShape = stSquare) then
begin EW:=0; EH:=0; end;
if (FShape = stRoundRect)or(FShape = stRoundSquare) then
begin EW:=FRectEllipse.x; EH:=FRectEllipse.y; end;
DrawShape( FBevelOuter, FBoldOuter, true );
InflateRect( R, -FBevelOffset, -FBevelOffset );
DrawShape(FBevelInner, FBoldInner, true );
Pen.Color:=clBtnFace;
Rect( R.Left, R.Top, R.Right, R.Bottom );
end;
stEllipse, stCircle:
begin
DrawShape( FBevelOuter, FBoldOuter, false );
InflateRect( R, -FBevelOffset, -FBevelOffset );
DrawShape(FBevelInner, FBoldInner, false );
end;
end;
end;
//-------------------------------------------------------
procedure TchHoleShape.CalcRGNs;
var
H, W, xOffs, yOffs      :integer;
R                 :TRect;

BmpInfo           :Windows.TBitmap;
BorderStyle: TFormBorderStyle;
procedure CalcShape( Bevel: TPanelBevel; fBold: boolean );
var
i: integer;
begin
i := integer( fBold );
case Bevel of
bvLowered: begin InflateRect( R, -1, -1 ); inc(R.Left,i); inc(R.Top,i); end;
bvRaised: begin InflateRect( R, -1, -1 ); dec(R.Right,i); dec(R.Bottom,i); end;
end;
end; procedure CalcBmpRgn(var rgn: HRGN );
var
i,j   :integer;
rgn2: HRGN;
TransparentColor: TColor;
begin
TransparentColor := FShapeBitmap.Canvas.Pixels[0, FShapeBitmap.Height-1];
for j:=0 to FShapeBitmap.Height do
for i:=0 to FShapeBitmap.Width do
begin
if FShapeBitmap.Canvas.Pixels[i,j] <> TransparentColor then continue;
RGN2 := CreateRectRgn(i, j, i+1, j+1);
CombineRgn( RGN, RGN2, RGN, RGN_OR );
DeleteObject( RGN2 );
end;
end;
begin
if not FShapeBitmap.Empty then
begin
if fNeedRebuildBitmapShape then} with FShapeBitmap do
begin
GetObject( FShapeBitmap.Handle, sizeof(Windows.TBitmap), @BmpInfo );
if RGNOuter <> 0 then DeleteObject( RGNOuter );
if RGNInner <> 0 then DeleteObject( RGNInner );
RGNInner := CreateRectRgn(0, 0, 0, 0);
CalcBmpRgn(RGNInner);
fNeedRebuildBitmapShape := false;
end;
```

```
end
else
begin
case FShape of
stRectangle, stRoundRect, stEllipse:
begin H:=Height; W:=Width; end
else
begin H:=min(Height,Width); W:=H; end;
end;
R := Bounds( 0, 0, W, H );
if RGNOuter <> 0 then DeleteObject( RGNOuter );
if RGNInner <> 0 then DeleteObject( RGNInner );

if FBevelOffset <> 0 then
begin
CalcShape( FBevelOuter, FBoldOuter );
OffsetRect(R,1,1);
end;
case FShape of
stRectangle, stSquare:
RGNOuter := CreateRectRgn( R.Left, R.Top, R.Right, R.Bottom );
stRoundRect, stRoundSquare:
RGNOuter := CreateRoundRectRgn( R.Left, R.Top, R.Right, R.Bottom,
FRectEllipse.x, FRectEllipse.y );
stEllipse, stCircle:
RGNOuter := CreateEllipticRgn( R.Left, R.Top, R.Right, R.Bottom );
end;
if FBevelOffset=0 then CalcShape( FBevelOuter, FBoldOuter );
InflateRect( R, -FBevelOffset, -FBevelOffset );
if FBevelOffset=0 then CalcShape( FBevelInner, FBoldInner )
else OffsetRect(R,-1,-1);
case FShape of
stRectangle, stSquare:
RGNInner := CreateRectRgn( R.Left+1, R.Top+1, R.Right+1, R.Bottom+1 );
stRoundRect, stRoundSquare:
RGNInner := CreateRoundRectRgn( R.Left+1, R.Top+1, R.Right+2, R.Bottom+2,
FRectEllipse.x, FRectEllipse.y );
stEllipse, stCircle:
RGNInner:=CreateEllipticRgn(R.Left+1,R.Top+1, R.Right+2, R.Bottom+2 );
end;
end;

if Owner is TForm then
begin
if csDesigning in ComponentState then BorderStyle := bsSizeable
     else BorderStyle := TForm(Owner).BorderStyle;
case BorderStyle of
bsSizeable:
begin
xOffs := GetSystemMetrics(SM_CXFRAME)-1;
yOffs := GetSystemMetrics(SM_CYFRAME)-1;
inc( yOffs, GetSystemMetrics(SM_CYCAPTION) );
end;
bsDialog:
begin
xOffs := GetSystemMetrics(SM_CXDLGFRAME)-1;
yOffs := GetSystemMetrics(SM_CYDLGFRAME)-1;
inc( yOffs, GetSystemMetrics(SM_CYCAPTION) );
end;
bsSingle:
begin
xOffs := GetSystemMetrics(SM_CXBORDER);
yOffs := GetSystemMetrics(SM_CYBORDER);
inc( yOffs, GetSystemMetrics(SM_CYCAPTION) );
end;
bsToolWindow:
begin

xOffs := GetSystemMetrics(SM_CXBORDER);
```

```
yOffs := GetSystemMetrics(SM_CYBORDER);
inc( yOffs, GetSystemMetrics(SM_CYSMCAPTION) );
end;
bsSizeToolWin:
begin
xOffs := GetSystemMetrics(SM_CXSIZEFRAME);
yOffs := GetSystemMetrics(SM_CYSIZEFRAME);
inc( yOffs, GetSystemMetrics(SM_CYSMCAPTION) );
end;
else
begin
xOffs := -1;
yOffs := -1;
end;
end;

OffsetRgn( RGNInner, Left+xOffs, Top+yOffs );
OffsetRgn( RGNOuter, Left+xOffs, Top+yOffs );
end;

fRunOnce:=false;
end;
//-------------------------------------------------------
procedure TchHoleShape.SayAllDTEnabledState( EnabledDT: boolean );
var
i: integer;
begin
for i:=0 to TWinControl(Owner).ControlCount-1 do with TWinControl(Owner) do
begin
if (Controls[i] is TchHoleShape) then
begin
TchHoleShape(Controls[i]).FEnabledDT := EnabledDT;
end;
end;

end;
//-------------------------------------------------------
procedure TchHoleShape.UpdateRGN;
var
i: integer;
NewRGN: HRGN;
begin
if not(Owner is TWinControl) then exit;
NewRGN := CreateRectRgn( 0, 0, 2000, 1000 );

for i:=0 to TWinControl(Owner).ControlCount-1 do with TWinControl(Owner) do
begin
if Controls[i] is TchHoleShape then
with TchHoleShape(Controls[i])do
if ((csDesigning in ComponentState)and FEnabledDT)
or ((not(csDesigning in ComponentState))and FEnabled) then
begin
CalcRGNs;
CombineRgn( NewRGN, NewRGN, RGNInner, aCombMode[ integer(FCombineMode) ] )
end;
end;

SetWindowRgn( TWinControl(Owner).Handle, NewRGN, true );
fNeedUpdateRGN:=false;
end;

procedure TchHoleShape.Update_;
begin
      if csLoading in ComponentState then exit;
      UpdateRGN();
      Refresh;
end;
procedure TchHoleShape.SmthChanged(Sender: TObject);
      begin
```

```
      Update_;
end;
procedure TchHoleShape.SetEnabled( Value: boolean );
begin
      if (FEnabled = Value)or not(Owner is TWinControl) then exit;
      FEnabled := Value; Update_;
end;
procedure TchHoleShape.SetEnabledDT( Value: boolean );
begin
      if (FEnabledDT = Value)or not(Owner is TWinControl) then exit;
      FEnabledDT := Value; SayAllDTEnabledState( FEnabledDT );
      Update_;
end;
procedure TchHoleShape.SetShape( Value: THoleShapeType );
begin
      if FShape = Value then exit;
      FShape := Value; Update_;
end;
procedure TchHoleShape.SetShapeBitmap( Value: TBitmap );
begin
      if FShapeBitmap = Value then exit;
      fNeedRebuildBitmapShape := true;
      FShapeBitmap.Assign(Value);
      if Assigned(FShapeBitmap) then
      begin
            Width := FShapeBitmap.Width;
            Height := FShapeBitmap.Width;
      end;
      Update_();
end;
procedure TchHoleShape.SetBevelInner( Value: TPanelBevel );
begin
      if FBevelInner = Value then exit;
      FBevelInner := Value; Update_;
end;
procedure TchHoleShape.SetBevelOuter( Value: TPanelBevel );
begin
      if FBevelOuter = Value then exit;
      FBevelOuter := Value; Update_;
end;
procedure TchHoleShape.SetBoldInner( Value: boolean );
begin
      if FBoldInner = Value then exit;
      FBoldInner := Value; Update_;
end;
procedure TchHoleShape.SetBoldOuter( Value: boolean );
begin
      if FBoldOuter = Value then exit;
      FBoldOuter := Value; Update_;
end;
procedure TchHoleShape.SetCombineMode( Value: TRGNCombineMode );
begin
      if FCombineMode = Value then exit;
      FCombineMode := Value; Update_;
end;
procedure TchHoleShape.SetBevelOffset( Value: integer );
begin
      if (FBevelOffset = Value)or(Value < 0) then exit;
      if (Value > width-2)or(Value > height-2) then Value:=min(width,height)-2;
      FBevelOffset := Value; Update_;
end;
procedure TchHoleShape.Loaded;
begin
      inherited;
      fNeedRebuildBitmapShape := true;
      UpdateRGN(); Refresh;
end;
end.
```

Розроблена бібліотека файл glPage.pas

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
unit glPage;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,

ComCtrls, CommCtrl, glTypes, glUtils, DrawTab, TabComm, ExtCtrls, glCommCl
const
TCM_SETTEXTCOLOR  =     (TCM_FIRST + 36);
type
TchPageControl = class (TPageControl)
private
FGlyphs           : TImageList;
FSingleGlyph      : boolean;
FTabStyle           : TTabStyle;
FTabSelectedStyle  : TTabStyle;
FWallpaper  : TTabsWallpaper;
FDrawGlyphsOption : TchWallpaperOption;
FLookLikeButtons  : boolean;
FTabsPosition     : TchSide;
FOptions          : TchTabOptions;
FFontDirection    : TchLabelDir;
FOnGetItemColor    : TchOnGetItemColorEvent;
FOnGetItemFontColor: TchOnGetItemColorEvent;
FOnGetGradientColors: TchOnGetGradientColors;

GlyphsChangeLink  : TChangeLink;
DrawTabStr  : TDRAWTABSTRUCT;
GlyphTmpBitmap    : TBitmap;
FontNormal         : TFont;
FontSelected       : TFont;
fNotFirst   : boolean;
aTabColors         : array[0..100] of TColor;

function GeTchyphIndex(Index: Integer): Integer;
procedure SeTchyphIndex(Index: Integer; imgIndex: Integer);
procedure SeTchyphs(Value: TImageList);
procedure SetSingleGlyph(Value: boolean);
procedure SetDrawGlyphsOption(Value: TchWallpaperOption);
procedure SetLookLikeButtons(Value: boolean);
procedure SetTabsPosition(Value: TchSide);
procedure SetOptions(Value: TchTabOptions);
procedure SetFontDirection(Value: TchLabelDir);
function GetFont: TFont;
procedure SetFont(Value: TFont);
function GetTabColor( Index: integer ): TColor;
procedure SetTabColor(Index: integer; Value: TColor);
procedure SmthChanged(Sender: TObject);
procedure FontsChanged(Sender: TObject);
procedure DrawItem(lpDrawItemStr:PDRAWITEMSTRUCT);
procedure CNDrawItem(var Message: TWMDrawItem); message CN_DRAWITEM;
procedure CMFontChanged(var Message: TMessage); message CM_FONTCHANGED;
procedure SetTabStyle(const Value: TTabStyle);
procedure SetTabSelectedStyle(const Value: TTabStyle);
protected
procedure GlyphsListChanged(Sender: TObject);
procedure WndProc(var Message: TMessage); override;
procedure CreateParams(var Params: TCreateParams); override;
procedure Loaded; override;
procedure Notification(AComponent: TComponent; Operation: TOperation); override;
public
fSupressDraw       : boolean;
procedure RemakeFonts;
constructor Create (AOwner: TComponent); override;
destructor Destroy; override;
```

```
property GlyphIndex[Index: Integer]: Integer read GeTchyphIndex write
SeTchyphIndex;
property TabColor[ Index: integer ]: TColor read GetTabColor write SetTabColor;
property GlyphState[Index: Integer]: Integer read GeTchyphState write
SeTchyphState;
published
property Glyphs: TImageList read FGlyphs write SeTchyphs;
property SingleGlyph: boolean read FSingleGlyph write SetSingleGlyph
default false;
property TabStyle: TTabStyle read FTabStyle write SetTabStyle;
property TabSelectedStyle: TTabStyle read FTabSelectedStyle write
SetTabSelectedStyle;
property Wallpaper: TTabsWallpaper read FWallpaper write FWallpaper;
property DrawGlyphsOption: TchWallpaperOption
read FDrawGlyphsOption write SetDrawGlyphsOption default fwoNone;
property LookLikeButtons: boolean read FLookLikeButtons write SetLookLikeButtons
default false;
property TabsPosition: TchSide read FTabsPosition write SetTabsPosition
default fsdTop;
property Options: TchTabOptions read FOptions write SetOptions;
property FontDirection: TchLabelDir
read FFontDirection write SetFontDirection default fldLeftRight;
property Font: TFont read GetFont write SetFont;
property OnGetItemColor: TchOnGetItemColorEvent read FOnGetItemColor write
FOnGetItemColor;
property OnGetItemFontColor: TchOnGetItemColorEvent read FOnGetItemFontColor
write FOnGetItemFontColor;
property OnGetGradientColors: TchOnGetGradientColors read FOnGetGradientColors
write FOnGetGradientColors;
end;

procedure Register;

implementation
const FontDirs:array[TchSide]of TchLabelDir
=(fldDownUp,fldLeftRight, fldUpDown, fldLeftRight );
procedure Register;
begin
RegisterComponents('Proba', [TchPageControl]);
end;
constructor TchPageControl.Create (AOwner: TComponent);
begin
inherited Create(AOwner);
TabStop:=false;
FTabStyle        := TTabStyle.Create(self);
with FTabStyle do
begin
BackgrColor := clBtnShadow;
Font.Color := clBtnHighlight;
CaptionHAlign := fhaCenter;
end;
FTabSelectedStyle := TTabStyle.Create(self);
with FTabSelectedStyle do
begin
BackgrColor := clBtnFace;
Font.Color := clBtnText;
CaptionHAlign := fhaCenter;
end;

FWallpaper        := TTabsWallpaper.Create;
FontNormal            := TFont.Create;
FontSelected          := TFont.Create;
DrawTabStr.Font_      := TFont.Create;

FTabStyle.Font.Name:='Arial';
FTabSelectedStyle.Font.Name:='Arial';

GlyphTmpBitmap    := TBitmap.Create;
GlyphsChangeLink  := TChangeLink.Create;
```

```
GlyphsChangeLink.OnChange:=GlyphsListChanged;
DrawTabStr.Gradient := TGradient.Create;
FSingleGlyph:=false;
FDrawGlyphsOption:=fwoNone;
FTabsPosition:=fsdTop;
FOptions:=[ftoAutoFontDirection,ftoExcludeGlyphs];
FFontDirection:=fldLeftRight;
FTabStyle.OnChanged             := SmthChanged;
FTabSelectedStyle.OnChanged   := SmthChanged;
FTabStyle.OnFontChanged := FontsChanged;
FTabSelectedStyle.OnFontChanged := FontsChanged;
FWallpaper.OnChanged           := SmthChanged;
FillMemory( @aTabColors, sizeof(aTabColors), $FF );
end;

destructor TchPageControl.Destroy;
begin
FTabStyle.Free;
FTabSelectedStyle.Free;
GlyphTmpBitmap.Free;
FWallpaper.Free;
GlyphsChangeLink.Free;
FontNormal.Free;
FontSelected.Free;
DrawTabStr.Font_.Free;
if Assigned(DrawTabStr.Gradient) then DrawTabStr.Gradient.Free;
inherited;
end;

procedure TchPageControl.SmthChanged;
begin Invalidate; end;


procedure TchPageControl.FontsChanged;
begin RemakeFonts; Invalidate; end;

procedure TchPageControl.CreateParams(var Params: TCreateParams);
const PosStyles : array[TchSide]of DWORD =
( TCS_VERTICAL, 0, TCS_VERTICAL or TCS_RIGHT, TCS_BOTTOM or TCS_SCROLLOPPOSITE
or TCS_BUTTONS );
begin
inherited CreateParams(Params);
with Params do
begin
if LookLikeButtons then Style := Style or TCS_BUTTONS;
Style := Style or TCS_OWNERDRAWFIXED or PosStyles[FTabsPosition];
end;
end;

procedure TchPageControl.Loaded;
begin
inherited Loaded; RemakeFonts;
if Assigned(Wallpaper.Bitmap)and(not Wallpaper.Bitmap.Empty)
then Wallpaper.bmp := Wallpaper.Bitmap;
end;

procedure TchPageControl.Notification(AComponent: TComponent; Operation:
TOperation);
begin
inherited Notification(AComponent, Operation);
if Assigned(Wallpaper)and(AComponent = Wallpaper.Image) and (Operation =
opRemove) then Wallpaper.Image := nil;
if (AComponent = FGlyphs) and (Operation = opRemove) then Glyphs := nil;
end;

procedure TchPageControl.CNDrawItem(var Message: TWMDrawItem);
begin
DrawItem(Pointer(Message.DrawItemStruct));
end;
```

```
procedure TchPageControl.WndProc(var Message: TMessage);
var
GlyphID:integer;
begin
inherited WndProc(Message);
with Message do
case Msg of
TCM_INSERTITEM:
begin  result:=0;
if not Assigned(FGlyphs) then exit;
GlyphID:=-1;
if FSingleGlyph then GlyphID:=0
else if wParam < FGlyphs.Count then GlyphID:=wParam;
if GlyphID=-1 then exit;
TTCItem(Pointer(Message.lParam)^).iImage:=GlyphID;
TTCItem(Pointer(Message.lParam)^).Mask:=TCIF_IMAGE;

SendMessage( handle, TCM_SETITEM, wParam, lParam );
end;
TCM_DELETEITEM: begin end;
TCM_DELETEALLITEMS: begin end;
end;
end;

procedure TchPageControl.GlyphsListChanged(Sender: TObject);
begin
if HandleAllocated then SendMessage(Handle, TCM_SETIMAGELIST, 0,
Longint(TImageList(Sender).Handle));
end;

procedure TchPageControl.DrawItem(lpDrawItemStr:PDRAWITEMSTRUCT);
var
FontColor: TColor;
begin
if fSupressDraw then exit;
with lpDrawItemStr^ do
if CtlType=ODT_TAB then
begin
DrawTabStr.lpDrawItemStr := lpDrawItemStr;
DrawTabStr.Caption:=Tabs[ItemID];

if GlyphIndex[ItemID]<>-1 then
begin
FGlyphs.GetBitmap( GlyphIndex[ItemID], GlyphTmpBitmap );
DrawTabStr.Glyph:=GlyphTmpBitmap;
end else DrawTabStr.Glyph:=nil;

if (itemState and ODS_DISABLED)<>0 then
begin
DrawTabStr.BoxStyle := FTabStyle;
DrawTabStr.Font_.Assign(FontNormal);
end
else if (itemState and ODS_SELECTED)<>0 then begin
DrawTabStr.BoxStyle := FTabSelectedStyle;
DrawTabStr.Font_.Assign(FontSelected);
end
else begin
DrawTabStr.BoxStyle := FTabStyle;
DrawTabStr.Font_.Assign(FontNormal);
end;

if Assigned(OnGetItemFontColor) then
begin
OnGetItemFontColor( self, ItemID, FontColor );
DrawTabStr.Font_.Color := FontColor;
end;
DrawTabStr.GlyphOption := FDrawGlyphsOption;
DrawTabStr.Wallpaper:=FWallpaper;
```

```
DrawTabStr.ClientR:=ClientRect;
DrawTabStr.TabsCount:=Tabs.Count;
DrawTabStr.fButton:=LookLikeButtons;
DrawTabStr.Position:=TabsPosition;
DrawTabStr.Options:=Options;
DrawTabStr.FontDirection:=FontDirection;

if Assigned(OnGetGradientColors) then OnGetGradientColors( self, ItemID,
DrawTabStr.Gradient);

if Assigned(OnGetItemColor) then OnGetItemColor( self, ItemID,
DrawTabStr.BackgrColor_) else
if aTabColors[ItemID] <> -1 then DrawTabStr.BackgrColor_ := aTabColors[ItemID]
else DrawTabStr.BackgrColor_ := DrawTabStr.BoxStyle.BackgrColor;
DrawOwnTab( DrawTabStr );
end;
end;

procedure TchPageControl.CMFontChanged(var Message: TMessage);
begin
inherited;
if ftoInheriteTabFonts in Options then
begin
FTabStyle.Font.Assign(inherited Font);
FTabSelectedStyle.Font.Assign(inherited Font);
Disabled.Assign(inherited Font);
RemakeFonts;
end;
end;

procedure TchPageControl.RemakeFonts;
const
RadianEscapments:array [TchlabelDir] of integer = (0,-1800,-900,900);
begin
if csReading in ComponentState then exit;
if fNotFirst then DeleteObject( FTabStyle.Font.Handle );
fNotFirst:=true;

FontNormal.Handle := CreateRotatedFont( FTabStyle.Font,
RadianEscapments[FFontDirection]);
FontNormal.Color := FTabStyle.Font.Color;
FontSelected.Handle := CreateRotatedFont( FTabSelectedStyle.Font,
RadianEscapments[FFontDirection]);
FontSelected.Color := FTabSelectedStyle.Font.Color;
end;

procedure TchPageControl.SeTchyphs(Value: TImageList);
var i: integer;
label SkipAutoGlypsSet;
begin
if Assigned(FGlyphs) then  FGlyphs.UnregisterChanges(GlyphsChangeLink);
FGlyphs := Value;
if Assigned(FGlyphs) then
begin
FGlyphs.RegisterChanges(GlyphsChangeLink);
SendMessage(Handle, TCM_SETIMAGELIST, 0, Longint(FGlyphs.Handle));
for i:=0 to min( Tabs.Count-1, FGlyphs.Count-1) do

if GlyphIndex[i]<>-1 then goto SkipAutoGlypsSet;
SetSingleGlyph(FSingleGlyph);
SkipAutoGlypsSet:
end
else SendMessage(Handle, TCM_SETIMAGELIST, 0, Longint(0));
end;

procedure TchPageControl.SeTchyphIndex( Index: Integer; imgIndex: Integer);
var
r     : TRect;
Item  : TTCItem;
```

```
begin
Item.iImage := imgIndex;
Item.mask := TCIF_IMAGE;
SendMessage( Handle, TCM_SETITEM, Index, Longint(@Item) );
SendMessage( Handle, TCM_GETITEMRECT, Index, Longint(@r) );
InvalidateRect( Handle, @r, true );
end;

function TchPageControl.GeTchyphIndex( Index: Integer ): Integer;
var
imgItem: TTCItem;
begin
if Assigned(FGlyphs) then
begin
imgItem.mask := TCIF_IMAGE;
SendMessage( Handle, TCM_GETITEM, Index, Longint(@imgItem) );
Result := imgItem.iImage;
end
else Result := -1;
end;

procedure TchPageControl.SetSingleGlyph(Value: boolean);
var i: integer;
begin
FSingleGlyph:=Value;
if (Tabs=nil)or(FGlyphs=nil) then exit;
if FSingleGlyph then
for i:=0 to Tabs.Count-1 do GlyphIndex[i]:=0
else
for i:=0 to Tabs.Count-1 do
if FGlyphs.Count >= i then GlyphIndex[i]:=i else break;
end;

procedure TchPageControl.SetDrawGlyphsOption(Value: TchWallpaperOption);
begin
if FDrawGlyphsOption = Value then exit;
FDrawGlyphsOption := Value; Invalidate;
end;

procedure TchPageControl.SetLookLikeButtons(Value: boolean);
begin

if FLookLikeButtons = Value then exit;
FLookLikeButtons := Value;
RecreateWnd;
end;

procedure TchPageControl.SetTabsPosition(Value: TchSide);
begin
if FTabsPosition = Value then exit;
FTabsPosition := Value; RecreateWnd;
if (ftoAutoFontDirection in FOptions)and not(csLoading in ComponentState) then
FontDirection := FontDirs[TabsPosition];
end;

procedure TchPageControl.SetOptions(Value: TchTabOptions);
begin
if FOptions = Value then exit; FOptions := Value;
if ftoAutoFontDirection in FOptions then
FontDirection := FontDirs[TabsPosition];
Invalidate;
end;

procedure TchPageControl.SetFontDirection(Value: TchlabelDir);
begin
if FFontDirection = Value then exit;
FFontDirection := Value; RemakeFonts;
Invalidate;
end;
```

```
function TchPageControl.GetFont: TFont;
begin Result := inherited Font; end;

procedure TchPageControl.SetFont(Value: TFont);
begin
inherited Font := Value;
if ftoInheriteTabFonts in Options then
begin
FTabStyle.Font.Assign(inherited Font);
FTabSelectedStyle.Font.Assign(inherited Font);
end;
end;


function TchPageControl.GetTabColor( Index: integer ): TColor;
begin
if Index<100 then Result := aTabColors[Index] else Result := -1;
end;

procedure TchPageControl.SetTabColor(Index: integer; Value: TColor);
var
TCItem: TTCItem;
begin
if (Index<100)and(TabColor[Index] <> Value) then aTabColors[Index] := Value
else exit;
if not fSupressDraw then
begin
Repaint;
TCItem.mask := TCIF_TEXT;
TCItem.pszText := PChar(Tabs[Index]);
SendMessage( Handle, TCM_SETITEM, Index, Longint(@TCItem));
end;
end;

procedure TchPageControl.SetTabStyle(const Value: TTabStyle);
begin
FTabStyle := Value;
RemakeFonts;
end;

procedure TchPageControl.SetTabSelectedStyle(const Value: TTabStyle);
begin
FTabSelectedStyle := Value;
RemakeFonts;
end;

end.
```

Файл frmAbout.pas – Файл інформації про розробника програмного продукту

```
{Центральноукраїнський Національний Технічний Університет
Нетеса Владислав Юрійович 2021 рік}
unit frmAbout;
interface
uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls,  Buttons,
ExtCtrls, jpeg;
type
  TAboutBox = class(TForm)
    Panel1: TPanel;
    ProgramIcon: TImage;
    ProductName: TLabel;
    Version: TLabel;
    Copyright: TLabel;
    Comments: TLabel;
    Memo1: TMemo;
    OKButton: TButton;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var  AboutBox: TAboutBox;
implementation
{$R *.dfm}
procedure TAboutForm.FormCreate(Sender: TObject);
begin
Memo1.Clear;
Memo1.Lines.Add('КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА');
Memo1.Lines.Add('');
Memo1.Lines.Add('на  тему:');
Memo1.Lines.Add('');
Memo1.Lines.Add(' Програмне забезпечення системи кібербезпеки для моделювання
взламу протоколу WPA2 на базі реалізації атаки KRACK');
Memo1.Lines.Add('');
Memo1.Lines.Add('');
Memo1.Lines.Add('');
Memo1.Lines.Add('Керівник: Смірнов С.А.');
Memo1.Lines.Add('');
Memo1.Lines.Add('Розробив: студент Нетеса Владислав Юрійович');
Memo1.Lines.Add('                    гр. КБ-18-3СК');
Memo1.Lines.Add('');
Memo1.Lines.Add('');
Memo1.Lines.Add('м. Кропивницький 2021');
Memo1.Lines.Add('');
Memo1.Lines.Add('');
end;
procedure TAboutForm.Button1Click(Sender: TObject);
begin
AboutForm.Close;
end;
end.
```