Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра кібербезпеки та комп'ютерних систем і мереж

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Метод тестування на проникнення, як засіб забезпечення безпеки корпоративної мережі
Назва теми

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

КРМКІ.015047.19.01.07 ПЗ

Виконав: студент 2 курсу, група КІ1м-19-1 Підпис Гавронський В.Є.

Керівник доц., к. т. н, доцент кафедри КбКСМ Підпис Муляр І.В.

Нормоконтролер доц., к. т. н, доцент кафедри КбКСМ Підпис Муляр І.В.

До захисту допускаю:
Зав. кафедри КбКСМ, к.т.н., доц Підпис Кльоц Ю.П.

4 12 2020 р.

Хмельницький, 2020

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА ПІДГОТОВКИ МАГІСТРА

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц

"4" 09 2020 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Гавронському В.Є.

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод тестування на проникнення, як засіб забезпечення безпеки корпоративної мережі

2. Керівник проекту (роботи) к.т.н., доц. Муляр І.В.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом № 118 ректора університету додаток №23 від 01.09.2020

2. Строк подання студентом проекту (роботи) на кафедру 3.12.2020

3. Вихідні дані до проекту (роботи) Тестування безпеки програмного забезпечення корпоративної мережі, захист інформації, GERT-мережа

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

Аналіз методологій тестування безпеки програмного забезпечення корпоративної мережі

Розробка математичних моделей тестування на проникнення

Метод пошуку алгоритму з двійкового коду для діагностики безпеки програмного забезпечення корпоративної мережі

Перевірка результатів моделювання

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Загальна характеристика магістерської роботи, класифікація методів тестування на проникнення, загальна схема тестування, основні аспекти перевірки архітектури та дизайну ПЗ, порівняльна характеристика найбільш відомих підходів математичної формалізації процесу тестування безпеки ПЗ, алгоритм генерації коду кібератаки, стохастична GERT-мережа, процес виділення алгоритму з двійкового коду, висновки

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання пр
Відповідальний за оформлення кваліфікаційної роботи	Муляр І.В.		

7. Дата видачі завдання «1» лютого 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи	Термін виконання етапів кваліфікаційної роботи	Пр
1	Вибір напряму дослідження та узгодження тематики КРМ з керівником	2.02.2020	
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	2.03.2020	
3	Робота над розділом 1 – аналіз моделей та методів тестування безпеки програмних засобів корпоративної мережі; постановка задачі	1.04.2020	
4	Робота над розділом 2 – розробка математичних моделей тестування на проникнення	1.05.2020	
5	Робота над науковою статтею	1.06.2020	
6	Робота над розділом 3 – метод виділення алгоритму з двійкового коду для аналізу безпеки програмного забезпечення	1.09.2020	
7	Робота над розділом 4 – оцінювання достовірності отриманих результатів	1.10.2020	
8	Узгодження отриманих; оформлення пояснювальної записки згідно вимог	1.11.2020	
9	Оформлення графічної частини	5.11.2020	
10	Попередній захист КРМ	10.11.2020	
11	Захист КРМ на засіданні ЕК	8.12.2020	

Студент _____ В.С. Гавронський
Підпис Ініціали, прізвище

Керівник кваліфікаційної роботи _____ І.В. Муляр
Підпис Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: Метод тестування на проникнення, як засіб забезпечення безпеки корпоративної мережі

Автор роботи: Гавронський Віталій Євгенович

Керівник роботи: к.т.н., доц. Муляр Ігор Володимирович

Пояснювальна записка: 72 сторінки, 32 рисунка, 6 таблиць, 3 додатки, 64 посилання.

Перелік ключових слів: тестування безпеки програмного забезпечення корпоративної мережі, захист інформації, GERT-мережа.

Метою кваліфікаційної роботи є встановлення прийнятного рівня безпеки програмного забезпечення в корпоративній мережі шляхом розробки методів її тестування.

Дана кваліфікаційна робота присвячена розробці методу тестування на проникнення для виявлення вразливостей, що дозволить в подальшому забезпечити безпеку корпоративної мережі. Для цього розроблено математичну модель кібератаки, яка дозволяє дослідити основні етапи кібератаки та в подальшому надати практичні поради для захисту мережі від подібних атак, удосконалено математичну модель діагностики системи управління ресурсами мережі, що дозволило підвищити ефективність тестування безпеки програмних засобів корпоративної мережі. В підсумку, проведене математичне моделювання показує можливість підвищення ефективності тестування безпеки програмних засобів корпоративної мережі до 5%, а за допомогою методу виділення алгоритму з двійкового коду вдалося підвищити рівень безпеки ПЗ до 3 %.

3.12.2020 Гавронський В.Є.

ANNOTATION

The theme of kwalification work: The method of penetration testing as a means of ensuring the security of the corporate network

Author of the work: Gavronskiy Vitaliy Yevhenovych

Supervisor: Ph.D., Assoc. Mulyar Ihor Volodymyrovych

Explanatory note: 72 pages, 32 figures, 6 tables, 3 pages, 64 sources.

List of keywords: security testing of corporate network software, information security, GERT-network.

The purpose of the thesis is to establish an acceptable level of software security in the corporate network by developing methods for testing it.

This thesis is devoted to the development of a method of penetration testing to identify vulnerabilities, which will further ensure the security of the corporate network. To do this, a mathematical model of cyberattack was developed, which allows to investigate the main stages of cyberattack and further provide practical advice to protect the network from such attacks, improved mathematical model of network resource management system, which increased the effectiveness of security testing of enterprise network software. As a result, the mathematical modeling shows the possibility of increasing the efficiency of software security testing of the corporate network to 5%, and using the method of extracting the algorithm from the binary code managed to increase the level of software security to 3%.

3.12.2020 Gavronskiy V.E.

ЗМІСТ

ВСТУП

Сьогодні усі розуміють необхідність тестування безпеки даних корпоративних мереж. Особливо це відноситься до ПЗ комп'ютерних мереж критичного застосування. Для них тестування безпеки має проводитись на всіх етапах розробки та використання. Проте відсутність затверджених методів тестування безпеки ПЗ та відповідних методик проведення тестів на проникнення зменшує ефективність тестування безпеки ПЗ і збільшує кількість успішно проведених кібератак. Наприклад, за відомостями ЗМІ, більше 80 % компаній мають негативний досвід кібервтручання у власні ресурси, що призвів до величезних економічних, фінансових та іміджевих втрат.

Проте наявне ПЗ піддається перетворенням, що перешкоджають його аналізу. Це може бути стиснення, шифрування даних, самомодифікація тощо. Тому постає протиріччя між вимогами до безпеки ПЗ та можливостями існуючих методів тестування цього показника якості.

Зрозуміло, що покращення таких моделей та методів тестування безпеки програмного забезпечення для захисту інформації є надзвичайно важливим завданням, яким займаються багато вчених [8-15]. У вказаних працях розглядаються проблеми тестування безпеки даних. Проте, дані дослідження носять рекомендаційний характер. Окрім того, сьогодні існує необхідність удосконалення методів динамічного аналізу та виділення алгоритму з двійкового коду для аналізу безпеки даних у корпоративних мережах.

Таким чином, розроблення методу тестування безпеки програмного забезпечення для захисту інформації в корпоративній мережі є актуальним науковим завданням.

Мета й завдання дослідження.

Мета магістерської роботи полягає у встановленні прийнятного рівня безпеки програмного забезпечення в корпоративній мережі шляхом розробки методів її тестування.

Для досягнення поставленої мети було вирішено такі наукові завдання:

- проаналізовано необхідні вимоги до безпеки даних корпоративної мережі, існуючі методики тестування безпеки і чинники, що на неї впливають, встановлено, як реалізується математичне моделювання тестування безпеки, вибрано напрямок дослідження;

- розроблено математичну модель кібератаки на корпоративну мережу;

- удосконалено метод виділення алгоритму з двійкового коду для аналізування безпеки програмного забезпечення корпоративної мережі;

- розроблено модель виділення множини векторів-ознак із загальними ознаками.

Об'єкт дослідження – процес тестування безпеки програмного забезпечення корпоративної мережі.

Предмет дослідження - моделі та методи тестування безпеки програмного забезпечення для захисту інформації в корпоративній мережі.

Наукова новизна одержаних результатів полягає в тому, що в магістерській роботі:

1. Розроблено математичну модель кібератаки на корпоративну мережу, яка дозволяє дослідити основні етапи кібератаки та в подальшому надати практичні поради для захисту мережі від подібних атак.

2. Удосконалено математичну модель діагностики системи управління ресурсами мережі, що дозволило підвищити ефективність тестування безпеки програмних засобів корпоративної мережі.

Практичне значення одержаних результатів полягає в адаптації процесу тестування безпеки ПЗ для захисту інформації, а також у можливості застосування запропонованого методу під час реалізації гнучкої методики розроблення ПЗ.

Практична реалізація. Проведене математичне моделювання показує можливість підвищення ефективності тестування безпеки програмних засобів корпоративної мережі до 5%. За допомогою методу виділення алгоритму з двійкового коду вдалося підвищити рівень безпеки ПЗ до 3 %.

Публікації. За матеріалами магістерської роботи опубліковано 1 статтю у фахових наукових виданнях України та 2 тези доповідей на наукових конференціях.

1 АНАЛІЗ МЕТОДОЛОГІЙ ТЕСТУВАННЯ БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОРПОРАТИВНОЇ МЕРЕЖІ

1.1 Існуючі методології для тестування інформаційної безпеки

Проникнення комп’ютерів у всі сфери життя людини з одного боку покращує його, а з іншого – створює нові загрози, як для суспільства так і держави в цілому.

Основними видами загроз для комп’ютерних систем на даний час є загрози порушення доступності інформаційних ресурсів, що зберігаються в даних системах, та порушення конфіденційності і цілісності інформації [1]. Згідно дослідження компанії Cisco, кожна друга атака на комп’ютерні мережі завдали організаціям збитків більше ніж на 500 000 доларів США. Причому основною причиною даної ситуації в цій сфері є недостатня кількість спеціалістів в сфері інформаційної безпеки, що призводить до неможливості впровадження нових заходів з захисту та проводити розслідування інцидентів інформаційної безпеки [2].

Проаналізувавши повідомлення в засобах масової інформації, повідомлення команди на реагування на комп’ютерні надзвичайні події України (CERT-UA) за декілька останніх років варто констатувати, що організовані групи зловмисників здійснюють втручання в роботу комп’ютерних систем органів державної влади, органів місцевого самоврядування, установ, підприємств, організацій та об’єктів критичної інфраструктури України. Це в свою чергу призводить до блокування роботи установ, матеріальних та репутаційних збитків. В нашій країні найбільш резонансними стали атаки BlackEnergy на інформаційно-телекомунікаційні системи Міністерства фінансів, Державної казначейської служби, та атаки з застосуванням вірусу PetyaA [2]. Варто визнати, що на даний час в Україні законодавство в сфері захисту інформації застаріло і не відповідає вимогам сьогодення. Це в свою чергу призводить до того, що рівень захисту комп’ютерних систем не відповідає тим загрозам, які існують.

Для перевірки ступеня захисту інформації, проводять аудит інформаційної

безпеки, керуючись наступними документами: «IT Audit Framework 2nd Edition», ISO/IEC 27007: Guidelins for information security management systems auditing Cobit, Global Technology Audit Guide, International Professional Practices Framework for Internal Auditing Standards, Guide to the Assessment of IT Risk. Важливим чинником такого аудиту, є проведення тестування на проникнення в комп'ютерну мережу замовника.

Тест на проникнення - це віртуальна атака на мережу, з метою дослідження того, наскільки ця мережа вразлива для реального нападу. Процес тестування подібний процесу злому, який проводить зловмисник. При цьому пентестер пробує отримати доступ до інформації в комп'ютерній мережі, отримати контроль над роботою мережі та призвести до її непрацездатності. Пентестер виступає в ролі зловмисника і хоче знайти найбільші вразливості мережі, занотувати їх в звіті і передати замовнику. Під час тестування він визначає, як мережа реагує на атаку і які дані можна витягнути з неї. Потім пентестер надає замовнику протокол своїх дій та зазначає недоліки в інформаційній безпеці його мережі та дає рекомендації по усуненню виявлених вразливостей. Тестування дає можливість отримати інформацію щодо захищеності ПЗ мережі, дозволяє зрозуміти наскільки якісно побудовані процеси інформаційної безпеки, оцінити правильність налагоджень серверів і робочих станцій.

Найчастіше тести проводять для:

– оцінки захищеності нової інформаційної системи, де буде оброблятись чутлива інформація;

- оцінки захищеності системи після її модернізації;

– виконання вимог міжнародних стандартів;

– планування витрат на інформаційну безпеку.

На заході тестування на проникнення є досить поширеним і воно прописано відповідними законами, постановами, стандартами. Наприклад:

– для американської медичної галузі та компаній, які мають справу з інформацією про здоров'я людей – згідно вимог американського закону про охорону та відповідальність за інформацію отриману в результаті медичного

страхування HIPAA (Health Information Portability and Accountability Act);

– для банківської сфери – згідно вимог міжнародного стандарту безпеки даних індустрії платіжних карт PCI DSS (Payment Card Industry Data Security Standard), розробленому платіжними системами Visa і MasterCard [3];

– для міжнародних компаній – згідно вимог міжнародного стандарту ISO/IEC 27001.

В українськомі законодавстві можна знайти вимоги щодо проведення тестування на проникнення в «Положенні про організацію заходів із забезпечення інформаційної безпеки в банківській системі України», затвердженого Постановою Правління Національного банку України від 28.09.2017 №95.

Оскільки корпоративні мережі відмінні між собою, необхідно вибрати методологію проведення тестування на проникнення. Зараз найбільш пошареними методологіями проведення тестування на проникнення є:

– The Open Source Security Testing Methodology Manual (OSSTMM);

– The National Institute of Standards and Technology (NIST) Special Publication 800-115;

– OWASP Testing Guide;

– Penetration Testing Execution Standard (PTES);

– Information Systems Security Assessment Framework (ISSAF).

OSSTMM – міжнародна методологія призначена для тестування інформаційної безпеки, розроблена ISECOM (Institute for Security and Open Methodologies) [14]. OSSTMM надає керівні принципи для оцінки безпеки. Ця методологія описує три класи безпеки: COMSEC (communication security channel), PHYSSEC(physical security channel), SPECSES(spectrum security channel). Приведені класи в свою чергу діляться на п’ять каналів взаємодії з організацією, які має перевірити фахівець з питань безпеки: фізична безпека, бездротові мережі, інформаційні та телекомунікаційні мережі, людський фактор.

Відомими перевагами методології OSSTMM є: конкретний опис методики підготовки тестування, покроково описані алгоритми тестування, перераховані головні терміни та поняття інформаційної безпеки. Нажаль, OSSTMM-

методологія не має опису інструментів, якими повинен оперувати пентестер.

NIST-методологія визначає три етапи в оцінюванні інформаційної безпеки: планування, виконання та експлуатація. При цьому проводиться аналізування одержаних даних, визначення причини, яка призвела до певної вразливості, розроблення рекомендацій щодо ліквідації вразливості та написання звіту. Перевагою NIST-методології є те, що вона в загальному описує алгоритм перевірки безпеки корпоративної мережі. Це може бути сніфінг мережі, перевірка log-файлів, настройок системи, цілісності файлів, сканування вразливостей та бездротових мереж. Недоліком є те, що NIST-методологія була розроблена ще в 2008 році, тому вона він не відповідає теперішньому стану розвитку IT-галузі та методам проникнення у корпоративні мережі.

OWASP-методологія є детально описана та включає питання безпеки веб-сайтів і веб-додатків. Однак, якщо такий веб-сайт або веб-додатки організації не є критичними з точки зору бізнесу, то тестування на проникнення з застосуванням OWASP-методології не є доцільним.

PTES-методологія базується тільки на проведенні тестування на проникнення. Вона описує сім етапів тестування: опис початкових даних для тестування, підбір необхідної інформації, виявлення напрямку атаки, аналізування вразливості мережі, використання цієї вразливості для можливості обходу захисної системи, пост-експлуатація, написання звітну. Перевагою PTES-методології є покроково опрацьований алгоритм дій щодо перевірки захищеності корпоративної мережі. Недоліком є те, що не приділяється достатньої уваги питанню фізичного захисту корпоративної мережі.

За допомогою ISSAF-методології можна моделювати вимоги до внутрішньої безпеки, оскільки вона направлена на безпеку комп’ютерних мереж, систем та ПЗ.

1.2 Класифікація методів тестування та їх використання

Відомо, що кожна з цих методологій має свою специфіку, недоліки і

переваги.

При тестуванні на проникнення необхідно вибрати відповідний метод, узгодити його з замовником та врахувати особливості самої системи. З урахуванням вищевказаного можна привести таку класифікацію методів тестування на проникнення (табл. 1.1).

Таблиця 1.1 – Класифікація методів тестування на проникнення

№	Ознака тестування	Види тестування
1.	За розташуванням програмно-апаратних засобів та пентестера відносно периметру організації-замовника	Зовнішнє
		Внутрішнє
2.	За обізнаністю пентестера про цільову систему	Білий ящик
		Чорний ящик
		Сірий ящик
3.	За обізнаністю технічних працівників організації-замовника про проведення тестування	Відкрите
		Приховане
4.	За характером заходів, що проводяться	Пасивне
		Агресивне
		Обережне
		Прораховане
5.	За повнотою виконання тестування	Повне
		Обмежене
		Фокусоване
6.	За видом інструментів, що використовуються	З застосуванням програмно-апаратних засобів
		З застосуванням методів соціальної інженерії та проникнення на контрольовану територію

За розміщенням пентестера стосовно контрольованої мережі, виділяють як зовнішнє так і внутрішнє тестування.

Зовнішне тестування подібне до атаки на корпоративну мережу через Інтернет, тому воно дозволяє отримати та оцінити ризики такої атаки. Фахівцю з безпеки надають початкову інформацію. Це може бути назва організації-замовника тестування, його офіційний сайт або адреси електронних скриньок. Пентестеру необхідно отримати доступ до корпоративної мережі замовника та дослідити її.

Якщо замовник хоче перевірити, які вразливості має його мережа зі сторони недобросовісних співробітників, він надає пентестеру можливість підключитись

безпосередньо до корпоративної мережі. Таке тестування називається внутрішнім. За даними компанії Infowatch, 58,3% всіх випадків витоку інформації з комп'ютерних мереж організацій по всьому світу сталося з вини персоналу [11] В свою чергу, за даними компанії Positive Technologies, в 100% випадків при внутрішньому тестуванні корпоративних мереж вдавалось отримувати повний контроль над мережою. При чому для отримання повного доступу, висока кваліфікація зловмисника не потребувалась.

Причинами вразливостей безпеки корпоративної мережі можуть бути брак оновлення операційної системи, використання словарних паролів, недоліки протоколів які дозволяли перенаправляти дані та отримувати інформацію про конфігурацію мережі [15].

Відповідно до відомостей пентестера про цільову систему розрізняють наступні методи тестування: білий, чорний та сірий ящики. Якщо використовується метод білого ящика, замовник надає пентестеру усю необхідну інформацію про цільову систему, засоби її захисту, потрібну технічну документацію. Йому можуть надати права адміністратора для доступу до мережі. Пентестер у своїй роботі співпрацює з адміністатором корпоративної мережі. Пентестеру необхідно знайти вразливості мережі, дати оцінку можливості проникнення в неї. Метод білого ящика дозволяє отримати найбільш повну картину вразливостей об'єкта тестування та виявляти найбільшу кількість векторів атаки. При цьому методі тестування проходить швидко, але зрозуміло, що пентестер знаходиться в більш привігілійнішому стані, ніж справжній зловмисник.

Білий ящик рекомендують використовувати в корпоративних мережах, які працюють у відповідальній сфері і зупинка їх роботи призведе до значних матеріальних втрат та інших локдаунів.

При використанні метода чорного ящика пентестеру не дають жодної інформації про цільову систему, окрім назви організації. Тому пентестер знаходиться в рівних умовах зі справжнім зловмисником. Про проведення тестування повідомляють керівників служби безпеки. Пентестеру необхідно

проникнути в мережу максимально непомітно для інших користувачів. На відміну від методу білого ящика, результати тестування цим методом залежать від кваліфікації пентестера і не завжди відображають дійсну ситуацію щодо інформаційної безпеки. При цьому методі не всі додатки можуть бути проаналізовані.

При тестуванні методом сірого ящика фахівцю з тестування безпеки надається певна інформація про корпоративну мережу, яку він має перевірити. Тому цей метод є оптимальним за швидкістю і якістю, так як пентестер не втрачатиме час на пошук інформації про ір-адреси серверів, адреси електронної пошти та іншу конфіденційну інформацію. Тобто пентестер в правах прирівнюється до працівника підприємства, який може з легкістю дістати подібну інформацію. Задачею пентестера тепер є розширення своїх прав в системі, наприклад, якщо він зайшов під звичайним користувачем, то має отримати права адміністратора, щоб отримати доступ до іншої службової інформації.

Відповідно до знань працівників підприємства-замовника про тестування їх мережі розрізняють приховане та відкрите тестування. Приховане тестування доцільно застосовувати, якщо замовник хоче перевірити як первинну систему безпеки, так і систему попередження вторгнення, наприклад чи відбудеться попередження керівництва про проникнення в систему. За приховане сканування знають лише керівники підрозділів безпеки та безпосередньо адміністратори мережі. У зв’язку з цим приховане тестування дозволяє також оцінити дії і професіональні навички працівників підприємства, взаємодію між підрозділами та якість реалізації політики безпеки [23].

Якщо використовується метод білого ящика, то доцільно застосовувати відкрите тестування. Його також застосовують у закладах, де небажана зупинка мережі. Перевагою відкритого тестування є можливість внутрішнім фахівцям з безпеки ознайомитись з методами і засобами проведення тестування та з ймовірним алгоритмом дій зловмисників.

За характером заходів що проводяться розрізняють:

– пасивне тестування – при якому мережа сканується з метою виявлення її

вразливостей. При їх знаходженні, вони не експлуатуються, а лише фіксуються;

– агресивне тестування – фахівець з безпеки використовує всі можливі вразливості, наприклад це може бути переповнення буфера навіть на тих системах, призначення яких невідомо, може вивести з ладу систему безпеки за допомогою атаки «відмова в обслуговуванні». При цьому пентестера не цікавить, що можуть вийти з ладу і сусідні системи та мережі;

– обережне тестування – при якому вразливості мережі будуть задіяні лише тоді, коли не постраждає вся мережа;

– прораховане тестування – пентестер має передбачти наслідки використання знайдених вразливостей, що можуть вивести з ладу мережу, і на основі цієї інформації планує свою подальшу роботу.

Відповідно до повноти виконання тестування розрізняють:

– повне тестування, яке проводиться для систем, що тестуються вперше. Повне тестування дозволяє виявити вразливості в усіх мережах підприємства, оскільки кожна мережа обстежується окремо;

– обмежене тестування, при якому замовник задає, які саме системи або додатки мають тестуватися;

– фокусоване тестування, тобто пентестер фокусується на тестуванні якоїсь конкретної мережі, системи або додатку. Фокусоване тестування використовують після зміни конфігурації, нарощування корпоративної мережі.

Стосовно інструментів, що використовуються при тестуванні, розрізняють тестування з застосуванням програмно-апаратних засобів та з застосуванням методів соціальної інженерії і проникненням на територію підприємства.

Зрозуміло, що тестування на проникнення здійснюється через комп’ютерну мережу. Для його реалізації задіюють, як окремі програмно-апаратні засоби так і набори таких програм, які є встроєними у спеціалізовані операційні системи.

Соціальна інженерія використовує людський фактор, оскільки люди є найслабшою ланкою в системі безпеки підприємства. Зрозуміло, що це пов’язано з низькою обізнаністю з інформаційною безпекою та довірливістю співробітників. Знання законів соціальної інженерії дозволяє пентестеру змусити користувача

корпоративної мережі, наприклад, запустити програму надіслану в поштовому повідомленні, перейти за посиланням, або повідомити пароль за телефоном. Тобто прийоми соціальної інженерії дозволяють як пентестеру, так і зловмиснику одержати потрібні результати з мінімальними затратами сил на відміну від класичних методів злому. При такому способі тестування фахівцю з безпеки необхідно зафіксувати такі атаки, щоб в подальшому проінформувати про них співробітників.

Зрозуміло, що сучасні засоби захисту забезпечують високий рівень захисту мережі, які не завжди може обійти пентестер. Тому постає завдання знайти доступ до даних іншими шляхами, наприклад через фізичний доступ до елементів мережі, що тестується. Така фізична атака передбачає те, що фахівець з безпеки отримує доступ до певної робочої станції після потрапляння тими чи іншими шляхами до приміщення.

Пентестери рекомендують застосовувати цей метод тоді, коли на підприємстві є відпрацьована та надійна політика безпеки, впроваджені необхідні процедури, пройдені всі інструкції. При цьому використання методів соціальної інженерії дає зробити оцінку ефективності прийнятих мір безпеки.

Підсумовуючи вищевказане, загальну схему тестування можна зобразити у вигляді рис. 1.1.

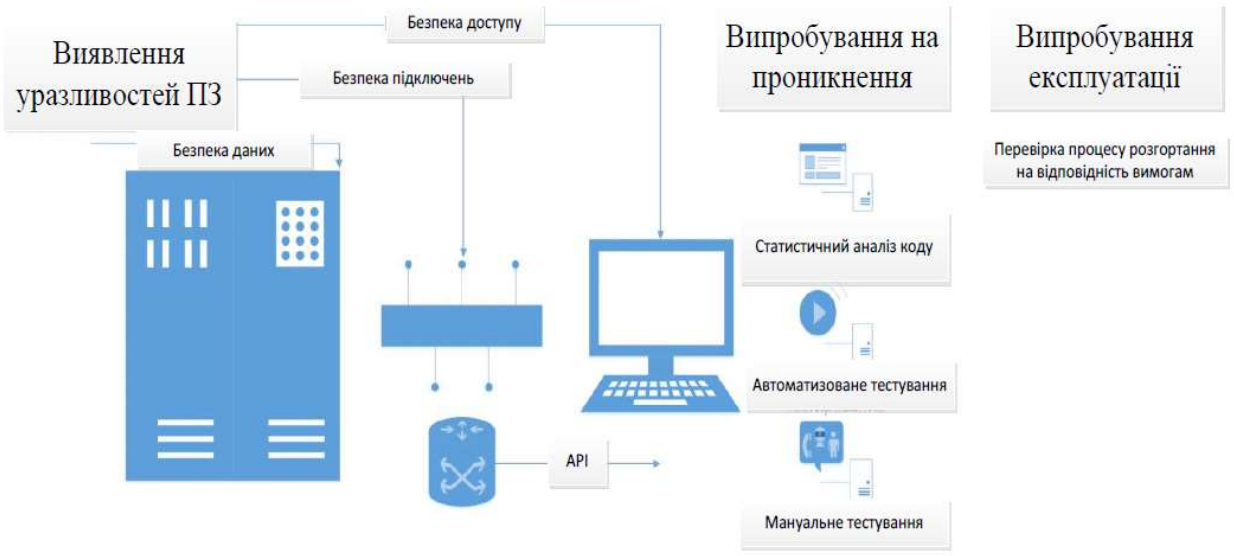


Рисунок 1.1 – Загальна схема тестування

З рис. 1.1 бачимо, що тестування безпеки ПЗ можна поділити на три основних складових: виявлення уразливостей ПЗ; випробування на проникнення; випробування експлуатації. Серед основних методів тестування безпеки ПЗ виділяють наступні:

1) аналіз архітектури та дизайну ПЗ;

2) побудова моделі загроз;

3) пошук уразливостей в початковому коді;

4) тест на проникнення;

5) тестування, що засноване на ризиках;

6) fuzzi-тестування;

7) перевірка процесу розгортання на відповідність вимогам.

Розглянемо більш детально перелічені методи тестування безпеки ПЗ.

Як зазначено у працях [48, 62], перевірка архітектури та дизайну ПЗ дозволяє зробити його аналіз на відповідність до вимог безпеки. До вказаного аналізу включають питання щодо розгортання, інфраструктури, загальної архітектури додатків, їх дизайну.

При цьому розгляд архітектури ПЗ повинен відбуватися відповідно до інфраструктури й сценаріїв розгортання, розгляд дизайну ПЗ має відбуватися згідно з усіма категоріями вразливостей, визначених на попередніх фазах роботи над проєктом, поетапний компонентний аналіз ПЗ передбачає розгляд механізмів забезпечення безпеки в кожному з ключових компонентів системи.

На рис. 1.2 показані три основні аспекти перевірки архітектури та дизайну ПЗ.

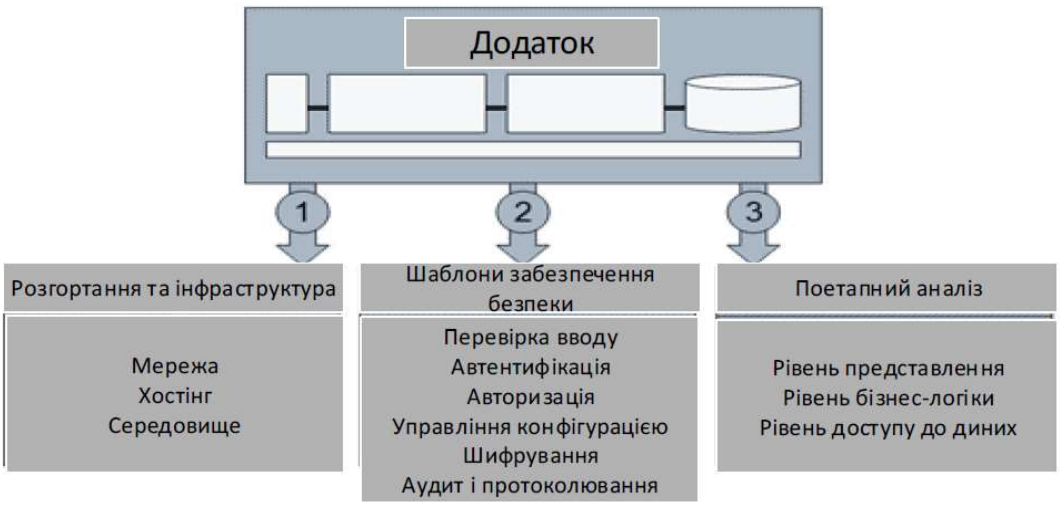


Рисунок 1.2 – Основні аспекти перевірки архітектури та дизайну ПЗ

Як видно з цього рисунку, метод аналізу архітектури та дизайну ПЗ охоплює основні питання розгортання та застосування додатків у складі корпоративних мереж, але не дає оцінити більш детальні речі, пов’язані з методикою та алгоритмами розробки програмного забезпечення.

Метод побудови моделей ризиків також відноситься до методів передпроектної розробки. Як зазначено в [17, 49], застосування моделей загроз допомагає описати вимоги щодо забезпечення безпеки, можливі загрози, вразливості та способи їх запобігання.

Таким чином моделювання загроз складається з п'яти основних кроків (рис. 1.3).

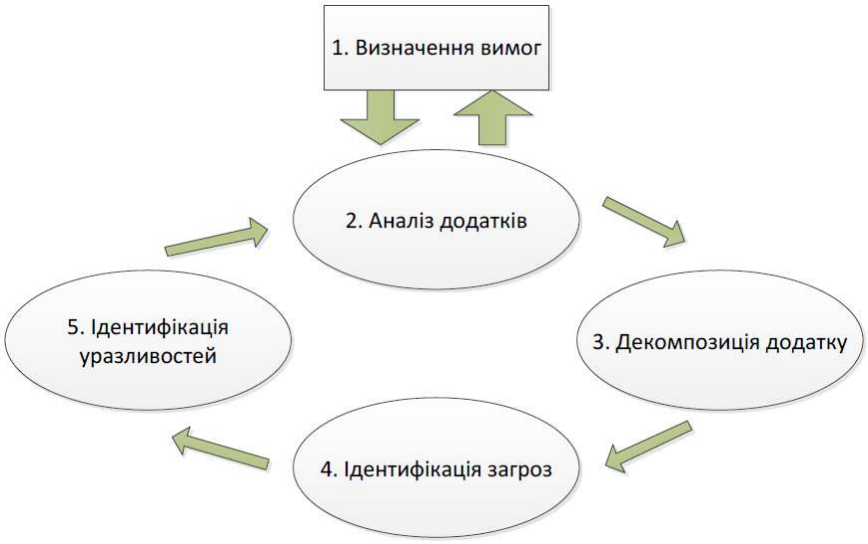


Рисунок 1.3 – Процес моделювання загроз

П'ять основних кроків, які виконуються при моделюванні загроз, наведені в додатку А.

1.3 Моделювання процесу передачі інформації каналом зв`язку

Однією з головних вимог, що ставиться до мереж, є інформаційна захищеність. Криптографічні методи, що базуються на математичних алгоритмах, використовуються для шифрування даних з подальшою передачею їх відкритими каналами зв’язку. Додатковим ступенем захисту є приховання самого факту передачі інформації, наприклад, за допомогою методів цифрової стеганографії [1]. Іншим підходом до вирішення цієї задачі є використання в якості носіїв інформації хаотичних сигналів, які характеризуються широким неперервним спектром та високою інформаційною ємністю [2-5]. Пристрої, побудовані на основі відносно нескладних математичних моделей, здатні генерувати неперіодичні електромагнітні коливання складної форми та дозволяють керувати хаотичними режимами за рахунок малих змін параметрів системи. Серед методів введення інформації в хаотичні сигнали, окрім модуляції параметрів нелінійної системи генератора, в науково-технічній літературі запропоновано ряд таких підходів, як хаотичне маскування (англ. – chaotic masking), перемикання хаотичних режимів (англ. – chaos shift keying), нелінійне підмішування (англ. – nonlinear mixing) тощо [2, 6]. Таким чином, каналом зв’язку передається шумоподібний хаотичний сигнал, когерентний прийом якого здійснюється за рахунок синхронізації хаотичних систем з подальшою демодуляцією хаотичних коливань. Отже, для прийому та обробки хаотичних сигналів на приймальній стороні повинні бути повністю або частково відтворені електричні кола генератора передавача та власне динамічний режим його роботи, що можна розглядати як додатковий ступінь захисту при передачі інформації.

Розглянемо в якості математичної моделі генератора хаотичних коливань динамічну систему Лоренца, яка складається з трьох звичайних диференціальних рівнянь першого порядку:

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(r - z) - y \\ \dot{z} = xy - bz \end{cases}; \tag{1.1}$$

де σ, r, b – дійсні додатні параметри системи.

Система Лоренца є нелінійною динамічною системою, яка при певних значеннях параметрів σ, r, b має нетривіальні розв'язки складної форми та високу чутливість до початкових умов [7, 8].

Так-як праві частини рівнянь системи (1.1) не містять вільних членів, то система є однорідною. В результаті заміні $x \to -x$, $y \to -y$ не змінюється вигляд рівнянь системи (1.1), що є свідченням симетричності системи Лоренца.

Для системи (1.1) дивергенція фазового потоку:

$$div(\dot{x}, \dot{y}, \dot{z}) = -\sigma - 1 - b < 0, \tag{1.2}$$

тоді, згідно теореми Ліувіля, фазовий потік стискає деякий об'єм фазового простору $V(t)$ згідно наступного співвідношення:

$$V(t) = V(0)e^{-(\sigma+b+1)t}, \tag{1.3}$$

отже, системи Лоренца є дисипативною.

Система Лоренца має нульову точку рівноваги $M_0 = (0;0;0)$ при довільних значеннях параметрів, а при $r > 1$ ще дві відмінні від нуля точки рівноваги:

$$M_{1,2} = \left(\pm\sqrt{b(r-1)}, \pm\sqrt{b(r-1)}, r-1\right) \tag{1.4}$$

У випадку, якщо $0 < r < 1$, єдиною точкою рівноваги, що притягує всі траєкторії у фазовому просторі, є точка M_0. При досягненні параметра значення $r = 1$ відбувається вилкоподібна біфуркація, що супроводжується (при $r > 1$)

втратою стійкості точки M_0 та появою пари стійких положень рівноваги $M_{1,2}$. Точки $M_{1,2}$ є стійкими вузлами при $1 < r < 1{,}345$ та стійкими фокусами при $1{,}345 < r < 24{,}737$. Розмах коливань у фазовому просторі відносно положень рівноваги збільшується із зростанням параметра r. Досягнувши значення $r \approx 13{,}927$ спостерігається перестроювання атрактора у фазовому просторі: при нульових початкових умовах, здійснивши оберт навколо однієї з точок рівноваги, траєкторія повернеться у початок координат. Далі, зі зростанням параметра r, в залежності від напрямку, траєкторія приходить в одну з точок $M_{1,2}$, гомоклінічні траєкторії переходять у граничні цикли, а розмах коливань зменшується. Досягнувши значення $r \approx 24{,}06$ відбувається наступне перестроювання атрактора: разом із стійкими точками $M_{1,2}$ у фазовому просторі виникає складна притягаюча структура, яка відповідає хаотичному режиму системи – «дивному атрактору» Лоренца. Точки $M_{1,2}$ втрачають стійкість після досягнення значення $r = r_k$. Для значень параметрів системи $\sigma = 10$ та $b = 8/3$, значення $r_k \approx 24{,}74$.

В загальному випадку значення параметра r_k при заданих σ та b визначається співвідношенням:

$$r_k = \frac{\sigma(\sigma + b + 3)}{\sigma - b - 1} \tag{1.5}$$

На рис. 1.4 показані характерні траєкторії системи (1.1) при різних значеннях параметра r для двох наборів початкових умов: I_1(0,1; 1; 1) та I_2(–0,1; –1; 1).

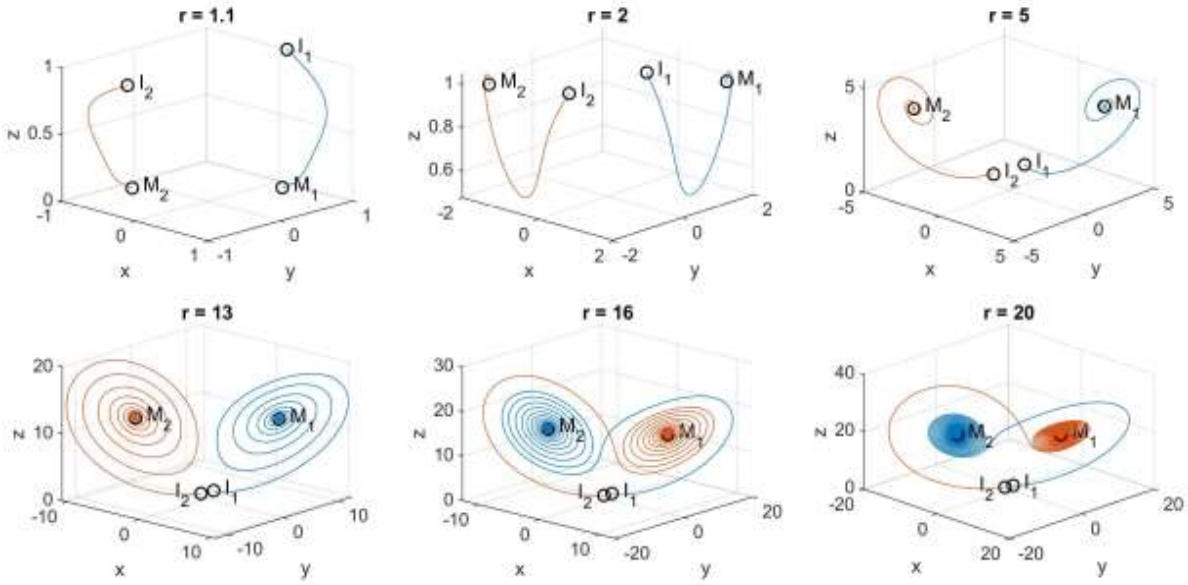
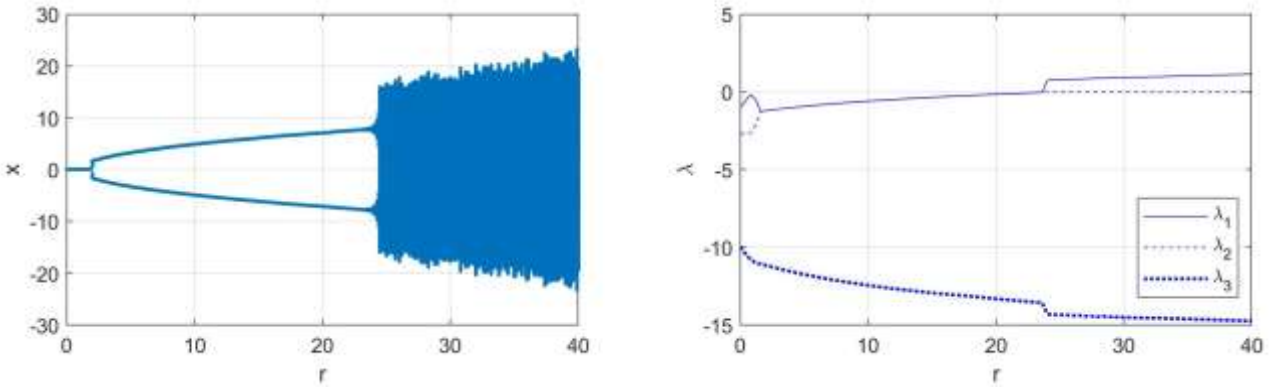


Рисунок 1.4 - Фазові траєкторії системи Лоренца для різних значеннях параметра *r*

Зміну динамічного режиму системи Лоренца за координатою x при зміні параметра $r \in [0{,}1;\ 40]$ ілюструє біфуркаційна діаграма, зображена на рис. 1.5, а.



а) б)

Рисунок 1.5 - Зміна динамічного режиму та ступеня хаотичності системи Лоренца в залежності від параметра *r*:
біфуркаційна діаграма – а), спектр показників Ляпунова – б)

Для кількісної оцінки хаотичності системи були розраховані показники Ляпунова [9, 10] λ_k, $k = 1..3$, для різних значеннях параметра $r \in [0{,}1;\ 40]$. Спектр

показників Ляпунова для вказаного діапазону значень параметра r зображено на рис. 1.5, б.

При значеннях параметрів $\sigma = 10$, $b = 8/3$, починаючи із значення $r \approx 24,74$, система Лоренца генерує хаотичні коливання, про що свідчить форма біфуркаційної діаграми [11] (рис. 1.5, а) та додатній знак старшого показника Ляпунова (рис. 1.5, б).

В якості набору параметрів, при яких системи (1.1) демонструє хаотичну поведінку було прийнято: $\sigma = 10$, $r = 35$, $b = 8/3$. Фазовий портрет атрактора, форму широкосмугового сигналу та його амплітудний спектр показано на рис. 1.6.

У хаотичному режимі система Лоренца генерує широкосмугові сигнали складної форми із неперервним спектром та високими кореляційними та ортогональними властивостями [8]. Висока інформаційна ємність та сильна залежність від початкових умов обумовлює використання сигналів такого типу в системах прихованої передачі інформації із шифруванням даних.

Головною проблемою, яку необхідно вирішити для ефективного використання хаотичних сигналів для передачі інформації в телекомунікаційних системах є задача синхронізація хаотичних генераторів на передавальній та приймальній сторонах – «ведучої» та «веденої» систем відповідно (див. рис. 1.5, б).

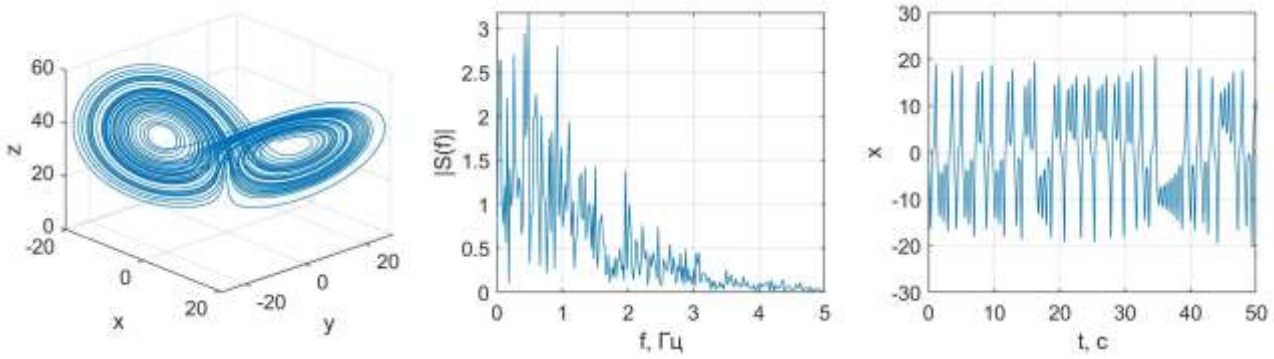
Розглянемо пару зв’язаних систем Лоренца з однаковими значеннями параметрів σ, r та b, що описується наступною системою рівнянь:

$$\begin{cases} \dot{x}_1 = \sigma(y_1 - x_1) \\ \dot{y}_1 = x_1(r - z_1) - y_1 \\ \dot{z}_1 = x_1 y_1 - b z_1 \\ \dot{x}_2 = \sigma(y_2 - x_2) \\ \dot{y}_2 = x_2(r - z_2) - y_2 \\ \dot{z}_2 = x_2 y_2 - b z_2 \end{cases} \qquad (1.6)$$

Синхронізація цих двох систем можлива якщо існують фазові траєкторії $\mathbf{U}_1$ та $\mathbf{U}_2$ такі, що при $t \to \infty$ відстань між траєкторіями $\delta \to 0$, тобто

$$\lim_{t\to\infty} \mathbf{U}_2 = \mathbf{U}_1 \qquad (1.7)$$

Крім того, рух системи (1.6) цією траєкторією повинен бути стійким по відношенню до завад. Необхідною умовою цього є від'ємне значення старшого показника Ляпунова «веденої» системи [7].



а) б) в)

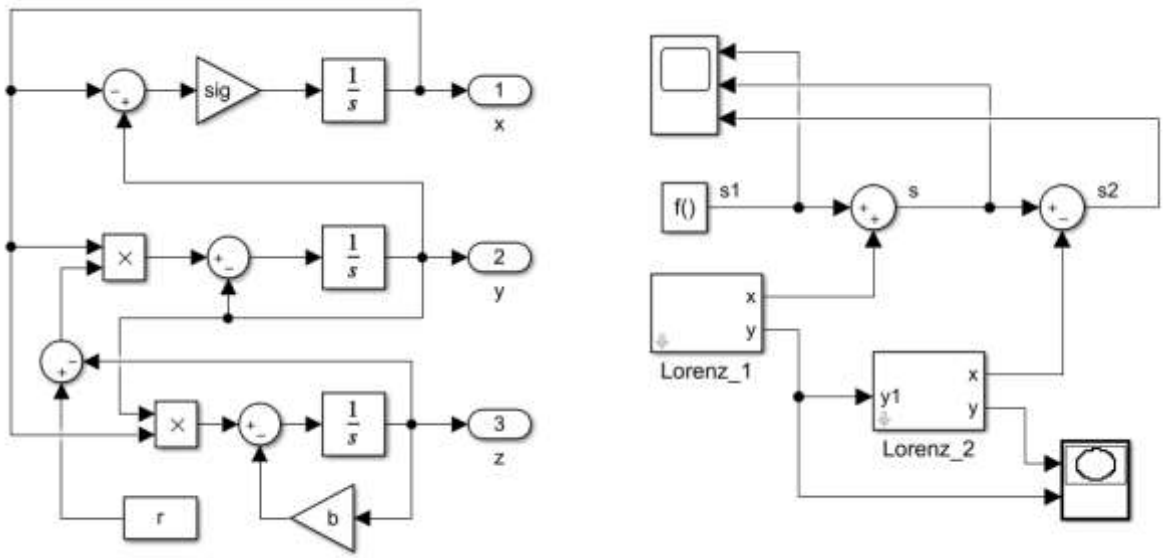
Рисунок 1.6 - Хаотичний режим системи Лоренца для набору параметрів $\sigma = 10$, $r = 35$, $b = 8/3$:
атрактор у фазовому просторі – а), амплітудний спектр сигналу координати x – б), часовий графік сигналу x

Моделювання процесу синхронізації зв'язаних систем Лоренца було виконано засобами MATLAB/Simulink.

Комп'ютерна Simulink-модель, що складається з трьох інтеграторів, які призначені для чисельного розв'язку динамічної системи Лоренца (1), зображена на рис. 1.7, а. Вихідними сигналами системи є часові значення координат x, y, z.

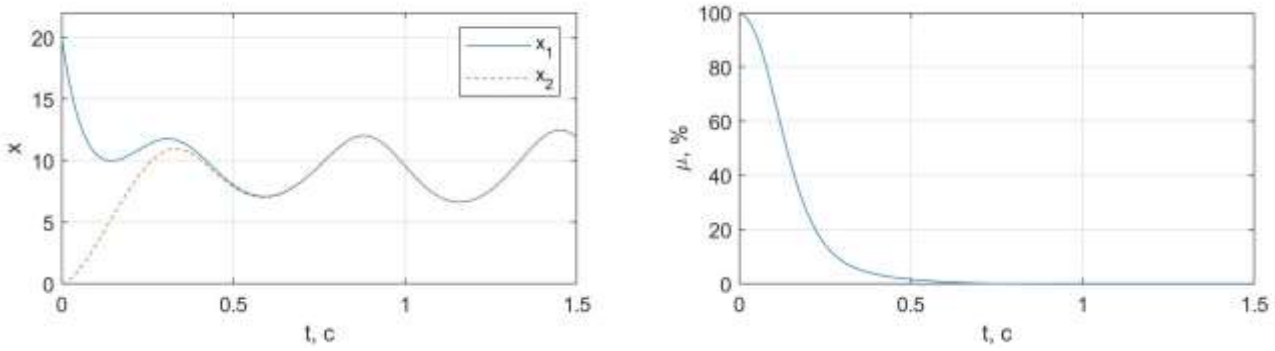
Система синхронізації двох зв'язаних систем Лоренца, *Lorenz_1* та *Lorenz_2*, показана на рис. 1.7, б. Для реалізації синхронного відгуку системи *Lorenz_2* використовується сигнал y «ведучої» системи *Lorenz_1*, а сигнал x слугує носієм інформаційного сигналу s_1.

Перехідний процес синхронізації динамічних систем Лоренца для сигналу x показано на рис. 1.8, а. На рис. 1.8, б зображено графік відносної похибки синхронізації μ.



а) б)

Рисунок 1.6 - Simulink-модель динамічної системи Лоренца – а) та системи синхронізації двох зв’язаних систем Лоренца з хаотичним маскуванням інформаційного сигналу – б)



а) б)

Рисунок 1.7 - Процес синхронізації зв’язаних динамічних систем Лоренца: часові діаграми вихідних сигналів x_1 та x_2 зв’язаних систем – а), відносна похибка синхронізації – б)

Представимо довільний вузькосмуговий сигнал у вигляді:

$$s\left[u(t),t\right]=S\left[u(t),t\right]\cdot\sin\left(\omega_0 t+\Psi\left[u(t),t\right]\right), \tag{1.8}$$

де $S\left[u(t),t\right]$ – амплітуда сигналу,

$\Psi\left[u(t),t\right]=\Phi\left[u(t),t\right]+\varphi_0$ – повна фаза сигналу.

Нехай $u(t)$ – повільно зростаючою функцією часу, тоді при диференціюванні вважатимемо $u(t)=u=const$.

Продиференціювавши вираз (1.8) двічі по часу, ввівши заміни $\sin\Psi=s/S$ та $\cos\Psi=\left(\dot{s}-\dot{S}\,s/S\right)\big/S\dot{\Psi}$, отримаємо лінійне диференціальне рівняння зі змінними коефіцієнтами:

$$\ddot{s}-\left[\frac{\ddot{\Psi}}{\dot{\Psi}}+\frac{2\dot{S}}{S}\right]\dot{s}+\left[\dot{\Psi}^2+\frac{1}{S}\left(\frac{2\dot{S}^2}{S}+\frac{\dot{S}\ddot{\Psi}}{\dot{\Psi}}-\ddot{S}\right)\right]s=0 \tag{1.9}$$

Форма сигналу $s(t)$, який є розв'язкам рівняння (1.9), залежить від типу модуляції, яка в свою чергу, в рамках описуваної моделі, задається законом зміни амплітуди $S\left[u(t),t\right]$ та повної фази $\Psi\left[u(t),t\right]$.

Наприклад, для сигналу з частотною модуляцією (ЧМ):

$$S\left[u(t),t\right]=S_0=const, \tag{1.10}$$

$$\Psi\left[u(t),t\right]=\omega_0 t+m_{чм}\int_0^t u(t)\,dt+\varphi_0, \tag{1.11}$$

де ω_0 – частота несучого коливання;

$m_{чм}$ – індекс модуляції;

φ_0 – початкова фаза.

Комп'ютерна Simulink-модель генератора вузькосмугових сигналів, побудована згідно рівняння (1.9) представлена на рис. 1.8. Вхідними сигналами для моделі є амплітуда S та повна фаза Ψ.

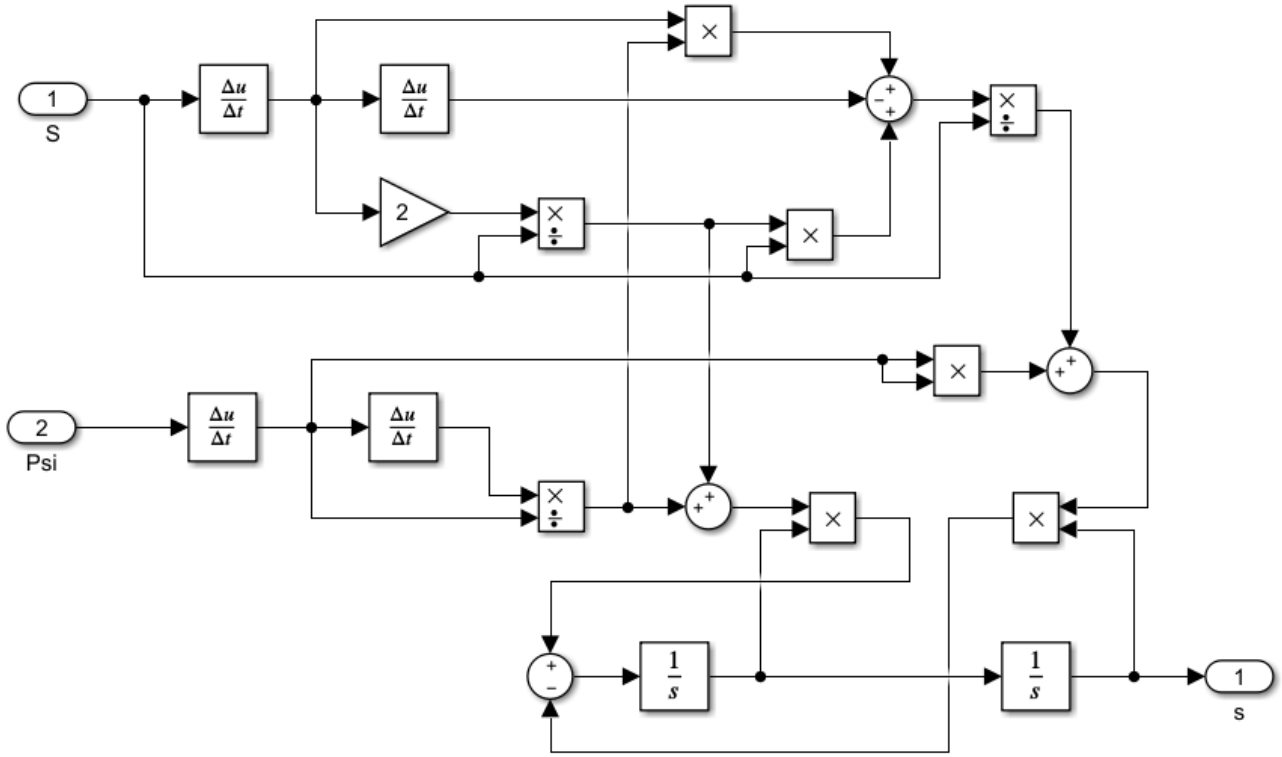
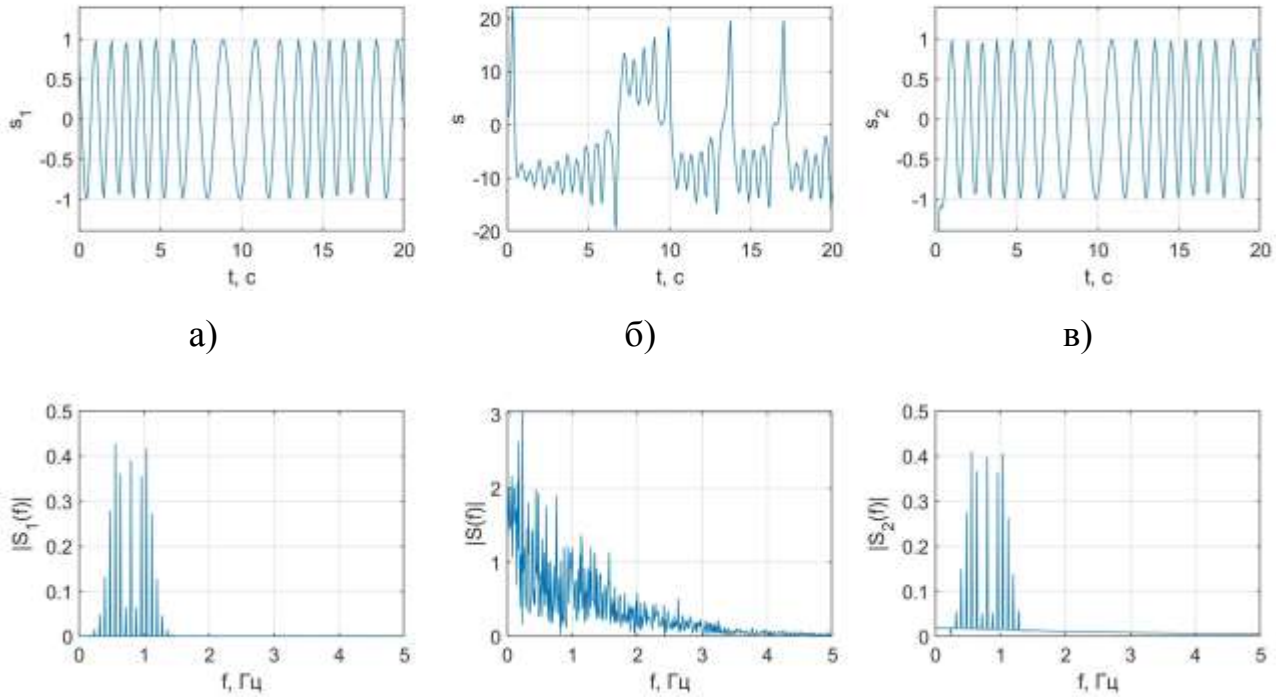


Рисунок 1.8 - Simulink-модель генератора вузькосмугових сигналів із заданою формою амплітуди та фази

Часові залежності та амплітудні спектри вихідних сигналів систем передачі та прийому, а також сигналу в каналі зв’язку, показано на рис. 1.9.

Тестовий ЧМ-сигнал s_1, отриманий на виході моделі, показаної на рис. 6, адитивно підмішується до вихідного хаотичного сигналу x_1 системи *Lorenz_1* (рис. 1.6, б) та разом із сигналом синхронізації y передається каналом зв’язку (рис. 1.9, б). На приймальній стороні тестовий сигнал виділяється шляхом віднімання від прийнятого хаотичного сигналу s сигналу x_2, згенерованого системою *Lorenz_2* (рис. 4, б), що синхронізується сигналом y. По завершенню перехідного процесу, після встановлення режиму синхронізації, виділений сигнал s_2 (рис. 1.9, е) співпадає з оригінальним сигналом повідомлення s_1 (рис. 1.9, а). Параметри систем *Lorenz_1* та *Lorenz_2* вважаються ідентичними, а канал зв’язку – ідеальним.



а) б) в)

г) д) е)

Рисунок 1.9 - Хаотичне маскування тестового вузькосмугового ЧМ-сигналу: сигнал на вході системи передачі та його амплітудний спектр – а), г); сигнал, переданий каналом зв’язку та його спектр – б), д); сигнал на виході приймальної системи та його спектр – в), е)

Таким чином, перспектива використання пристроїв із хаотичної динамікою обумовлена рядом факторів, серед яких висока інформаційна ємність, широкий спектр частот та конфіденційність передачі повідомлень. Можливість реалізації на базі одного пристрою великої кількості хаотичних режимів в перспективі дає можливість побудови багатоканальних систем передачі інформації. Сильна залежність від початкових умов та нестійкість фазових траєкторій дозволяє за рахунок малих впливів керувати динамікою хаотичних генераторів та здійснювати модуляцію з великою швидкістю.

Не дивлячись на простоту реалізації, метод хаотичного маскування має ряд суттєвих недоліків. Так, при наявності завад в каналі зв’язку інформаційний сигнал, потужність якого апріорі є нижчою порівняно із несучим хаотичним сигналом, стає співрозмірним із шумами каналу. Збільшення рівня

інформаційного сигналу призводить до втрати конфіденційності, оскільки можливим стає несанкціонований перехват інформаційного повідомлення шляхом відфільтровування хаотичної складової. Таким чином, при проектуванні системи передачі, основаній на хаотичному маскуванні, необхідно визначити оптимальне співвідношення сигнал/хаос виходячи із оцінки можливого рівня шумів в каналі та потрібної якості передачі.

1.4 Постановка завдання дослідження

Проведені дослідження показали, що існує широкий спектр варіантів розроблення та використання тестування безпеки ПЗ корпоративних мереж. Ці варіанти можуть відрізнятися технологіями впровадження, вартісними та іншими тактико-технічними показниками, характеристиками їх окремих елементів тощо.

Основним завданням розроблення методу тестування безпеки ПЗ корпоративної мережі є розробка, удосконалення та вибір моделей і методів, що забезпечують максимальні показники безпеки ПЗ.

Під час розробки математичних моделей тесту на проникнення необхідно виділити такі завдання:

- розробка та дослідження стохастичної моделі початкової генерації коду кібератаки до ресурсів корпоративної мережі та моделі процесів активного аналізу системи управління ресурсом і впровадження в корпоративну мережу;
- постановка та вирішення завдання математичного моделювання;
- оцінка ефективності розроблених математичних моделей.

При розробленні методу виділення алгоритму з двійкового коду для аналізу безпеки програмного забезпечення виникає необхідність в оцінюванні та виборі методологічних підходів до розробки та побудови організаційних структур, розробленні загальної схеми виділення алгоритму з двійкового коду для аналізу безпеки програмного забезпечення, розробці моделі виділення множини векторів-ознак із загальними ознаками, синтезі

пов'язаних векторів-ознак, розробленні моделі виявлення наявності динамічного коду в частині вектора-ознаки.

1.5 Висновки до розділу

У розділі проведено аналіз основних вимог щодо безпеки ПЗ, зазначено на пріоритетність вимог безпеки програмного забезпечення та обов'язковість дотримання цих вимог на всіх етапах життєвого циклу ПЗ.

Проведено дослідження основних методологій тестування безпеки ПЗ та факторів, що впливають на цей процес. Визначено актуальність завдання та зафіксовано, що пріоритетним напрямком дослідження є розроблення методів і засобів тестування безпеки ПЗ, що мають на меті оцінювання веб-служб, тестування проникнення до веб-додатків.

Проаналізовані основні напрямки математичного моделювання. Визначено, що перспективним напрямком є моделювання за допомогою стохастичних мереж. Указано на доцільність удосконалення наявних методологій тестування безпеки ПЗ шляхом розробки нових методів та засобів з урахуванням підвищених вимог щодо якості ПЗ. Сформульовано завдання магістерського дослідження.

2 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

2.1 Постановка задачі моделювання

Згідно з вимогами системного підходу до захисту інформації, набір взаємозв'язаних елементів, робота яких спрямована на забезпечення безпеки функціонування, утворює систему захисту інформації. Такими елементами є математичні, технічні та програмні рішення, а також людські ресурси.

Проблеми при побудові системи захисту інформації виникають не лише внаслідок складності її побудови, а й через зовнішні чинники та нетипові дії зловмисників.

Оскільки усе більший інтерес у хакерів викликають електронні інформаційні ресурси, то спостерігаємо розширення діапазону поведінкових портретів хакерів у межах кібератак.

Для реалізації кібератаки хакер має змоделювати цю подію, яка має призвести до вдалого для нього результату. Проведені дослідження показали, що сьогодні існує низка математичних моделей [58-72] у цій галузі. Однак відомі моделі кібератак не використовують такий компонент, як «дії зловмисника». Це призводить до того, що ефективність систем захисту інформації від несанкціонованого доступу знижується, і вони не завжди можуть виявити такий різновид кібератак. Тому розроблення математичної моделі кібератак є актуальним науковим завданням.

Кібератака містить такі етапи, як генерація коду кібератаки, перехоплення конфіденційної інформації, аналізування системи управління та безпосередньо вхід в корпоративну мережу.

В результаті проведеного аналітичного огляду математичних моделей кібератак приходимо до висновку, що найбільш доцільно вибрати стохастичні моделі, які представляються методом графічного відображення (Graphical Evaluation and Review Technique, тобто GERT-мережі). Тому в даному

розділі розглянемо питання побудови такої математичної моделі для початкової генерації коду кібератаки та системи управління ресурсом корпоративної мережі.

2.2 Математична модель кібератаки

Початковим етапом несанкціонованого злочинного впливу є кібератака до ресурсів корпоративної мережі. Тому необхідно провести аналіз та розробити відповідну GERT-модель. На рисунку 2.1. приведена структурна схема кібератаки

Стохастична мережа, що реалізує наведений алгоритм представлена на рисунку 2.2. У цій мережі переходи зі стану в стан характеризують виконання операцій алгоритму кібератаки та описуються випадковою величиною, що має відомий закон розподілу.

Так перехід (1,2) описує вибір обладнання в мережі для злому. Перехід (2,3) описує вибір методу атаки на комп’ютер жертви під ОС Windows. Перехід (2,4) описує вибір методу атаки на комп’ютер жертви під ОС Linux.

Якщо хакер не зміг вибрати метод атаки протягом визначеного часу або характеристики знайденого злочинного ПЗ не відповідають умовам і меті кібератаки, то здійснюються переходи (2,1) і (3,1) відповідно.

Пошук програмного забезпечення мережі Internet, скачування та установка його для ПК під ОС Windows представлений переходом (3,5). Аналогічну дію для ПК під ОС Linux відображає перехід (4,7). Якщо в мережі Internet відсутнє відповідне програмне забезпечення, то виконується кодування та налагодження ПЗ під ОС Windows (перехід 3,6) і Linux (перехід 4,6).

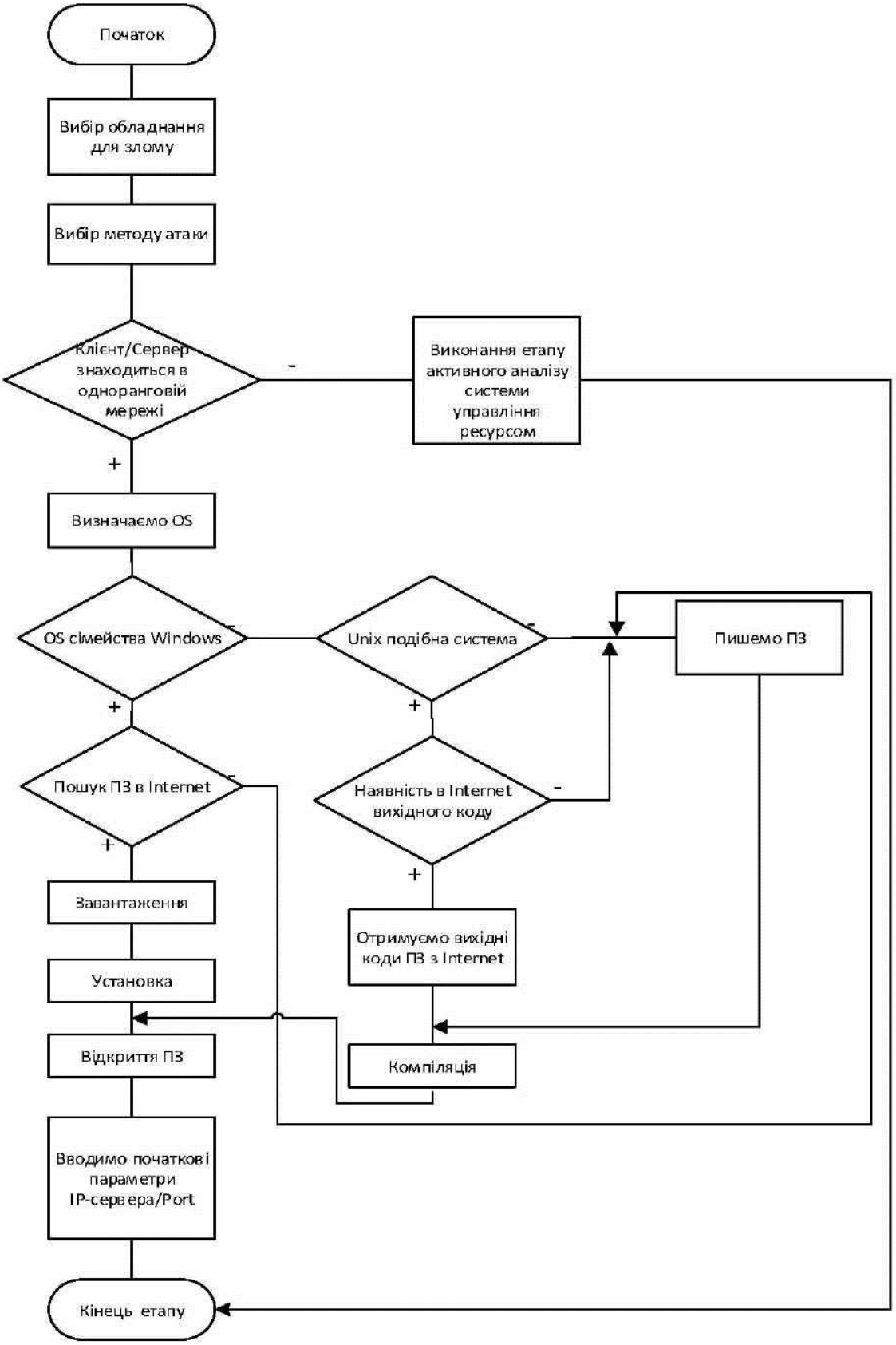


Рисунок 2.1 - Алгоритм проведення кібератаки

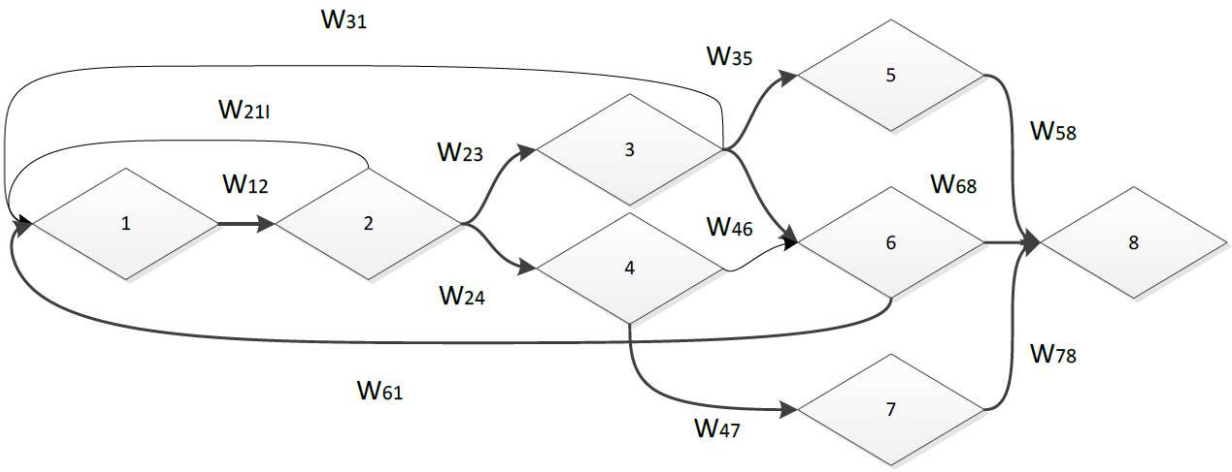


Рисунок 2.2 – Стохастична мережа, побудована за алгоритмом проведення кібератаки

Якщо хакеру не вдалося провести операцію кодування та налагодження свого програмного забезпечення під час атаки, здійснюється перехід (6,1). Відкривання злочинного програмного забезпечення і ввід параметрів IP-серверу комп'ютера-жертви відображають переходи (5,8), (6,8) і (7,8).

Дослідивши процеси проведення кібератаки, були сформовані характеристики гілок стохастичної моделі (таблиця 2.1).

Таблиця 2.1 – Характеристики гілок стохастичної моделі

№ з/п	Гілка	W-функція	Ймовірність	Похідна функція моментів
1.	(1,2)	W_{12}	P_1	$\lambda_1 /(\lambda_1 - s)$
2.	(2,3)	W_{23}	P_2	$\lambda_2 /(\lambda_2 - s)$
3.	(2,1)	W_{21}	P_3	$\lambda_3 /(\lambda_3 - s)$
4.	(2,4)	W_{24}	$1- P_1- P_2- P_3$	$\lambda_4 /(\lambda_4 - s)$
5.	(3,5)	W_{35}	P_4	$\lambda_2 /(\lambda_2 - s)$
6.	(3,1)	W_{31}	P_3	$\lambda_3 /(\lambda_3 - s)$
7.	(3,6)	W_{36}	$1- P_3- P_3$	$\lambda_4 /(\lambda_4 - s)$

Продовження таблиці 2.1

№ З/п	Гілка	W-функція	Ймовірність	Похідна функція моментів
8.	(4,6)	W_{46}	P_5	$\lambda_4/(\lambda_4 - s)$
9.	(4,7)	W_{47}	$1\text{-}P_5$	$\lambda_2/(\lambda_2 - s)$
10.	(5,8)	W_{58}	P_4	$\lambda_2/(\lambda_2 - s)$
11.	(6,8)	W_{68}	P_6	$\lambda_5/(\lambda_5 - s)$
12.	(6,1)	W_{61}	$1\text{-}P_6$	$\lambda_6/(\lambda_6 - s)$
13.	(7,8)	W_{78}	$1\text{-}P_5$	$\lambda_2/(\lambda_2 - s)$

Відповідно до характеристик гілок GERT-мережі еквівалентну W-функцію часу кібератаки на ресурси корпоративної мережі можна представити як:

$$W_E(s) = \frac{\left(\begin{array}{l} W_{12}W_{23}W_{35}W_{58} + W_{12}W_{23}W_{36}W_{68} + W_{12}W_{24}W_{46}W_{68} + \\ + W_{12}W_{24}W_{47}W_{78} \end{array}\right)}{1 - W_{12}W_{21} - W_{12}W_{23}W_{31} - W_{12}W_{23}W_{36}W_{61} - W_{12}W_{24}W_{46}W_{61}} =$$

$$= \frac{\left(\begin{array}{l} p_1(\lambda_1/\lambda_1 - s)\left(\begin{array}{l} p_2 p_4^2 \lambda_2^3 (\lambda_1 - s)(\lambda_4 - s)^2(\lambda_5 - s) + \\ + p_1 p_2 p_6 q_2 \lambda_1 \lambda_2 \lambda_4 \lambda_5 (\lambda_2 - s)^2(\lambda_4 - s) + \\ + p_1 p_5 p_6 q_1 \lambda_1 \lambda_4^2 \lambda_5 (\lambda_2 - s)^3 + \\ + p_1 q_1 q_3^2 \lambda_1 \lambda_2^2 \lambda_4 (\lambda_2 - s)(\lambda_5 - s) \end{array}\right) \end{array}\right)}{\left(\begin{array}{l} (\lambda_1 - s)(\lambda_2 - s)(\lambda_3 - s)(\lambda_4 - s)^2(\lambda_6 - s) - \\ - p_1 p_2 \lambda_1 \lambda_3 (\lambda_2 - s)(\lambda_4 - s)^2(\lambda_6 - s) - \\ - p_1 p_2 p_3 \lambda_1 \lambda_2 \lambda_3 (\lambda_4 - s)^2(\lambda_6 - s) - \\ - p_1 p_2 q_1 q_2 p_3 \lambda_1 \lambda_2 \lambda_4 \lambda_6 (\lambda_4 - s)(\lambda_3 - s) - \\ - p_1 p_5 q_1 q_4 p_3 \lambda_1 \lambda_4^2 \lambda_6 (\lambda_2 - s)(\lambda_3 - s) \end{array}\right)}{(\lambda_1 - s)(\lambda_2 - s)(\lambda_3 - s)(\lambda_4 - s)^2(\lambda_6 - s)}.$$

де $q_1 = 1 - p_2 - p_3$; $q_2 = 1 - p_3 - p_4$; $q_3 = 1 - p_5$; $q_4 = 1 - p_6$.

Після проведених розрахунків, можна зробити висновки, що в складних

стохастичних мережах відсутні прості методи знаходження особливих точок функції *Φ(z)*, оскільки для їх знаходження необхідно розв'язати нелінійні рівняння, які ускладнюються пропорційно нарощуванні GERT-мережі [49].

Таким чином, густина можливого розподілу ймовірності процесу кібератаки, буде визначатися

$$\phi(\mathrm{x}) = \frac{1}{2\pi i}\int_{-i\infty}^{i\infty} e^{zx}\frac{-\mathrm{y}z^6 + \mathrm{b}z^5 - tz^4 + \mathrm{u}z^3 - kz^2 + wz - h}{\left(-z^6 + jz^5 - dz^4 + gz^3 - vz^2 + rz - c\right)\times} \times \left(\lambda_1 + z\right)\left(\lambda_2 + z\right)\left(\lambda_5 + z\right), \qquad (2.2)$$

Функція *Φ(z)* може мати як прості полюси, що визначаються коренем рівняння $-z^6 + jz^5 - dz^4 + gz^3 - vz^2 + rz - c = 0$, так і може мати й полюси вищих порядків. При цьому густина розподілу часу тестування $\varphi(\mathrm{x})$ дорівнює

$$\gamma_{-1} = \frac{1}{(m-1)!}\lim_{z\to z_n}\frac{d^{m-1}\left[(z-z_n)^m e^{zx}\Phi(z)\right]}{dz^{m-1}}$$

Функція Φ(z) має полюси в точках $z_1 = -\lambda_1,\ z_2 = -\lambda_2,\ z_3 = -\lambda_5$.

Многочлен $-z^6 + jz^5 - dz^4 + gz^3 - vz^2 + rz - c$ породжує ще шість полюсів. Розв'язавши рівняння

$$-z^6 + jz^5 - dz^4 + gz^3 - vz^2 + rz - c = 0 \qquad (2.3)$$

отримаємо іще шість особливих точок z_4, z_5, z_6, z_7, z_8.

Підсумовуючи вищевикладене, можна констатувати, що була розроблена модель процесу кібератаки. Вона відрізняється від відомих тим, що враховує генерацію коду для ОС Windows і Linux та дозволяє здійснити пошук рішень в Internet. Модель може використовуватися для дослідження кібератаки.

2.3 Математична модель діагностики системи управління ресурсом корпоративної мережі

Як показує аналіз літератури, проведення заходів тестування безпеки корпоративних мереж свідчить, що тривалість етапів аналізу системи управління ресурсом корпоративної мережі може займати більше 45% від усієї тривалості тестування.

З аналізу джерел [6, 23, 58, 64] дізнаємося, що однією з особливостей процесу аналізу системи управління ресурсом є неспецифічність тестових переборів під час атаки. Це пояснюється тим, що пентестер часто виконує роль хакера-зломщика й починає працювати з різними ресурсами корпоративної мережі. Наприклад, це може бути і спроби дізнатися пароль, і аналізування системи на захищеність від DoS-атак, і введення помилок за допомогою яких можна буде проникнути в систему, і перегляд загальнодоступних даних з метою знайти пароль для входу в систему тощо. Описані методи, разом зі знанням основних уразливостей корпоративної мережі, роблять атаку успішною, а тестові приклади не завжди можуть відобразити реальну атаку.

Вихідними даними для математичного опису алгоритму аналізу діагностики системи управління ресурсом мережі є те, що тестування безпеки мережі проводять при наявності доступу до корпоративної мережі. Взаємодія пентестера з корпоративною мережею повинна бути безпечною для самої мережі те не має призводити до виходу її з ладу. Взаємодіючи з корпоративною мережею, ми отримуємо інформацію про її можливості. Необхідно врахувати, що факт подачі однакового тестового перебору, призводить до різної реакції мережі.

На рис. 2.3 представлено математичну модель діагностики системи управління ресурсом корпоративної мережі.

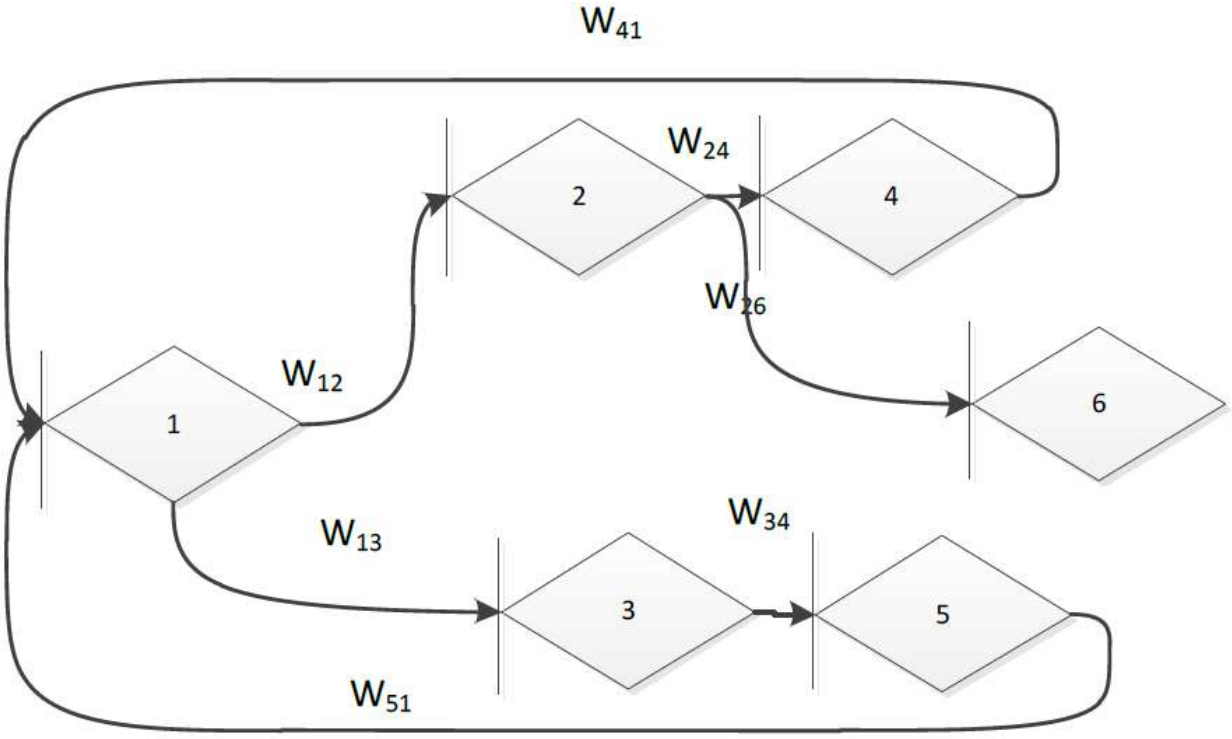


Рисунок 2.3 – Математична модель діагностики системи управління ресурсом корпоративної мережі

У приведеній моделі перший вузол характеризує початковий стан мережі перед впливом на неї алгоритму аналізу системи управління ресурсом. Другий вузол показує стан мережі, коли неможливо перехопити ресурси для її управління. Третій вузол відображає зворотну ситуацію, тобто можливість перехоплення ресурсів для управління мережею. Четвертий вузол показує на виправлення помилок. П’ятий вузол показує неповне виконання алгоритму аналізу системи управління ресурсом. Шостий вузол відображає факт повної перевірки корпоративної мережі.

Перехід (1-2) показує вплив на мережу описаного алгоритму, що призвів до позитивного результату для пентестера. Виправлення помилок відображає перехід (2-4). Перехід (4-1) забезпечує перехід в початковий стан мережі, який був до запуску тесту. Перехід (2-6) описує імплементацію шкідливого коду в корпоративну мережу. Перехід (1-3) показує вплив на мережу описаного алгоритму, що призвів до негативного результату для пентестера. Перехід (3-5) представляє оцінку повноти охоплення аналізованих впливом елементів, що були

задіяні алгоритмом управління ресурсом корпоративної мережі.

Дослідивши алгоритм управління ресурсом корпоративної мережі, були сформовані характеристики гілок стохастичної моделі (таблиця 2.2).

Таблиця 2.2 – Характеристики гілок стохастичної моделі діагностики системи управління ресурсом корпоративної мережі

№ З/п	Гілка	*W*-функція	Ймовірність	Похідна функція моментів
1.	(1,2)	W_{12}	p_1	$\lambda_1 / (\lambda_1 - s)$
2.	(2,4)	W_{24}	p_2	$\lambda_2 / (\lambda_2 - s)$
3.	(2,6)	W_{26}	$1 - p_2\ (q_1)$	$\lambda_3 / (\lambda_3 - s)$
4.	(4,1)	W_{41}	p_3	$\lambda_4 / (\lambda_4 - s)$
5.	(1,3)	W_{13}	p_4	$\lambda_1 / (\lambda_1 - s)$
6.	(3,5)	W_{35}	p_4	$\lambda_5 / (\lambda_5 - s)$
7.	(5,1)	W_{51}	$1 - p_4\ (q_2)$	$\lambda_6 / (\lambda_6 - s)$

Відповідно до характеристик гілок GERT-мережі еквівалентну W-функцію алгоритмів системи управління ресурсом корпоративної мережі можна представити як:

$$W_E(s) = \frac{W_{12}W_{26} + W_{12}W_{24}W_{41}W_{12}W_{26} + W_{13}W_{35}W_{51}W_{12}W_{26}}{1 - W_{12}W_{24}W_{41}W_{45}W_{51} - W_{13}W_{35}W_{51}} =$$

$$= \left(\frac{\begin{pmatrix} \left(p_1 q_1 \lambda_1 \lambda_3 (\lambda_1 - s)(\lambda_2 - s)(\lambda_4 - s)(\lambda_5 - s)(\lambda_6 - s)\right) + \\ + \left(p_1^2 p_2 p_3 q_1 \lambda_1^2 \lambda_2 \lambda_3 \lambda_4 (\lambda_5 - s)(\lambda_6 - s)\right) + \\ + \left(p_1 p_4^2 q_1 q_2 \lambda_1^2 \lambda_3 \lambda_5 \lambda_6 (\lambda_2 - s)(\lambda_4 - s)\right) \end{pmatrix}}{(\lambda_1 - s)^2 (\lambda_2 - s)(\lambda_3 - s)(\lambda_4 - s)(\lambda_5 - s)(\lambda_6 - s)} \right) \Bigg/ \left(1 - \left(\frac{\begin{matrix} p_1 p_2 p_3 \lambda_1 \lambda_2 \lambda_4 (\lambda_5 - s)(\lambda_6 - s) + \\ + p_4^2 q_2 \lambda_1 \lambda_5 \lambda_6 (\lambda_2 - s)(\lambda_4 - s) \end{matrix}}{(\lambda_1 - s)(\lambda_2 - s)(\lambda_4 - s)(\lambda_5 - s)(\lambda_6 - s)} \right) \right)$$

Спростивши W-функцію алгоритмів системи управління ресурсом корпоративної мережі, маємо:

$$W_E(s)=\frac{a+bs+cs^2+ds^3+gs^4-s^5}{(\lambda_1-s)(\lambda_3-s)\left(k+ms+ns^2+hs^3+rs^4-s^5\right)},$$

де $a=p_1q_1\lambda_1\lambda_3\left(\lambda_1\lambda_2\lambda_4\lambda_5\lambda_6\left(p_1p_2p_3+p_4^2q_2+1\right)\right)$

$$b=-p_1q_1\lambda_1\lambda_3\left(\begin{array}{l}\lambda_4\lambda_5\lambda_6\left(\lambda_2+\lambda_1+p_4^2q_2\right)+\lambda_1\lambda_2\lambda_5\lambda_6\left(1+p_4^2q_2\right)+\\+\lambda_1\lambda_2\lambda_4\left(\lambda_6+\lambda_5+p_1p_2p_3\left(\lambda_6+\lambda_5\right)\right)\end{array}\right)$$

$$c=p_1q_1\lambda_1\lambda_3\left(\begin{array}{l}\lambda_4\lambda_5\lambda_6+\lambda_2\lambda_5\lambda_6+\lambda_2\lambda_4\lambda_6+\lambda_1\lambda_5\lambda_6\left(1+p_4^2q_2\right)+\\+\lambda_2\lambda_4\lambda_5+\lambda_1\lambda_4\lambda_6+\lambda_1\lambda_4\lambda_5+\lambda_1\lambda_2\lambda_5+\\+\lambda_1\lambda_2\lambda_4\left(1+p_1p_2p_3\right)\end{array}\right)$$

$$d=-p_1q_1\lambda_1\lambda_3\left(\begin{array}{l}\lambda_5\lambda_6+\lambda_4\lambda_6+\lambda_4\lambda_5+\lambda_2\lambda_6+\lambda_2\lambda_5+\\+\lambda_2\lambda_4+\lambda_1\lambda_6+\lambda_1\lambda_5+\lambda_1\lambda_4+\lambda_1\lambda_2\end{array}\right)$$

$$g=p_1q_1\lambda_1\lambda_3\left(\lambda_6+\lambda_5+\lambda_4+\lambda_2+\lambda_1\right)$$

$$k=\lambda_1\lambda_2\lambda_4\lambda_5\lambda_6\left(p_1p_2p_3+p_4^2q_2-1\right)$$

$$m=-\left(\begin{array}{l}\lambda_1\left(p_4^2q_2\lambda_5\lambda_6\left(\lambda_4+\lambda_2\right)\right|+\lambda_2\lambda_4p_1p_2p_3\left(\lambda_6+\lambda_5\right)\big)+\\+\left(\lambda_4\lambda_5\lambda_6\left(\lambda_2+\lambda_1\right)+\lambda_1\lambda_2\left(\lambda_5\lambda_6+\lambda_4\lambda_6+\lambda_4\lambda_5\right)\right)\end{array}\right)$$

$$n=\left(\begin{array}{l}\lambda_1\left(p_4^2q_2\lambda_5\lambda_6+\lambda_2\lambda_4p_1p_2p_3\right)-\left(\begin{array}{l}\lambda_5\lambda_6\left(\lambda_4+\lambda_2\right)+\lambda_2\lambda_4\left(\lambda_5+\lambda_6\right)+\\+\lambda_1\lambda_6\left(\lambda_5+\lambda_4\right)+\lambda_1\lambda_4\left(\lambda_5+\lambda_2\right)+\\+\lambda_1\lambda_2\left(\lambda_5+\lambda_6\right)\end{array}\right)\end{array}\right)$$

$$h=-\left(\begin{array}{l}\lambda_6\left(\lambda_5+\lambda_4+\lambda_2+\lambda_1\right)+\lambda_5\left(\lambda_4+\lambda_2+\lambda_1\right)+\\+\lambda_4\left(\lambda_2+\lambda_1\right)+\lambda_2\lambda_1\end{array}\right)$$

$$r=\lambda_6+\lambda_5+\lambda_4+\lambda_2+\lambda_1.$$

Після проведених розрахунків, можна зробити висновки, що в складних стохастичних мережах відсутні прості методи знаходження особливих точок функції *Ф(z)*, оскільки для їх знаходження необхідно розв'язати нелінійні рівняння, які ускладнюються пропорційно нарощуванні GERT-мережі.

Розв’язавши дане рівняння, маємо:

$$\Phi(z)=\frac{z^5-gz^4-dz^3-cz^2-bz-a}{\left((\lambda_1+z)(\lambda_3+z)\left(z^5-rz^4-hz^3-nz^2-mz-k\right)\right)} \tag{2.5}$$

Густина розподілу ймовірностей часу виконання алгоритмів системи управління ресурсом корпоративної мережі

$$\phi(x)=\frac{1}{2\pi i}\int_{-i\infty}^{i\infty}e^{zx}\frac{z^5-gz^4-dz^3-cz^2-bz-a}{\left((\lambda_1+z)(\lambda_3+z)\left(z^5-rz^4-hz^3-nz^2-mz-k\right)\right)}dz, \tag{2.6}$$

Густина розподілу ймовірностей Φ(z) має полюси в точках $z_1=-\lambda_1,\quad z_2=-\lambda_3$. Многочлен $z^5-rs^4-hz^3-nz^2-mz-k$ породжує ще п'ять полюсів. Розв'язавши рівняння

$$z^5-rz^4-hz^3-nz^2-mz-k \tag{2.7}$$

отримаємо ще п'ять особливих точок $z_3,\ z_4,\ z_5,\ z_6,\ z_7$

Знайдемо густину розподілу ймовірностей при таких параметрах гілок моделі: $\lambda_1=0{,}9$, $\lambda_2=0{,}9$, $\lambda_3=0{,}6$, $\lambda_4=0{,}3$, $\lambda_5=0{,}91$, $\lambda_6=0{,}9$ $p_1=0{,}9$, $p_2=0{,}9$, $p_3=0{,}6$, $p_4=0{,}8$.

Отримаємо, що $r=0{,}07$, $h=1{,}872$, $n=3{,}705$ $m=5{,}97$, $k=-3{,}91$

А функція $\Phi(z)$ має полюси $z_1=-0{,}9$, $z_2=-0{,}6$, $z_3=0{,}479$.

Крім дійсних коренів, є чотири комплексно-спряжених: $z_4=-1{,}124+i\cdot 0{,}963$, $z_5=-1{,}124-i\cdot 0{,}963$, $z_6=0{,}847-i\cdot 1{,}741$, $z_7=0{,}847+i\cdot 1{,}741$

Підставивши отримані дані, отримаємо:

$$\phi(\mathrm{x}) = 2\frac{e^{-1,124x}\left[\frac{(1,341\cos(0,963x))}{10^{-3}} + \frac{-1,322\sin(-0,963x)}{10^{-3}}\right]}{8,348\times10^{5}}$$

Таким чином, розроблено математичну модель системи управління ресурсом корпоративної мережі. Модель можна використовувати для дослідження тестування безпеки корпоративної мережі для того щоб знизити її вразливості.

2.4 Висновки до розділу

У розділі розглянуто модель реалізації кібератаки на корпоративну мережу. Встановлено, що забезпечення безпеки корпоративної мережі неможливе без попередньої роботи пентестера засобами виявлення відповідних вразливостей мережі.

У розділі розроблено математичні моделі початкової генерації коду кібератаки та діагностики системи управління ресурсом корпоративної мережі. Розроблена математична модель генерування кодів кібератаки на корпоративну мережу, дозволяє дослідити основні етапи генерування таких кодів та в подальшому надати практичні поради для захисту мережі від подібних атак. Удосконалена математична модель діагностики системи управління ресурсами мережі, дозволила підвищити ефективність тестування безпеки програмних засобів корпоративної мережі.

3 МЕТОД ПОШУКУ АЛГОРИТМУ З ДВІЙКОВОГО КОДУ ДЛЯ ДІАГНОСТИКИ БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОРПОРАТИВНОЇ МЕРЕЖІ

3.1 Схема пошуку алгоритму з двійкового коду для діагностики безпеки програмного забезпечення корпоративної мережі

Зрозуміло, що сьогодні не існує мереж, які б не були в той чи інший спосіб вразливі до кібератак. Аналіз IT-індустрії показує, що кількість вдалих кібератак постійно збільшується. Цю негативну тенденцію можна пояснити легковажним ставленням до питань тестування безпеки даних мереж та неоднозначності трактування самого поняття тестування безпеки даних. Наприклад, ряд авторів [38-42] тестування безпеки даних корелюють з пошуком та усуненням небезпек, що загрожують роботі корпоративної мережі. До таких небезпек можна віднести визначення паролю зовнішніми засобами; атаку на мережу за допомогою аналізуючих захист додатків; атаки відмови обслуговування інших клієнтів; цілеспрямований ввід помилок для проникнення в мережу під час відновлення; аналізування загальнодоступної інформації для пошуку паролю для входження в мережу.

Проте, існують і інші методи проникнення в мережу. Наприклад, це може бути метод відновлення в машинно-незалежному вигляді алгоритму з набору двійкових векторів-ознак [53]. Цей метод дозволяє знайти неоголошені розробником вразливості в ПЗ, помилки реалізації, виявити шкідливий код, що може бути комп'ютерним вірусом.

Крім цього, є ряд спеціалізованих додатків, які дозволяють виділити деякі алгоритми з двійкового коду, наприклад IxChariot. Проте такі додатки досліджують, зазвичай, лише ту частину програми, що використовується під час запусків й залишає певні «сліди» - вектори ознак. Тому розробка методу пошуку алгоритму з двійкового коду для діагностики безпеки даних корпоративної мережі є актуальною.

Схему пошуку алгоритму з двійкового коду приведено на рис. 3.1.

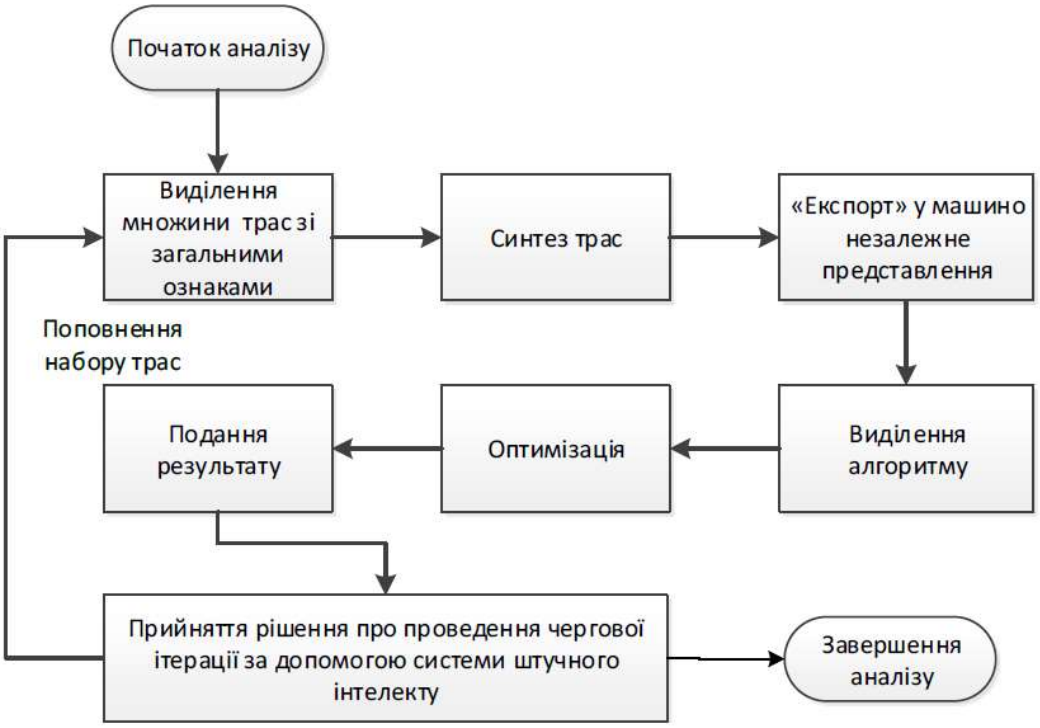


Рисунок 3.1 - Схема пошуку алгоритму

Якщо аналізові піддаються прості програми, то достатньо один раз пройти схему. Якщо ж аналізуються складні програми, то виникає необхідність у повторенні алгоритму.

З рисунку бачимо, що схема пошуку алгоритму починається з виділення множини трас - векторів із загальними ознаками. Далі йде синтез трас, тобто інформація про об’єкт дослідження представляється за допомогою методу графів. Потім інформація експортується в машино-незалежне представлення та проводиться виділення алгоритму, тобто частини коду, що ми аналізуємо. Після цього йде оптимізація, для спрощення одержаного результату. Блок Подання результату забезпечує представлення коду у формі, придатній для пентестера, а також для подачі в систему штучного інтелекту для прийняття відповідного рішення про потребу проходження чергової ітерації або про завершення аналізу.

3.2 Модель виділення множини векторів із загальними ознаками

Двійковий вектор виконання, отриманий за допомогою відповідного

симулятора є послідовністю кроків, які містять код поданої на виконання команди.

Опис позначень, пов'язаних з двійковими векторами, приведений в таблиці 3.1.

Таблиця. 3.1 – Позначення вектору виконання

Позначення	Опис
$len[t]$ $t^{(i)}$	Кількість кроків у атракторі t Крок атрактору t з номером $i \in \{1,2,...,len[t]\}$
$addr[t^{(i)}]$ $inst[t^{(i)}]$	Адреса, що виконувалася на кроці $t^{(i)}$ інструкції Інструкція, що виконувалася на кроці $t^{(i)}$
$process[t^{(i)}]$ $thread[t^{(i)}]$	Ідентифікатор процесу на кроці $t^{(i)}$ Ідентифікатор потоку виконання на кроці $t^{(i)}$
$read[t^{(i)}]$ $write[t^{(i)}]$	Множина адрес пам'яті, що читаються на кроці $t^{(i)}$ Множина адрес пам'яті, що записуються на кроці $t^{(i)}$
$size[j]$ $branch[j]$	Розмір у байтах коду інструкції j Ознака, чи є інструкція j передачею управління

Для опису поведінки програми, що тестується, крім представлених векторів, для дослідження необхідно знати про те, як ці вектори між собою співвідносяться. Тому під пов'язаними векторами розуміємо набір векторів, отриманих з одного й того ж початкового стану формування системи. Такі вектори очевидно відрізняються сценаріями, що реалізовуються в аналізованій системі в кожному досліді, що також визначається вхідними даними. Якщо використовуються інтерактивні програми, то вхідними даними можуть бути послідовність дій на графічний інтерфейс.

Головними вихідними даними для запропонованого методу відновлення алгоритму буде виступати набір пов'язаних векторів.

Першим кроком методу пошуку алгоритму є об'єднання набору пов'язаних векторів у загальний вираз $G = (V, C)$. Цей вираз – це орієнтований граф із петлями, що відповідають набору міжпроцедурних графів потоку для управління

окремими потоками виконання з додатковими позначеннями. У загальному випадку, коли вектори включають кілька потоків виконання, цей граф виявиться незв'язним, і кожному потоку виконання буде відповідати своя компонента зв'язності.

Для графа G, як і для звичайного графа потоку керування, вершина *V* відповідає базовому блоку (лінійній ділянці інструкції приведеного потоку виконання), а ребро C - можливим передаванням керування між цими відрізками. Так як в запропонованій методиці єдиним джерелом знань про потік управління інструкції є її вектори, граф буде містити тільки ті ребра, переходи за якими дійсно спостерігалися.

Окрім простих вершин базових блоків, кожному потоку до виконання в графі *G* визначено той, який точно не містить будь-яких команд вхідної та вихідної вершини, перша вершина з котрих домінує, а друга вершина з них постдомінує над всіма другими базовими блоками цього потоку до виконання.

При кожній такій вершині зберігається початковий адрес, послідовність команд та номер покоління. Номер покоління – це ціле число, яке показує стан коду команди. Воно дозволяє коректно представити код, який змінюється під час виконання. У кожному взятому окремо потоці до виконання відповідний номер покоління збільшується на одиницю при перезаписі коду цього потоку.

Позначення, які будуть далі використовуватися, приведено в таблиці 3.2.

Адреса кінця базового блоку $end[B]$ обчислюється за формулою

$$end[B] = start[B] + \sum_{j \in insn[B]} size[j]$$

Окрім позначень у табл. 3.2, у коді приведених далі алгоритмів будуть вважатися доступними такі функції, реалізація яких залежить від обраного методу зберігання графа.

Таблиця 3.2 - Основні позначення та функції

Позначення	Опис
$entry[G]_T$	Вхідна вершина потоку виконання T в графі G
$exit[G]_T$	Вихідна вершина потоку виконання T в графі G
$start[B]$	Початкова адреса базового блоку B
$end[B]$	Адреса кінця базового блоку B, не включно
$insn[B]$	Послідовність інструкцій в базовому блоці B
$gen[B]$	Номер покоління базового блоку B
$succ[B]$	Множина базових блоків, у які ведуть ребра з B
$pred[B]$	Множина базових блоків, з яких ведуть ребра в B
$from[e]$	Базовий блок, з якого виходить ребро e
$to[e]$	Базовий блок, у який входить ребро e

1. Функція *F1* призначена для створення та повертання нового порожнього графу.

2. Функція *F2 (G, T, n, a)* призначена для створення нового основного блоку у графі G, який приналежний до покоління з номером n потоку виконання *T,* та який виконує призначення початкової адреси *a.* Список команд новоствореного базового блоку спочатку задається порожнім. Адреса *a* може мати значення $\emptyset$, що відповідає за не прив'язані до адреси початкові та кінцеві вершини.

3. Функція *F3 (G, B)* відповідає за видалення основного блоку *B* з графа *G* разом з усіма його ребрами.

4. Функція *F4 (G, T, n)* відповідає за повернення відсортованого за адресами списку основних блоків покоління за номером n потоку *T* в графі *G.*

5. Функція *F5 (G, T, n, a)* відповідає за пошук основного блоку *B* в графі G, що відноситься до покоління з номером *n* потоку *T* так, що до нього належить адреса *a*: $start[B] \le a < end[B]$. Якщо цуй блок не можливо

знайти, відбувається присвоєння йому значення 0.

6. Функція *F6 (G, B, a)* відповідає за розподілення основного блоку *B* на два так, що команди *B, з* адресами меншими за адреси *a*, потрапляють у перший блок, а інші - в другій. Адреса *a* має належати основному блоку *B*. *F6* повертає два блоки *(B_x, B_2)*, при адресі в блоці B_1 меншій, ніж *a,* та при адресі в B_2 - більшій, ніж *a*.

7. Функція *F7 (G, T, n)* відповідає за повернення сукупності ребер, що з'єднують основні блоки з номером n в графі *G*.

Зрозуміло, що граф *G* пов’язаний з набором векторів. Якщо такий набір порожній, то порожній і граф.

Для кращого розуміння приведемо опис алгоритмів виконання побудови графа *G*.

Нехай у векторі, що ми опрацьовуємо, не буде зміни коду при його виконанні. Це буде тоді, коли вектор показує роботу головної частини програми. При цьому передбачається, що ми уже завантажили саму програму та відповідні бібліотеки.

Програма алгоритму Відтворювання графа представлено на рисунку 3.2.

1: **function** Static graph construction (t)
2: $G \leftarrow$ F1
3: **for** $T \in \bigcup_{i=1}^{len[t]} \left\{ thread\left[t^{(i)} \right] \right\}$ **do**
4: $entry\left[G\right]_T \leftarrow$ F2 (G, T, 1, 0)
5: $exit\left[G\right]_T \leftarrow$ F2 (G, T, 1, 0)
6: $E \leftarrow$ Static graph construction funiculus (t)
7: $G \leftarrow$ F1
8: ($G, entry\left[G\right]_T, t$, 1, $len\left[t\right]$, T, 1)
9: F7 ($G, E, exit\left[G\right]_T$)
10: **end for**
11: **return** G
12: **end function**

Рисунок 3.2 - Програма алгоритму Відтворювання графа

Вхідними даними є вектор t, а на виході програма будує граф *G*. Потім ми переходимо до алгоритму Продовження відтворення графа, приведеного на рисунку 3.3.

Він має наступні параметри: *G* - граф, який ми маємо побудувати, *S* – вершина цього графу, *t* – вектор з діапазоном кроків *[a,b]*. Розроблений алгоритм призначений для повернення основного блоку.

```
1: function Static graph construction funiculus (G, S, t, a, b,
   T, n)
2: E ← S ; m ← 0
3: for i = a, a + 1,..., b do
4: if thread[t^(i)] = T then
5: if m = 0 then
6: B ← F5 (G, T, n, addr[t^(i)])
7: if B = 0 then
8: B ← F2 (G, T, n, addr[t^(i)])
9: i ← i − 1
10:else if start[B] ≠ addr[t^(i)] then
11:⟨B′, B⟩ ← F6 (G, B, addr[t^(i)])
12:end if F7 (G, T E, B)
13:E ← B; m ← |insn[B]| − 1
14:else if m > 0 then
15:m ← m − 1
16:else if m < 0 then
17:B ← F5 (G, T, n, addr[t^(i)])
18:if B = 0 then
19:insn[E] ← insn[E] ∪ insn[t^(i)]
20:if branch[insn[t^(i)]] then
21:m ← 0
22:end if
23:else
24:F7 (G, E, B)
25:E ← B; m ← |insn[B]| − 1
26:end if
27:end if
28:end if
29:end for
30:if (E ≠ S) ∧ (addr[t^(b)] + size[insn[t^(b)]] ≠ end[E]) then
31:⟨E, E′⟩ ← F6 (G, E, addr[t^(b)] + size[insn[t^(b)]])
32:end if
33:return E
34:end function
```

Рисунок 3.3 - Програма алгоритму Продовження відтворювання графа

Алгоритм виконує покроковий прохід по вектору з потоку *T*. Ми будемо відстежувати, у якому основному блоці міститься виконання. Величина m описує алгоритм: якщо m = 0 розуміємо, що було передано керування; якщо *m* > 0 керування міститься всередині основного блоку; якщо *m* < 0 керування міститься всередині базового блоку, який до цього не зустрічався.

При передачі керування з блоку *E* можливі такі ситуації:

1. Керування передається за адресою початку відомого основного блоку.

2. Керування передається за адресою у відомому основному блоці, проте не на початок.

3. Керування передається за адресою, що не належить якомусь відомому основному блоці. При цьому створюється новий основний блок з заданою адресою початку та додається відповідне ребро.

Якщо переглядати команди нового основного блоку в третій ситуації, то кожна чергова команда додається і його список. Це буде тривати доти, поки не виконається умова закінчення основного блоку. При чому адреса наступної команди має відповідати за початок основного блоку.

Якщо було переглянуто усі команди потоку в заданому діапазоні кроків, то можливе проведення поділу останнього основного блоку.

Розглянемо питання наявності у програмі динамічного коду. Динамічний код може з’являтися внаслідок різноманітних причин. Наприклад, виконання редактором операції завантаження та вивантаження динамічних бібліотек. Це відбувається внаслідок того, що діапазон адрес завантаженої бібліотеки може перекритися з діапазоном вивантаженої бібліотеки. При цьому, в тих самих адресах пам'яті може знаходитися неоднаковий код.

Проблеми можуть виникнути і внаслідок поліморфної природи програми. Це спостерігається тоді, коли наступний варіант коду програми побудовано на основі першого. Зрозуміло, що такий сценарій дещо ускладнить аналіз. Зловмисники використовують цю процедуру для

імплементації в програму шкідливого коду, і такий сценарій не дасть провести сигнатурний аналіз антивірусними програмами.

У магістерській роботі не ставиться за мету знайти причини та розділити динамічну зміну коду. Динамічна зміна коду проходить за наступних сценарієм.

Один з периферійних пристроїв змінює сторінки фізичної пам'яті шляхом виконання DMA-транзакції. Під час цієї зміни у векторі не спостерігається запису інформації. Таким чином, опишемо критерій наявності динамічного коду в частині вектору. Нехай в заданому векторі маємо діапазон номерів кроків та зафіксовано певний процес. Представимо множину цих кроків через $R_p = \left\{ r \in R : proced\left[t^{(r)} \right] = R \right\}$. Побудуємо наступні множини:

$$\Re(R_p) = \bigcup_{r \in R_p} read\left[t^r \right] \tag{3.1}$$

$$W(R_p) = \bigcup_{r \in R_p} write\left[t^r \right] \tag{3.2}$$

$$X(R_p) = \bigcup_{r \in R_p} \left[addr\left[t^r \right], addr\left[t^r \right] + size\left[insn\left[t^r \right] \right] - 1 \right] \tag{3.3}$$

Зрозуміло, що множини містить усі адреси, в які проводився запис команд в заданих кроках. Нехай відома множина адрес віртуальної пам'яті в певному процесі, перезаписаних у результаті DMA-транзакцій в заданому діапазоні. Позначимо цю множину через $D_p(R)$.

Таким чином, множина не матиме динамічного коду тоді і тільки тоді, коли виконано

$$\left(\Re(R_p) \cup W(R_p) \cup D_p(R) \right) \cap X(R_p) = 0 \tag{3.4}$$

При цьому отримуємо код, який є статичним та відповідно може аналізуватися методами статичного аналізу. До нього використаємо алгоритм Статичного відтворення графу. Для цього необхідно розбити вектор на відрізки статичного коду. При цьому відрізки будуємо починаючи від менших кроків до більших. Кожний наступний відрізок розширюємо поки виконується умова (3.4).

Програма алгоритму Статичного відтворення графу представлена на рисунку 3.4.

1: **function** *CTG-Full* (t)
2: $G \leftarrow$ F7
3: **for** $T \in \bigcup_{i=1}^{len[t]} \{thread[t^i]\}$ **do**
4: $entry[G]_T \leftarrow$F2 $(G, T, 1, 0)$
5: $n_T \leftarrow 1$; $E_T \leftarrow entry[G]_T$;
6: **end for**
7: **for** $P \in \bigcup_{i=1}^{len[t]} \{process[t^i]\}$ **do**
8: **for** $\langle a, b \rangle \in$ *CTG-Full-Розбиття* (t, P) **do**
9: **for** $T \in \bigcup_{i=a}^{b} \{thread[t^i]\}$ **do**
10: $E_T \leftarrow$ Static graph construction funiculus $(G, E_T\, t, a, b, T, n_T)$
11: $n_T \leftarrow n_T + 1$
12: **end for**
13: **end for**
14: **end for**
15: **for** $T \in \bigcup_{i=1}^{len[t]} \{thread[t^i]\}$
16: $exit[G]_T \leftarrow$F2 $(G, T, n_T, 0)$
17: F7 $(G, E_T, exit[G]_T)$
18: **end for**

Рисунок 3.4 - Алгоритм Статичного відтворення графу

У приведеному алгоритмі на вхід подаємо вектор *t*, а на виході отримуємо побудований граф *G*.

Алгоритм розбиття вектору на відрізки статичного коду наведено на рисунку 3.5.

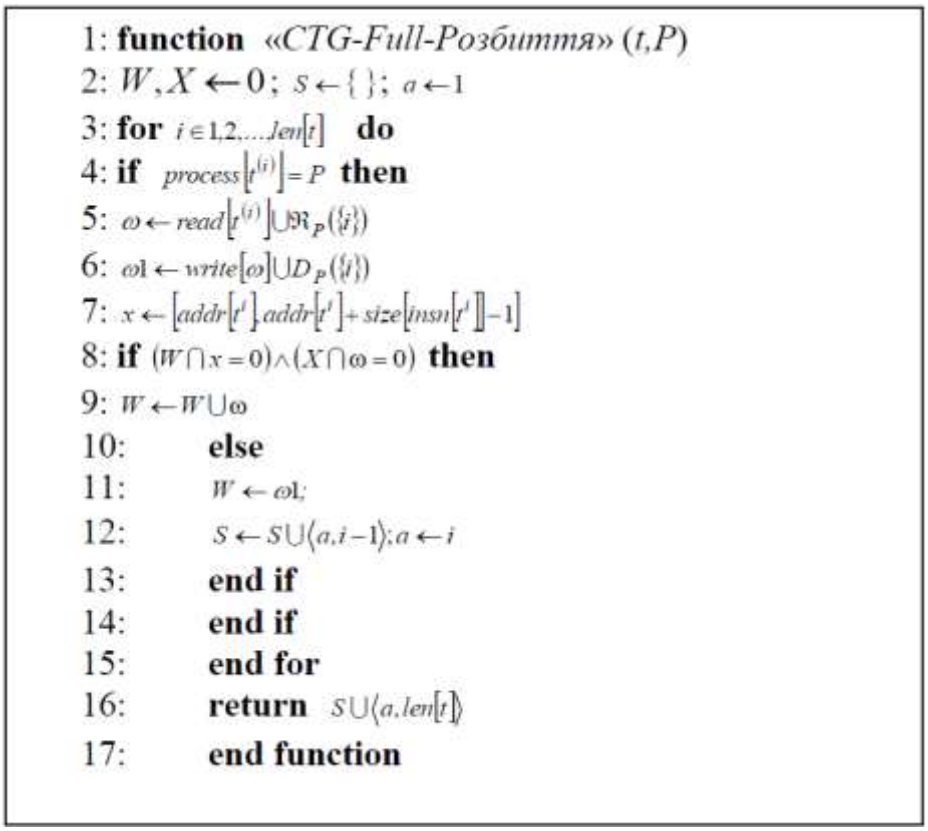


Рисунок 3.5 - Алгоритм розбиття вектору на відрізки статичного коду

Приведений на рисунку 3.5 алгоритм задає впорядковану послідовність відрізків *S* по вектору *t* та ідентифікатору процесу *P*.

Таким чином, першим етапом методу пошуку алгоритму з двійкового коду для діагностики безпеки даних корпоративної мережі є розроблені алгоритми. Ми змогли побудувати граф для довільного вектору. Це допоможе досліджувати як статичні, так і динамічні коди.

3.3 Побудова алгоритму утворення трас

Вихідними даними для алгоритму утворення трас виступає набір таких графів, які будуть зв'язані з трасою *t*. Розглянемо спочатку потік виконання $T = thread[t^{(1)}]$.

Якщо для основних блоків $B \in V_i$, $B' \in V_J$ команди в перетині збігаються, то такі блоки називаються сумісними. На рисунку 3.6 приведено

приклад таких сумісних множин. Сумісними є перша та друга множина, друга та третя, а от перша і третя між собою не сумісні.

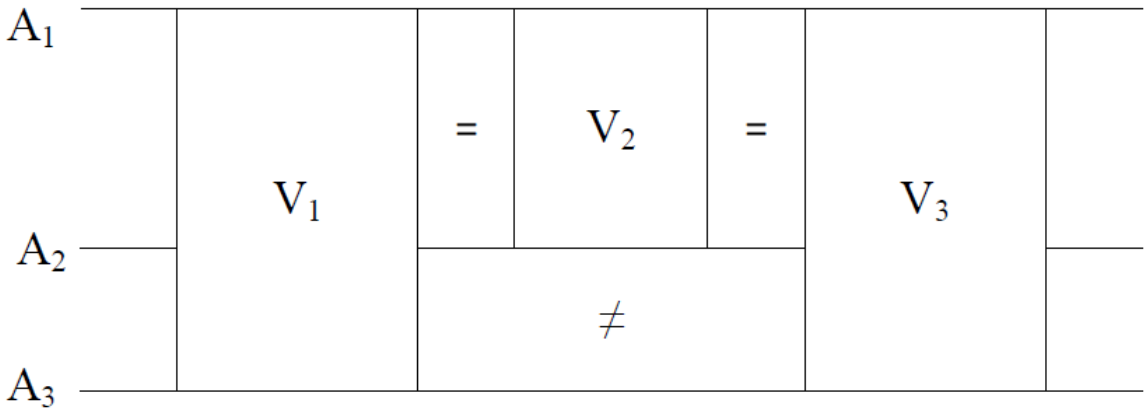


Рисунок 3.6 - Приклад сумісних множин

Якщо основні блоки B і B' перетинаються, але не збігаються повністю, тобто $start[B] \neq start[B']$, тоді утворяться різні шляхи у двох графах. Такий приклад приведено на рисунку 3.7.

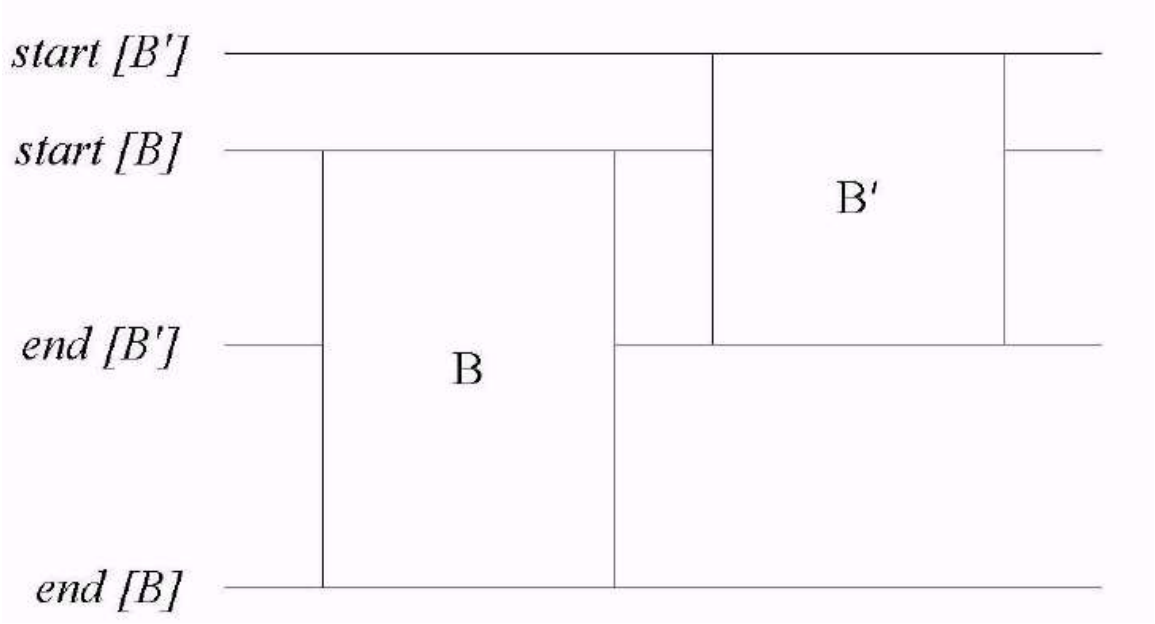


Рисунок 3.7 – Приклад часткового перетину основних блоків

Сумісність притаманна не лише множинам, але і для поколінь. Покоління називають сумісними, якщо сумісні їх множини в основних блоках.

Приклад побудови сумісних поколінь зображено на рисунку 3.8.

```
1: function «Генератор» (G*, n*, T, N)
2:         тCFG -Merge BBs (G*, n*, T, N)
3:         тCFG -Merge Edges(G*, n*, T, N)
4:     end function
```

Рисунок 3.8 - Сумісні покоління

Приведений алгоритм містить два етапи. На першому етапі ми розбиваємо та утворюємо основні блоки. Цей алгоритм приведено на рисунку 3.9. На другому етапі основні блоки сполучають ребрами. Цек алгоритм приведений на рисунку 3.10.

```
1:  function «Базовий блок» (G*, n*, T, N)
2:    for ⟨G_i, n_i⟩ ∈ N do
3:      for B ∈ «Генератор» (G_i, T, n_i) do
4:        B* ← «Синтез» (G*, T, n*, start[B])
5:        insn[B*] ← insn[B]
6:        for E* ∈ «Генератор» (G_i, T, n_i) \ {B*} do
7:          if (start[B*] = start[E*]) ∧ (end[B*] = end[E*]) then
8:            «Вилучення» (G*, B*)                    ▷ випадок (a)
9:            exit for
10:         else if (start[B*] > start[E*]) ∧ (start[B*] < end[E*]) then
11:           ⟨E1*, E2*⟩ ← «Розділ» (G*, E*, start[B*])
12:           if end[B*] = end[E2*] then
13:             «Вилучення» (G*, B*)                  ▷ випадок (b)
14:             exit for
```

15: *else if* $end[B^*] < end[E_2^*]$ *then*
16: $\langle E_{2,1}^*, E_{2,2}^* \rangle \leftarrow$ «Розділ» $(G^*, E_2^*, end[B^*])$
17: «Вилучення» (G^*, B^*) ▷ випадок (c)
18: *exit for*
19: *else if* $end[B^*] > end[E_2^*]$ *then*
20: $\langle B_1^*, B_2^* \rangle \leftarrow$ «Розділ» $(G^*, B^*, end[E_2^*])$
21: «Вилучення» (G^*, B_1^*) ▷ випадок (d)
22: $B^* \leftarrow B_2^*$
23: *end if*
24: *else if* $(end[B^*] > start[E^*]) \wedge (end[B^*] < end[E^*])$ *then*
25: $\langle E_1^*, E_2^* \rangle \leftarrow$ «Розділ» $(G^*, E^*, end[B^*])$
26: *if* $start[B^*] = start[E_1^*]$ *then*
27: «Вилучення» (G^*, B^*) ▷ випадок (e)
28: *exit for*
29: *else if* $start[B^*] < start[E_1^*]$ *then*
30: $\langle B_1^*, B_2^* \rangle \leftarrow$ «розділ» $(G^*, B^*, start[E_1^*])$ ▷ випадок(f)
31: тCFG- BB-Delete (G^*, B_2^*)
32: *exit for*
33: *end if*
34: *else if* $(start[B^*] < start[E^*]) \wedge (end[B^*] > end[E^*])$ *then*
35: $\langle B_1^*, B_2^* \rangle \leftarrow$ «Розділ»$(G^*, B^*, start[E^*])$ ▷ випадок (g)
36: $\langle B_{2,1}^*, B_{2,2}^* \rangle \leftarrow$ «Розділ» $(G^*, B_2^*, end[E^*])$
37: «Вилучення» $(G^*, B_{2,1}^*)$
38: $B^* \leftarrow B_{2,2}^*$
39: *end if*
40: *end for*
41: *end for*
42: *end for*
43: *end function*

Рисунок 3.9 – Алгоритм Базовий блок

```
1:  function «Синтез ребер» (G*, n*, T, N)
2:      for ⟨G_i, n_i⟩ ∈ N do
3:          for B ∈ «Генератор» (G_i, T, n_i) do
4:              B_1* ← «Знаходження» (G*, T, n*, end[from[e]]−1)
5:              B_2* ← «Знаходження» (G*, T, n*, start[to[e]])
6:              «Побудова» (G*, B_1*, B_2*)
7:          end for
8:      end for
9:  end function
```

Рисунок 3.10 - Алгоритм Синтез ребер

Множина можливих випадків взаємного розташування основного блоку наведена на рисунку 3.11.

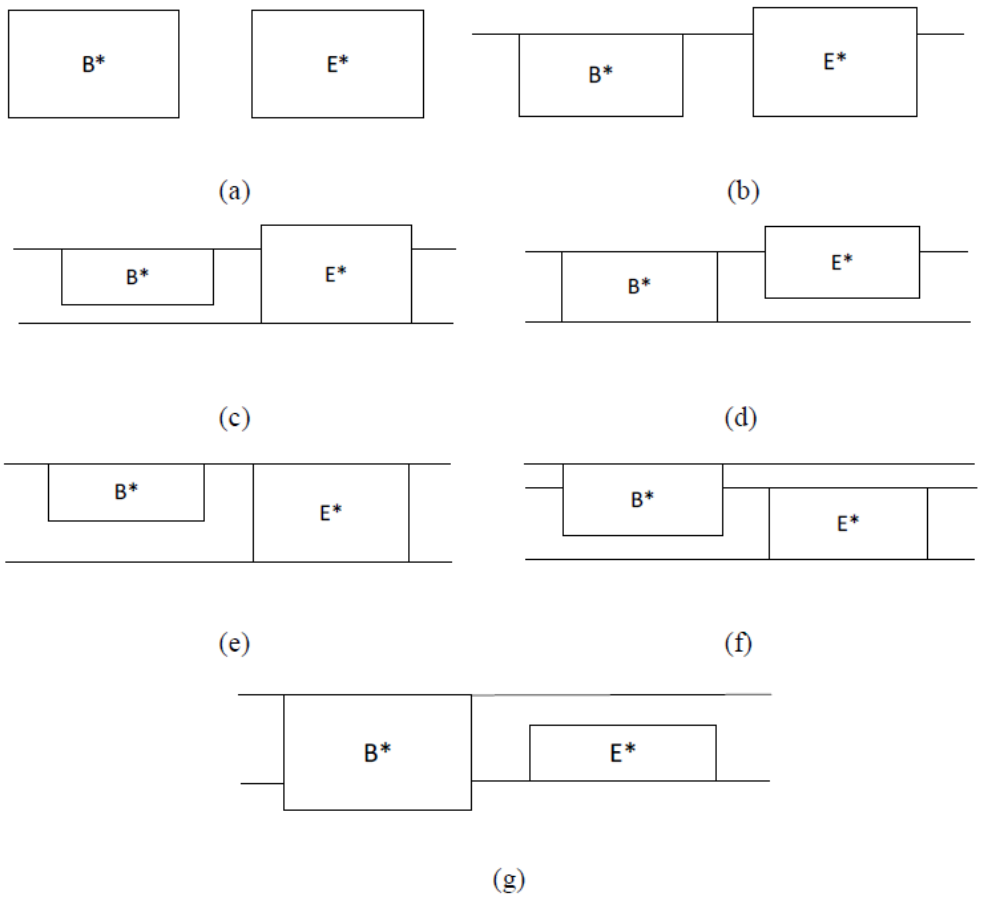


Рисунок 3.11 - Множина можливих випадків взаємного розташування основного блоку

Тепер постає питання як треба вибирати множини *N* для об'єднання поколінь відповідного потоку виконання. Кожному графу G_i необхідно привести його відображення f_i.

При побудові відображень необхідно забезпечити мінімальну кількість поколінь, щоб змогти побачити взаємозв'язки між конфігураціями і програмі. Це можна зробити шляхом розв’язку задачі попарної редукції. Тобто замість аналізу всіх послідовностей, що необхідно вирівняти потрібно розглядати лише одну їх пару. Цю операцію виконуємо доти, поки послідовності не стануть редуковані до єдиної.

Розглянемо приклад вирівнювання покоління для довільної кількості графів шляхом заповнення їх матриць. Якщо маємо два графа $G \in \Gamma$ і $G' \in \Gamma$, для яких потік виконання *T* має *k* і *k'* поколінь. Побудуємо алгоритм відображень *f* і *f* '. Спочатку необхідно побудувати матриці *L* і *D* розмірності $(k+1)\times(k'+1)$. Елемент *L* має мінімальну довжину.

Заповняємо дані матриці за допомогою алгоритму Нідлмана-Вунша, який належить до динамічного програмування, та є глобальним вирівнюванням [59].

Даний алгоритм складається із трьох послідовних етапів. На першому необхідно побудувати ініціюючі матриці. Для цього дві послідовності, що ми порівнюємо, розташовуємо у верхньому і нижньому рядках. Окрім цього перед кожною послідовністю виставляємо пропуск. Далі заповнюємо перший стовпчик і перший рядок. Заповнення здійснюємо за допомогою методу штрафу за пенальті.

На другому етапі заповнюємо безпосередньо таблицю. Значення комірки розраховується за формулою:

$$F_{ij} = \max(F_i - 1, j - 1 + S(A_i, B_j), F_i, j - 1 + d, F_i - 1, j + d)$$

де F_{ij} - значення в певній комірці,

$S(A_i, B_j)$ - очки за збіжність комірок в певних рядках,

d - штаф пенальті (заданий).

На основі цієї матриці будуємо матрицю локалізації. Слідкуємо за процесом її заповнення.

На третьому етапі виконуємо пошук максимального вирівнювання. Пошук починаємо із останньої кутової комірки, а завершуємо завжди першою коміркою. Алгоритм вирівнювання: необхідно на основі матриці локалізації створюємо шлях, який базується на інформації кожної комірки. Буква D — діагональ, при цьому необхідно перейти на комірку, яка розташована у діагоналі, Т — вершина, при цьому необхідно перейти на 1 комірку вгору, L — вліво, при цьому необхідно перейти на комірку, розташовану ліворуч. Якщо комірка має два значення, відповідно можливі два напрямки руху, три — три.

Таким чином, псевдо-код алгоритма для обчислення матриці F буде мати наступний вигляд, представлений на рисунку 3.12.

```
for i=0 to length(A)
  F(i,0) ← d*i
for j=0 to length(B)
  F(0,j) ← d*j
for i=1 to length(A)
  for j = 1 to length(B)
  {
    Match ← F(i-1,j-1) + S(Ai, Bj)
    Delete ← F(i-1, j) + d
    Insert ← F(i, j-1) + d
    F(i,j) ← max(Match, Insert, Delete)
  }
```

Рисунок 3.12 - Псевдо-код алгоритма для обчислення матриці F

Коли матриця F розрахована, її F_{ij} дає максимальну оцінку серед усіх можливих вирівнювань. Для обчислення самого вирівнювання, що отримало таку оцінку, потрібно почати з правої нижньої клітинки і порівнювати значення в ній з трьома можливими джерелами, щоб побачити, звідки воно з'явилось. У випадку відповідності L_i і D_j вирівняні.

На рисунку 3.13 приведений приклад вирівнювання матриць.

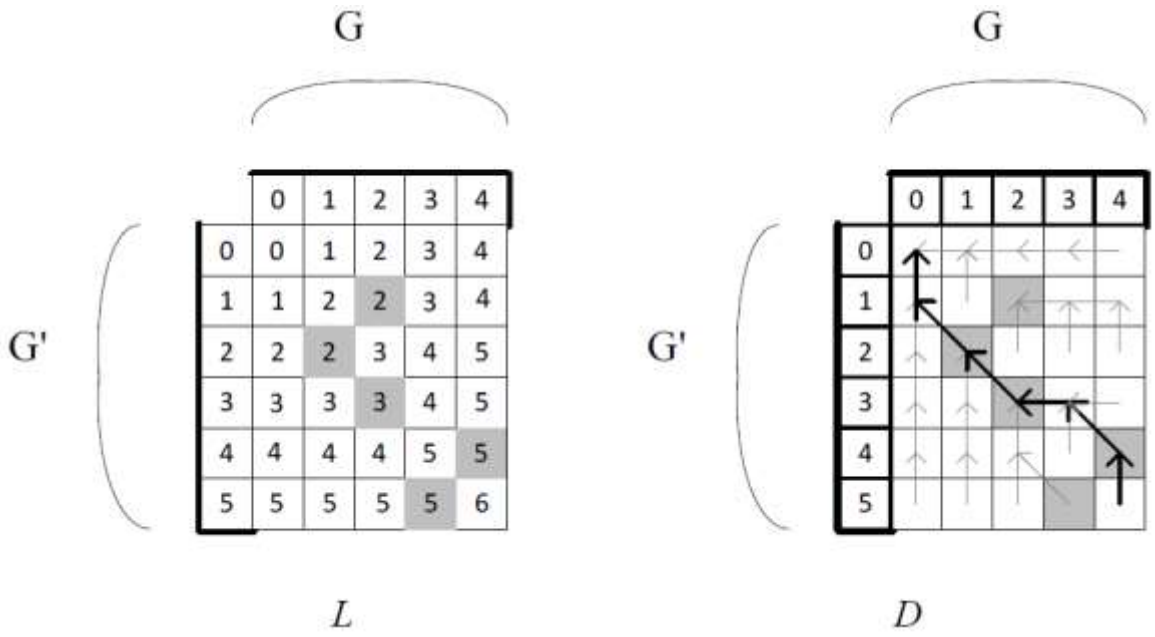


Рисунок 3.13 – Приклад вирівнювання матриць

Код вищеописаного вирівнювання має вигляд, наведений на рисунку 3.14

```
AlignmentA ← ""
AlignmentB ← ""
i ← length(A)
j ← length(B)
while (i > 0 or j > 0)
{
  Score ← F(i,j)
  ScoreDiag ← F(i - 1, j - 1)
  ScoreUp ← F(i, j - 1)
  ScoreLeft ← F(i - 1, j)
  if (Score == ScoreDiag + S(Ai, Bj))
  {
    AlignmentA ← Ai + AlignmentA
    AlignmentB ← Bj + AlignmentB
    i ← i - 1
    j ← j - 1
  }
```

```
  else if (Score == ScoreLeft + d)
  {
    AlignmentA ← A_i + AlignmentA
    AlignmentB ← "-" + AlignmentB
    i ← i - 1
  }
  otherwise (Score == ScoreUp + d)
  {
    AlignmentA ← "-" + AlignmentA
    AlignmentB ← B_j + AlignmentB
    j ← j - 1
  }
}
```

```
while (i > 0)
{
  AlignmentA ← A_i + AlignmentA
  AlignmentB ← "-" + AlignmentB
  i ← i - 1
}
while (j > 0)
{
  AlignmentA ← "-" + AlignmentA
  AlignmentB ← B_j + AlignmentB
  j ← j - 1
}
```

Рисунок 3.14 - Код вирівнювання матриць

3.4 Висновки до розділу

У розділі розроблено схему пошуку алгоритму з двійкового коду для діагностики безпеки даних корпоративної мережі. Встановлено, що пошук алгоритму починається з виділення множини трас - векторів із загальними ознаками. Далі йде синтез трас, тобто інформація про об’єкт дослідження представляється за допомогою методу графів. Потім інформація експортується в машино-незалежне представлення та проводиться виділення алгоритму, тобто частини коду, що ми аналізуємо. Після цього йде оптимізація, для спрощення одержаного результату та приведення інформації до форми, придатної для пентестера, а також для подачі в систему штучного інтелекту для прийняття

відповідного рішення про потребу проходження чергової ітерації або про завершення аналізу

Побудовано граф для довільного вектору, що допоможе досліджувати як статичні, так і динамічні коди

4 ПЕРЕВІРКА РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ

4.1 Перевірка моделі кібератаки

Проведемо перевірку математичної моделі кібератаки. Встановимо, яка буде густина розподілу ймовірностей часу тестування даних корпоративної мережі $\varphi(x)$ при наступних параметрах гілок: $\lambda_1 = 0{,}9$, $\lambda_2 = 0{,}15$, $\lambda_3 = 0{,}99$, $\lambda_4 = 0{,}1$, $\lambda_5 = 0{,}9$, $\lambda_6 = 0{,}99$ $p_1 = 0{,}9$, $p_2 = 0{,}5$, $p_3 = 0{,}5$, $p_4 = 0{,}9$, $p_5 = 0{,}8$, $p_6 = 0{,}9$.

Розв'язавши рівняння, яке стоїть у знаменнику

$$\phi(\mathrm{x}) = \frac{1}{2\pi i}\int_{-i\infty}^{i\infty} e^{zx}\frac{-\mathrm{yz}^6 + \mathrm{bz}^5 - tz^4 + \mathrm{uz}^3 - kz^2 + wz - h}{\left(-z^6 + jz^5 - dz^4 + gz^3 - vz^2 + rz - c\right)\times} \times\left(\lambda_1 + z\right)\left(\lambda_2 + z\right)\left(\lambda_5 + z\right)$$

встановили, що функція $\Phi(z)$ має прості полюси: $z_1 = -0{,}5$, $z_2 = -0{,}811$, $z_3 = -1{,}022$ та чотири комплексно-спряжених: $z_4 = -0{,}14 + i\cdot 0{,}021$, $z_5 = a - bi = -0{,}14 - i\cdot 0{,}021$, $z_6 = a - bi = -0{,}016 - i\cdot 0{,}042$, $z_7 = a - bi = -0{,}016 + i\cdot 0{,}042$

Значення густини розподілу ймовірностей часу початкової генерації коду кібератаки приведені в таблиці 4.1.

Таблиця 4.1 - Значення густини розподілу ймовірностей часу кібератаки

№	$\varphi_e(x)$	$\varphi_i(x)$	№	$\varphi_e(x)$	$\varphi_i(x)$
1	0.289	0.288	20	0.04	0.04
2	0.275	0.275	21	0.034	0.034
3	0.259	0.259	22	0.029	0.029
4	0.243	0.242	23	0.024	0.024
5	0.225	0.224	24	0.02	0.02

Продовження таблиці 4.1

№	$\varphi_в(x)$	$\varphi_i(x)$	№	$\varphi_в(x)$	$\varphi_i(x)$
6	0.208	0.208	25	0.016	0.015
7	0.191	0.19	26	0.013	0.012
8	0.174	0.173	27	0.01	0.01
9	0.158	0.158	28	0.008	0.008
10	0.143	0.142	29	0.006	0.005
11	0.129	0.128	30	8.783×10^{-3}	8.782×10^{-3}
12	0.115	0.115	31	7.206×10^{-3}	7.206×10^{-3}
13	0.103	0.102	32	5.853×10^{-3}	5.853×10^{-3}
14	0.091	0.09	33	4.699×10^{-3}	4.699×10^{-3}
15	0.08	0.08	34	3.718×10^{-3}	3.718×10^{-3}
16	0.071	0.07	35	2.89×10^{-3}	2.88×10^{-3}
17	0.062	0.062	36	2.15×10^{-3}	2.15×10^{-3}
18	0.054	0.054	37	1.616×10^{-3}	1.616×10^{-3}
19	0.046	0.046	38	1.136×10^{-3}	1.136×10^{-3}

Відповідна графічна залежність наведена на рисунку 4.1.

Точність знаходження функції $\varphi_i(x)$ залежить від похибка обраної інтерполяції. Наприклад, якщо застосовувати многочлен Лагранжа третього степеня, то похибка не перевищить 0,002. Чого буде достатньо для завдань, поставлених у даній магістерській роботі.

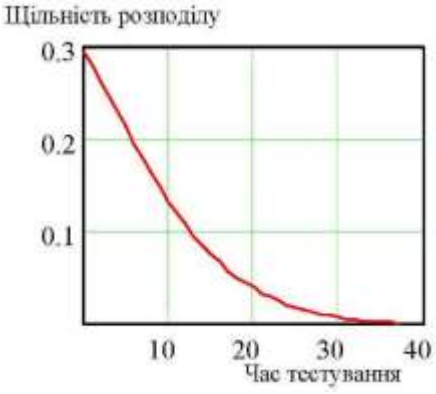


Рисунок 4.1 – Залежність щільності розподілу від часу

4.2 Перевірка моделі діагностики системи управління ресурсом корпоративної мережі

Проведемо дослідження моделі діагностики системи управління ресурсом корпоративної мережі. З другого розділу відомо, що густина розподілу ймовірностей часу виконання алгоритмів системи управління ресурсом корпоративної мережі визначається

$$\phi(\mathrm{x}) = \frac{1}{2\pi i}\int_{-i\infty}^{i\infty} e^{zx}\frac{z^5 - gz^4 - \mathrm{d}z^3 - cz^2 - bz - a}{\left((\lambda_1 + z)(\lambda_3 + z)\left(z^5 - rz^4 - \mathrm{h}z^3 - nz^2 - mz - k\right)\right)}dz,$$

Графік залежності функції розподілу та густини розподілу ймовірностей часу виконання алгоритмів системи управління ресурсом корпоративної мережі при знайдених полюсах від часу приведено на рисунку 4.2 та 4.3 відповідно.

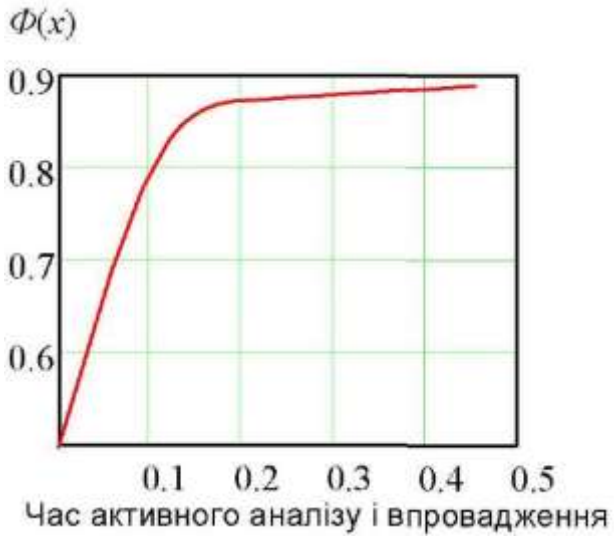


Рисунок 4.2 - Графік функції розподілу

Рисунок 4.3 – Залежність густини розподілу ймовірностей часу виконання в корпоративній мережі від часу

4.3 Перевірка методу пошуку алгоритму з двійкового коду для діагностики безпеки даних корпоративної мережі

Для перевірки методу пошуку алгоритму з двійкового коду для діагностики безпеки даних корпоративної мережі було проведено імітаційне моделювання з застосуванням у середовищі MathCAD 15 спеціальної програми, яка маже виконувати з одного боку захист, а з іншого генерування кібератаки, – IxChariot 9. Структурну схему моделі тестування безпеки даних корпоративної мережі представлено на рисунку 4.4.

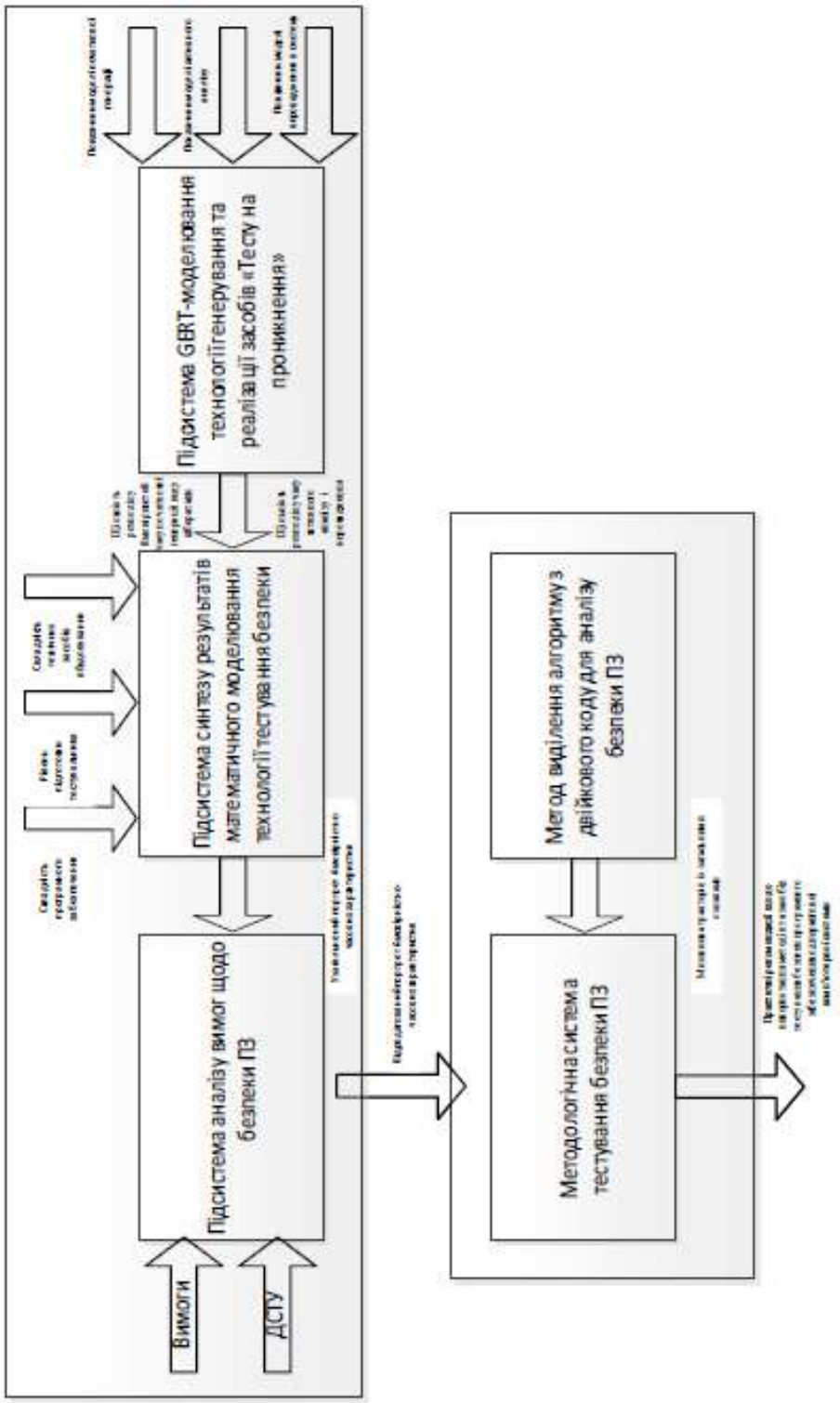


Рис. 4.4. Узагальнена структурна схема імітаційної моделі технології тестування безпеки програмного забезпечення

Результати оцінювання розробленої моделі приводяться на рисунку 4.5.

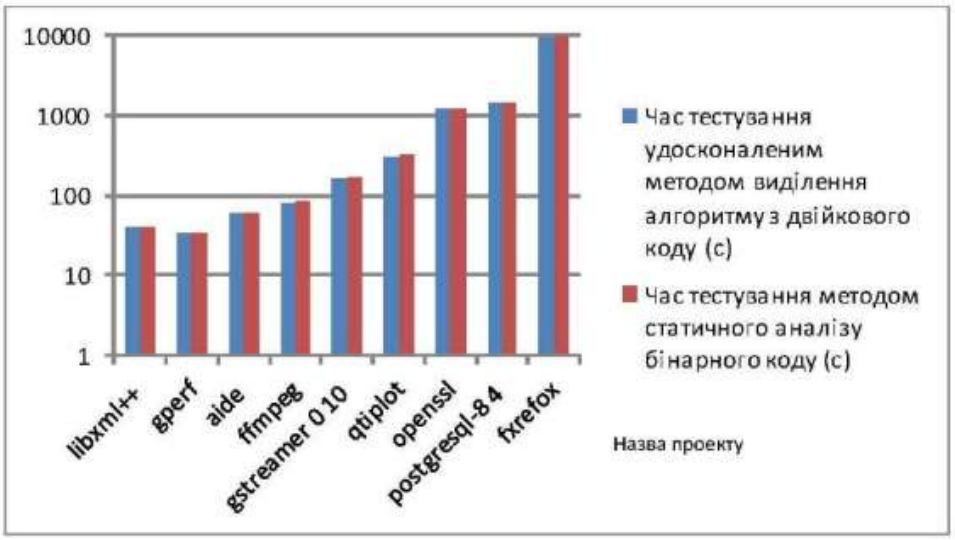


Рисунок 4.5 – Порівняння часу тестування безпеки даних удосконаленим та відомим методами

Під час тестування використовувалися програми з відкритим кодом для одержання результатів тестування даних з застосуванням удосконаленого методу та відомого. З рисунка видно, що удосконалений метод показує приблизно на 5% кращі показники по часу, ніж відомий.

Проведено також порівняння за рівнем безпеки удосконаленого та відомого методу (рисунок 4.6). Бачимо, що удосконалений метод підвищив безпеку даних до 3%.

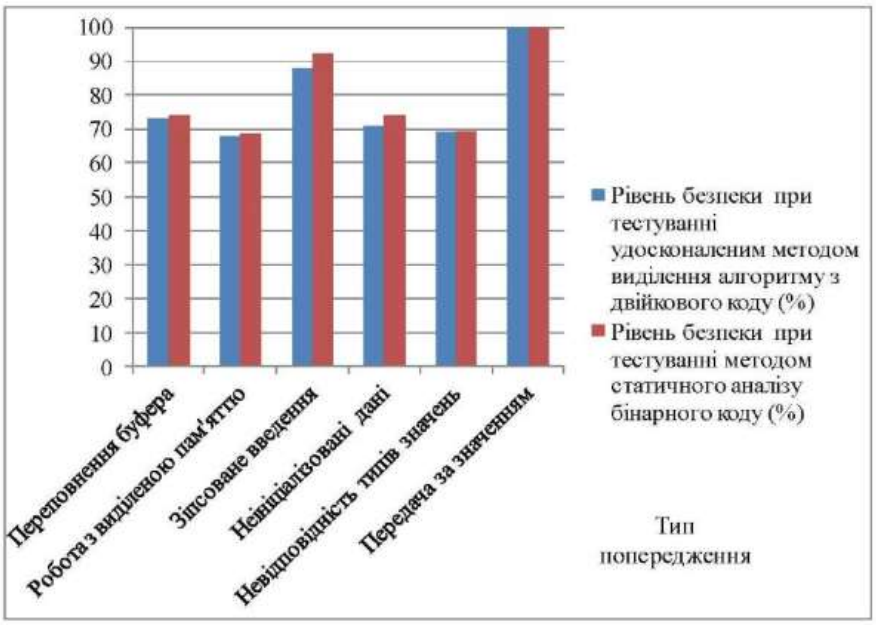


Рисунок 4.6 - Порівняння рівня безпеки удосконаленим та відомим методами

4.4 Висновки до розділу

У розділі проведено перевірку моделі кібератаки. Встановлено, щл точність знаходження функції $\varphi_i(x)$ залежить від похибка обраної інтерполяції. Наприклад, якщо застосовувати многочлен Лагранжа третього степеня, то похибка не перевищить 0,002. Чого буде достатньо для завдань, поставлених у даній магістерській роботі.

Перевірку моделі діагностики системи управління ресурсом корпоративної мережі та приведені графіки функції розподілу і залежності густини розподілу ймовірностей часу виконання в корпоративній мережі від часу.

Перевірку методу пошуку алгоритму з двійкового коду для діагностики безпеки даних корпоративної мережі. Для цього було проведено імітаційне моделювання з застосуванням у середовищі MathCAD 15 спеціальної програми, яка маже виконувати з одного боку захист, а з іншого генерування кібератаки, – IxChariot 9.

ВИСНОВКИ

У магістерській роботі вирішено важливе наукове завдання щодо розроблення методу тестування безпеки даних для захисту інформації в корпоративній мережі. В результаті проведених досліджень можна зробити наступні висновки:

1. Аналіз основних вимог до безпеки даних корпоративної мережі показав, що наявні методики тестування безпеки не дозволяють забезпечити прийнятний рівень їх захисту. Дослідження математичних моделей технологій тестування безпеки дозволили зробити вибір і сформулювати наукове завдання магістерських досліджень.

2. Розроблено математичну модель кібератаки на корпоративну мережу, яка дозволяє дослідити основні етапи кібератаки та в подальшому надати практичні поради для захисту мережі від подібних атак.

3. Удосконалено математичну модель діагностики системи управління ресурсами мережі, що дозволило підвищити ефективність тестування безпеки програмних засобів корпоративної мережі.

4. Удосконалено метод виділення алгоритму з двійкового коду для аналізування безпеки даних корпоративної мережі.

5. Розроблено модель виділення множини векторів-ознак із загальними ознаками.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Абазина Е.С. Цифровая стеганография: состояние и перспективы / Е.С. Абазина, А.А. Ерунов // Системы управления, связи и безопасности. – 2016. – № 2. – С. 182–201.

2. Агуреев К.И. Применение детерминированного хаоса для передачи информации / К.И. Агуреев // Известия ТулГУ. Технические науки. – 2014. – Вып. 11. Ч. 2. – С.197–212.

3. Анастасия Гришина, Андрей Куликов – Анализируем защищенность IT-систем. Positive Research 2018. Сборник исследований по практической безопасности. [Електронний ресурс]. – Режим доступу: https://www.ptsecurity.com/upload/ corporate/ru-ru/analytics/Positive-Research-2018-rus.pdf (дата звернення: 21.09.2020)

4. Барабаш О. В. Методи пошуку оптимальних маршрутів графа структури розгалуженої інформаційної мережі за заданим критерієм оптимальності при різних обмеженнях / О. В. Барабаш, І. П. Саланда, А. П. Мусієнко // Наукові записки Українського науково-дослідного інституту зв'язку. - К.: УНДІЗ, 2016. - №2 (42). - С 99-106.

5. Барабаш О.В. Забезпечення функціональної стійкості інформаційних мереж на основі розробки методу протидії DDoS-атакам / О.В. Барабаш, Н.В. Лукова-Чуйко, А.П. Мусієнко, В.В. Собчук. // Сучасні інформаційні системи Advanced Information Systems. - 2018. - Т. 2 ,№ 1.-С. 56-63.

6. Блинов А.М. Информационная безопасность / А.М. Блинов. - СПб.: Изд-во СПбГУЭФ, 2010. - 96 с.

7. Гавронський В.Є., Муляр І.В., Яцків В.В. Метод тестування на проникнення, як засіб забезпечення безпеки корпоративної мережі // НПК МНІС ІП-2020 Науково-практична інтернет-конференція молодих науковців і студентів «Інтелектуальний потенціал - 2020». – С. 18-21.

8. Гагарина Л.Г. Технология разработки программного обеспечения / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул - М: ИД «ФОРУМ», ИНФРА-М, 2008. - 400 с.

9. Глобальное исследование утечек конфиденциальной информации в 2017 году. Аналитический центр InfoWatch. [Електронний ресурс]. – Режим доступу: https://www.infowatch.ru/sites/default/files/report/analytics/russ/ InfoWatch_Global_Report_2017_year.pdf (дата звернення: 21.09.2020)

10.Горбенко І.Д. Прикладна криптологія. Теорія. Практика. Застосування / І.Д. Горбенко, Ю.І. Горбенко. -Х.:Форт, 2012. - 870 с.

11.Гусятников В.Н. Стандартизация и разработка программных систем / В.Н. Гусятников, А.И. Безруков - М: Финансы и статистика, 2010. - 288 с.

12.Департамент QA: Ошибки управления [Электронный ресурс] – Режим доступу: http://blog.alsedi.com/departament-qa-oshibki-upravleniya/

13.Дмитриев А.С. Динамический хаос: новые носители информации для систем связи / А.С. Дмитриев, А.И. Панас. – М. : Изд-во Физико-математической литературы, 2002. – 252 с.

14.Дмитриенко В.Д. Исследование потоковых свойств трафика, циркулирующего в компьютерных сетях систем критического применения для определения интервалов времени управления сетевыми ресурсами / В. Д. Дмитриенко, М. І. Науменко, С. Г. Семенов // Системи управління, навігації та зв'язку. - К.:ЦНДІ навігації і управління. -2009. -Вип. 3(11).-С. 198-201.

15.Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» [Електронний ресурс]. – Режим доступу до ресурсу: http://zakon4.rada.gov.ua/laws/show/2594-15

16.Земляный О.В. Передача информации на основе манипуляции спектром широкополосного хаотического сигнала / О. В. Земляный // Радиофизика и электроника. – 2015. – Т. 6(20), № 3. – С. 72–78.

17.Иванюк П.В. Хаотическое маскирование информационных сигналов с использованием генератора на базе системы Лю / П.В. Иванюк, Л.Ф. Политанский, Р.Л. Политанский, О.М. Элияшив // Технология и конструирование в электронной аппаратуре. – 2012. – № 3. – С. 11–17.

18.Колесов В.В. Применение дискретных хаотических алгоритмов в широкополосных телекоммуникационных системах / В.В. Колесов,

А.И. Полубехин, Е.П. Чигин, А.Д. Юрин // Вестник СибГУТИ. – 2016. – № 3. – С. 77–92.

19.Конформность и механизмы социального влияния [Електронний ресурс]. - Режим доступу до ресурсу: http://www.elitarium.ru/konformnost_mekhanizmy_vlijanija/

20.Кузнецов О.О. Протоколи захисту інформації у комп'ютерних системах та мережах / О.О. Кузнецов, С.Г. Семенов. - Х.: ХНУРЕ, 2009.-184 с.

21.Липаев В.В. Надежность и функциональная безопасность комплексов программ реального времени / В.В. Липаев. Институт системного программирования Российской академии наук -М: 2013. 176 с.

22.Лисиця Д.О. Імітаційна модель процесу генерації та реалізації засобів «тесту на проникнення» / Д.О. Лисиця, СВ. Козелков, С.Г. Семенов, А.О. Лисиця //Телекомунікаційні та інформаційні технології. - 2018. - Вип. 2(59).-С. 81 -90.

23.Лужецький В.А. Організаційно-правові питання безпеки інформації Концептуальна модель системи інформаційного впливу / В.А. Лужецький // Безпека інформації Ukrainian Scientific Journal of Information Security Том 23, № 1 (2017)

24.Мірошніченко О.В., Гавронський В.Є., Гурман І.В., Муляр І.В., Жиров Б.Г. Побудова інтелектуальних засобів оцінювання параметрів якості функціонування телекомунікаційних мереж // Всеукраїнська науково-практичної конференція молодих вчених, ад‟юнктів, слухачів, курсантів і студентів "Молодіжна військова наука у Київському національному університеті імені Тараса Шевченка" 24 квітня 2020 року.- С. 141-142.

25.Надеждин Е.Н. Оценка эффективности механизма защиты сетевых ресурсов на основе игровой модели информационного противоборства. / Е.Н.Надеждин // Научный вестник -Тамбов: ООО "Консалтинговая компания Юком". - 2007. - №2(4) . -С. 49-58

26.Певнев В.Я. Методы обеспечения целостности информации в инфокоммуникационных системах / В.Я. Певнев // Вісник Національного

технічного університету ХПІ. Серія: Техніка та електрофізика високих напруг. – 2015. - № 51. – С. 74-77

27.Передерий Ю.А. Метод оценки спектра ляпуновских показателей по временной реализации / Ю.А. Передерий // Известия вузов. ПНД. – 2012. – Т. 20, вып. 1. – С. 99–104.

28.Піскозуб А.З. Використання вільного програмного забезпечення для підвищення рівня захищеності комп'ютерних мереж та систем // Матеріали другої міжнародної науково-практичної конференції FOSS Lviv 2012,. – Львів, 2012.- с.86-90.

29.Полицын С. А. Подходы к вычислению временных затрат на проекты в сфере разработки программного обеспечения на основе использования прецедентов / С.А. Полицын // Программная инженерия. -2011 №7. -С.9-14

30.Положення про організацію заходів із забезпечення інформаційної безпеки в банківській системі України, затвердженого Постановою Правління Національного банку України від 28.09.2017 №95. [Електронний ресурс]. – Режим доступу до положення: https://bank.gov.ua/document/download?docId=56426049 (дата звернення: 21.09.2020)

*31.*Постанова Кабінету Міністрів України від 29.03.2006 №373 «Про затвердження Правил забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно- телекомунікаційних системах» [Електронний ресурс]. – Режим доступу до ресурсу: http://zakon2.rada.gov.ua/laws/show/373-2006-n.

32.Прикладне застосування теорії хаотичних систем у телекомунікаціях: монографія / [Ю.Я. Бобало, С.Д. Галюк, М.М. Климаш, Р.Л. Політанський]; Нац. ун-т «Львів. політехніка». – Львів: Коло, 2015. – 178 с.

33.Пятін І.С. Конфіденційна система зв'язку / І.С. Пятін, В.І. Лужанський, Л.В. Карпова // Вісник Хмельницького національного університету. Технічні науки. –2015. – № 1. – С. 207–212.

34.Савин Р. Тестирование Дот Ком, или пособие по жестокому обращению с багами в интернет-стартапах / Р.Савин - М.: Дело, 2007. -312с.

35.Семенов С.Г. Анализ и синтез защищенных компьютерных систем и сетей / С.Г. Семенов, А.А. Подорожняк, А.И. Баленко. - Х. :НТУ«ХШ» -2012.-204 с.

36.Семенов С.Г. Методика математического моделирования защищенной ИТС на основе многослойной GERT-сети / С.Г. Семенов // Вісник Національного технічного університету «Харківський політехнічний інститут». -Х.:НТУ «ХШ». -2012. -№62 (968). - С 173-181.

37.Сирота А.А. Компьютерное моделирование и оценка эффективности сложных систем / А.А. Сирота. -М.: Техносфера, 2006. -280 с.

38.Скляров Д. В. Искусство защиты и взлома информации / Д. В. Скляров - СПб.: БХВ-Петербург, 2004. - 288 с.

39.Слободян М.О., Таранчук А.А., Гавронський В.Є. Генерування широкосмугових хаотичних сигналів для прихованої передачі інформації в телекомунікаційних системах // Вісник Хмельницького національного університету. – 2020. – № 4.

40.Стефінко Я.Я., Піскозуб А.З. Використання відкритих операційних систем для тестування на проникнення в навчальних цілях/Я.Я. Стефінко, А.З. Піскозуб //Вісник Національного університету “Львівська політехніка” Комп’ютерні системи та мережі. – 2014. - № 806. – С. 258-263.

41.Тарасов В.А. Интеллектуальные системы поддержки принятия решений / В.А. Тарасов, Б.М. Герасимов, И.А. Левин, В.А. Корнейчук К.: МАКНС. -2007. - 335 с.

42.Халифе К. Комплекс математичних моделей процесу розробки програмного забезпечення / Кассем Халифе, С.Г. Семенов // Інформаційні технології та комп'ютерна інженерія. - Вінниця: ВИТУ, 2017-№2-С 14-20

43.Черушева Т. В. Проектирование программного обеспечения / Т. В. Черушева. -Пенза : Изд-во ПГУ, 2014. - 172 с.

44.Cisco 2018. Годовой отчет по безопасности. [Електронний ресурс]. – Режим доступу: https://www.cisco.com/c/dam/global/ru_ru/assets/offers/assets/cisco_2018_acr_ru.pdf (дата звернення: 21.09.2020)

45.Information Systems Security Assessment Framework (ISSAF). [Електронний ресурс]. – Режим доступу: http://www.oissg.org/files/issaf0.2.1.pdf (дата звернення: 21.09.2020)

46.ISO 9001:1994 «Системы качества Модель для гарантии качества в проектировании, разработке, изготовлении, установки и обслуживании».

47.ISO/IEC «Информационная технология Методы и средства обеспечения безопасности - Критерии оценки безопасности ИТ – Часть 1: Введение и общая модель». ISCMEC JTC 1/SC27 №2738, 02.2001 г.

48.ISO/IEC 15408 3: 1999 «Информационная технология - Методы и средства обеспечения безопасности - Критерии оценки безопасности ИТ -Часть 3: Гарантийные требования безопасности».

49.IxChariot: Тестирование в условиях атак отказа в обслуживании (DOS) [Електронний ресурс]. - Режим доступу до ресурсу: http://ixchariot.ru/tests/.

50.Kali Linux. // http://www.kali.org/

51.Kim Peter. The Hacker Playbook 3: Practical Guide To Penetration Testing / Securety planet LLC 2018 359 p.

52.Kniberg Henrik Scrum and XP from the Trenches - 2nd Edition / Henrik Kniberg - InfoQ, 2015. - 94p.

53.Krishnan M. Soumya Software Development Risk Aspects and Success Frequency on Spiral and Agile Model / M. Soumya Krishnan // International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization). -2015. -Vol.3. -№1. -pp.301-310.

54.M.-F. Danca. Matlab code for Lyapunov exponents of fractional order systems / Marius-F. Danca, N.V. Kuznetsov // International Journal of Bifurcation and Chaos. – 2018. – Vol. 28, No. 05, 1850067, – 14 p.

55.O WASP Testing Guide v4. [Електронний ресурс]. – Режим доступу: https://www.owasp.org/index.php/ OWASP_Testing_Project (дата звернення: 21.09.2020)

56.PCI DSS. Requirements and Security Assessment Procedures. Version 3.2. [Електронний ресурс]. – Режим доступу: https://www.pcisecuritystandards.org/document_library (дата звернення: 21.09.2020)

57.Putu Adi Guna Permana Scrum Method Implementation in a Software Development Project Management/ Putu Adi Guna Permana // International Journal of Advanced Computer Science and Applications. - 2015. - Vol. 6. №9.-P. 199-205.

58.Semenov S. The concept definition of mathematical modelling of the secured information-telecommunication system with regard to conditions of the posterior uncertainty / S. Semenov, O. Dorokhov, D. Grynov // Transport and Telecommunication.-2013.-Vol. 14, №2.-P. 167-174.

59.Technical Guide to Information Security Testing and Assessment. Recommendations of the National Institute of Standards and Technology. Special Publication 800-115. [Електронний ресурс]. – Режим доступу: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf (дата звернення: 21.09.2020)

60.The Open Source Security Testing Methodology Manual (OSSTMM). [Електронний ресурс]. – Режим доступу: http://www.isecom.org/mirror/OSSTMM.3.pdf (дата звернення: 21.09.2020)

61.The Penetration Testing Execution Standard (PTES). [Електронний ресурс]. – Режим доступу: http://www.penteststandard. org/index.php/Main_Page (дата звернення: 21.09.2020)

62.https://uk.wikipedia.org/wiki Алгоритм_Нідлмана_—_Вунша

63.Zeng Y. Risk Management For Enterprise Resource Planning System Implementations in Project-Based / Y. Zeng // Firms : dis. for the degree of PHD, Maryland, 2010. - P. 210.

64.Zhang Peng. Security in Network Coding / Peng Zhang, Chuang Lin. - Springer, 2016.-98 p.

ДОДАТОК А
(Обов’язковий)

Таблиця А.1 - Основні кроки при моделюванні загроз

Крок	Діяльність	Опис
1	2	3
Крок 1	Визначення вимог щодо забезпечення безпеки додатка	Чітко визначені вимоги допомагають не тільки виконати цей крок, але й визначити обсяг зусиль для виконання наступних кроків
Крок 2	Створення загального уявлення про програму	Каталогізація основних характеристик програми та основних типів користувачів («акторів») допоможе ідентифікувати значущі загрози
Крок 3	Декомпозиція додатку	Розуміння механіки роботи аналізованого додатку допоможе у виявленні найбільш значущих загроз із відповідним рівнем деталізації
Крок 4	Ідентифікація загроз	Використовуючи інформацію, зібрану на кроках 2 і 3, можна ідентифікувати загрози, значимі для додатка, можливих сценаріїв його використання та контексту, у якому воно буде виконуватися
Крок 5	Ідентифікація вразливостей	Аналіз ланок додатку для визначення вразливостей у контексті ідентифікованих загроз допоможе сфокусуватися на тих областях, у яких найбільш ймовірно допустити помилки при реалізації

ДОДАТОК Б
(Обов’язковий)

Перелік публікацій за темою магістерської роботи

к.т.н., с.н.с. Мірошніченко О.В. (ВІКНУ)
к.т.н. Гавронський В.Є. (ХмНУ)
к.т.н. Гурман І.В. (ХмНУ)
к.т.н. Муляр І.В. (ХмНУ)
Жиров Б.Г. (ВІКНУ)

ПОБУДОВА ІНТЕЛЕКТУАЛЬНИХ ЗАСОБІВ ОЦІНЮВАННЯ ПАРАМЕТРІВ ЯКОСТІ ФУНКЦІОНУВАННЯ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖ

Розвиток сучасних телекомунікаційних систем і мереж обумовлює необхідність створення та надійного функціонування великого набору комунікаційних сервісів, що забезпечують ефективну роботу користувача із різнорідною інформацією у телекомунікаційній мережі. Разом з тим, неоднорідність мережевих інформаційних ресурсів, так і аудиторії користувачів, якій вся ця інформація адресована, ускладнює об'єктивний аналіз та моніторинг телекомунікаційних ресурсів. Тому актуально, що при експлуатації телекомунікаційних мереж повинен бути використаний широкий спектр сучасних та науково обґрунтованих технічних і технологічних рішень їх аналізу та моніторингу.

Якість телекомунікаційної послуги включає усю сукупність характеристик послуги, які визначають її здатність задовольнити встановлені чи очікувані потреби споживача послуг та визначають сукупність специфічних показників, які характеризують споживчі властивості цієї послуги та визначають її здатність задовольнити заявлені, встановлені та замовлені потреби споживача послуг.

Для оцінки якості функціонування телекомунікаційної мережі, виявлення та запобігання збоїв необхідно вести постійний статистичний контроль та аналіз стану мереж, що беруть участь у передачі інформації. Для оцінки якості телекомунікаційної мережі необхідна побудова системи характеристик якості мережі. Характеристики якості, якісні і кількісні, являють собою функції, задані на множині об'єктів, і приймають значення на деяких підмножинах - шкалах критеріїв.

Після проведення дослідження, виявлення та аналізу факторів, що впливають на побудову та процес функціонування телекомунікаційних мереж, подальший розвиток та удосконалення побудови інтелектуальних засобів оцінювання параметрів якості функціонування телекомунікаційних мереж пропонується вдосконалення методу оптимізації параметрів мережі, що дозволяє зменшити середнє значення помилки обробки проекційних даних для відновлення інформації, що надходить із вимірювальних ліній.

Розроблений метод оптимізації телекомунікаційних мереж на основі інтелектуальних технологій дозволяють забезпечити задану якість послуг при передачі інформаційних даних та підвищують ефективність використання обладнання і каналів передачі.

Метод тестування на проникнення, як засіб забезпечення безпеки корпоративної мережі

Гавронський В.Є., Муляр І.В.
Хмельницький національний університет

Необхідним елементом безпеки корпоративної мережі є тестування її програмного забезпечення як на етапах розробки, так і його використання. Проте не існує єдиної ефективної методики тестування на проникнення. Наприклад, існують криптографічні методи, що базуються на математичних алгоритмах та використовуються для шифрування даних з подальшою передачею їх відкритими каналами зв’язку. Додатковим ступенем захисту є приховання самого факту передачі інформації, наприклад, за допомогою методів цифрової стеганографії [1]. Іншим підходом до вирішення цієї задачі є використання в якості носіїв інформації хаотичних сигналів, які характеризуються широким неперервним спектром та високою інформаційною ємністю [2].

Таким чином, розроблення методу тестування безпеки програмного забезпечення для захисту інформації в корпоративній мережі є актуальним науковим завданням.

Провівши аналіз наявних методів тестування безпеки ПЗ корпоративних мереж, було виявлено певні недоліки стосовно їх застосування при умові підвищеної уваги до ПЗ у хакерів.

На основі аналізу сучасних методів забезпечення безпеки ПЗ та систематизації сучасних підходів у предметній галузі захисту даних сформульовано актуальне наукове завдання, що полягає в розробленні методу

тестування безпеки програмного забезпечення для захисту інформації в корпоративній мережі.

Для цього розроблено математичну модель кібератаки. Стохастичну мережу, що працює відповідно до розробленої моделі, представлено на рис. 1.

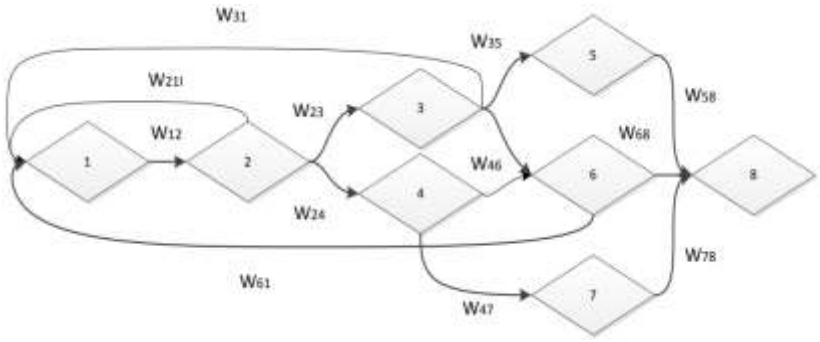


Рис. 1. GERT-мережа алгоритму генерації коду кібератаки несанкціонованого доступу

На рис. 1 перехід (1,2) характеризує операції вибору обладнання-жертви для злому. Переходи (2,3) (2,4) описують процес вибору методу атаки з урахуванням визначення операційної системи на вузлі-жертві (Windows або Linux, відповідно).

У цій мережі переходи зі стану в стан характеризують виконання операцій алгоритму генерації коду кібератаки та описуються випадковою величиною, що має відомий закон розподілу.

Так перехід (1,2) описує вибір обладнання в мережі для злому. Перехід (2,3) описує вибір методу атаки на комп'ютер жертви під ОС Windows. Перехід (2,4) описує вибір методу атаки на комп'ютер жертви під ОС Linux.

Якщо хакер не зміг вибрати метод атаки протягом визначеного часу або характеристики знайденого злочинного ПЗ не відповідають умовам і меті кібератаки, то здійснюються переходи (2,1) і (3,1) відповідно.

Пошук програмного забезпечення мережі Internet, скачування та установка його для ПК під ОС Windows представлений переходом (3,5). Аналогічну дію для ПК під ОС Linux відображає перехід (4,7). Якщо в мережі Internet відсутнє відповідне програмне забезпечення, то виконується кодування та налагодження ПЗ під ОС Windows (перехід 3,6) і Linux (перехід 4,6).

Якщо хакеру не вдалося провести операцію кодування та налагодження свого програмного забезпечення під час атаки, здійснюється перехід (6,1). Відкривання злочинного програмного забезпечення і ввід параметрів IP-серверу комп'ютера-жертви відображають переходи (5,8), (6,8) і (7,8).

Проведені дослідження показали, що в складних GERT-мережах з можливими циклами відсутні прості методи знаходження особливих точок функції. Це пов'язано з тим, що для знаходження особливих точок необхідно вирішувати нелінійні рівняння, і чим складнішою є структура GERT-мережі, тим складнішим є і вихідне рівняння [3].

Синтез трас – це наступний етап розробленого методу. Вихідними даними для алгоритму синтезу трас «Синтез» виступає набір графів $\Gamma = \{G_1, G_2, ..., G_M\}$, таких, що всі вони споріднені до деякої траси t

Алгоритм складається з двох кроків. На першому кроці виконується розбиття та синтез базових блоків (алгоритм «Базовий блок»). На другому кроці новостворені базові блоки з'єднуються ребрами (алгоритм «Синтез ребер»).

Спочатку враховуються внутрішні команди управління, обумовлені операторами BRANCH. Прямим проходом по операторах цього блоку будується орієнтований ациклічний граф, що описує вирази, які зустрічаються в цьому базовому блоці. Листові вершини в такому графі відповідають операторам INIT.

При побудові такого графа паралельно підтримується хеш-таблиця, ключі в якій відповідають виразам. Під час підрахунку хешу враховується код операції, а також хеші підвиразів. Тоді доповнення графа при перегляді чергового оператора зводиться до перевірки знаходження еквівалентного виразу в хеш-таблиці. Якщо такий вираз знайдено, то нові вершини не створюються. В іншому випадку необхідно створити нову вершину та додати вихідні ребра підвиразів, якщо такі в даному операторі є.

Після того, як граф побудовано, можна виключити повторне обчислення підвиразів: послідовно проглядаються оператори та виключаються ті, які обчислюють підвирази повторно. Слід зазначити, що додатково необхідно враховувати залежності, які проходять через біти слова стану: якщо будь-яка операція виставляє деякий біт, а інша операція, розташована далі, його читає, то першу з них виключати не можна.

У сукупності з уже наявними в середовищі можливостями реалізовані програмні компоненти дозволили повною мірою проводити запропоновану процедуру виділення алгоритму, у тому числі й ітеративно, з поповненням набору розглянутих трас в процесі аналізу.

Основною відмінністю розробленого методу є можливість його використання в ітеративному сценарії, коли в розгляд додаються нові траси за умови відсутності обмежень на природу аналізованого коду. Це дає можливість отримати уявлення про природу динамічної модифікації коду програми за допомогою побудови її еволюційного графа, розміри якого можуть розглядатися як одна з метрик складності програми.

Таким чином, розроблено математичну GERT-модель процесу генерації коду кібератаки несанкціонованого

доступу. Запропонована математична модель відрізняється від відомих урахуванням у процесі математичної формалізації GERTмережі основних етапів генерації коду для операційних систем Windows або Linux з можливістю пошуку сучасних рішень у мережі Інтернет. Модель може бути використано для дослідження основних етапів генерації коду кібератаки з метою вироблення практичних рекомендацій протидії процесу несанкціонованого доступу до ресурсів корпоративної мережі.

Література

1. Абазина Е.С. Цифровая стеганография: состояние и перспективы / Е.С. Абазина, А.А. Ерунов // Системы управления, связи и безопасности. – 2016. – № 2. – С. 182–201.

2. Колесов В.В. Применение дискретных хаотических алгоритмов в широкополосных телекоммуникационных системах / В.В. Колесов, А.И. Полубехин, Е.П. Чигин, А.Д. Юрин // Вестник СибГУТИ. – 2016. – № 3. – С. 77–92.

3. Барабаш О. В. Методи пошуку оптимальних маршрутів графа структури розгалуженої інформаційної мережі за заданим критерієм оптимальності при різних обмеженнях / О. В. Барабаш, І. П. Саланда, А. П. Мусієнко // Наукові записки Українського науково-дослідного інституту зв'язку. -К.: УНДІЗ, 2016. - №2 (42). - С 99-106.

УДК 621.391.01

М.О. СЛОБОДЯН, А.А. ТАРАНЧУК, В.Є. ГАВРОНСЬКИЙ
Хмельницький національний університет

ГЕНЕРУВАННЯ ШИРОКОСМУГОВИХ ХАОТИЧНИХ СИГНАЛІВ ДЛЯ ПРИХОВАНОЇ ПЕРЕДАЧІ ІНФОРМАЦІЇ В ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМАХ

Стаття присвячена математичному моделюванню генератора широкосмугових хаотичних коливань на основі класичної нелінійної динамічної системи Лоренца. На базі двох синхронно зв'язаних динамічних генераторів Лоренца була побудована модель системи передачі інформації з хаотичним маскуванням довільного вузькосмугового сигналу широкосмуговим хаотичним сигналом. Чисельний розв'язок диференціальних рівнянь системи та амплітудний спектр вихідного сигналу хаотичного генератора було розраховано за допомогою програмного забезпечення Matlab. З метою дослідження зміни динамічного режиму в залежності від параметрів моделі було розраховано спектр показників Ляпунова та побудовано діаграму біфуркацій. Модель системи синхронізації зв'язаних динамічних систем та модель генератора вузькосмугових сигналів з довільним типом модуляції були побудовані в середовищі імітаційного моделювання Simulink.

Ключові слова: *хаос, хаотичне маскування, динамічні системи, синхронізація.*

Вступ

Однією з головних вимог, що ставиться до телекомунікаційних систем та мереж, є інформаційна захищеність. Криптографічні методи, що базуються на математичних алгоритмах, використовуються для шифрування даних з подальшою передачею їх відкритими каналами зв'язку. Додатковим ступенем захисту є приховання самого факту передачі інформації, наприклад, за допомогою методів цифрової стеганографії [1]. Іншим підходом до вирішення цієї задачі є використання в якості носіїв інформації хаотичних сигналів, які характеризуються широким неперервним спектром та високою інформаційною ємністю [2-5]. Пристрої, побудовані на основі відносно нескладних математичних моделей, здатні генерувати неперіодичні електромагнітні коливання складної форми та дозволяють керувати хаотичними режимами за рахунок малих змін параметрів системи. Серед методів введення інформації в хаотичні сигнали, окрім модуляції параметрів нелінійної системи генератора, в науково-технічній літературі запропоновано ряд таких підходів, як хаотичне маскування (англ. – chaotic masking), перемикання хаотичних режимів (англ. – chaos shift keying), нелінійне підмішування (англ. – nonlinear mixing) тощо [2, 6]. Таким чином, каналом зв'язку передається шумоподібний хаотичний сигнал, когерентний прийом якого здійснюється за рахунок синхронізації хаотичних систем з подальшою демодуляцією хаотичних коливань. Отже, для прийому та обробки хаотичних сигналів на приймальній стороні повинні бути повністю або частково відтворені електричні кола генератора передавача та власне динамічний режим його роботи, що можна розглядати як додатковий ступінь захисту при передачі інформації в телекомунікаційних системах.

Математична модель генератора хаотичних коливань

Розглянемо в якості математичної моделі генератора хаотичних коливань динамічну систему Лоренца, яка складається з трьох звичайних диференціальних рівнянь першого порядку:

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(r - z) - y; \\ \dot{z} = xy - bz \end{cases} \quad (1)$$

де σ, r, b – дійсні додатні параметри системи.

Система Лоренца є нелінійною динамічною системою, яка при певних значеннях параметрів σ, r, b має нетривіальні розв'язки складної форми та високу чутливість до початкових умов [7, 8].

Так-як праві частини рівнянь системи (1) не містять вільних членів, то система є однорідною. В результаті заміни $x \to -x$, $y \to -y$ не змінюється вигляд рівнянь системи (1), що є свідченням симетричності системи Лоренца.

Для системи (1) дивергенція фазового потоку:

$$div(\dot{x}, \dot{y}, \dot{z}) = -\sigma - 1 - b < 0, \tag{2}$$

тоді, згідно теореми Ліувіля, фазовий потік стискає деякий об'єм фазового простору $V(t)$ згідно наступного співвідношення:

$$V(t) = V(0)e^{-(\sigma+b+1)t}, \tag{3}$$

отже, системи Лоренца є дисипативною.

Система Лоренца має нульову точку рівноваги $M_0 = (0;0;0)$ при довільних значеннях параметрів, а при $r > 1$ ще дві відмінні від нуля точки рівноваги:

$$M_{1,2} = \left(\pm\sqrt{b(r-1)}, \pm\sqrt{b(r-1)}, r-1\right) \tag{4}$$

У випадку, якщо $0 < r < 1$, єдиною точкою рівноваги, що притягує всі траєкторії у фазовому просторі, є точка M_0. При досягненні параметра значення $r = 1$ відбувається вилкоподібна біфуркація, що супроводжується (при $r > 1$) втратою стійкості точки M_0 та появою пари стійких положень рівноваги $M_{1,2}$. Точки $M_{1,2}$ є стійкими вузлами при $1 < r < 1{,}345$ та стійкими фокусами при $1{,}345 < r < 24{,}737$. Розмах коливань у фазовому просторі відносно положень рівноваги збільшується із зростанням параметра r. Досягнувши значення $r \approx 13{,}927$ спостерігається перестроювання атрактора у фазовому просторі: при нульових початкових умовах, здійснивши оберт навколо однієї з точок рівноваги, траєкторія повернеться у початок координат. Далі, зі зростанням параметра r, в залежності від напрямку, траєкторія приходить в одну з точок $M_{1,2}$, гомоклінічні траєкторії переходять у граничні цикли, а розмах коливань зменшується. Досягнувши значення $r \approx 24{,}06$ відбувається наступне перестроювання атрактора: разом із стійкими точками $M_{1,2}$ у фазовому просторі виникає складна притягаюча структура, яка відповідає хаотичному режиму системи – «дивному атрактору» Лоренца. Точки $M_{1,2}$ втрачають стійкість після досягнення значення $r = r_k$. Для значень параметрів системи $\sigma = 10$ та $b = 8/3$, значення $r_k \approx 24{,}74$.

В загальному випадку значення параметра r_k при заданих σ та b визначається співвідношенням:

$$r_k = \frac{\sigma(\sigma + b + 3)}{\sigma - b - 1} \tag{5}$$

На рис. 1 показані характерні траєкторії системи (1) при різних значеннях параметра r для двох наборів початкових умов: I_1 (0,1; 1; 1) та I_2 (–0,1; –1; 1).

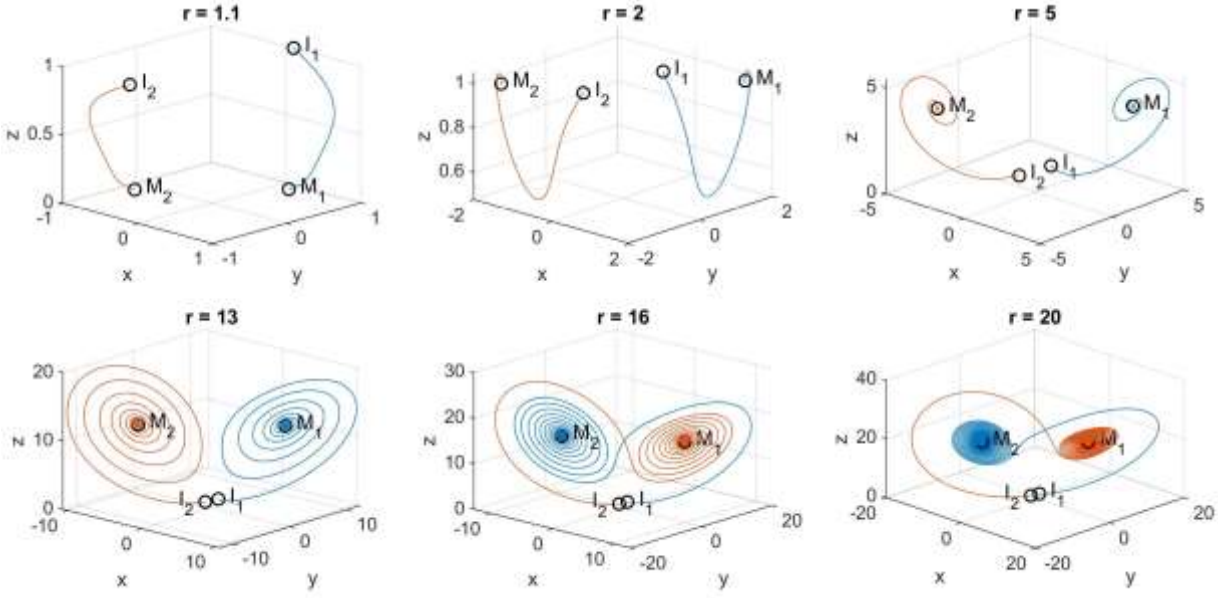
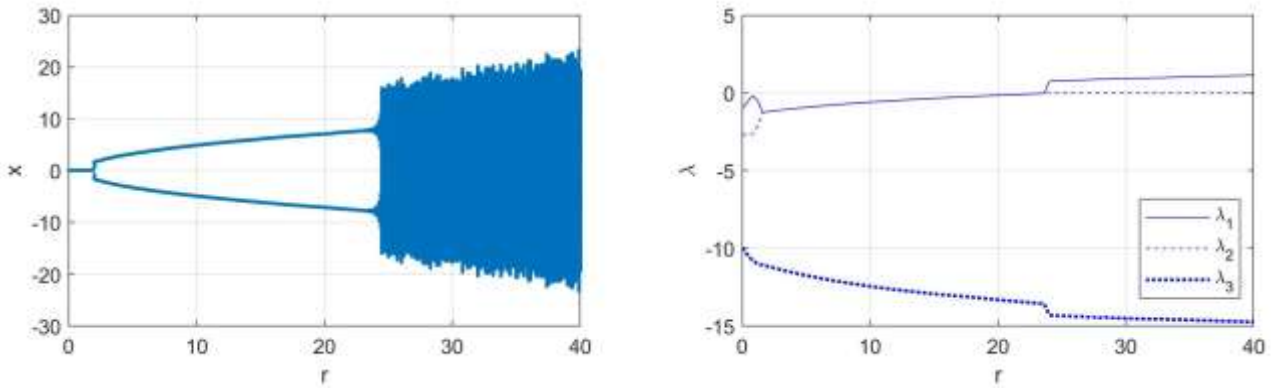


Рис. 1. Фазові траєкторії системи Лоренца для різних значеннях параметра *r*

Зміну динамічного режиму системи Лоренца за координатою x при зміні параметра $r \in [0{,}1;\ 40]$ ілюструє біфуркаційна діаграма, зображена на рис. 2, а.



а) б)

Рис. 2. Зміна динамічного режиму та ступеня хаотичності системи Лоренца в залежності від параметра *r*: біфуркаційна діаграма – а), спектр показників Ляпунова – б)

Для кількісної оцінки хаотичності системи були розраховані показники Ляпунова [9, 10] λ_k, $k = 1..3$, для різних значеннях параметра $r \in [0{,}1;\ 40]$. Спектр показників Ляпунова для вказаного діапазону значень параметра r зображено на рис. 2, б.

При значеннях параметрів $\sigma = 10$, $b = 8/3$, починаючи із значення $r \approx 24{,}74$, система Лоренца генерує хаотичні коливання, про що свідчить форма біфуркаційної діаграми [11] (рис. 2, а) та додатній знак старшого показника Ляпунова (рис. 2, б).

В якості набору параметрів, при яких системи (1) демонструє хаотичну поведінку було прийнято: $\sigma = 10$, $r = 35$, $b = 8/3$. Фазовий портрет атрактора, форму широкосмугового сигналу та його амплітудний спектр показано на рис. 3.

У хаотичному режимі система Лоренца генерує широкосмугові сигнали складної форми із неперервним спектром та високими кореляційними та ортогональними властивостями [8]. Висока інформаційна ємність та сильна залежність від початкових умов обумовлює використання сигналів такого типу в системах прихованої передачі інформації із шифруванням даних.

Синхронізація зв'язаних динамічних систем та маскування вузькосмугового сигналу широкосмуговим хаотичним сигналом

Головною проблемою, яку необхідно вирішити для ефективного використання хаотичних сигналів для передачі інформації в телекомунікаційних системах є задача синхронізація хаотичних генераторів на передавальній та приймальній сторонах – «ведучої» та «веденої» систем відповідно (див. рис. 4, б).

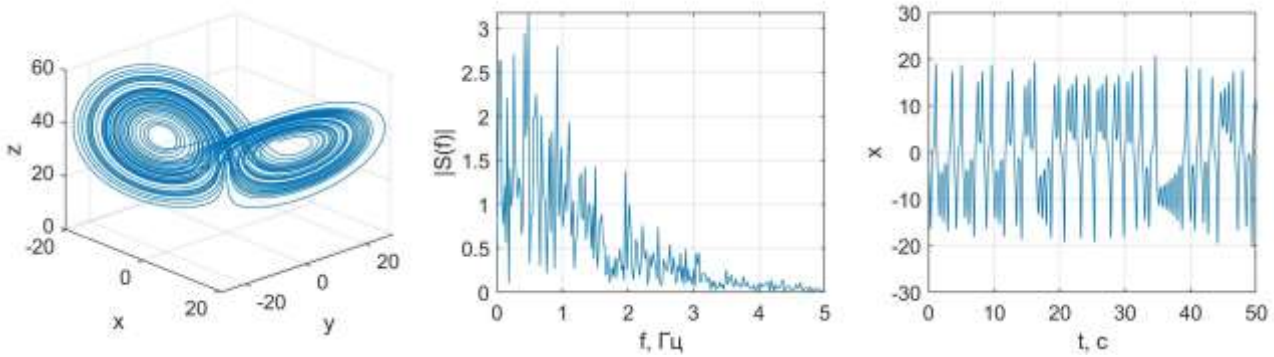
Розглянемо пару зв'язаних систем Лоренца з однаковими значеннями параметрів σ, r та b, що описується наступною системою рівнянь:

$$\begin{cases} \dot{x}_1 = \sigma(y_1 - x_1) \\ \dot{y}_1 = x_1(r - z_1) - y_1 \\ \dot{z}_1 = x_1 y_1 - b z_1 \\ \dot{x}_2 = \sigma(y_2 - x_2) \\ \dot{y}_2 = x_2(r - z_2) - y_2 \\ \dot{z}_2 = x_2 y_2 - b z_2 \end{cases} \tag{6}$$

Синхронізація цих двох систем можлива якщо існують фазові траєкторії $\mathbf{U}_1$ та $\mathbf{U}_2$ такі, що при $t \to \infty$ відстань між траєкторіями $\delta \to 0$, тобто

$$\lim_{t \to \infty} \mathbf{U}_2 = \mathbf{U}_1 \tag{7}$$

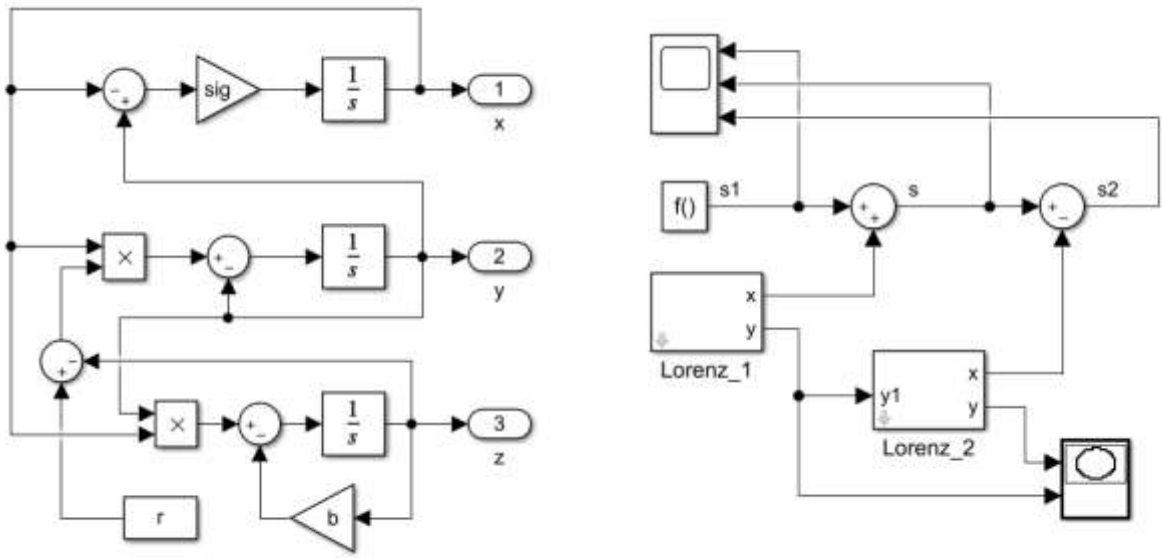
Крім того, рух системи (6) цією траєкторією повинен бути стійким по відношенню до завад. Необхідною умовою цього є від'ємне значення старшого показника Ляпунова «веденої» системи [7].



а) б) в)

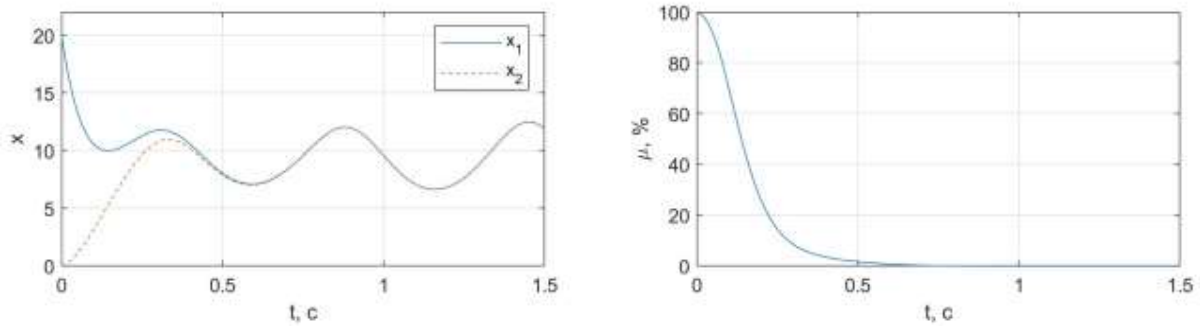
Рис. 3. Хаотичний режим системи Лоренца для набору параметрів $\sigma = 10$, $r = 35$, $b = 8/3$: атрактор у фазовому просторі – а), амплітудний спектр сигналу координати x – б), часовий графік сигналу x

Моделювання процесу синхронізації зв'язаних систем Лоренца було виконано засобами MATLAB/Simulink.
Комп'ютерна Simulink-модель, що складається з трьох інтеграторів, які призначені для чисельного розв'язку динамічної системи Лоренца (1), зображена на рис. 4, а. Вихідними сигналами системи є часові значення координат x, y, z.
Система синхронізації двох зв'язаних систем Лоренца, *Lorenz_1* та *Lorenz_2*, показана на рис. 4, б. Для реалізації синхронного відгуку системи *Lorenz_2* використовується сигнал y «ведучої» системи *Lorenz_1*, а сигнал x слугує носієм інформаційного сигналу s_1.
Перехідний процес синхронізації динамічних систем Лоренца для сигналу x показано на рис. 5, а. На рис. 5, б зображено графік відносної похибки синхронізації μ.



а) б)

Рис. 4. Simulink-модель динамічної системи Лоренца – а), та системи синхронізації двох зв'язаних систем Лоренца з хаотичним маскуванням інформаційного сигналу – б)



а) б)

Рис. 5. Процес синхронізації зв'язаних динамічних систем Лоренца: часові діаграми вихідних сигналів x_1 та x_2 зв'язаних систем – а), відносна похибка синхронізації – б)

Представимо довільний вузькосмуговий сигнал у вигляді:

$$s\left[u(t),t\right]=S\left[u(t),t\right]\cdot\sin\left(\omega_0 t+\Psi\left[u(t),t\right]\right), \tag{8}$$

де $S\left[u(t),t\right]$ – амплітуда сигналу,

$\Psi\left[u(t),t\right]=\Phi\left[u(t),t\right]+\varphi_0$ – повна фаза сигналу.

Нехай $u(t)$ – повільно зростаючою функцією часу, тоді при диференціюванні вважатимемо $u(t)=u=const$.

Продиференціювавши вираз (8) двічі по часу, ввівши заміни $\sin\Psi=s/S$ та $\cos\Psi=\left(\dot{s}-\dot{S}\,s/S\right)\Big/S\dot{\Psi}$, отримаємо лінійне диференціальне рівняння зі змінними коефіцієнтами:

$$\ddot{s}-\left[\frac{\ddot{\Psi}}{\dot{\Psi}}+\frac{2\dot{S}}{S}\right]\dot{s}+\left[\dot{\Psi}^2+\frac{1}{S}\left(\frac{2\dot{S}^2}{S}+\frac{\dot{S}\ddot{\Psi}}{\dot{\Psi}}-\ddot{S}\right)\right]s=0 \tag{9}$$

Форма сигналу $s(t)$, який є розв'язкам рівняння (9), залежить від типу модуляції, яка в свою чергу, в рамках описуваної моделі, задається законом зміни амплітуди $S\left[u(t),t\right]$ та повної фази $\Psi\left[u(t),t\right]$.

Наприклад, для сигналу з частотною модуляцією (ЧМ):

$$S\left[u(t),t\right]=S_0=const, \tag{10}$$

$$\Psi\left[u(t),t\right]=\omega_0 t+m_{чм}\int_0^t u(t)\,dt+\varphi_0, \tag{11}$$

де ω_0 – частота несучого коливання;

$m_{чм}$ – індекс модуляції;

φ_0 – початкова фаза.

Комп'ютерна Simulink-модель генератора вузькосмугових сигналів, побудована згідно рівняння (9) представлена на рис. 6. Вхідними сигналами для моделі є амплітуда S та повна фаза Ψ.

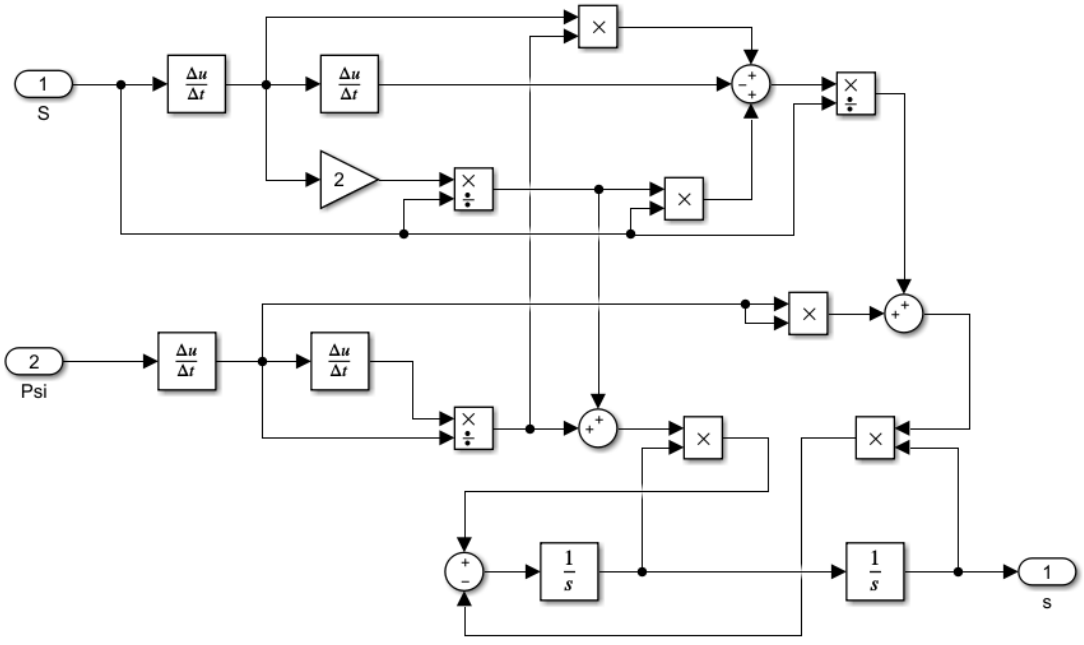
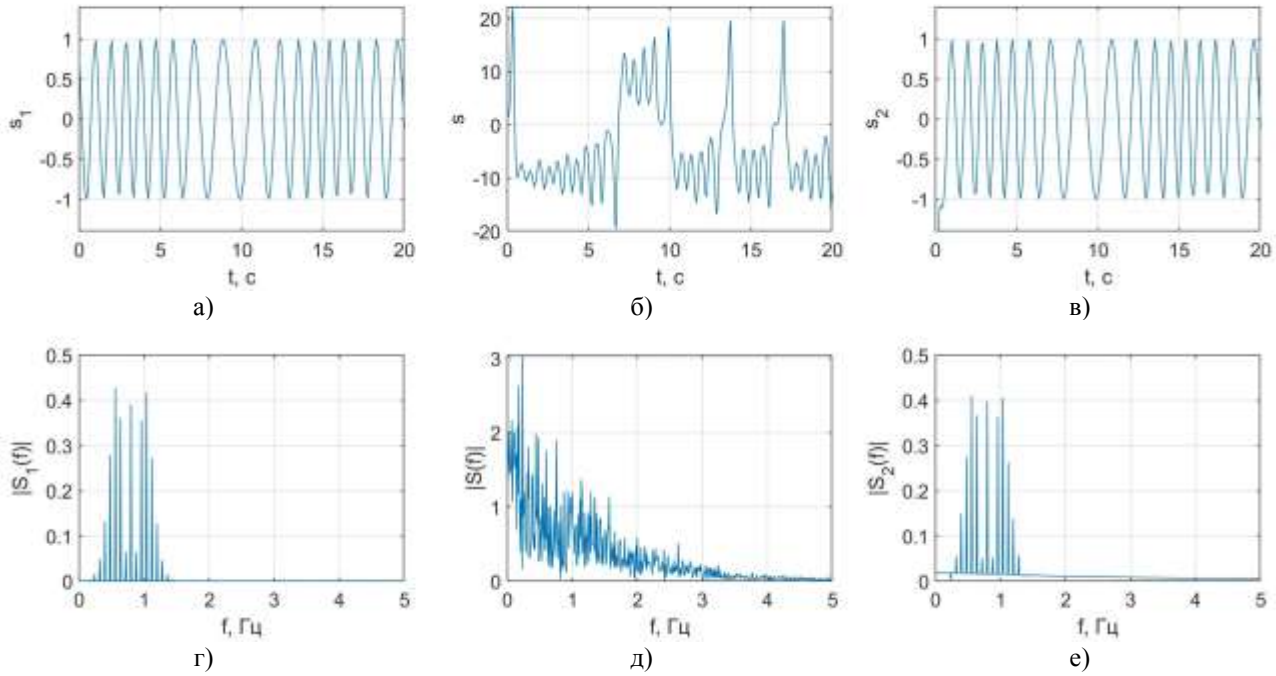


Рис. 6. Simulink-модель генератора вузькосмугових сигналів із заданою формою амплітуди та фази

Часові залежності та амплітудні спектри вихідних сигналів систем передачі та прийому, а також сигналу в каналі зв'язку, показано на рис. 7.



а) б) в)

г) д) е)

Рис. 7. Хаотичне маскування тестового вузькосмугового ЧМ-сигналу:
сигнал на вході системи передачі та його амплітудний спектр – а), г); сигнал, переданий каналом зв'язку та його спектр – б), д);
сигнал на виході приймальної системи та його спектр – в), е)

Тестовий ЧМ-сигнал s_1, отриманий на виході моделі, показаної на рис. 6, адитивно підмішується до вихідного хаотичного сигналу x_1 системи *Lorenz_1* (рис. 4, б) та разом із сигналом синхронізації y передається каналом зв'язку (рис. 7, б). На приймальній стороні тестовий сигнал виділяється шляхом віднімання від прийнятого хаотичного сигналу s сигналу x_2, згенерованого системою *Lorenz_2* (рис. 4, б), що синхронізується сигналом y. По завершенню перехідного процесу, після встановлення режиму синхронізації, виділений сигнал s_2 (рис. 7, е) співпадає з оригінальним сигналом повідомлення s_1 (рис. 7, а). Параметри систем *Lorenz_1* та *Lorenz_2* вважаються ідентичними, а канал зв'язку – ідеальним.

Висновки

1. Перспектива використання пристроїв із хаотичної динамікою в сучасних засобах телекомунікації обумовлена рядом факторів, серед яких висока інформаційна ємність, широкий спектр частот та конфіденційність передачі повідомлень. Можливість реалізації на базі одного пристрою великої кількості хаотичних режимів в перспективі дає можливість побудови багатоканальних систем передачі інформації. Сильна залежність від початкових умов та нестійкість фазових траєкторій дозволяє за рахунок малих впливів керувати динамікою хаотичних генераторів та здійснювати модуляцію з великою швидкістю.

2. Не дивлячись на простоту реалізації, метод хаотичного маскування має ряд суттєвих недоліків. Так, при наявності завад в каналі зв'язку інформаційний сигнал, потужність якого апріорі є нижчою порівняно із несучим хаотичним сигналом, стає співрозмірним із шумами каналу. Збільшення рівня інформаційного сигналу призводить до втрати конфіденційності, оскільки можливим стає несанкціонований перехват інформаційного повідомлення шляхом відфільтровування хаотичної складової. Таким чином, при проектуванні системи передачі, основаній на хаотичному маскуванні, необхідно визначити оптимальне співвідношення сигнал/хаос виходячи із оцінки можливого рівня шумів в каналі та потрібної якості передачі.

3. Описаний підхід може бути застосований для багатоканальної передачі вузькосмугових сигналів з кутовою модуляцією, наприклад, в безпровідних інфокомунікаційних системах з використанням кварцових сенсорів фізичних параметрів з модульованим міжелектродним зазором збудження п'єзорезонатора [12].

Література

1. Абазина Е.С. Цифровая стеганография: состояние и перспективы / Е.С. Абазина, А.А. Ерунов // Системы управления, связи и безопасности. – 2016. – № 2. – С. 182–201.

2. Земляный О.В. Передача информации на основе манипуляции спектром широкополосного хаотического сигнала / О. В. Земляный // Радиофизика и электроника. – 2015. – Т. 6(20), № 3. – С. 72–78.

3. Иванюк П.В. Хаотическое маскирование информационных сигналов с использованием генератора на базе системы Лю / П.В. Иванюк, Л.Ф. Политанский, Р.Л. Политанский, О.М. Элияшив // Технология и конструирование в электронной аппаратуре. – 2012. – № 3. – С. 11–17.

4. Пятін І.С. Конфіденційна система зв'язку / І.С. Пятін, В.І. Лужанський, Л.В. Карпова // Вісник

Хмельницького національного університету. Технічні науки. –2015. – № 1. – С. 207–212.

5. Колесов В.В. Применение дискретных хаотических алгоритмов в широкополосных телекоммуникационных системах / В.В. Колесов, А.И. Полубехин, Е.П. Чигин, А.Д. Юрин // Вестник СибГУТИ. – 2016. – № 3. – С. 77–92.

6. Агуреев К.И. Применение детерминированного хаоса для передачи информации / К.И. Агуреев // Известия ТулГУ. Технические науки. – 2014. – Вып. 11. Ч. 2. – С.197–212.

7. Дмитриев А.С. Динамический хаос: новые носители информации для систем связи / А.С. Дмитриев, А.И. Панас. – М. : Изд-во Физико-математической литературы, 2002. – 252 с.

8. Прикладне застосування теорії хаотичних систем у телекомунікаціях: монографія / [Ю.Я. Бобало, С.Д. Галюк, М.М. Климаш, Р.Л. Політанський]; Нац. ун-т «Львів. політехніка». – Львів: Коло, 2015. – 178 с.

9. Передерий Ю.А. Метод оценки спектра ляпуновских показателей по временной реализации / Ю.А. Передерий // Известия вузов. ПНД. – 2012. – Т. 20, вып. 1. – С. 99–104.

ДОДАТОК В

(Обов’язковий)

Презентація

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Гавронський Віталій

Метод тестування на проникнення, як засіб забезпечення безпеки корпоративної мережі

Науковий керівник
к.т.н., доцент Муляр І.В.

кафедра кібербезпеки та комп'ютерних систем і мереж

Тема: Метод тестування на проникнення, як засіб забезпечення безпеки корпоративної мережі

Мета магістерської роботи полягає у встановленні прийнятного рівня безпеки програмного забезпечення в корпоративній мережі шляхом розробки методів її тестування..
Об'єкт дослідження: процес тестування безпеки програмного забезпечення корпоративної мережі.
Предмет дослідження: моделі та методи тестування безпеки програмного забезпечення для захисту інформації в корпоративній мережі

Задачі досліджень у роботі формулюються наступним чином:

- проаналізувати необхідні вимоги до безпеки даних корпоративної мережі, існуючі методики тестування безпеки і чинники, що на неї впливають, встановити, як реалізується математичне моделювання тестування безпеки, вибрати напрямок дослідження;
- розробити математичну модель кібератаки на корпоративну мережу;
- удосконалити метод виділення алгоритму з двійкового коду для аналізування безпеки програмного забезпечення корпоративної мережі;
- розробити модель виділення множини векторів-ознак із загальними ознаками.

Наукова новизна роботи

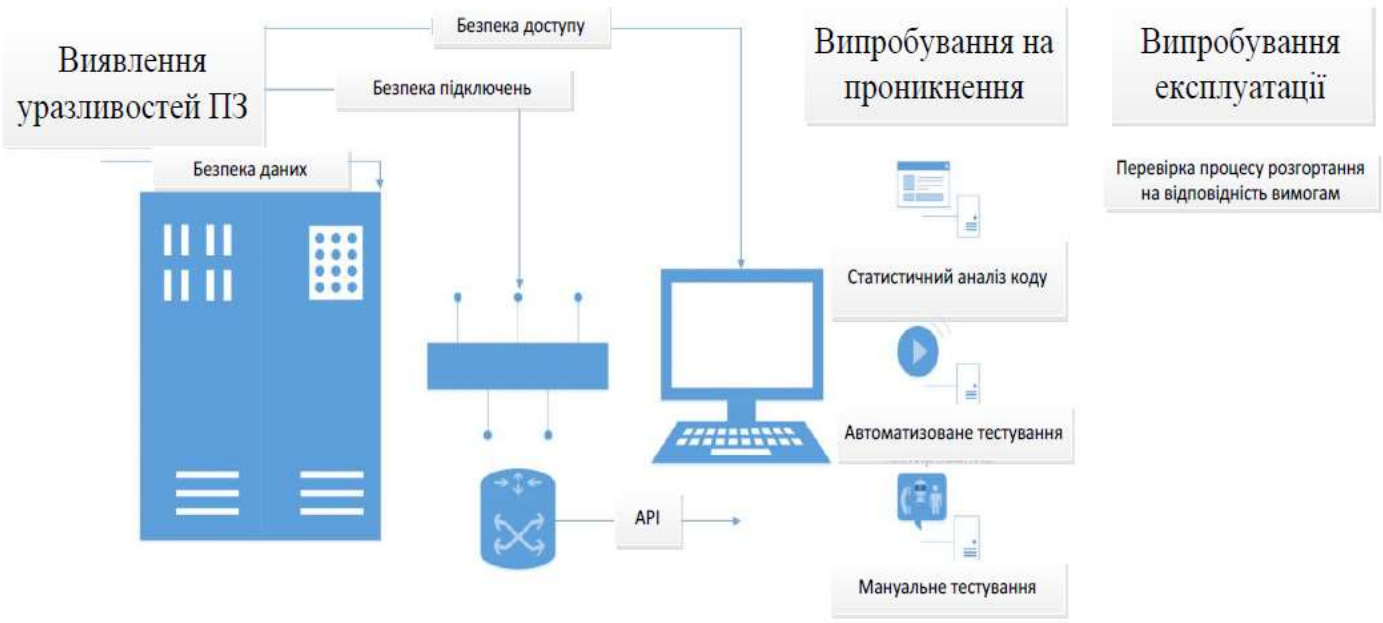
1. Розроблено математичну модель генерування кодів кібератаки на корпоративну мережу, яка дозволяє дослідити основні етапи генерування таких кодів та в подальшому надати практичні поради для захисту мережі від подібних атак.
2. Удосконалено математичну модель діагностики системи управління ресурсами мережі, що дозволило підвищити ефективність тестування безпеки програмних засобів корпоративної мережі.

Практична цінність. Проведене математичне моделювання показує можливість підвищення ефективності тестування безпеки програмних засобів корпоративної мережі до 5%. За допомогою методу виділення алгоритму з двійкового коду вдалося підвищити рівень безпеки ПЗ до 3 %.
Апробація роботи. Наукові результати і основні положення магістерської роботи доповідались і обговорювались на всеукраїнських науково-технічних конференціях.
Публікації. По темі магістерської роботи опублікована стаття у фаховому журналі, 2 - тези доповідей на всеукраїнських конференціях.

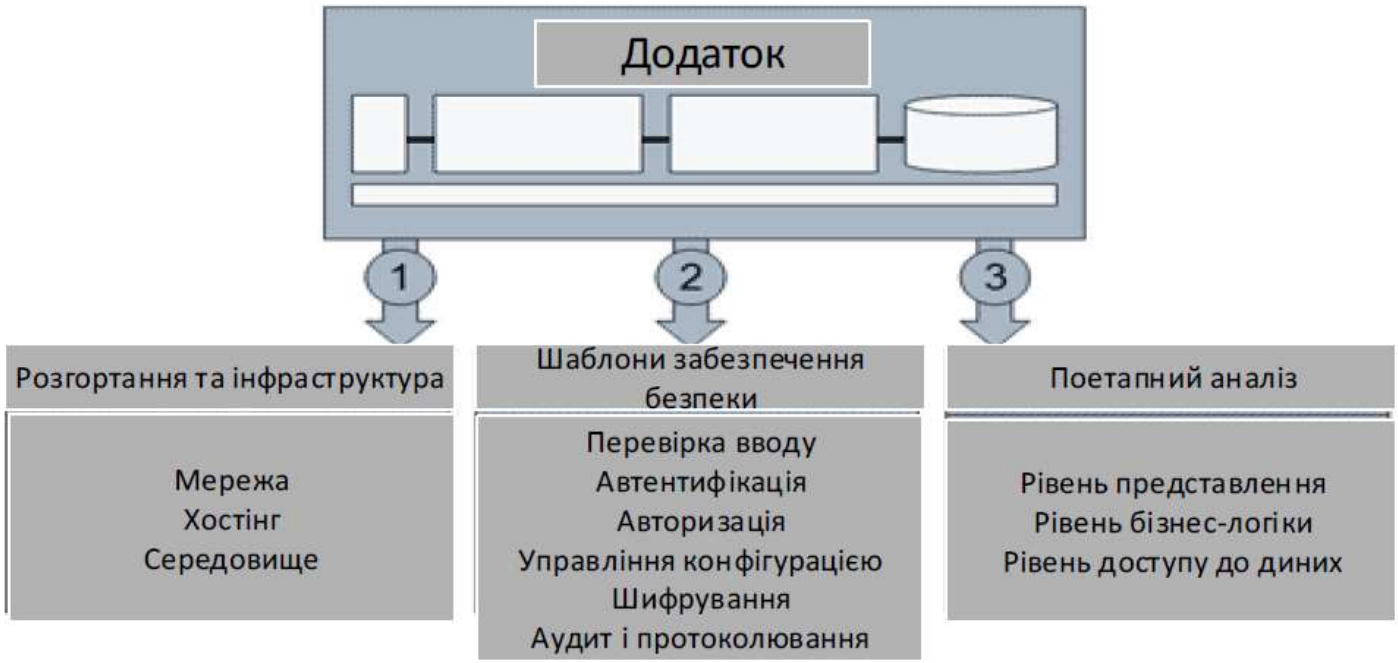
Класифікація методів тестування на проникнення

№	Ознака тестування	Види тестування
1.	За розташуванням програмно-апаратних засобів та пентестера відносно периметру організації-замовника	Зовнішне
		Внутрішне
2.	За обізнаністю пентестера про цільову систему	Білий ящик
		Чорний ящик
		Сірий ящик
3.	За обізнаністю технічних працівників організації-замовника про проведення тестування	Відкрите
		Приховане
4.	За характером заходів, що проводяться	Пасивне
		Агресивне
		Обережне
		Прораховане
5.	За повнотою виконання тестування	Повне
		Обмежене
		Фокусоване
6.	За видом інструментів, що використовуються	З застосуванням програмно-апаратних засобів
		З застосуванням методів соціальної інженерії та проникнення на контрольовану територію

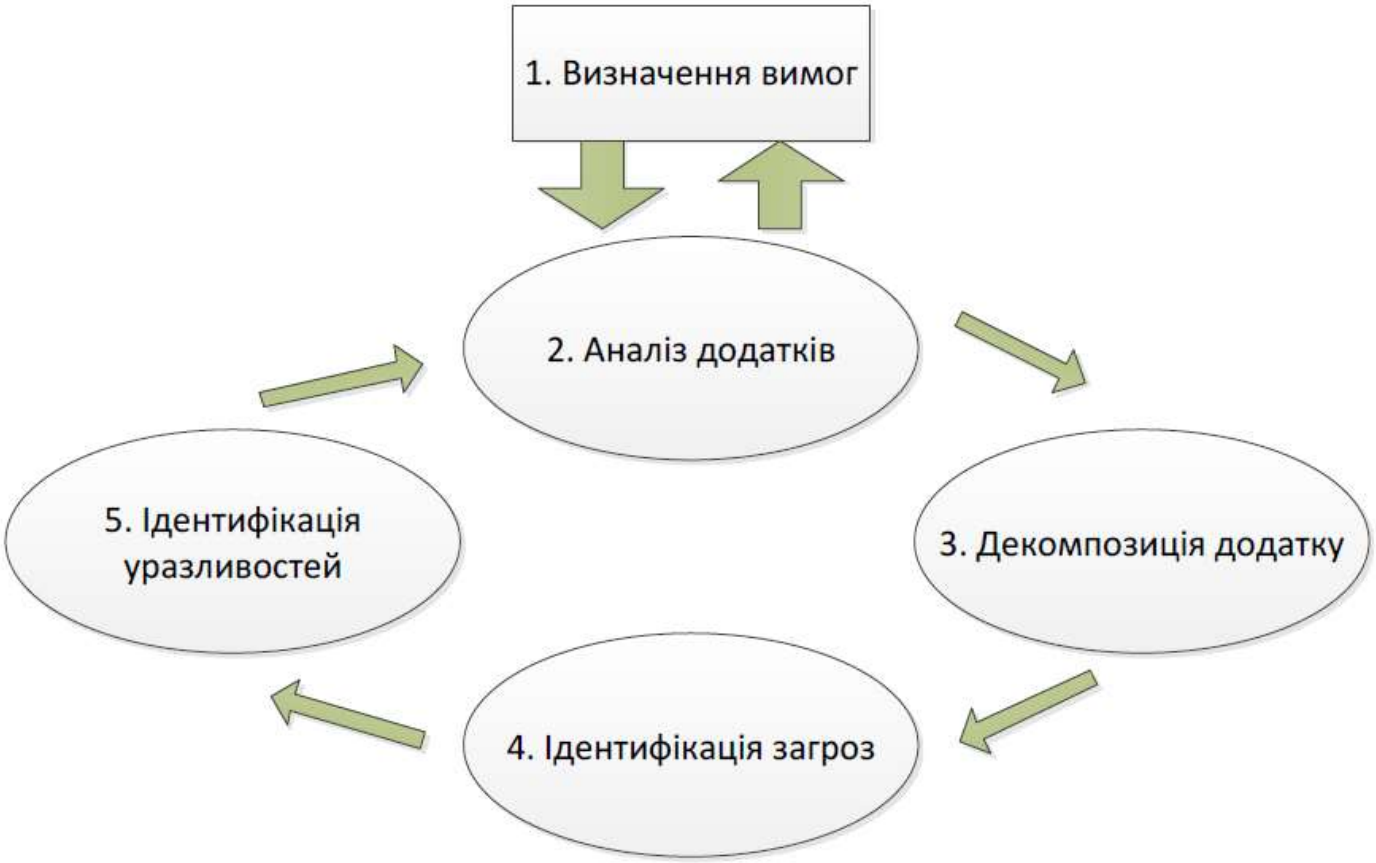
Загальна схема тестування



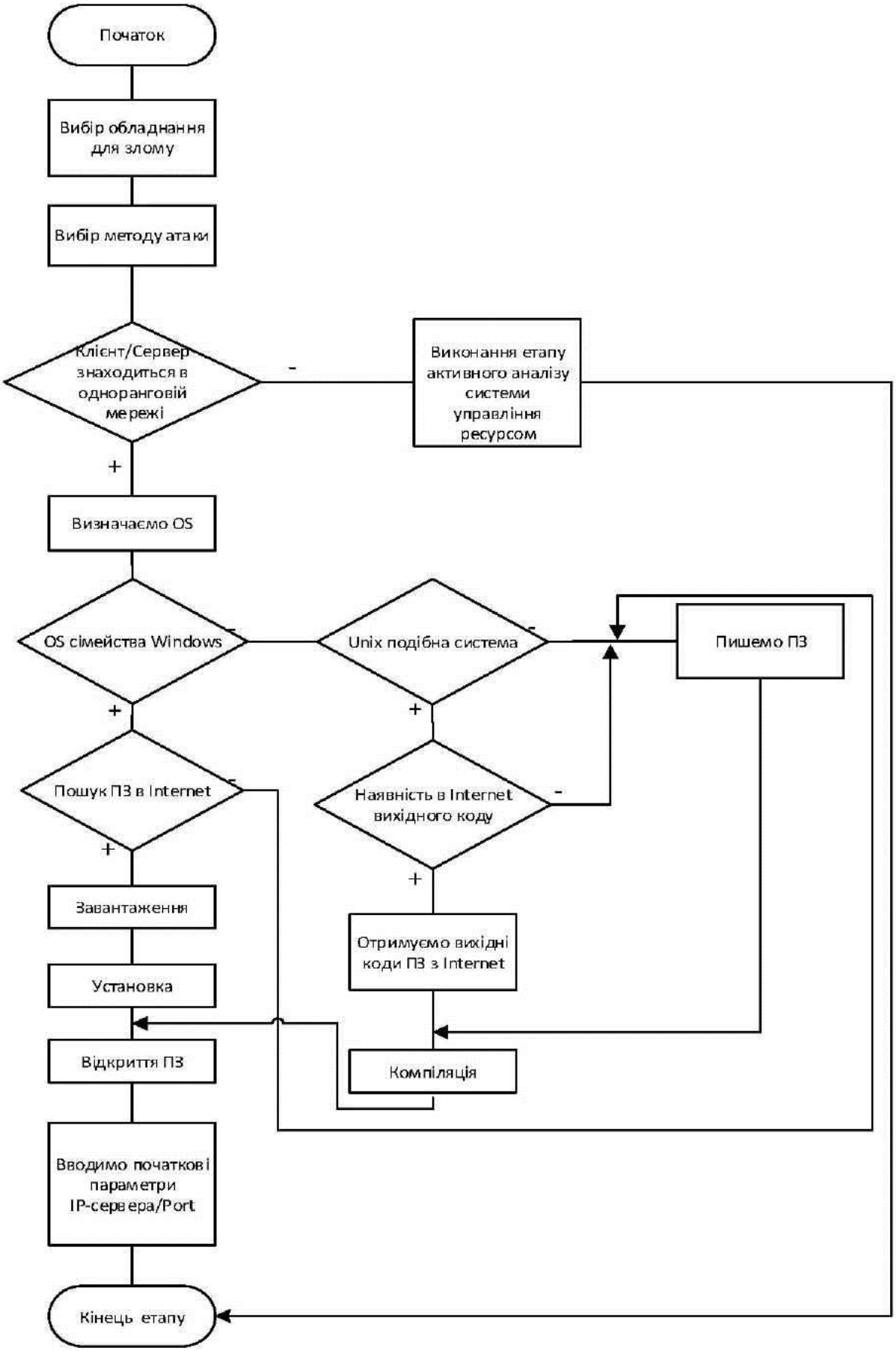
Основні аспекти перевірки архітектури та дизайну ПЗ



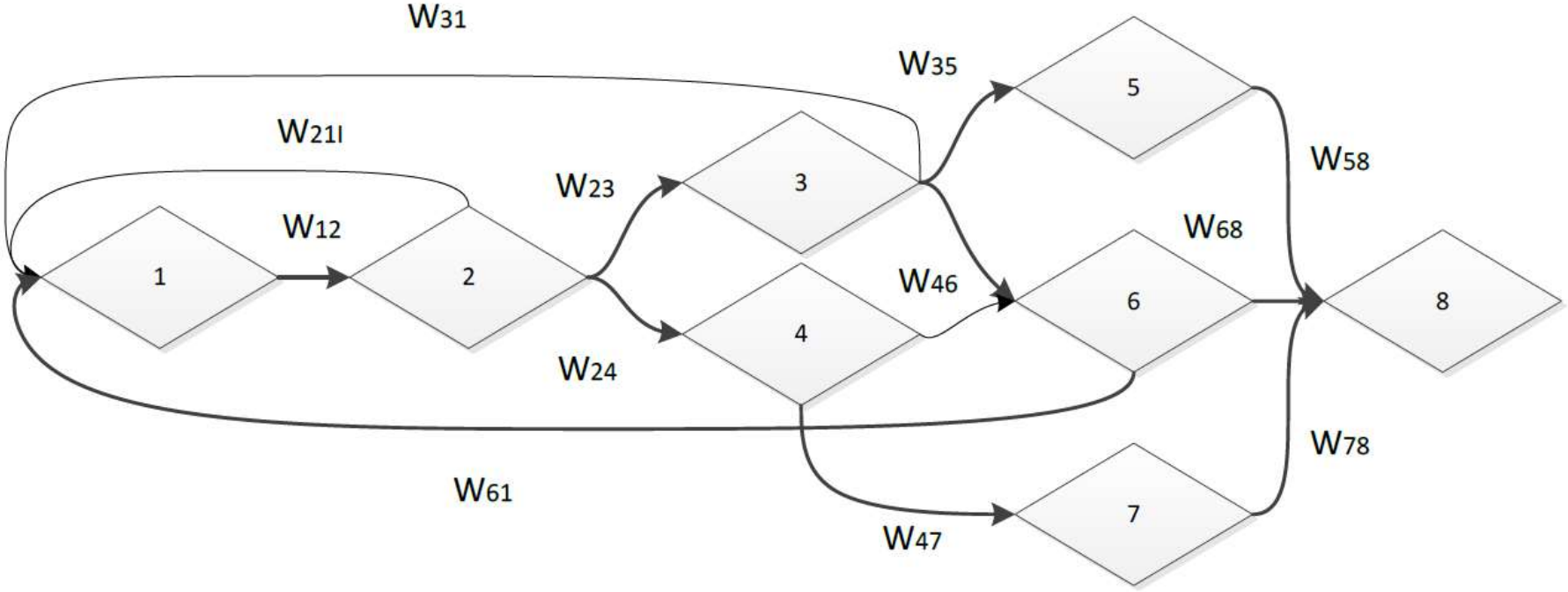
Процес моделювання загроз



Алгоритм проведення кібератаки



Стохастична мережа, побудована за алгоритмом проведення кібератаки

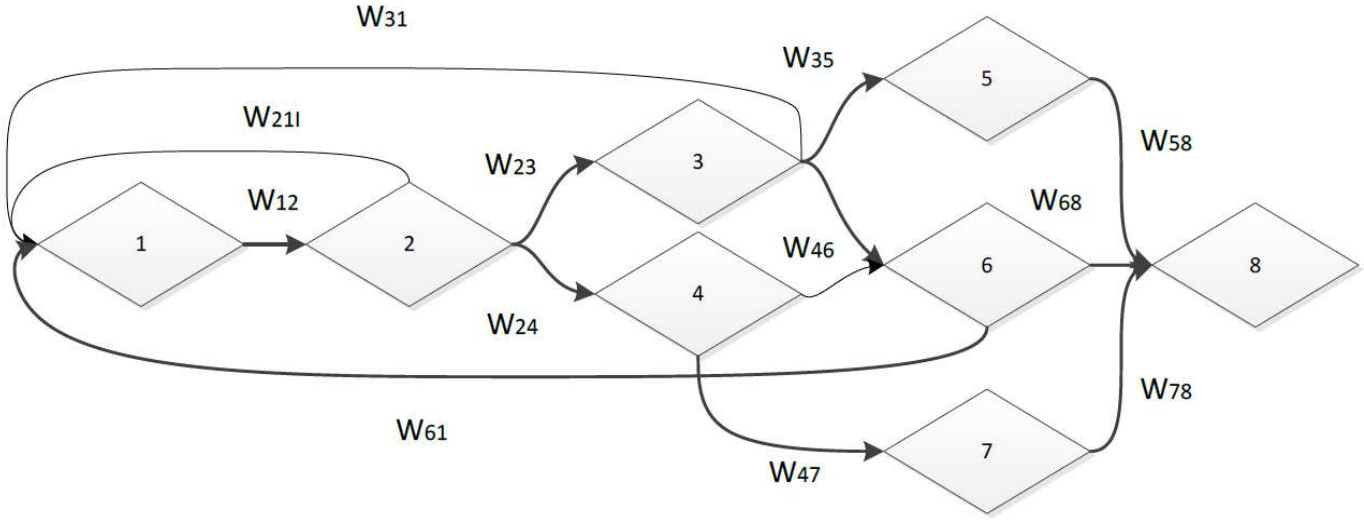


Так перехід (1,2) описує вибір обладнання в мережі для злому. Перехід (2,3) описує вибір методу атаки на комп'ютер жертви під ОС Windows. Перехід (2,4) описує вибір методу атаки на комп'ютер жертви під ОС Linux.

Якщо хакер не зміг вибрати метод атаки протягом визначеного часу або характеристики знайденого злочинного ПЗ не відповідають умовам і меті кібератаки, то здійснюються переходи (2,1) і (3,1) відповідно.

Якщо хакеру не вдалося провести операцію кодування та налагодження свого програмного забезпечення під час атаки, здійснюється перехід (6,1). Відкривання злочинного програмного забезпечення і ввід параметрів IP-серверу комп'ютера-жертви відображають переходи (5,8), (6,8) і (7,8).

Математична модель діагностики системи управління ресурсом корпоративної мережі



У приведеній моделі перший вузол характеризує початковий стан мережі перед впливом на неї алгоритму аналізу системи управління ресурсом. Другий вузол показує стан мережі, коли неможливо перехопити ресурси для її управління. Третій вузол відображає зворотну ситуацію, тобто можливість перехоплення ресурсів для управління мережею. Четвертий вузол показує на виправлення помилок. П’ятий вузол показує неповне виконання алгоритму аналізу системи управління ресурсом. Шостий вузол відображає факт повної перевірки корпоративної мережі.

Перехід (1-2) показує вплив на мережу описаного алгоритму, що призвів до позитивного результату для пентестера. Виправлення помилок відображає перехід (2-4). Перехід (4-1) забезпечує перехід в початковий стан мережі, який був до запуску тесту. Перехід (2-6) описує імплементацію шкідливого коду в корпоративну мережу. Перехід (1-3) показує вплив на мережу описаного алгоритму, що призвів до негативного результату для пентестера. Перехід (3-5) представляє оцінку повноти охоплення аналізованих впливом елементів, що були задіяні алгоритмом управління ресурсом корпоративної мережі..

Процес виділення алгоритму з двійкового коду

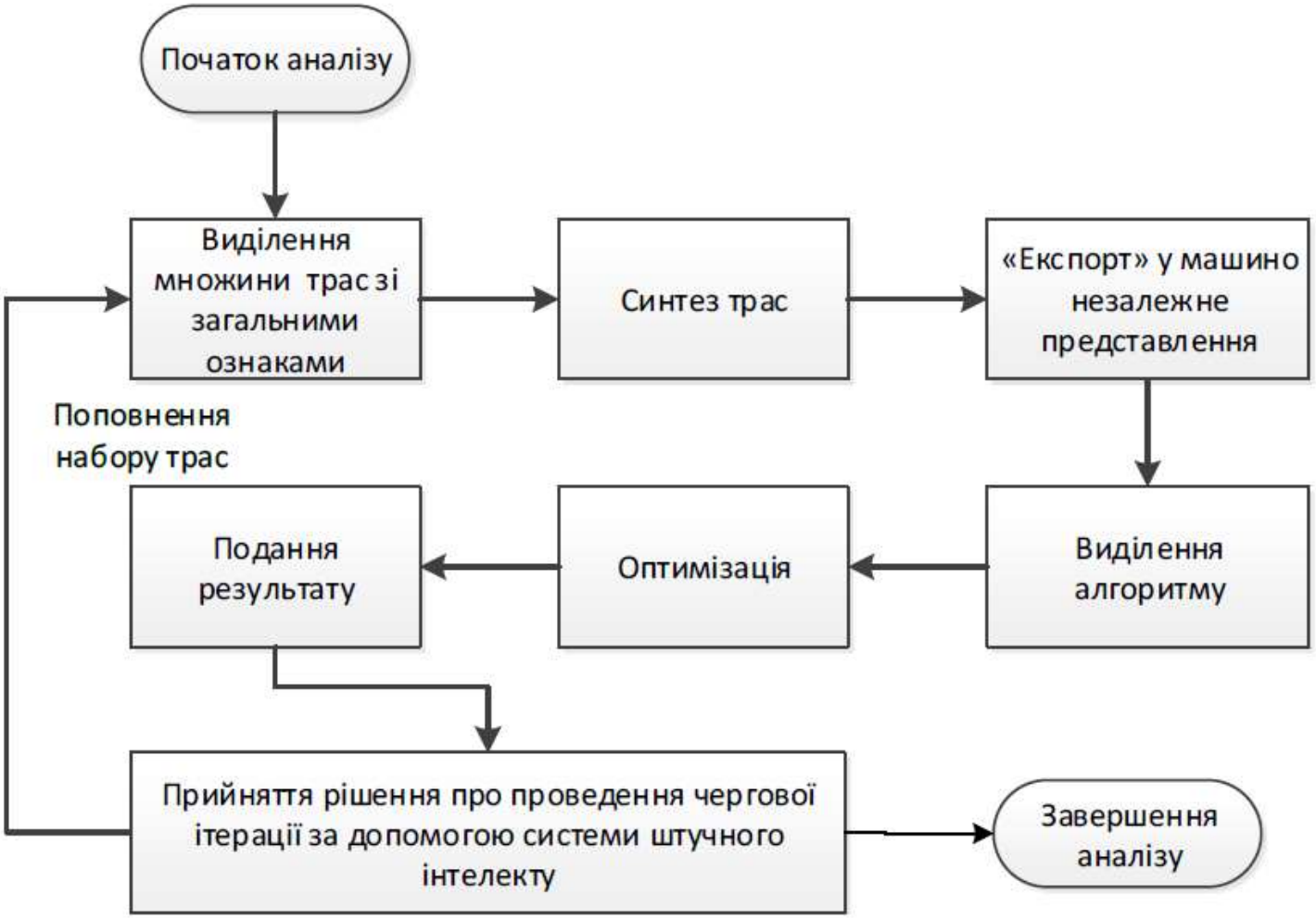
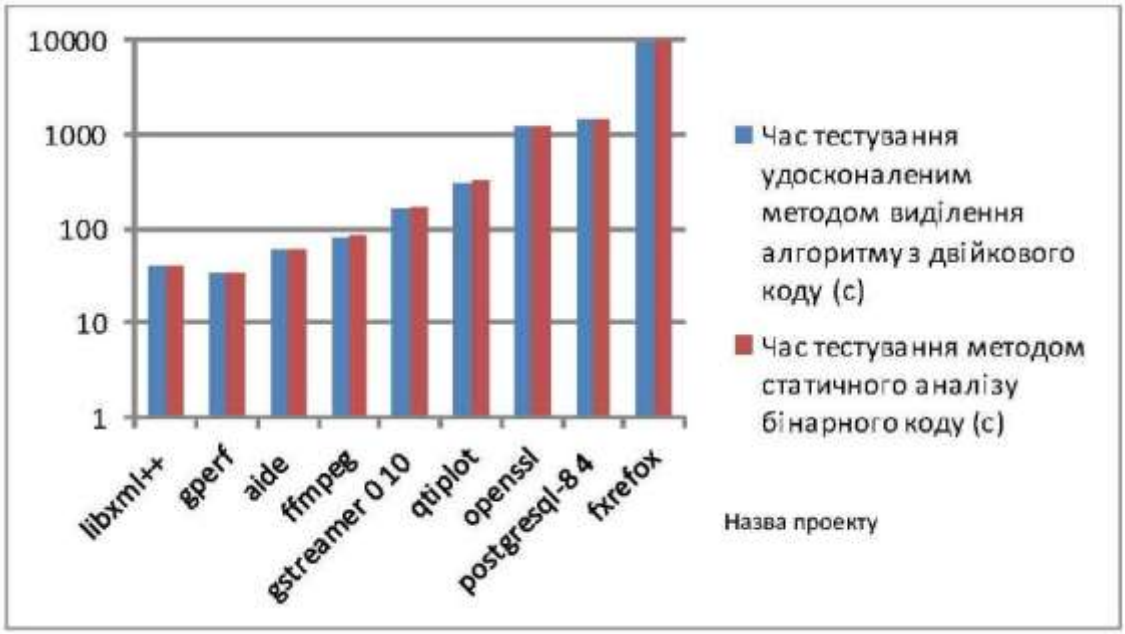
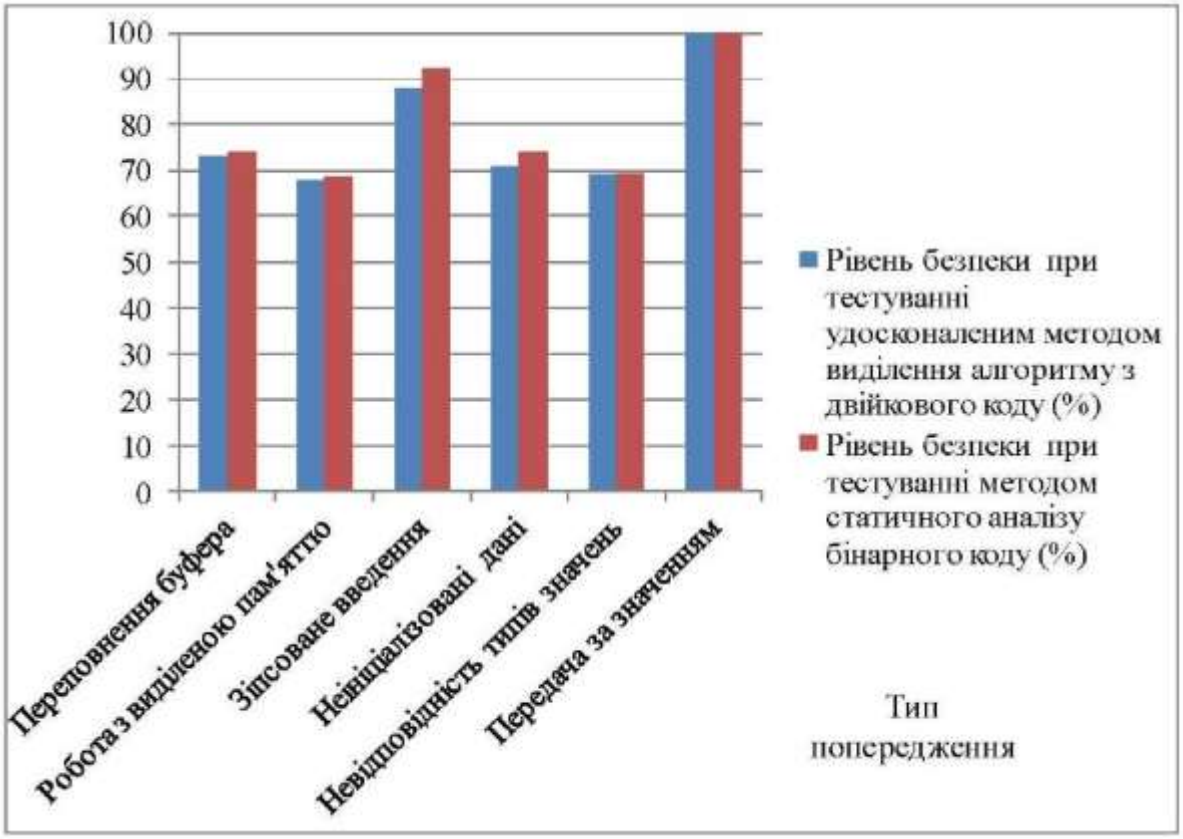


Схема пошуку алгоритму починається з виділення множини трас - векторів із загальними ознаками. Далі йде синтез трас, тобто інформація про об’єкт дослідження представляється за допомогою методу графів. Потім інформація експортується в машино-незалежне представлення та проводиться виділення алгоритму, тобто частини коду, що ми аналізуємо. Після цього йде оптимізація, для спрощення одержаного результату. Блок Подання результату забезпечує представлення коду у формі, придатній для пентестера, а також для подачі в систему штучного інтелекту для прийняття відповідного рішення про потребу проходження чергової ітерації або про завершення аналізу

Порівняння часу тестування безпеки даних удосконаленим та відомим методами



Порівняння за рівнем безпеки удосконаленого та відомого методу



Висновки

У магістерській роботі вирішено важливе наукове завдання щодо розроблення методу тестування безпеки даних для захисту інформації в корпоративній мережі. В результаті проведених досліджень можна зробити наступні висновки:

6. Аналіз основних вимог до безпеки даних корпоративної мережі показав, що наявні методики тестування безпеки не дозволяють забезпечити прийнятний рівень їх захисту. Дослідження математичних моделей технологій тестування безпеки дозволили зробити вибір і сформулювати наукове завдання магістерських досліджень.

7. Розроблено математичну модель кібератаки на корпоративну мережу, яка дозволяє дослідити основні етапи кібератаки та в подальшому надати практичні поради для захисту мережі від подібних атак.

8. Удосконалено математичну модель діагностики системи управління ресурсами мережі, що дозволило підвищити ефективність тестування безпеки програмних засобів корпоративної мережі.

9. Удосконалено метод виділення алгоритму з двійкового коду для аналізування безпеки даних корпоративної мережі.

10. Розроблено модель виділення множини векторів-ознак із загальними ознакам

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «магістр»

Магістр Гавронський В.Є.
Тема Метод тестування на проникнення, як засіб забезпечення безпеки корпоративної мережі
Галузь знань 12 – Інформаційні технології
Спеціальність 123 –Комп'ютерна інженерія

Обсяг кваліфікаційної роботи:

кількість листів креслень 10 ; кількість сторінок записки 72

1. Короткий зміст КР та прийнятих рішень В рамках магістерської роботи розроблено метод тестування безпеки програмного забезпечення для захисту інформації в корпоративній мережі, що дозволить в подальшому забезпечити безпеку мережі та унеможливить кібервтручання у її ресурси

2. Висновок про відповідність КР дипломному завданню кваліфікаційна робота у повній мірі відповідає поставленому завданню як в теоретичній, так і в практичній частині кваліфікаційної роботи.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі висвітлюється актуальність теми роботи, дається аналіз досліджуваної проблеми і обґрунтовується застосовуваний підхід до її вирішення, формулюються цілі і завдання дослідження, описується наукова новизна і практична значимість отриманих результатів. У першому розділі проведено аналіз основних вимог щодо безпеки ПЗ, зазначено на пріоритетність вимог безпеки програмного забезпечення та обов'язковість дотримання цих вимог на всіх етапах життєвого циклу ПЗ. Наступні розділи присвячені розробці моделей кібератаки, діагностики системи управління ресурсом мережі, розроблено схему пошуку алгоритму з двійкового коду для діагностики безпеки програмного забезпечення корпоративної мережі. Розглянуто питання застосування розробленого методу.

4. Позитивні сторони кваліфікаційної роботи Кваліфікаційна робота містить ряд інноваційних рішень, зокрема, розроблено математичну модель кібератаки на корпоративну мережу, яка дозволяє дослідити основні етапи кібератаки та в подальшому надати практичні поради для захисту мережі від подібних атак

5. Негативні сторони КР При розробці стохастичних мереж, автор не показує їх обмежень. Проте вони існують та пов'язані з ускладненням аналітичного опису при додаванні додаткових вузлів у таку мережу

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до теми дипломної роботи з дотриманням стандартів. В загальному графічне оформлення виконане на достатньому рівні. Пояснювальна записка відповідає нормам для її оформлення.

7. Відгук про роботу в цілому В загальному кваліфікаційна робота заслуговує позитивної оцінки. Весь матеріал дипломної роботи структурований, чіткий та послідовний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики дипломної роботи. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для досягнення поставленої задачі.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінку «відмінно», 4.85, А

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)

Підченко С.К., зав. каф. ТНІТ ХНУ, д.ф.-м.н., доцент

« 3 » 12 2020 . (підпис)

ХНУ. Зам. 54, тир. 6000, 2007

Tue Dec 01 15:12:42 EET 2020, Муляра І.В, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 9%**

ID: 81940 Название: Метод тестування на проникнення, як засіб забезпечення безпеки корпоративної мережі Добавлено в БД: 2020-12-01 Авторы: Гавронський В.Є. Руководители: Муляр І.В. Консультанты: Оппоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	66409	592	196 (0%)	4 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

UNICHECK

User name:
Kafedra kiberbezpeky

Check date:
02.12.2020 09:20:09 EET

Report date:
02.12.2020 09:27:08 EET

Check ID:
1005324366

Check type:
Doc vs Internet

User ID:
100005590

File name: MAG33_ю

Page count: 65 Word count: 9083 Character count: 68362 File size: 7.45 MB File ID: 1005447318

Text modifications detected (similarity score might be affected)

10% Matches

Highest match: 8.14% with Internet source (http://www.dut.edu.ua/uploads/p_[illegible].pdf)

No Library search was conducted

0% Quotes

Exclusion of quotes is off

Exclusion of references is off

0% Exclusions

No exclusions

Modifind

Text modifications detected. Find more details in the online report.

Suspicious formatting 23 Pages

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод тестування на проникнення, як засіб забезпечення безпеки корпоративної мережі

Автор: Гавронський Віталій Євгенович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: Програмування та захист комп'ютерних систем і мереж

Науковий керівник: Муляр Ігор Володимирович, к.т.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (дані зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (дані – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В цьому випадку мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, перешкоджувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
3) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 10% і адресується до 13 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи І.В. Муляр

Завідувач кафедри КібКСМ, гарант ОПП Ю.Л. Клюн

Дата: 03.12.2020

ЗАТВЕРДЖУЮ
Голова циклової комісії комп'ютерної інженерії
ХПК НУ «ЛП»
_____ Б.І. Гетьман
« 2 » грудня 2020 р.

АКТ

Про впровадження результатів магістерського дослідження Гавронського В.Є. на тему «Метод тестування на проникнення, як засіб забезпечення безпеки корпоративної мережі»

Науково-технічна комісія склала цей акт про те, що результати магістерського дослідження, а саме метод тестування безпеки програмного забезпечення комп'ютерної мережі, впроваджено для використання.

Очікується, що застосування цього методу забезпечить стабільну роботу мережі, а також унеможливить кібервтручання у власні ресурси. При цьому Хмельницький політехнічний коледж Національного університету «Львівська політехніка» отримає значний економічний ефект, оскільки не потребує купівлі і щорічного оновлення ліцензії.

Голова комісії _____ Гетьман Б.І.
(підпис) (прізвище та ініціали)