МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

САВЕНКО ОЛЕГ СТАНІСЛАВОВИЧ

УДК 004.75:004.49

ТЕОРІЯ ТА ПРАКТИКА СТВОРЕННЯ РОЗПОДІЛЕНИХ СИСТЕМ ВИЯВЛЕННЯ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ЛОКАЛЬНИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ

05.13.05 – комп'ютерні системи та компоненти

Автореферат
дисертації на здобуття наукового ступеня
доктора технічних наук

Львів – 2019

Дисертацією є рукопис.

Робота виконана в Хмельницькому національному університеті Міністерства освіти і науки України.

Наукові консультанти:

доктор технічних наук, професор
Саченко Анатолій Олексійович,
завідувач кафедри інформаційно-обчислювальних систем і управління Тернопільського національного економічного університету, м. Тернопіль;

доктор наук, професор
Марковський Георгій,
професор кафедри комп'ютерних наук Міссурійського університету наук і технологій, м. Ролла, штат Міссурі, США.

Офіційні опоненти:

доктор технічних наук, професор
Мельник Анатолій Олексійович,
завідувач кафедри електронно-обчислювальних машин Національного університету "Львівська політехніка", м. Львів;

доктор технічних наук, професор
Мухін Вадим Євгенійович,
професор кафедри математичних методів системного аналізу Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського", м. Київ;

доктор технічних наук, професор
Дрозд Олександр Валентинович,
професор кафедри інтелектуальних комп'ютерних систем та мереж Одеського національного політехнічного університету, м. Одеса.

Захист відбудеться *25 жовтня 2019 р.* о *14-00* годині на засіданні спеціалізованої вченої ради Д 35.052.08 у Національному університеті "Львівська політехніка" (79013, м. Львів, вул. С. Бандери, 12, ауд. 226 головного корпусу).

З дисертацією можна ознайомитися у бібліотеці Національного університету "Львівська політехніка" (79013, м. Львів, вул. Професорська, 1).

Автореферат розісланий *16 вересня 2019 р.*

Учений секретар спеціалізованої вченої ради, д.т.н., професор *Я. Т. Луцик*

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність роботи. Тенденції розвитку технологій створення і поширення зловмисного програмного забезпечення (ЗПЗ) демонструють активне розширення технічних можливостей таких засобів. Сучасне зловмисне програмне забезпечення представляє собою складні багатофункційні програмні системи та комплекси, які побудовані з використанням ефективних методів створення програмних засобів та методів поширення зловмисного коду.

Виявлення зловмисного програмного забезпечення здійснюється за допомогою різноманітних засобів. Ефективність та достовірність виявлення суттєво залежать від архітектури таких засобів, а також їх позиціонування та місця розміщення в комп'ютерних системах (КС), зокрема, і локальних мережах. Дослідження відомих антивірусних методів та засобів вказують, що реалізація нових принципів, моделей та методів виявлення конкретних типів зловмисного програмного забезпечення шляхом створення відповідних систем потребує подальшого розвитку. Перспективним напрямом досліджень створення ефективних систем виявлення ЗПЗ є мережні системи, які використовуються в локальних комп'ютерних мережах організацій та підприємств. Такі засоби виявлення мають розподілену архітектуру і, тому, покращення ефективності виявлення ЗПЗ можливе за рахунок залучення груп КС локальної мережі.

Розвитком теоретичних основ та науково-практичного застосування розподілених систем різного призначення активно займаються Бернс Б., Луцький Г. М., Марковський Г., Мельник А. О., Мухін В. Є.

Фахівцями, які проводять наукові дослідження в сфері виявлення ЗПЗ є Андерсон Б., Вінод П., Головко В. А., Дудикевич В. Б., Дейгон Д., Дітріх Д., Дрозд О. В., Ентонакакіс М., Елаєн К., Ішайз Х., Іслахі М., Карпінський М. П., Касперський Є. В., Котенко І. В., Крістодонеску Д., Лі З., Максимович В. М., Малан Д. Дж., Мартінсен Є. А., Махавер Д. К., Подловченко Р. І., Рувінська В. М., Погребенник В. Д., Рад Б. Б., Сочор Т., Сцор П., Саченко А. О., Сиротинський О. І., Тодерічі А. Г., Собейкіс В. Г., Целік З., Хольц Т., Хюнсанг Чоі та ін.

У сучасних системах виявлення ЗПЗ застосовуються методи та засоби, що використовують знання про його функціонування та поведінку. Дослідження функціонування ЗПЗ у різних КС локальної мережі розподіленими системами виявлення дає змогу здійснити більш детальний аналіз програмних об'єктів та процесів на наявність ЗПЗ за рахунок залучення інших КС. Важливим при цьому є вибір архітектури таких розподілених систем, яка дозволить ефективно організувати процес виявлення у локальній комп'ютерній мережі порівняно з виявленням в окремій КС.

Тому, в роботі пропонується використання розподілених багаторівневих систем на основі децентралізації та самоорганізації для виявлення ЗПЗ у локальних комп'ютерних мережах. У такому випадку, залучення інших компонентів розподіленої системи до виявлення ЗПЗ в певній КС підвищує ймовірність його прояву і, відповідно, покращує ефективність його виявлення.

Зв'язок роботи з науковими програмами, планами, темами. Дослідження, представлені у дисертації, проводились в рамках держбюджетних НДР

Хмельницького національного університету № 1Б-2001 «Методологія тестового комбінованого діагностування мікропроцесорних пристроїв та систем (МПП та С) на базі компонентів штучного інтелекту» (номер державної реєстрації 0101U005058), № 4Б-2012 «Розвиток теоретичних основ та розробка методів статико-динамічного спектрального оцінювання сигналів в радіолокації» (номер державної реєстрації 0112U002247), № 1Б-2018 «Розроблення високоефективних методів відбору енергії від фотоелектричних модулів» (номер державної реєстрації 0116U001548), № 1Б-2019 «Агентно-орієнтована система підвищення безпеки та якості програмного забезпечення комп'ютерних систем» (номер державної реєстрації 0119U100662).

Мета і завдання дослідження. Метою дисертаційної роботи є покращення ефективності виявлення зловмисного програмного забезпечення шляхом розвитку теорії і практики створення розподілених систем у локальних комп'ютерних мережах на основі принципів децентралізації та самоорганізації.

Задачі дослідження формулюються в роботі наступним чином:

1) дослідити особливості функціонування зловмисного програмного забезпечення в локальних комп'ютерних мережах та проаналізувати ефективність роботи сучасних розподілених систем виявлення, а також їх архітектуру і компоненти;

2) удосконалити модель архітектури розподіленої системи виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах, в якій синтезувати вимоги розподіленості, децентралізованості, багаторівневості та самоорганізованості, для створення на її основі розподілених систем та їх компонентів, що функціонуватимуть автономно і самостійно прийматимуть рішення про наявність зловмисного програмного забезпечення та нарощення своїх функціональних можливостей;

3) розробити модель архітектури типових компонентів розподіленої багаторівневої системи виявлення зловмисного програмного забезпечення на основі структур Кріпке з представленням компонентів через стани, в яких вони можуть перебувати під час функціонування, для визначення стану безпеки всієї розподіленої системи та її компонентів;

4) розробити метод взаємодії компонентів розподіленої багаторівневої системи виявлення зловмисного програмного забезпечення для підтримки її цілісності, визначення порядку передачі знань між її компонентами і використання встановлених аналітичних залежностей між рівнями безпеки програмних модулів (ПМ) та рівнем безпеки всієї розподіленої багаторівневої системи (РБС), на основі якого система змогла б автономно змінювати свою архітектуру та функції без втручання користувача, а також визначати стратегію своєї подальшої роботи;

5) удосконалити моделі зловмисного програмного забезпечення шляхом їх подання алгебрами поведінки, що дозволило б створити базис поведінкових сигнатур, врахувати особливості функціонування зловмисного програмного забезпечення в локальних комп'ютерних мережах та здійснити його класифікацію за типами поведінки;

6) розробити метод виявлення бот-мереж у локальних комп'ютерних мережах, який би включав здійснення активного моніторингу системних подій та організацію узгодженої взаємодії компонентів розподіленої системи при прийнятті рішень, для

створення на його основі засобів, здатних інтегруватись в розподілену систему та класифікувати бот-мережі за їх поведінковими сигнатурами, сформованими закладеними в їх компоненти функціями;

7) розробити метод виявлення файлового зловмисного програмного забезпечення в локальних комп'ютерних мережах, суть якого полягає в поєднанні роботи програмних агентів, що здійснюють виявлення зловмисного програмного забезпечення в окремих комп'ютерних системах, відповідно до імплементованих в них методів: динамічного формування поведінкової сигнатури шляхом відстеження викликів прикладного програмного інтерфейсу, знаходження поліморфного та метаморфного програмного коду, сканування виконуваних програм шляхом створення для них автономних процесів та відповідних програмних агентів у розподіленій системі;

8) розробити метод виявлення файлового зловмисного програмного забезпечення на основі динамічного формування поведінкової сигнатури шляхом відстеження викликів прикладного програмного інтерфейсу, в якому поведінкову сигнатуру формувати на основі критичних викликів прикладного програмного інтерфейсу за групами зловмисної активності з відображенням частоти їх входження та характеру взаємодії критичних функцій, для виявлення нових версій відомого зловмисного програмного забезпечення не тільки за наявністю критичних викликів, але й за їх взаємодією між собою;

9) розробити метод виявлення поліморфних та метаморфних вірусів з використанням функцій заплутування програмного коду на основі поетапного аналізу і порівняння функціональних блоків програмного об'єкта та його змінених версій, отриманих, в тому числі, від різних компонентів розподіленої системи шляхом їх взаємодії між собою;

10) розробити програмне забезпечення розподіленої багаторівневої системи та її апаратно-програмні компоненти захисту інформації для реалізації запропонованих теоретичних основ таких систем, підтвердження можливості їх практичного створення та використання в експериментальних дослідженнях для порівняння з відомими системами виявлення і впровадити розроблену розподілену багаторівневу систему виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах організацій (підприємств).

Об'єкт дослідження – процес виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах.

Предмет дослідження – методи і засоби виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах.

Методи дослідження. Для розв'язання поставлених задач використовуються основні положення:

1) теорії розподілених систем для побудови моделей архітектури розподіленої системи та її компонентів, на основі яких можуть бути розроблені програмні та апаратно-програмні засоби;

2) теорії абстрактної алгебри для представлення зловмисного програмного забезпечення, як елементів підмножин із заданими характеристичними властивостями, у вигляді алгебр поведінки;

3) теорії множин і теорії графів для деталізованого подання зловмисного

програмного забезпечення, як об’єктів дослідження;

4) методи класифікації для здійснення віднесення програмних об’єктів до класів зловмисного програмного забезпечення;

5) теорії комп’ютерних мереж для представлення і організації функціонування розподіленої системи, зокрема її програмних та апаратно-програмних компонентів.

Наукова новизна одержаних результатів полягає в наступному:

1) удосконалено модель архітектури розподіленої системи виявлення зловмисного програмного забезпечення в локальних комп’ютерних мережах, яка відрізняється від відомих комплексним врахуванням вимог розподіленості, децентралізованості, багаторівневості та самоорганізованості, що дозволяє створювати на її основі розподілені системи та їх компоненти, які функціонуватимуть автономно і самостійно прийматимуть рішення про наявність зловмисного програмного забезпечення та нарощення своїх функціональних можливостей;

2) вперше розроблено модель архітектури типових компонентів розподіленої багаторівневої системи виявлення зловмисного програмного забезпечення на основі структур Кріпке з представленням компонентів через стани, в яких вони можуть перебувати під час функціонування, що дає змогу враховувати перебування їх в різних станах і є основою для визначення стану безпеки всієї розподіленої системи та її компонентів;

3) вперше розроблено метод взаємодії компонентів розподіленої багаторівневої системи виявлення зловмисного програмного забезпечення на основі підтримки її цілісності та визначення порядку передачі знань між її компонентами і використання встановлених аналітичних залежностей між рівнями безпеки програмних модулів та рівнем безпеки всієї розподіленої багаторівневої системи, що дозволяє системі автономно змінювати свою архітектуру та функції без втручання користувача, а також визначати стратегію своєї подальшої роботи;

4) удосконалено моделі зловмисного програмного забезпечення шляхом їх подання алгебрами поведінки, що дозволило створити базис поведінкових сигнатур, і, на відміну від відомих представлень, врахувати особливості функціонування зловмисного програмного забезпечення в локальних комп’ютерних мережах та здійснити його класифікацію за типами поведінки;

5) вперше розроблено метод виявлення бот-мереж у локальних комп’ютерних мережах, який базується на здійсненні активного моніторингу системних подій та узгодженій взаємодії компонентів розподіленої системи при прийнятті рішення, і, на відміну від відомих методів, дає можливість створення на його основі засобів, здатних інтегруватись в розподілену систему та класифікувати бот-мережі за їх поведінковими сигнатурами, що формуються закладеними в їх компоненти функціями;

6) вперше розроблено метод виявлення файлового зловмисного програмного забезпечення в локальних комп’ютерних мережах, який полягає в поєднанні роботи програмних агентів, що здійснюють виявлення зловмисного програмного забезпечення в окремих комп’ютерних системах, відповідно до імплементованих в них методів: динамічного формування поведінкової сигнатури шляхом відстеження викликів прикладного програмного інтерфейсу, знаходження поліморфного та метаморфного програмного коду, сканування виконуваних програм шляхом

створення для них автономних процесів та відповідних програмних агентів у розподіленій системі, що, на відміну від аналогів, дозволяє покращити аналіз і підвищити достовірність виявлення зловмисного програмного забезпечення;

7) вперше розроблено метод виявлення файлового зловмисного програмного забезпечення на основі динамічного формування поведінкової сигнатури шляхом відстеження викликів прикладного програмного інтерфейсу, в якому, на відміну від відомих методів, поведінкова сигнатура формується на основі критичних викликів прикладного програмного інтерфейсу за групами зловмисної активності та відображає частоту їх входження і характер взаємодії критичних функцій, що дає змогу виявляти нові версії відомого зловмисного програмного забезпечення не тільки за наявністю критичних викликів, але й за їх взаємодією між собою;

8) вперше розроблено метод виявлення поліморфних та метаморфних вірусів з використанням функцій заплутування програмного коду, відмінністю якого є поетапний аналіз і порівняння функціональних блоків програмного об’єкта та його змінених версій, отриманих, в тому числі, від різних компонентів розподіленої системи шляхом їх взаємодії між собою.

Обґрунтованість і достовірність наукових положень, висновків і рекомендацій. Наукові положення, висновки і рекомендації дисертації обґрунтовані коректним та доцільним використанням математичного апарату, алгоритмами здійснення виявлення, успішною програмною реалізацією розробленої розподіленої багаторівневої системи виявлення зловмисного програмного забезпечення в локальних комп’ютерних мережах, ефективним практичним впровадженням результатів дисертаційного дослідження на підприємствах, що експлуатують комп’ютерні системи, яке продемонструвало відповідність теоретичних досліджень з реальними результатами застосування.

Практичне значення одержаних результатів. У результаті виконаного дисертаційного дослідження розроблено архітектуру і компоненти розподіленої багаторівневої системи виявлення зловмисного програмного забезпечення в локальних комп’ютерних мережах, здійснено їх програмну реалізацію, а також розроблено апаратно-програмні засоби захисту інформації в складі розподіленої багаторівневої системи, використання яких регламентується вимогами безпеки. Результати експериментальних досліджень підтверджують ефективність розроблених програмних засобів, а також правильність наукових положень теорії розподілених систем, оскільки впровадження розподіленої багаторівневої системи виявлення зловмисного програмного забезпечення дозволяє підвищити достовірність виявлення на 5–12 % порівняно з відомими аналогами та знизити рівень помилок першого роду до 5 %.

Теоретичні та практичні результати роботи впроваджено в Державному підприємстві «Новатор» (м. Хмельницький, акт про впровадження від 29.03.2019 р., відділ автоматизованих систем управління), ТОВ «ІТТ – telecommunication company» (м. Хмельницький, акт про впровадження від 21.02.2019 р.), ТОВ «ЮКС++» (м. Хмельницький, акт про впровадження від 26.02.2019 р.), компанії CYPRESS SEMICONDUCTOR (м. Львів, акт про впровадження від 28.02.2019 р.) та навчальному процесі Хмельницького національного університету при викладанні дисциплін «Безпека та захист комп’ютерних систем», «Технічна діагностика і надійність комп'ютерних пристроїв та систем», «Паралельні та розподілені обчислення» та «Системне програмне

забезпечення» (акт про впровадження від 05.04.2019 р.).

Особистий внесок здобувача. Всі основні результати дисертаційного дослідження, які представлені до захисту, одержані автором особисто. В роботах, опублікованих одноосібно автором, отримано наступні результати: [15] – розроблено модель розподіленої системи на основі синтезу багаторівневої, децентралізованої та самоорганізованої архітектури та реалізовано метод виявлення метаморфних вірусів; [2] – розроблено представлення файлового зловмисного програмного забезпечення матрицями, в яких поєднано середовища можливого перебування ЗПЗ та шляхи його поширення; [14] – розроблено систему оцінки безпеки початкового стану комп'ютерних систем у локальній мережі залежно від використовуваного системного програмного забезпечення; [16] – систематизовано критерії класифікації методів виявлення файлового ЗПЗ та здійснено їх класифікацію, виділено типові недоліки класів методів; [17, 43] – запропоновано модель розподіленої багаторівневої системи виявлення ЗПЗ та формалізовано її типову архітектуру; [18, 45] – розроблено алгебраїчні системи та алгебри поведінки для представлення об'єктів класів ЗПЗ; [20] – розроблено модель архітектури компонент РБС на основі структур Кріпке та метод їх взаємодії; [37] – розроблено метод виявлення бот-мереж самоорганізованими розподіленими системами; [41, 46] – розроблено метод взаємодії компонентів РБС; [51] – розроблено програмне забезпечення РБС.

У роботах, опублікованих у співавторстві, автору належать основні ідеї, теоретична та практична розробка положень, відображених у характеристиці наукової новизни отриманих результатів, а саме: [22, 23, 30, 40] – розроблено методологію використання агентних систем при здійсненні виявлення бот-мереж у комп'ютерних мережах, а також принципи комунікації між агентами; [24] – розроблено метод виявлення бот-мереж, які містять поліморфний програмний код; [25] – запропоновано методологію отримання різних зразків ЗПЗ із залученням різних КС мережі для отримання його функціональних блоків; [26] – розроблено метод виявлення бот-мереж на основі врахування його поведінки в мережі та можливої присутності файлового ЗПЗ; [27] – розроблено метод виявлення файлового ЗПЗ, яке використовує техніки заплутування програмного коду, та здійснено вибір метрик для порівняння функціональних блоків; [28, 35] – розроблено модель та архітектуру РБС і її програмні модулі; [10] - розроблено методологію вирішення проблеми виявлення файлового ЗПЗ розподіленими системами; [1, 3] – розроблено систему представлення файлового ЗПЗ на основі матриць та механізм їх отримання; [36] – розроблено концепцію та механізми організації розподілених систем; [4, 42] – розроблено типову структуру мережного ЗПЗ; [5, 6] – здійснено аналіз та формалізовано представлення станів процесів; [7] – розроблено представлення граничних станів процесів та структуру сигнатури процесу; [8] – розроблено базові характеристики компонентів розподілених систем виявлення ЗПЗ; [9, 12, 38, 39] – розроблено концепцію поділу файлового ЗПЗ на класи за їх поведінкою; [11] – здійснено визначення граничного стану для групи процесів, які наближаються до стану взаємоблокування; [13] – розроблено деталізоване множиною дій представлення файлового ЗПЗ, яке здійснює заплутування програмного коду, на основі застосування його моделей; [19] – розроблено архітектуру РБС та її представлення UML-діаграмами; [21] – розроблено метод отримання поведінкової сигнатури на основі трасування API-викликів; [47–49,

31–33] – розроблено методологію виявлення ЗПЗ у корпоративних мережах; [50] – розроблено модуль пошуку функціональних блоків у виконуваних програмах; [29] – розроблено матричне представлення класів ЗПЗ; [34] – розроблено формальне представлення функціональних блоків виконуваних програм та деталізацію зловмисних дій критичними API-функціями; [44] – розроблено метод взаємодії компонентів РБС.

Апробація результатів дисертації. Основні положення та результати проведених у дисертаційній роботі досліджень доповідалися та обговорювалися на 45 міжнародних та всеукраїнських конференціях, а саме: 6-th, 7-th, 8-th, 9-th IEEE Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications IDAACS'2011 (Prague, Chech Republic, 2011), IDAACS'2013 (Berlin, Germany, 2013), IDAACS'2015 (Warszawa, Poland, 2015), IDAACS'2017 (Bucharest, Romania, 2017); 19-th, 20-th, 21-th, 22-th, 23-th, 24-th, 25-th International Conference on Computer Networks: CN'2012 (Szczyrk, Poland, 2012), CN'2013 (Lwówek Slaski, Poland, 2013), CN'2014 (Brunów, Poland, 2014), CN'2015 (Brunów, Poland, 2015), CN'2016 (Brunów, Poland, 2016), CN'2017 (Brunów, Poland, 2017), CN'2018 (Gliwice, Poland, 2018); 4-th, 5-th, 7-th, 13-th International Scientific and Technical Conference on Computer Science and Information Technologies: CSIT (Lviv, Ukraine, 2007, 2008, 2012, 2018); International Conference Advanced Computer Systems and Networks: Design and Application ACSN (Lviv, Ukraine, 2009, 2011); 1-st International Academic Conference on Science and Education in Australia, America and Eurasia: Fundamental and Applied Science (Melbourne, Australia, 2014); Міжнародній науково-практичній конференції «Сучасні інформаційні і електронні технології (СІЕТ)» (Одеса, 2009, 2010, 2011, 2012, 2013); Проблемно-науково-міжгалузевій конференції «Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, економіки, моделювання та управління (ПНМК)» (Бучач – Яремча, 2011, 2014); Міжнародній конференції «Контроль і управління в складних системах (КУСС)» (Вінниця, 2010, 2012, 2014, 2018); Міжнародній науково-практичній конференції «Комп'ютерні системи в автоматизації виробничих процесів» (Хмельницький – Головченці, 2002, 2003, 2004, 2005, 2007); Міжнародній науково-технічній конференції «Захист інформації і безпека інформаційних систем» (Львів, 2012); Міжнародному науково-практичному семінарі молодих вчених та студентів «Програмовані логічні інтегральні схеми та мікропроцесорна техніка в освіті і виробництві» (Луцьк, 2018); V Міжнародній науково-практичній конференції «Інформаційні технології та взаємодії» (Київ, 2018); Міжвузівській науково-практичній конференції «Прогресивні інформаційні технології в науці та освіті» (Вінниця, 2007); 1-st International Workshop «Critical infrastructure safety and security (CrlSS-DESSERT'11)» (Kirovograd, 2011); Міжнародній конференції «Інтелектуальний аналіз інформації (IAI)» (Київ, 2008, 2010, 2013); Міжнародній науково-технічній конференції «Системний аналіз та інформаційні технології (САІТ)» (Київ, 2008, 2009, 2012); щорічних наукових конференціях професорсько-викладацького складу Хмельницького національного університету.

Публікації. За результатами проведених досліджень основні наукові результати опубліковано у 21 науковій праці [1–21], з яких 2 статті у періодичних зарубіжних виданнях [10, 15] і 3 статті індексовані у наукометричних базах [10, 13, 14], 19 статей

у фахових наукових виданнях України [1–9, 11–14, 16–21]. Апробація засвідчена публікаціями 7 статей у періодичних зарубіжних серійних виданнях [22–28] і 7 праць в матеріалах зарубіжних та українських конференцій [29–35], індексованих у наукометричній базі Scopus, з яких 9 індексовані у наукометричній базі Web of Science, 11 статей та тез доповідей у журналах та збірниках праць конференцій [36-46]. Опубліковано 3 патенти на корисну модель [47–49] та 2 свідоцтва про реєстрацію авторського права на твір (програму) [50–51].

Структура дисертації. Дисертація складається з анотації, вступу, шести розділів, висновків, списку використаних джерел з 399 найменувань на 49 сторінках та шести додатків на 46 сторінках. Загальний обсяг дисертації становить 425 сторінок, з них 304 сторінки основного тексту, 54 рисунки, 46 таблиць.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У **вступі** обґрунтовано актуальність теми дисертації, визначено об’єкт та предмет, мету і завдання дослідження, визначено наукову новизну та практичну цінність одержаних результатів. Вказано зв’язок роботи з науковими програмами та науково-дослідними роботами за місцем виконання роботи. Надано інформацію щодо кількості публікацій та апробації результатів дисертації.

У **першому розділі** проведено аналіз і дослідження технологій розвитку зловмисного програмного забезпечення, систем виявлення зловмисного програмного забезпечення та достовірності результатів їх роботи.

Дослідження функціонування ЗПЗ в КС локальних комп’ютерних мереж (ЛКМ) та антивірусних засобів їх виявлення показало наступні результати: розробники ЗПЗ володіють значними технічними засобами для його реалізації; ЗПЗ може використовувати різносторонні засоби для поширення, що підвищує його життєздатність і можливості для поширення та ускладнює виявлення антивірусними засобами; певні типи ЗПЗ мають модульну структуру, що знижує достовірність виявлення відомими антивірусними засобами; застосування стандартних методів (сигнатурний аналізатор, метод контрольних сум тощо) не гарантує високого рівня достовірності виявлення ЗПЗ в КС через необхідність постійного оновлення існуючих баз вже відомого ЗПЗ; використання модулів шифрування у ЗПЗ не дозволяє застосовувати сигнатурний аналізатор та ускладнює виявлення через створення різних копій одного і того самого ЗПЗ; застосування сучасних евристичних аналізаторів вимагає значних ресурсів КС та, водночас, з невисокою швидкодією не гарантує належних результатів виявлення ЗПЗ в КС і, при цьому, можливі хибні спрацювання, кількість яких збільшується при встановленні високого рівня підозрілості у налаштуваннях системи; використання методів на основі контрольних сум не завжди дає однозначну відповідь щодо того, чи відбулося інфікування КС; використання мережних систем виявлення для підвищення достовірності виявлення, через залучення адміністратора мережі до прийняття рішення, знижує оперативність в прийнятті рішення; відомі мережні системи побудовані переважно з використанням централізованої архітектури, що активізує зловмисників до виявлення центру для зупинки системи; відомі мережні системи виявлення і антивірусні засоби переважно є хост-орієнтованими і не враховують можливостей ЗПЗ виконуватись в декількох КС

одночасно. Також було проаналізовано методи виявлення зловмисного програмного забезпечення та критерії їх класифікації. Виділено недоліки відомих методів та запропоновано стратегію для розробки нових методів виявлення з їх реалізацією в розподілених системах як засобів протидії ЗПЗ у локальних комп'ютерних мережах.

Відомі антивірусні засоби та системи виявлення ЗПЗ не забезпечують його повного виявлення: у КС залишається певний відсоток невиявленого ЗПЗ. Враховуючи актуальність його виявлення для підприємств і організацій, необхідна розробка нових методів та засобів виявлення ЗПЗ у локальних комп'ютерних мережах, особливо для нового ЗПЗ. З метою підвищення достовірності та ефективності процесу виявлення ЗПЗ у локальних комп'ютерних мережах актуальним є розроблення теорії і практики створення розподілених систем виявлення ЗПЗ як напряму подальших досліджень.

У **другому розділі** представлено удосконалену модель архітектури розподіленої системи виявлення ЗПЗ в ЛКМ, розроблені модель архітектури типових компонентів РБС на основі структур Кріпке та метод взаємодії компонентів РБС.

Згідно з аналізом поставленої задачі та сформованих вимог до систем виявлення ЗПЗ архітектуру РБС було синтезовано на основі комплексного врахування вимог розподіленості, децентралізованості, багаторівневості та самоорганізованості. Архітектура розробленої системи зображена як узагальнена схема взаємозв'язку основних складових рис. 1.

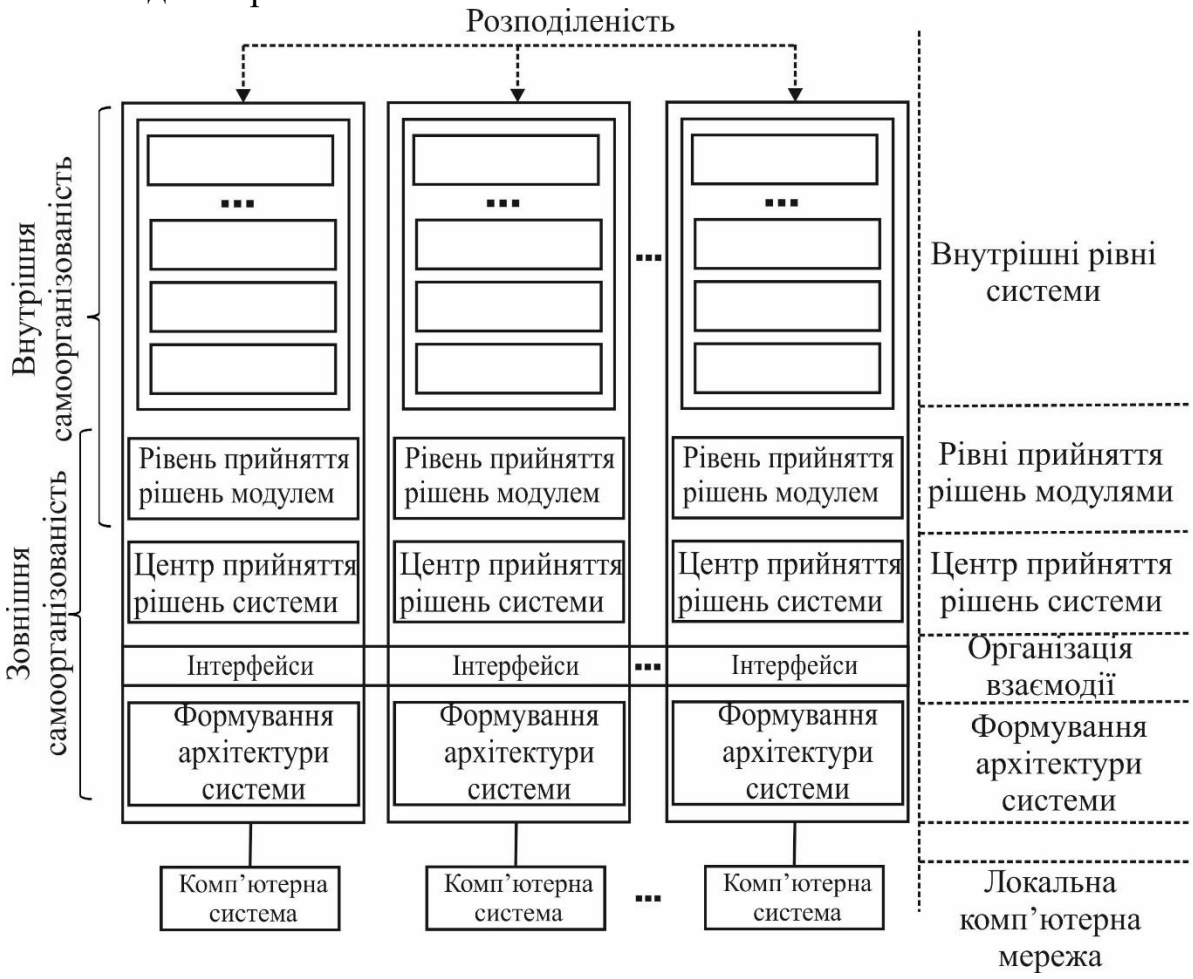

Рис. 1. Узагальнена схема взаємозв'язку основних компонентів в архітектурі системи

Модель РБС, що враховує її архітектурні складові та їх стани і зв'язки між ними, представимо за формулою (1):

$$M_A^S = \langle S, G_A \rangle, \quad (1)$$

де S – множина станів системи; G_A – орієнтований граф узагальнених станів РБС.

РБС складається із сукупності однакових програмних модулів. Кожен програмний модуль системи має однакову структуру, в якому виділено чотири основних рівні залежно від функційного призначення та згрупованих у них завдань: на першому рівні здійснюється моніторинг подій, визначаються переходи до наступних рівнів та оброблюється інформація, що надходить від інших програмних модулів; на другому – здійснюється перевірка виконуваних файлів, запущених процесів та мережної активності без залучення інформації від інших програмних модулів системи; на третьому рівні виконуються завдання другого рівня із залученням інформації з інших програмних модулів системи; на четвертому рівні виконується оброблення, оптимізація та вилучення інформації з бази програмного модуля КС. Кожен рівень та відповідні йому узагальнені підсистеми в свою чергу теж представляються наборами підрівнів, в які закладено виконання певних функціоналів. Поетапно ця розподілена архітектура буде наповнюватись підсистемами, в яких реалізовуватимуться інші підмоделі, що включатимуться до системи. Структура системи з урахуванням моделі розподіленої архітектури зображена на рис. 2. Представимо архітектуру РБС моделлю на основі структури Кріпке за формулою (2):

$$M_{\text{РБС}} = \langle S, S_0, R, F_{\text{РБС}} \rangle, \quad (2)$$

де S – скінченна множина станів РБС; S_0 – множина початкових станів (причому їх може бути декілька); R – множина переходів між станами; $F_{\text{РБС}}$ – функція, що відображає кожен стан з множини S у підмножину атомів, які є істинними у відображуваних станах.

Оскільки система є розподіленою, тоді множину станів системи представимо через підмножини станів, які відносяться до програмних модулів A_i, i = 1, 2, ..., n, а саме: $S = \bigcup_{i=1}^{n} S_i$, тобто множини станів ПМ формуватимуть множину станів, в якій перебуватиме система. Аналогічно, множина початкових станів РБС $S_0 = \bigcup_{i=1}^{n} S_{0i}$, причому серед їх елементів не може бути однакових. З будь-якого стану $s \in S$ існує мінімум один перехід до іншого стану, тобто справедливим є відношення $R \subseteq S \times S$, що означає для будь-якого стану з множини станів існує відповідний йому стан з цієї самої множини.

Нехай T_A – множина атомарних тверджень, пов'язаних із станами, які є істинними тільки в цих станах. Якщо множина атомарних тверджень T_A скінчена, тоді множина її підмножин 2^{T_A} буде відображенням $F_{\text{РБС}}$ для множини S в ті підмножини атомів, які будуть істинними в $s \in S$. Представимо модель архітектури структурної компоненти РБС програмного модуля A_i за формулою (3):

$$M_{A_i} = \langle S_i, S_{0i}, R_i, F_{A_i} \rangle, \quad (3)$$

де i = 1, 2, ..., n, R_i – множина переходів між станами $s_{ij} \in S_i$; j – кількість i – тих станів; F_{A_i} – функція відображення множини станів S_i в множину підмножин множини T_{A_i}, тобто $2^{T_{A_i}}$.

Кожен стан обов'язково має зв'язок з деякими іншими станами. Перехід з одного стану до іншого, якщо між ними є зв'язок, відобразимо послідовністю $s_{ij}s_{ip}$, де i – номер програмного модуля, j та p - стани цього самого модуля. Тоді, послідовності $s_{ij}s_{ip}s_{ij}s_{ih}s_{iy}s_{iu}s_{ie}s_{ik}\ldots$ означатимуть переходи ПМ з одного стану в інший протягом часу його функціонування. РБС у процесі функціонування характеризується множиною послідовностей переходів із стану в стан програмних модулів. Діаграма переходів, зображена на рис. 3, відповідає структурі Кріпке для ПМ. У сукупності такі діаграми для всіх програмних модулів РБС утворюють її представлення через стани, в яких вона може бути. Представлення компонентів РБС через їх стани, в яких вони можуть перебувати під час функціонування, дало змогу визначати стан безпеки всієї розподіленої системи та її компонентів. Розроблена модель передбачає можливість збільшення кількості рівнів системи без зміни її архітектури. Основою архітектури РБС виступають програмні модулі з однаковими архітектурами, але при цьому кожен з них може самостійно приймати рішення на основі різних даних, зібраних у ЛКМ.

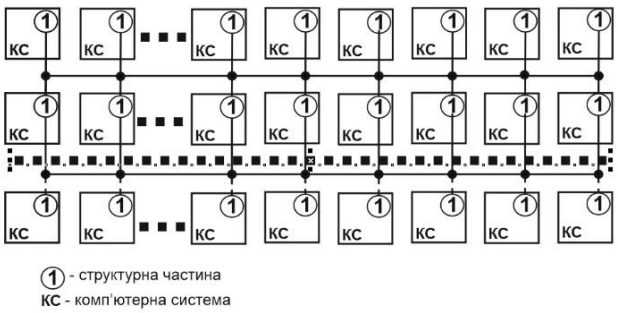

Рис. 2. Структура системи з урахуванням моделі розподіленої архітектури

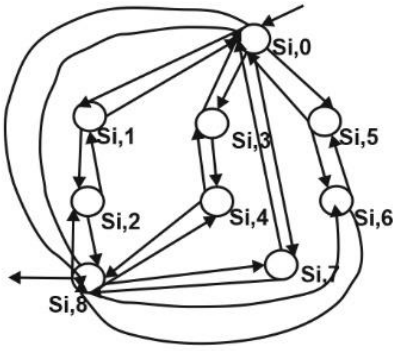

Рис. 3. Структура Кріпке для станів програмного модуля

Для встановлення порядку здійснення комунікації між частинами РБС та обміну знаннями між ними розроблено метод взаємодії компонентів РБС виявлення ЗПЗ. Він застосовуватиметься для розв'язання задач верхнього рівня організації взаємодії, тобто тільки для організації взаємодії частин системи і представлення її цілісною. Для вирішення проблеми з безпосереднього виявлення ЗПЗ у локальних комп'ютерних мережах застосовуватимуться методи, які відноситимуться до нижчого рівня системи, що включатимуть архітектурні особливості розподіленої системи і технології виявлення ЗПЗ.

Для розгляду основних кроків методу вважатимемо, що не менше, ніж в двох КС локальної мережі стартова функція виконалась успішно. Тоді подальші процеси, що протікатимуть у мережі, які пов'язані з функціонуванням системи, представимо такими кроками методу взаємодії компонентів РБС:

1. Визначити стани програмних модулів РБС.

1.1. Визначити стан кожного програмного модуля в КС.

1.2. Запис інформації (крок 1.1) у внутрішні бази кожного ПМ.

1.3. Порівняти результати сканування КС (крок 1.1) з попередніми результатами сканувань за певний період і здійснити їхню обробку.

1.4. Сформувати пакет повідомлення про свій стан кожним ПМ системи.

1.5. Запис повідомлення пакета (крок 1.5) у базу повідомлень ПМ.

1.6. Надіслати пакети повідомлень про свій стан у інші ПМ РБС.

2. Обробити відповіді від ПМ КС на відправлені пакети.

2.1. Отримати відповіді на надіслане повідомлення про стан ПМ та обробити їх.

2.2. Запис отриманої інформації (крок 2.1) до бази повідомлень ПМ.

2.3. Підтвердити формування РБС з активних програмних модулів.

3. Обробити програмним модулем невизначеності, які пов’язані з відсутністю відповідей на відправлені пакети.

4. Сканувати задані порти КС, через які буде здійснюватись обмін повідомленнями між програмними модулями, та обробити результати сканувань.

5. Здійснити розрахунок стану програмного модуля та інших ПМ РБС на етапі обміну повідомленнями і обробити результати.

6. Визначити стан безпеки розподіленої багаторівневої системи.

6.1. Визначити стан безпеки кожного ПМ окремо.

6.2. Здійснити обмін повідомленнями про свої стани всіма активними ПМ.

6.3. Провести підтвердження про обмін повідомленнями.

6.4. Обчислити (за формулами (4) і (5)) стан безпеки РБС кожним ПМ.

6.5. Здійснити обмін повідомленнями між ПМ про визначений ними стан безпеки РБС.

6.6. Здійснити аналіз і обробку отриманих результатів кожними ПМ РБС.

7. Прийняти рішення про подальшу роботу РБС на основі розрахованого рівня безпеки.

8. Вилучити програмний модуль з РБС у результаті вимкнення КС.

9. Застосувати методи виявлення ЗПЗ при відповідній зміні стану ПМ РБС.

9.1. Застосувати метод виявлення файлового ЗПЗ при переході ПМ на рівень 2.

9.2. Застосувати метод виявлення мережного ЗПЗ при переході ПМ на рівень 3.

10. Оптимізувати статистичні дані, які накопичені в системі кожним ПМ.

11. Здійснити обмін знаннями всередині РБС між ПМ, які стосуються виявлення ЗПЗ.

12. Здійснити колективне виконання завдань ПМ РБС при перебуванні одного з ПМ на рівні 3 та обмін отриманими результатами (крок 11).

13. Виконувати завдання першого і другого рівнів ПМ, якщо в складі РБС всього один ПМ.

14. Поповнити РБС новими програмними модулями при розширенні РБС або при ввімкненні КС, в які встановлені ПМ.

Кроки методу можуть виконуватись не послідовно, а залежно від настання певних подій у системі, які вимагатимуть реагування на них. Кроки 7-12, 14 пов’язані з кроком 1 і можуть викликатись після повного його виконання або повертатись після повного свого виконання до нього. Така їх взаємопов’язаність спричинена тим, що РБС є реагуючою системою на певні події, що протікають в КС. Настання таких подій відслідковується динамічно в КС та змінює стан ПМ, який як компонента РБС

повідомляє про зміну свого стану іншим компонентам системи, що відображається в постійній зміні станів і рівнів РБС. Крок 13 відображає здатність РБС бути в складі всього одного ПМ після виконання кроку 1 або після виконання кроку 8. Перехід з кроку 4 до кроку 1 здійснюється не одним і тим самим ПМ, а двома різними ПМ РБС.

Виведено формулу (4) для обчислення рівня безпеки РБС на першому етапі:

$$R_{b,\text{РБС},1} = \frac{\sum_{l=1}^{n}(1-\sum_{s=1}^{m} k_{s,l}\cdot p_{s,l})}{n}, \tag{4}$$

де $R_{b,\text{РБС},1}$ – рівень безпеки РБС, визначений на першому етапі; b – позначення безпеки; l – номер ПМ РБС; n – кількість ПМ РБС; $k_{s,l}$ – коефіцієнт загрози бути ураженим ЗПЗ s – того стану ПМ, значення якого встановлюється з відрізку $[0;1]$ в залежності від того, які функційні навантаження закладено у певний s – й стан; $p_{s,l}$ – ймовірність бути ураженим ЗПЗ; m – кількість станів ПМ.

Враховуючи необхідність досягнення однозначності у визначенні подальших дій для РБС після першого етапу досліджень свого стану і знаходження центру в поточний момент, встановлено аналітичні залежності на основі результатів з формування центру для визначення подальших дій РБС за функцією, результат якої вказуватиме на подальші кроки РБС.

На основі числових величин характеристик РБС встановлено аналітичну залежність, яка виражає інтегроване значення рівня її безпеки на другому етапі за формулою (5), вважаючи частку кожної з характеристик, які враховано, рівноцінною:

$$R_{b,\text{РБС},2} = \frac{1}{4}\cdot\left(\sum_{s=1}^{m}\left(1-\prod_{\substack{j=1,\\ p_{s,j}<1}}^{n}(1-p_{s,j})\right)\cdot k_s + \sum_{j=1}^{n}\sum_{\substack{s=1,\\ t_{s,j}>0,\\ w_{s,j}>0}}^{m}\left(\frac{t_{s,j}}{\sum_{s=1}^{m} t_{s,j}}\cdot \frac{w_{s,j}}{\sum_{s=1}^{m} w_{s,j}}\right) + \sum_{s=1}^{m}\left((1+k_s)\cdot\frac{\sum_{j=1}^{n} w_{s,j}}{\sum_{s=1}^{m}\sum_{j=1}^{n} w_{s,j}}\cdot\frac{\sum_{j=1}^{n} t_{s,j}}{\sum_{s=1}^{m}\sum_{j=1}^{n} t_{s,j}}\right)\right), \tag{5}$$

де $R_{b,\text{РБС},2}$ – рівень безпеки РБС, визначений на другому етапі; b – позначення безпеки; s – номер ПМ РБС; n – кількість програмних модулів РБС; m – кількість станів ПМ; k_s – коефіцієнт загрози бути ураженим ЗПЗ s – того стану ПМ, значення якого встановлюється з відрізку $[0;1]$ залежно від того, які функційні навантаження закладено у певний s – й стан; $p_{s,j}$ – ймовірність бути ураженим ЗПЗ; $w_{s,j}$ - кількість перебувань ПМ з номером j в стані s; $i = 1,2,\dots,n$; $s = 1,2,\dots,m$; $t_{s,j}$ – сумарний час перебування ПМ з номером j у стані s.

Значення $p_{s,j}$ отримуються на основі результатів функціонування закладених у програмні модулі підсистем виявлення певних типів ЗПЗ.

Модель спроектованої РБС виявлення ЗПЗ може бути реалізована програмно або апаратно-програмно залежно від вимог політики безпеки організації (підприємства), тому функціонування РБС може бути в двох режимах. Метод взаємодії компонентів РБС використовується однаково в обох режимах, бо він регламентує порядок взаємодії компонентів РБС тільки для верхнього рівня програмної частини, тобто зовнішнього

рівня. Наявність апаратно-програмних пристроїв для підвищення рівня захищеності розподіленої системи передбачає виконання ними тільки функцій внутрішнього рівня, тому вони не здійснюють взаємодію з програмною частиною системи зовнішнього рівня, яка є зв'язуючим програмним забезпеченням РБС.

Таким чином, використання розробленого методу дозволяє організовувати підтримку цілісності розподіленої системи та здійснення передачі знань, отриманих окремими структурними компонентами системи, іншим компонентам. Отримані аналітичні залежності рівнів безпеки ПМ і РБС дозволяють здійснювати зміну архітектури РБС динамічно та впливати на подальші її дії без втручання користувача. Розроблений метод є основою для розробки зв'язуючої частини програмного забезпечення РБС виявлення ЗПЗ.

У **третьому розділі** удосконалено моделі зловмисного програмного забезпечення шляхом їх подання алгебрами поведінок. Це представлення стало основою для створення базису поведінкових сигнатур, в яких враховано особливості функціонування ЗПЗ у локальних комп'ютерних мережах.

Формалізацію ЗПЗ здійснено для подальшого його представлення поведінковими сигнатурами. В якості об'єкту для дослідження було розглянуто множину ЗПЗ, яке, за певних обставин та протягом певного часу експлуатації ЛКМ, проникло в комп'ютерні системи, змогло подолати певні системи захисту і функціонує там, тобто, те ЗПЗ, яке на момент виявлення вже перебуває в КС, і в ЛКМ. ЗПЗ в ЛКМ, особливістю якого є втілення у виконувані файли, завантажувальний сектор жорсткого диска, оперативний запам'ятовуючий пристрій та поширення мережею своїх копій, представлено алгебраїчною системою типу $\tau = (\alpha,\ \beta)$:

$$\mathfrak{A}_V = \langle V,\ \ \Omega_F,\ \ \Omega_P \rangle, \tag{6}$$

де V – множина всього розглядуваного ЗПЗ; $\Omega_F = \{F_0,\ F_1,\ F_2, \dots, F_{\alpha_1}, \dots\}$ – множина операцій заданих на множині V для кожного $\alpha_1 = 0, 1, 2, \dots$; $\Omega_P = \{P_0,\ P_1,\ P_2, \dots, P_{\beta_1}, \dots\}$ – множина предикатів, заданих на множині V для кожного $\beta_1 = 0, 1, 2, \dots$; $\alpha = 1,\ \beta = 1$ – арності операцій, тому тип системи $\tau = (1,\ 1)$.

Елементами множини $v_j \in V$ $(j = 1, 2, \dots)$ вважатимемо всі об'єкти файлової системи, завантажувального сектора диска, оперативної пам'яті, мережні пакети, які відносяться до розглядуваного ЗПЗ. Елементи $v_0 \in V_0 \subseteq V$ є одиничними елементами, тобто такими, що містять єдиний функціонал, вміст якого полягає у необхідності здійснення самокопіювання з метою поширення, але без конкретного функціонального наповнення для виконання технічно цих дій. Решта операцій представлені іншими функціями. Елементи, що формують множину V_0, є породжуючими для решти різних елементів множини V. Функції з множини Ω_F виконуються на елементах v_0, що формують інші об'єкти, які належатимуть множині V, а також можуть виконуватись на інших елементах множини V, які не належать множині V_0. Функції з множини Ω_F не завжди успішно виконуватимуться по відношенню до елементів з множини V, тому для представлення ЗПЗ в ЛКМ вибрано також множину предикатів, яка відображатиме результат успішного/неуспішного виконання функцій. Функції F_{α_1} $(\alpha_1 = 0, 1, 2, \dots)$ з множини Ω_F визначимо як такі, що

здійснюватимуть відображення елементів з множини V на неї. Їх конкретне визначення залежатиме від поділу множини Ω_F на підмножини за різними характеристичними властивостями ЗПЗ. Предикати P_{β_1} $(\beta_1 = 0, 1, 2, \dots)$ з множини Ω_P визначимо як такі, що будуть істинними при успішному виконанні операцій, і хибними – в іншому випадку. Множину Ω_F представимо її підмножинами $\Omega_{F_{s,t}}$, які відображатимуть такі характеристичні для ЗПЗ властивості та закладені в його функціонал особливості: зберігання знань про механізм місця розміщення своїх наступних копій; пошук місця у пам'яті для розміщення своєї копії; знання про механізми втілення у виконувані програми; механізми запису в оперативну пам'ять; технології приховування свого перебування у комп'ютерних системах; пошук інших вузлів мережі для свого поширення; механізми для формування і відправлення мережних пакетів; алгоритми подолання механізмів захисту; техніки запису своїх копій у головний завантажувальний сектор; виконання деструктивних дій. Ці характеристичні властивості ЗПЗ, пов'язані із системними викликами, які відносяться до роботи з файлами, оперативною пам'яттю та командами роботи в мережі: створення, відкриття, закриття, видалення, читання, записування, додавання, знаходження, отримання атрибутів і встановлення атрибутів, команди доступу до ОП, команди для роботи в мережі. Реалізація характеристичних властивостей ЗПЗ, пов'язана з системними викликами та командами для роботи в мережі, визначатиме наповнення функцій з множини Ω_F і залежатиме від них, що є основою для ідентифікації таких дій.

Виділимо в множині Ω_F підмножини Ω_{F_p} таким чином, щоб $\Omega_F = \bigcup_{p=1}^{k} \Omega_{F_p}$, де k – кількість характеристичних властивостей ЗПЗ. Тоді алгебраїчні системи для кожної властивості задамо за формулою (7):

$$\mathfrak{A}_{V,p} = \langle V,\ \Omega_{F_p},\ \Omega_{P_p} \rangle, \tag{7}$$

де Ω_{F_p}– множина операцій, заданих на множині V; Ω_{P_p}– множина предикатів, заданих на множині V; $p = 1, \dots, k;\ k = 10$.

Нехай $V = \bigcup_{s=1}^{n} V_s$, тобто виділимо в кожній КС локальної мережі підмножину ЗПЗ V_s, де $s = 1, 2, \dots, n$. Для виділених підмножин з множини V у момент часу $t = 0$ буде справедливе твердження $\bigcap_{s=1}^{n} V_s = \emptyset$. Дійсно, із самого початкового встановлення програмного забезпечення у всі КС мережі, в них відсутнє ЗПЗ. У процесі збільшення тривалості їх роботи, тобто при $t > 0$, ймовірність появи певного ЗПЗ в різних КС мережі зростає, тому справедливим може бути твердження $\bigcap_{s=1}^{n} V_s \neq \emptyset$. Задамо алгебри для всіх десяти властивостей за формулою (8):

$$\mathfrak{B}_{V_{s,p}} = \langle V_s,\ \Omega_{F_p} \rangle, \tag{8}$$

де s – кількість вузлів ЛКМ; Ω_{F_p} – множина функцій заданих на множині V, які відповідають властивості p; $p = 1, \dots, k;\ k = 10$.

Аналогічно для кожного типу ЗПЗ задамо їх моделі за формулою (9):

$$\mathfrak{M}_{V,p} = \langle V; \quad \Omega_{P_{p,k}} \rangle, \tag{9}$$

де $\Omega_{P_{p,k}}$ - множина предикатів, заданих на множині V; $p = 1, \dots, 10$.

Крім того, для розробки відповідних алгебр поведінки було розглянуто та враховано в них види загроз, які можуть бути здійснені у ЛКМ. Їх аналіз пов'язаний з вимогами, які висуваються до безпеки ЛКМ. Для порушення цих вимог розробники ЗПЗ закладають в нього механізми здійснення загроз у вигляді таких атак: переривання, перехоплення, зміна, підробка. В ЛКМ здійснення таких атак або їх комбінацій відбувається по відношенню до апаратного забезпечення, програмного забезпечення, ліній зв'язку та даних. Об'єкти в ЛКМ, які можуть бути піддані атакам за певним типами загроз: множина файлів, множина користувацьких процесів, множина запитів користувачів, множина мережних пакетів. Функціонування комп'ютерних систем у локальних мережах пов'язане з обробленням, зберіганням та поширенням інформації. Саме при виконанні цих дій можливим є здійснення атак, узагальнені види яких виділимо наступним чином: множина несанкціонованих змін; множина підроблених об'єктів, розміщених у системі; множина перехоплень з боку зловмисника засобами програм або комп'ютерів; множина переривань, яка здійснена для виведення з ладу компонентів системи. Усі категорії атак задано матрицями, в яких зберігаються типові шаблони поведінки.

Таким чином, удосконалено моделі типів зловмисного програмного забезпечення поданням їх алгебрами поведінки, які стали основою створення базису поведінкових сигнатур. Вони враховують особливості, які проявлятимуться при виконанні функцій, їх функціонування в ЛКМ та є основою для здійснення класифікації за типами поведінки.

У **четвертому розділі** представлено розроблений метод виявлення бот-мереж у локальних комп'ютерних мережах, який базується на здійсненні активного моніторингу системних подій та узгодженій взаємодії компонентів розподіленої системи при прийнятті рішення.

В якості мережного ЗПЗ як об'єктів дослідження розглянуто керовані бот-мережі. Функції, які закладені в їх алгоритми функціонування, виконують типові дії в комп'ютерних мережах. Характерна особливість, яка чітко відрізняє цю підмножину ЗПЗ від інших, полягає в тому, що, переважно, не відбуваються спроби копіювання зловмисного коду у всі файли для поширення. Тому для виявлення такого ЗПЗ необхідним є пошук та виокремлення характерних для цього типу ознак, їх подальша формалізація та використання. Оскільки ЗПЗ такого типу є складними програмними комплексами, які функціонують у глобальних комп'ютерних мережах, то для їх виявлення було розроблено метод, застосування та реалізація якого передбачена саме в розподілених системах.

Для мережного ЗПЗ здійснено формалізацію закладених в нього функцій. За своєю структурою в бот-мережах виділяють вузли, які відносяться до керування мережею і підтримки її цілісності, та вузли, які є кінцевими і з яких здійснюється виконання зловмисних дій. Вся бот-мережа представляє собою розподілену програмну систему, в якій є рівень зв'язуючого програмного забезпечення. Було виділено основні складові бот-мережі: командно-керуючі центри, контролюючі

центри, базові елементи мережі (боти). Узагальнена структура бот-мережі зображена на рис. 4. Базові елементи бот-мережі позначимо підмножинами E_{3,i_3}, де $i_3 = 1, 2, \dots, n_3$, n_3 – кількість базових елементів мережі. Контролюючі центри бот-мережі позначимо підмножинами E_{2,i_2}, де $i_2 = 1, 2, \dots, n_2$, n_2 – кількість контролюючих центрів бот-мережі. Командно-контролюючі центри бот-мережі займають третій рівень ієрархії і позначимо їх підмножинами E_{1,i_1}, де $i_1 = 1, 2, \dots, n_1$, n_1 – кількість командно-керуючих центрів бот-мережі. Розроблено еталонну модель бот-мереж, в основі якої її типові компоненти задані відповідними функціями відображають типові дії ЗПЗ у ЛКМ. Їх представлення базується на характеристичних ознаках бот-мережі і включає компоненти трьох рівнів. Представимо цілісну бот-мережу, як об'єднання її складових частин за формулою (10):

$$E = \bigcup_{i_1=1}^{n_1} E_{1,i_1} \bigcup_{i_2=1}^{n_2} E_{2,i_2} \bigcup_{i_3=1}^{n_3} E_{3,i_3}, \qquad (10)$$

де E – множина складових частин бот-мережі.

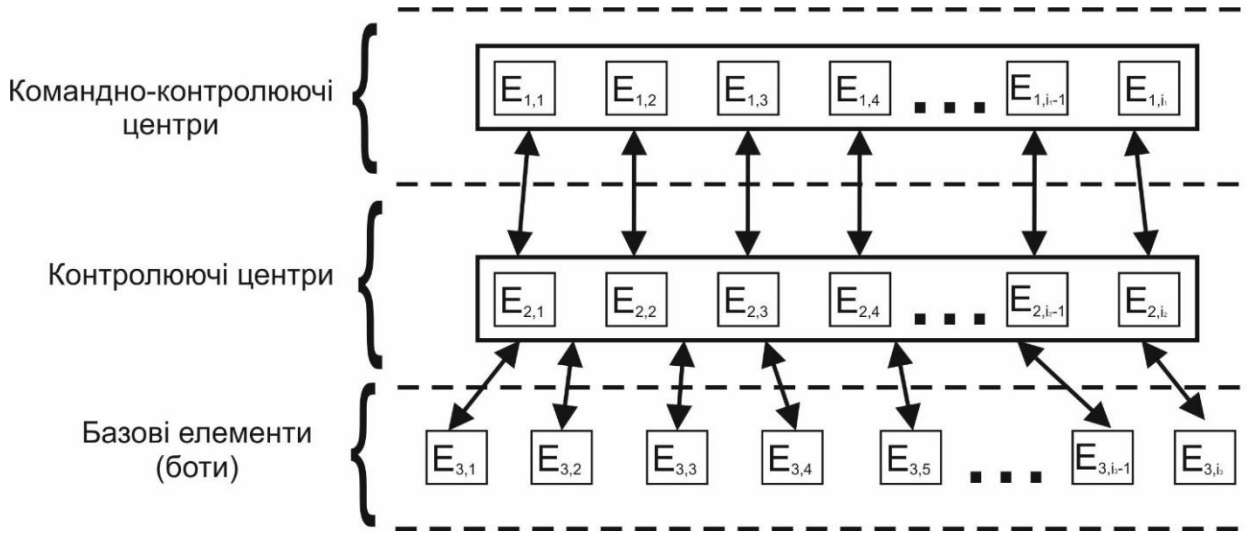

Рис. 4. Структура розподіленої керованої бот-мережі

Елементами підмножин $E_{w,i}$ є функції, з яких сформовано елемент $E_{w,i}$ бот-мережі. В різних елементах $E_{w,i}$ можуть бути однакові функції, тобто елементи $E_{w,i}$ можуть формуватись з однакових блоків (функцій). Таке представлення дало змогу систематизувати відомі бот-мережі в еталонну модель бот-мережі, яка охоплює різні топології та стала основою для деталізації до рівня функцій API-викликів.

Було проаналізовано можливі варіанти подій у КС локальної мережі для застосування РБС, яка містить інформацію про всі КС та події, що відбуваються в них, які аналізуються і на основі такого аналізу приймається рішення про подальші дії. Варіанти подій є основою для розробки методу виявлення бот-мереж у ЛКМ в частині відображення в ньому особливостей різного стану КС, в якому вона може перебувати. На основі варіантів здійснюється перехід до однієї з шести стратегій для визначення подальшої роботи ПМ та РБС. Ймовірності в станах ПМ залежать від варіантів подій, які відбуватимуться в КС, пов'язаних з присутністю ЗПЗ, та визначаються встановленими аналітичними виразами.

Для здійснення оброблення подій у КС програмним модулем РБС було здійснено формалізоване представлення розроблених функцій бот-мереж через поведінкові сигнатури на основі API-функцій. Підготовка таких еталонних моделей полягає у формуванні знань на основі функцій, які вказують на наявність бот-мереж. З цією метою потрібно виконати такі основні кроки:

1. Здійснити для кожної функції еталонної моделі бот-мережі її реалізацію в різних операційних системах.

2. Для кожної функції задати вектор зловмисних дій та атак, виділивши його компоненти. Узагальнити компоненти різних векторів, які відповідають однаковим функціоналам.

3. Здійснити трасування складових функцій для отримання їх представлення через API-функції для кожної операційної системи чи її актуальної версії.

4. Пронумерувати всі отримані, задіяні в представленні API-функції, або всі API-функції відповідно до переліку для кожної операційної системи.

5. Здійснити нумерування API-функцій для кожної компоненти вектора зловмисних дій та атак еталонної моделі бот-мережі для підготовки бітової матриці з врахуванням різних представлень компонент вектора зловмисних дій та атак.

6. Сформувати вектори для функцій ознак за номерами API-функцій, які їх формують, для трьох типів елементів бот-мережі.

7. Маркувати критичні API-функції у векторах.

8. Побудувати вектори кількості входжень API-функцій у векторах зловмисних дій та атак.

9. Здійснити побудову бітової матриці з врахуванням різних представлень компонент векторів зловмисних дій та атак.

10. Побудувати вектор зворотних дій для кожної функції.

11. Задати бітову матрицю зворотних дій на основі відношення.

12. Визначити кількість компонентів векторів для кожної з функцій і занести в кожен вектор додатковою останньою компонентою.

Отримані результати за цими кроками заносимо до бази зловмисних дій та атак і визначаємо цю інформацію, як таку, що характеризує клас «0». Для відомих типів бот-мереж, які сформовано на основі їх виявлення, проводимо виконання кроків 1–12 для кожного типу і відносимо отриману інформацію в базу зловмисних дій та атак відповідно за класами «1» – «6». Для тих функцій типів бот-мереж класів «1» – «6», які не є у відкритому доступі, використовуємо функції-аналоги з еталонної моделі бот-мережі. Збільшення кількості класів є необхідним для розширення можливості самонавчання класифікаторів ПМ РБС.

Отримавши вектори, в яких представлені числові величини в якості їхніх компонентів, що дозволяє на їхній основі перейти до здійснення класифікації нових виконуваних об’єктів в КС, потрібно організувати збір інформації про виконувані процеси в КС на основі API-функцій. Здійснення моніторингу API-викликів у динамічному режимі проводиться програмним модулем РБС у кожній КС його компонентою-утилітою.

Для отримання чисельних ознак компонентів вектора ймовірно зловмисних подій здійснимо визначення відсоткової відповідності його до кожного вектора кожного класу з бази векторів зловмисних дій та атак.

Здійснення навчання наївного баєсівського класифікатора проводимо для кожної API-функції, підкласів та класів з урахуванням варіацій для функцій. Заданий порядок для організації навчання класифікатора використовуємо як основу для здійснення організації самонавчання, яке закладене в класифікатор кожного ПМ РБС. При додаванні нового елемента до класифікатора ПМ в одній з КС після цього здійснюється розсилання всім іншим ПМ РБМ інформації про ці дії і сам контейнер з новим елементом для доповнення класифікатора. Таким чином, знання, отримані одним ПМ, передаються іншим компонентам системи для використання.

Розроблений метод виявлення бот-мереж у ЛКМ, суть якого полягає в здійсненні активного моніторингу системних подій та узгодженій взаємодії компонентів розподіленої системи при прийнятті рішення, складається з таких основних кроків:

1. Отримати дані про активні процеси та мережні пакети на основі активного моніторингу виконання команд в КС, починаючи з першої API-функції кожного процесу, що буде виконуватись після запуску КС.

2. Здійснити збір даних моніторингу після виявлення певних ймовірно зловмисних проявів в КС у вектор.

3. Сформувати вектор ознак ймовірно підозрілих дій для зібраних даних, компонентами якого є API-функції.

4. Прийняти рішення про місце обробки вектора ймовірно зловмисних дій.

5. Здійснити обробку вектора ймовірно зловмисних дій в КС, в якій він сформований, якщо аналіз ресурсів показав невеликий відсоток завантаженості, інакше надіслати для обробки в іншу визначену ПМ КС.

6. Здійснити класифікацію вектора ймовірно зловмисних дій.

7. Аналіз результатів кроку 6.

7.1. Якщо встановлено віднесення такого вектора до певного підкласу класу бот-мереж, тоді додати цю інформацію до класифікаторів усіх ПМ.

7.2. Якщо встановлено віднесення такого вектора до декількох підкласів класів бот-мереж, тоді здійснити аналіз із залученням інших ПМ РБС на основі обробки варіантів подій.

7.3. Якщо близькість для включення до певного підкласу є нечіткою, але додатково із залученням інших ПМ визначено, що вектор містить зловмисні дії, тоді здійснити створення нового класу для бот-мереж, занести дані, оновити налаштування класифікатора, передати результат іншим КС.

7.4. Якщо перевірка показала, що досліджуваний вектор не містить зловмисного навантаження, тоді зупинити дослідження процесу, на основі якого він був сформований.

7.5. Якщо встановлено, що досліджуваний вектор містить зловмисне навантаження, тоді зупинити відповідний процес.

7.6. Здійснити пошук і дослідження аналогічних процесів в інших КС мережі на основі отриманих відомостей, де встановлена РБС її програмними модулями.

8. Обробка варіантів подій із залученням решти ПМ РБС.

8.1. – 8.6. Для варіантів подій (1–6) задіяти відповідну стратегію та виконати дії.

9. Обчислити значення ймовірностей в станах ПМ і надіслати вимогу для інших ПМ здійснити обчислення ймовірності бути ураженою для всієї РБС. Цей крок

здійснюється позапланово через дослідження наявного зловмисного прояву в одній з КС.

10. Здійснити оптимізацію вектора, що додається до бази зловмисних дій та атак.

11. Сформувати значення ймовірностей перебування в станах для надсилання іншим ПМ щодо визначення стану РБС за встановленими аналітичними виразами.

12. Залучити засоби для здійснення самоконтролю, використовуючи внутрішній планувальник ПМ РБС, та забезпечення стійкості ПМ в КС при групі подій з кроку 8, які відносяться до зовнішніх впливів.

За розробленим методом програмні модулі РБС можуть досягати таких цілей: вилучення ймовірно уражених ПМ з РБС, встановлення відношення до ЗПЗ типу бот-мереж на основі обміну і обробки знань, створення нового класу бот-мереж на основі фрагмента програмного коду.

Таким чином, розроблений метод виявлення бот-мереж у ЛКМ, суть якого полягає в здійсненні активного моніторингу системних подій та узгодженій взаємодії компонентів розподіленої системи при прийнятті рішення, дає можливість створювати засоби, які здатні інтегруватись в розподілену систему та класифікувати бот-мережі за їх поведінковими сигнатурами, що формуються закладеними в їхні компоненти функціями.

У **п’ятому розділі** розроблено методи виявлення файлового ЗПЗ в ЛКМ, реалізація яких дозволила створювати засоби, що інтегруються в розподілені системи.

На основі розроблених алгебр поведінок ЗПЗ у ЛКМ було формалізовано файлове ЗПЗ. Його представлення стало основою в розроблених методах виявлення.

Розроблено метод виявлення файлового зловмисного програмного забезпечення на основі динамічного формування поведінкової сигнатури шляхом відстеження викликів API, який дозволяє враховувати наявність поліморфного та метаморфного коду у ЗПЗ для приховування від АПЗ. Використання у вірусах технології заплутування програмного коду унеможливлює виокремлення сталої частини коду, аналіз якої дозволив би прийняти рішення про можливе інфікування. Проте, це стає можливим при використанні в якості основи сигнатури API викликів функцій, тобто набору готових класів, процедур, функцій, структур і констант, що надаються застосунком або операційною системою для використання у зовнішніх програмних продуктах.

При побудові сигнатури вірусу, замість використання всіх API-викликів, що здійснює вірусна програма, розглядались лише критичні API-функції. Критичні API виклики містять усі виклики, які можуть призвести до порушення безпеки, зміни усталеної поведінки роботи системи або виклики, що використовуються для комунікації (модифікація значення системного реєстру, файлового вводу–виводу, API-доступу до мережних ресурсів).

Сигнатура поведінки програми на основі трасування API-викликів може бути представлена у вигляді сукупності двох складових: частоти виклику та характеру взаємодії критичних API-викликів. Аналіз першої складової дозволяє визначити розподіл критичних API-викликів за групами зловмисної активності і відображає кількісну складову сигнатури. Друга складова сигнатури передбачає відображення у векторний простір характеру взаємодії критичних API-функцій вірусної програми та

описує їх взаємозв'язок. Аналіз другої складової сигнатури надає можливість розмежувати вірусні програми від корисних застосунків не тільки за наявністю критичних API-викликів, але й за їх взаємодією між собою. Для формального подання сигнатури вірусної програми представимо її у вигляді кортежу за формулою (11):

$$S = \left\langle A, F, \left\langle D, d_G, n_E \right\rangle \right\rangle, \tag{11}$$

де A – множина API-викликів, що здійснює вірусна програма класу C_i, у процесі власного функціонування; F – множина частот виклику критичних API-функцій; D – вектор степенів вершин графа G_V; d_G – діаметр графа G_V; n_E – кількість ребер графа G_V.

При формуванні вірусної сигнатури, на основі трасування API-викликів спільним для обох етапів є категоризація API викликів за класами. Групування API-функцій за класами критичних дій надає можливість представити у вигляді одного позначення множину функцій, які є подібними за функціональними властивостями та можуть використовуватись різними екземплярами, що належать одному вірусному сімейству. Аналіз множини частот виклику критичних API-викликів дозволяє сформувати параметр приналежності до вірусного класу, що визначає зв'язок між вірусною програмою та одним із класів вірусних програм за кількістю критичних API-викликів. Це є достатньою умовою для віднесення підозрілої програми до одного із класів вірусних програм чи корисних застосунків. Проте, аналіз даного параметра не містить інформації про характер взаємодії між критичними API-викликами і, відповідно, не можливо віднести підозрілу програму до конкретної модифікації вірусу, а лише до певного класу. Тому, друга складова сигнатури вірусної програми покликана відображати характер взаємодії критичних API-функцій вірусної програми та описувати взаємозв'язок між ними, що дозволить здійснити розмежування вірусних програм всередині класу. Процес визначення параметру приналежності до вірусного класу заснований на відмінності між кількістю API-викликів, які здійснюють вірусні програми та довірені застосунки в процесі власного функціонування. Тому, розмежування між класами вірусних програм та довірених застосунків можливе за їх поведінкою, тобто послідовністю критичних API-викликів.

Перший крок методу виявлення передбачає визначення приналежності підозрілої програми до одного із класу вірусних програм або корисних застосунків (рис. 5). З цією метою за допомогою χ^2- тесту визначається різниця між пропорціями частоти критичних API-викликів підозрілої програми (перша частина сигнатури *S*) та частотою критичних API-викликів кожного класу.

У результаті даного кроку буде отримано множину значень параметра приналежності підозрілої програми до кожного з класів. Буде визначено клас C_i, який за частотою критичних API-викликів відповідає частоті викликів критичних API-функцій підозрілої програми. Наступний крок методу передбачає пошук вірусної сигнатури в межах класу C_i. Вектор $V = \left\langle D, d_G, n_E \right\rangle$ складається з 28 числових ознак, в якому 26 ознак визначають степені вершин графа (кожна вершина графа визначається класом критичних API-викликів), а останні дві – діаметр графа та

кількість ребер. Після формування вектора ознак здійснюється його класифікація засобами машинного навчання, що дозволяє віднести досліджувану підозрілу програму до однієї з модифікацій вірусу.

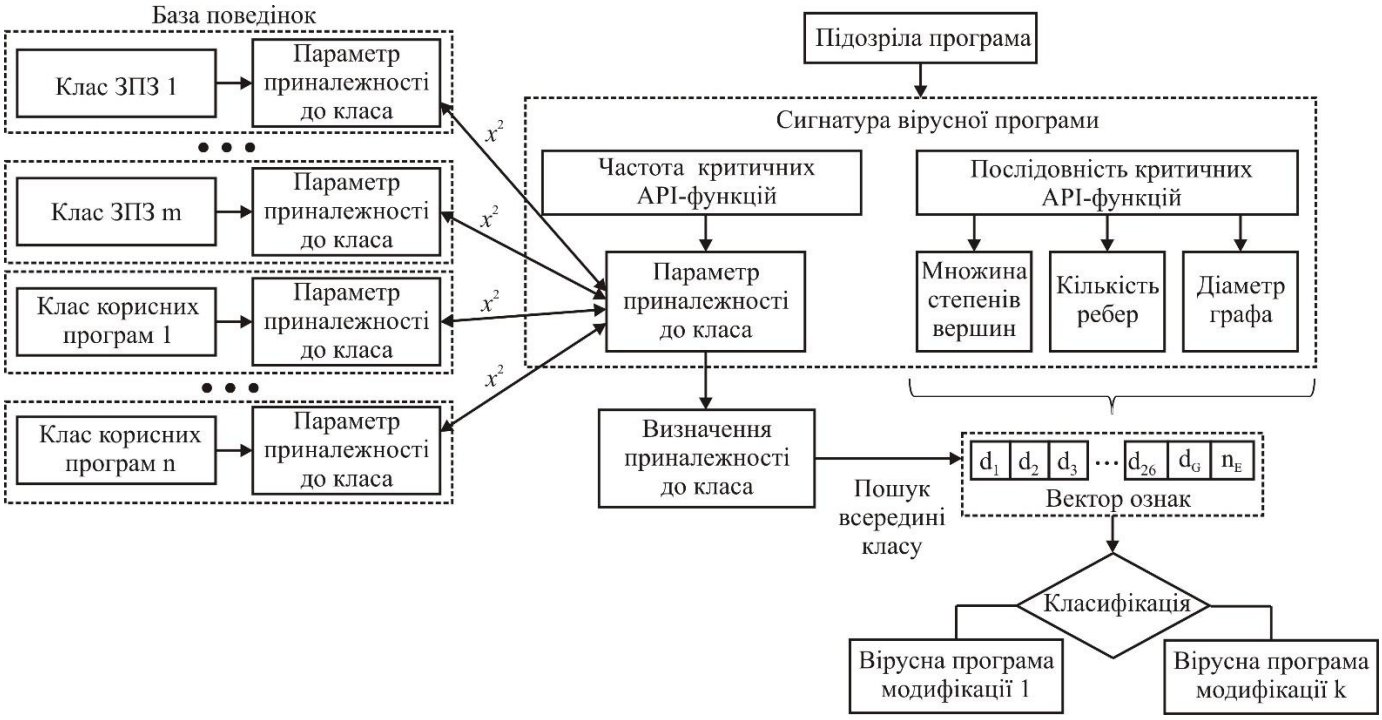

Рис. 5. Схема процесу виявлення вірусних програм
з використанням розробленої поведінкової сигнатури

Метод виявлення файлового ЗПЗ на основі динамічного формування поведінкової сигнатури, шляхом відстеження API-викликів, представимо такими основними кроками:

1. Формування сигнатури поведінки програми.

1.1. Формування сигнатури для класу зловмисних програм на основі відстеження API-викликів кожного екземпляра файлового ЗПЗ.

1.2. Визначення ступеня входження кожного зразка до класу зловмисних програм, тобто знаходження усередненої оцінки для кожного класу файлового ЗПЗ, яка показує відхилення значень в класі.

1.3. Створення бази даних для класів поведінки ЗПЗ та ступенів його приналежності для кожного класу.

2. Виявлення зловмисної програми, представленої поведінковою сигнатурою, на основі трасування викликів API-функцій.

2.1. Моніторинг виконуваних файлів та відстеження їх API-викликів з використанням емулятора процесору ПМ РБС.

2.2. Побудова поведінкової сигнатури підозрілої програми засобами ПМ.

2.3. Пошук сигнатур вірусів у класі та визначення того, чи належить підозріла програма до одного з класів ЗПЗ.

2.4. Віднесення ЗПЗ до відповідного класу ЗПЗ.

Для здійснення перевірки файлів виконуваних програм ПМ РБС створює для кожного програмного об’єкта процес, в якому виконується агент. Кількість таких

агентів, тобто породжених командою для перевірки файлів виконуваних програм, відповідає кількості необхідних для перевірки програмних об’єктів. У процесі постановки такого завдання з метою перевірки ПМ здійснює самоконтроль для уникнення блокування або сповільнення роботи КС. Для вирішення цього завдання будуються сигнатури процесів, і внутрішній планувальник РБС, за певними критеріями визначає процеси, які перевантажують ресурси КС протягом тривалого часу, або можуть привести її активні процеси до стану взаємоблокування. При виявленні таких ситуацій здійснюється призупинення частини процесів, які породжені ПМ РБС.

Таким чином, реалізація методу виявлення файлового ЗПЗ на основі динамічного формування поведінкової сигнатури, шляхом відстеження викликів API засобами РБС, дозволить його багатократне одночасне застосування в різних КС і в кожній КС (за наявності однакових файлів у різних каталогах) та обробку отриманих результатів з різних ПМ для прийняття рішення про наявність у програмному об’єкті вірусного коду.

З метою виявлення файлового ЗПЗ, з наявним у них поліморфним та метаморфним функціоналом, здійснимо аналіз потенційно підозрілої поведінки програм у різних КС та аналіз особливостей заплутування коду (обфускації), які демонструються під час функціонування програми. Функції обфускації можна отримати на основі еквівалентного функціонального блокового пошуку в підозрілій програмі та її модифікованому варіанті. Висновок про наявність поліморфних та метаморфних вірусів проводиться ПМ РБС з використанням зібраної інформації від всіх ПМ мережі, в яких воно досліджувалось.

Метод включає наступні кроки: попередню обробку даних; локалізацію місця для пошуку еквівалентних функціональних блоків (ЕФБ) у виконуваному файлі; пошук ЕФБс і вибір уточнення ЕФБс; одержання кількісних ознак заплутування коду, що базується на порівнянні ЕФБс з ймовірно зловмисною програмою та її видозміненою версією; висновок про наявність поліморфних або метаморфних вірусів. Схему кроків методу зображено на рис. 6.

Основні кроки методу такі:

1. Попередня обробка даних. Розбирання підозрілої програми та отримання зразка коду перед емуляцією. Для створення модифікованої версії підозрілої програми необхідно здійснити її запуск в емуляторах процесорів з різними налаштуваннями у різних КС. У результаті будуть отримані списки операційних кодів підозрілої програми та її модифіковані версії.

2. Локалізація місця для пошуку еквівалентних функціональних блоків у виконуваному файлі.

3. Пошук еквівалентних функціональних блоків. Розбиття списків операційних кодів на блоки для отримання двох наборів ФБ, які представлені у вигляді матриць оцінок появи кодів операцій у функціональних блоках ймовірно зловмисної програми та її видозміненої версії. Пошук ЕФБ передбачає попарне порівняння кожного ФБ з множини з кожним ФБ з використанням метрики подібності. Після здійснення порівняння буде отримано оцінку подібності між двома ФБ, за результатами аналізу якої буде прийнято рішення про подальші дії.

4. Вибір уточнення еквівалентних функціональних блоків і їх порівняння. З

метою вибору уточнення ЕФБ необхідно створити матрицю ймовірностей для послідовності операційних кодів функціональних блоків, які визначені як еквівалентні. Стовпці та рядки матриці визначають коди операцій, які присутні у функціональному блоці. Кожна клітинка матриці містить відношення числа виникнення пари опкодів до загальної кількості кодів операцій в рядку.

Процедура вибору уточнення ЕФБ включає порівняння матриць ймовірностей для послідовності операційних кодів кожного функціонального блоку з кожним ФБ з набору, які є еквівалентними, і вибором мінімального значення оцінки подібності.

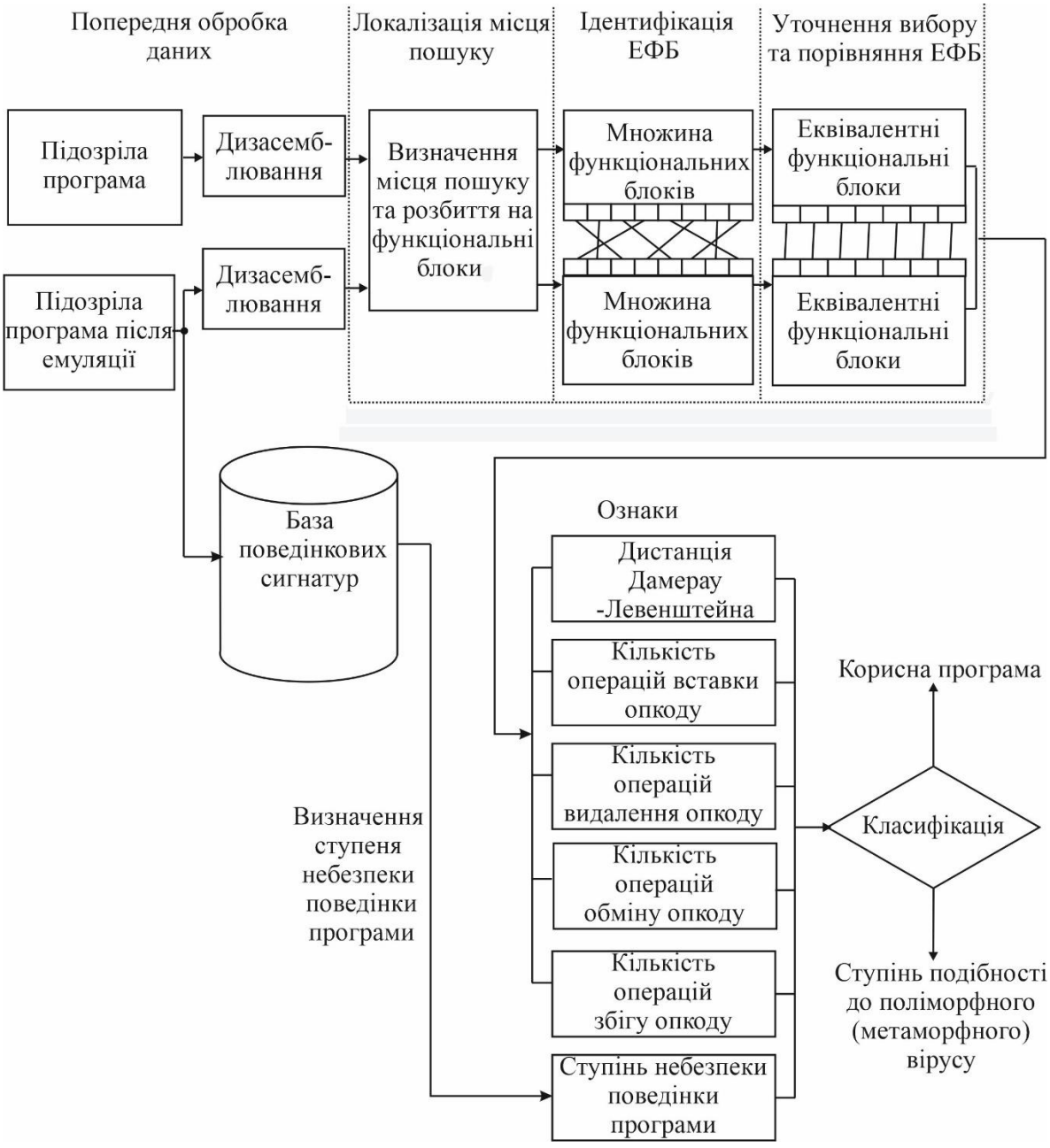

Рис. 6. Схема кроків методу

Для оцінки подібності використано метрику подібності. В результаті отримуємо кількісні особливості обфускації: відстань Дамерау-Левенштейна, кількість вставок, видалення, транспозиції та збігів опкодів.

Розроблений метод виявлення поліморфних та метаморфних вірусів на основі

аналізу функцій обфускації базується на використанні РБС для отримання різних модифікованих версій ймовірно зловмисної програми. Для встановлення факту обфускації здійснюється аналіз подібності функціональних блоків на еквівалентність, які отримані на основі пошуку у ймовірно зловмисній програмі та її модифікованій версії.

Метод виявлення файлового ЗПЗ у ЛКМ полягає в поєднанні роботи програмних агентів, що здійснюють виявлення зловмисного програмного забезпечення в окремих комп'ютерних системах, відповідно до імплементованих в них методів: динамічного формування поведінкової сигнатури шляхом відстеження викликів прикладного програмного інтерфейсу; знаходження поліморфного та метаморфного програмного коду; сканування виконуваних програм шляхом створення для них автономних процесів та відповідних програмних агентів у розподіленій системі. Це дозволяє покращити аналіз та підвищити достовірність виявлення зловмисного програмного забезпечення. Метод складається з таких основних кроків:

1. Здійснити сканування виконуваних файлів із створенням окремих процесів для кожного досліджуваного виконуваного файлу. Всі процеси створюються запуском одного компонента ПМ в КС, який на основі закладеного функціоналу для сканування приймає рішення про потребу застосування відповідних методів виявлення.
2. Здійснити збір даних моніторингу після виявлення певних ймовірно зловмисних проявів в КС у вектор.
3. Сформувати вектор ознак ймовірно підозрілих дій для зібраних даних, компонентами якого є API-функції.
4. Застосувати метод виявлення файлового ЗПЗ на основі динамічного формування поведінкової сигнатури шляхом відстеження викликів API-функцій.
5. Прийняти рішення про місце обробки вектора ймовірно зловмисних дій.
6. Здійснити аналіз завантаженості ресурсів КС, в якій ПМ виявив ймовірно підозрілі дії (крок 5).
7. Здійснити вибір подальших дій на основі результатів, які отримано (крок 6): вибір та застосування методів виявлення файлового ЗПЗ; зупинення досліджуваного процесу; дослідження аналогічних програмних об'єктів в інших КС, де встановлено програмні модулі РБС.
8. Здійснити обробку варіантів подій із залученням інших ПМ РБС та на її основі задіяти відповідну стратегію подальших дій РБС.
9. Обчислити значення ймовірностей у станах ПМ і надіслати вимогу для інших ПМ здійснити обчислення ймовірності бути ураженою для всієї РБС. Цей крок здійснюється позапланово через дослідження наявного зловмисного прояву в одній з КС.
10. Здійснити оптимізацію вектора, що додається в базу зловмисних дій та атак.
11. Сформувати значення ймовірностей перебування в станах з метою надсилання іншим ПМ для визначення стану РБС за встановленими аналітичними виразами.
12. Залучити засоби для здійснення самоконтролю, використовуючи внутрішній планувальник ПМ РБС, та забезпечення стійкості ПМ у КС при групі

подій з кроку 8, які відносяться до зовнішніх впливів.

Кроки 9–12 аналогічні зазначеним у методі виявлення бот-мереж (кроки 9–12).

Таким чином, розроблений метод виявлення файлового ЗПЗ в ЛКМ є основою для організації функціонування програмних модулів РБС і дозволяє здійснити вилучення ймовірно уражених ПМ з РБС, встановлення відношення програмного об’єкта до файлового ЗПЗ на основі обміну і обробки знань всередині РБС, сканування виконуваних файлів створенням для них окремих процесів.

У шостому розділі представлена програмна реалізація РБС, результати її застосування на практиці при виявленні різних типів ЗПЗ у ЛКМ, а також розроблена методика оцінки ефективності роботи РБС.

У розробленому варіанті програмного забезпечення РБС відображено ідеологію децентралізованості, самоорганонізованості, багаторівневості та розподіленості у просторі. Розроблене програмне забезпечення РБС Distributed Multilevel System дозволяє здійснювати його доповнення новими методами, реалізує зв’язуючу частину розподіленої системи і може бути використано для проведення експериментальних досліджень.

Метою експериментів з виявлення бот-мереж була перевірка застосування методу виявлення, роботи класифікатора в структурі розподіленої системи та визначення залежності відсотка виявлених вузлів бот-мережі від їх представлення векторами та різними класифікаторами. Для підготовки до проведення експериментів було здійснено конструювання 28 штучних бот-мереж та отримано коди відомих виявлених бот-мереж. Всі згенеровані бот-мережі згруповано за класами, в яких виділено 25 структурних елементів на трьох стадіях функціонування та 81 функцію. Кожну функцію задано векторами зловмисних дій та атак з врахуванням варіацій, на основі яких побудовано зразки для їх включення в підкласи і класи. Експеримент проводився для класифікатора без додавання екземплярів створених бот-мереж та з ними, тобто здійснювалась перевірка без навчання класифікатора на створених зразках і з попереднім їх віднесенням за класами.

Експерименти передбачали визначення наступних показників ефективності виявлення вузлів бот-мереж для класів і підкласів баєсівського класифікатора:

1) $P_{1,1}$ – відсоток векторів зловмисних дій та атак для вузлів бот-мереж, що належать даному класу відносно всіх тестових зразків, які система віднесла до цього класу з використанням попереднього навчання;

2) $P_{1,2}$ – аналогічно до п. 1, однак без використання попереднього навчання;

3) $P_{2,1}$ – відсоток векторів зловмисних дій та атак для вузлів бот-мереж, що належать даному підкласу класу відносно всіх тестових векторів, які система віднесла до цього підкласу класу в тестовій вибірці (ті, які були правильно віднесені до підкласів) з використанням попереднього навчання;

4) $P_{2,2}$ – аналогічно до п. 3, однак без використання попереднього навчання;

5) $P_{3,1}$ – відсоток правильно виявлених вузлів бот-мереж з використанням попереднього навчання;

6) $P_{3,2}$ – аналогічно до п. 5, однак без використання попереднього навчання;

7) $P_{4,1}$ – відсоток неправильно класифікованих вузлів бот-мереж як корисних додатків (помилка 1-го роду) з використанням попереднього навчання;

8) $P_{4,2}$ – аналогічно до п. 7, однак без використання попереднього навчання;

9) $P_{5,1}$ – відсоток неправильно класифікованих вузлів бот-мереж як таких, що є вузлами бот-мереж, але віднесені не до того класу (помилка 3-го роду), з використанням попереднього навчання;

10) $P_{5,2}$ – аналогічно до п. 9, однак без використання попереднього навчання.

Результати оцінки ефективності виявлення програмного забезпечення вузлів бот-мереж на основі роботи двох класифікаторів для введених класів та підкласів наведено у табл. 1.

Таблиця 1

Результати експерименту з виявлення бот-мереж

Показники експери-менту, %	Отримані значення для різних класів, %							Середні значен-ня, %
	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	
$P_{1,1}$	90,74	84,29	73,66	86,30	94,04	94,18	96,60	89,44
$P_{1,2}$	75,93	63,57	60,22	70,32	68,77	67,60	69,36	67,71
$\lvert P_{1,1}–P_{1,2}\rvert$	14,81	20,72	13,44	15,98	25,27	26,58	27,24	21,73
$P_{2,1}$	85,80	83,57	72,58	85,39	98,88	93,92	96,60	88,42
$P_{2,2}$	74,69	63,57	59,14	70,32	67,37	66,58	67,66	66,80
$\lvert P_{2,1}–P_{2,2}\rvert$	11,11	20	13,44	15,07	31,57	27,34	28,94	21,62
$P_{3,1}$	92,11	84,21	71,93	89,47	90,53	88,42	93,68	87,72
$P_{3,2}$	76,32	57,89	63,16	64,91	71,58	54,74	75,79	65,89
$\lvert P_{3,1}–P_{3,2}\rvert$	15,79	26,32	8,77	24,56	18,95	33,68	17,89	21,83
$P_{4,1}$	7,89	14,47	28,07	10,53	7,37	11,58	6,32	11,70
$P_{4,2}$	21,05	40,79	36,84	31,58	24,21	44,21	22,11	31,97
$\lvert P_{4,1}–P_{4,2}\rvert$	13,16	26,32	8,77	21,05	16,84	32,63	15,79	20,27
$P_{5,1}$	0	1,32	0	0	2,11	0	0	0,01
$P_{5,2}$	2,63	1,32	0	3,51	4,21	1,05	2,11	2,14
$\lvert P_{5,1}–P_{5,2}\rvert$	2,63	0	0	3,51	2,1	1,05	2,11	2,13

У результаті проведення експерименту одержано віднесення векторів, отриманих на основі моніторингу, до потрібного підкласу та класу з точністю до 66 % для класифікатора без введених векторів 28 штучно згенерованих бот-мереж та 88 % – для класифікатора, в який попередньо було додано вектори шляхом здійснення його навчання, зберігаючи в ньому шаблони попередніх наповнень. Перевірка здійснювалась окремо для класів, їх підкласів та в цілому для вузлів. Результати були усереднені і їх дисперсія відносно середнього значення становить 1 %. Різниця відхилення для двох класифікаторів за кожним класом, підкласом і вузлами бот-мереж в цілому становить 21,5 %. Відхилення між різницями відхилень для двох класифікаторів складає за кожним окремим класом, підкласом та вузлом бот-мережі становить менше 5 %, що вказує на точність визначення у різних класах і підкласах. Отже, результат виявлення програмного забезпечення вузлів бот-мереж збігається в

розрізі класів та підкласів для векторів зловмисних дій та атак.

Помилки 1-го роду склали для першого і другого класифікаторів 11,7 % та 31,97 %, відповідно, що пояснюється їх різним наповненням. Помилки 3-го роду – 0,01 % та 2,14 %, відповідно, що пояснюється більш ширшим полем класифікації другого класифікатора через менший обсяг навчальної вибірки. В цілому результати роботи класифікаторів показали можливість їх застосування для задач виявлення бот-мереж.

Розглянемо використання розробленої РБС на основі функціонування її підсистеми, що дозволяє виявляти файлове ЗПЗ за рахунок дослідження підозрілого коду програмними модулями, які розміщені у різних комп'ютерних системах. Для проведення експериментів було залучено ЛКМ, що складалась з 20 комп'ютерних систем. Кожна КС була обладнана віртуальним середовищем на основі Qemu, яке задіювалось ПМ розробленої системи для дослідження поведінки ймовірно зловмисних програм і отримання викликів API функцій. Дослідження виконуваних програм здійснювалось на трьох етапах їх функціонування: потрапляння в КС, активізації та виконання закладених функцій. Кожен ПМ використовував базу поведінкових сигнатур файлового ЗПЗ на її різних етапах функціонування. Було згенеровано 600 програмних об'єктів з функціональним навантаженням чотирьох типів файлового ЗПЗ, по 150 кожного. В досліджувану підмножину файлових вірусів було включено лише ті, які не містили поліморфного або метаморфного коду і не були троянськими програмами. Результати проведеного експерименту представлено в табл. 2.

Таблиця 2

Результати експерименту для файлового ЗПЗ

Програмні об'єкти з наявним в них ЗПЗ		Кількість програм, виявлених як підозрілі	Відсоток виявлення, %	Кількість ПМ, які залучались для дослідження протягом всього експерименту	Кількість ПМ, які були заблоковані іншими ПМ розробленої системи, під час виявлення
Файлові віруси	150	146	97,3	0	2
Поліморфні віруси	150	144	96,0	57	7
Метаморфні віруси	150	138	92,0	24	3
Троянські програми	150	142	94,7	2	5
Разом	600	570	95,0	83	17
Середні значення				20,75	5,7

Крім того, за результатами проведено експерименту було також встановлено кількість ПМ, які залучались для дослідження протягом всього експерименту, та

кількість ПМ, що були заблоковані іншими ПМ розробленої системи, під час виявлення ЗПЗ. Це підтверджує застосування решти компонентів розподіленої системи в процесі виявлення ЗПЗ окремим ПМ.

Для проведення порівняльного аналізу було обрано наступні відомі антивірусні засоби: Symantec Endpoint Protection (версія 14.2.1); Panda Endpoint Protection (версія 8.42.00); Malwarebytes Endpoint Security (версія 3.8.3); McAfee Endpoint Protection Suite (версія 10.6.0); AVG Internet Security Business Edition (версія 19.1.2360). Проведений розрахунок на основі формул ROC-аналізу достовірності виявлення існуючими мережними АПЗ на згенерованому наборі ЗПЗ і його результати представлено діаграмою на рис. 7, в якій, зокрема, відображено порівняльний аналіз розробленої системи Distributed Multilevel System з існуючими АПЗ щодо помилок першого роду (хибні спрацювання) та помилок другого роду (хибно-негативні результати).

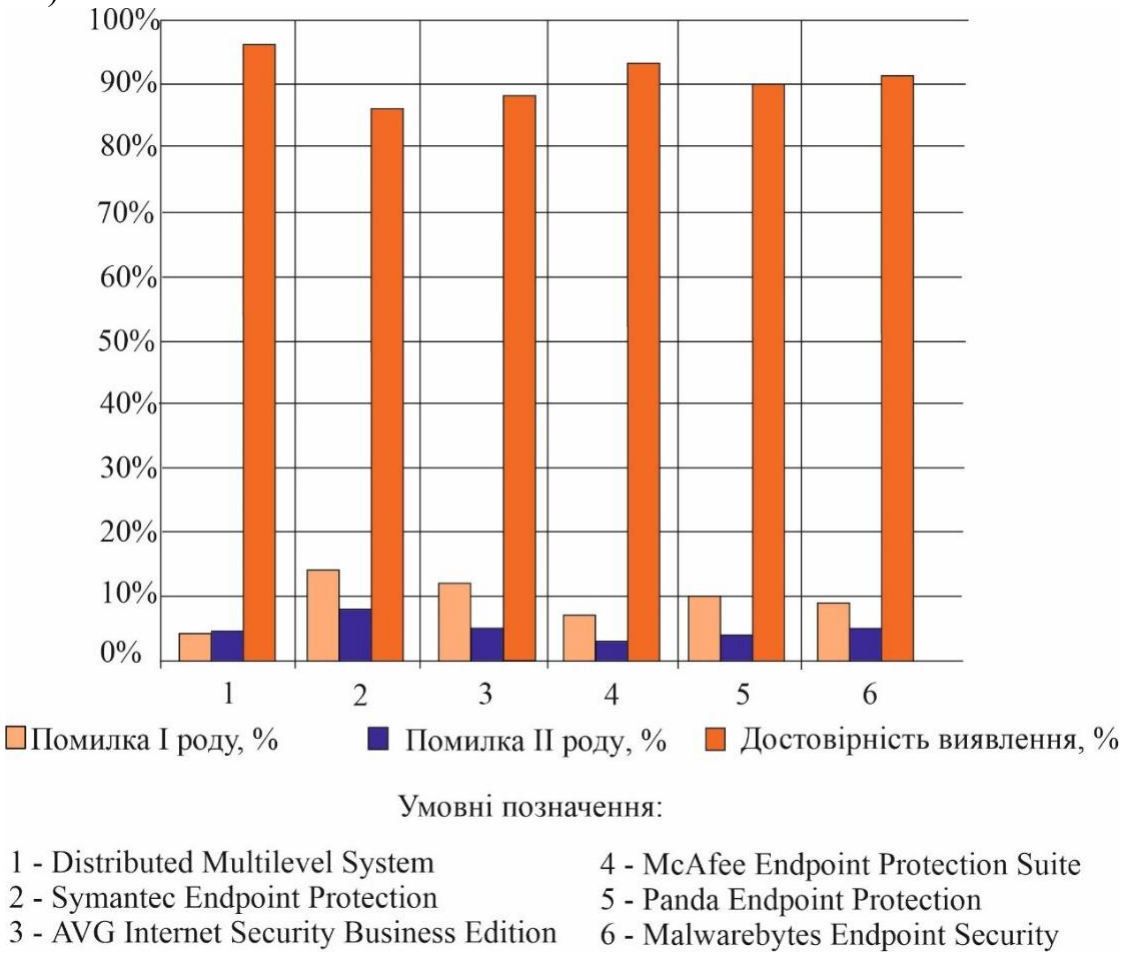

Рис. 7. Порівняльний аналіз розробленої системи Distributed Multilevel System з існуючими мережними АПЗ

Результати експериментальних досліджень з використанням розробленої системи Distributed Multilevel System підтверджують правильність наукових положень розроблених методів та ефективність архітектури РБС, оскільки її впровадження підвищує достовірність виявлення на 5–12 % в мережному представленні порівняно з хостовим, та на 2–4 % у порівнянні з існуючими мережними АПЗ виявлення файлового ЗПЗ. Результати проведених

експериментальних досліджень показують, що рівень достовірності виявлення ЗПЗ при застосуванні розробленою системою Distributed Multilevel System складає близько 95 %, що на 2–4 % вище порівняно з існуючими антивірусними програмними засобами (рис. 7). Застосування розробленої системи Distributed Multilevel System дозволяє досягти зниження рівня помилок першого роду до 5%, що на 3–4 % нижче порівняно з існуючими мережними антивірусними програмними засобами (рис. 7).

Здійснення оцінки ефективності розробленої РБС було проведено, порівнюючи її використання в хостовому представленні одним ПМ та РБС, в якій більше одного ПМ. Така оцінка дозволила встановити додаткові витрати за часом на операції обміну між ПМ та витратами на залучення додаткових ресурсів КС. До критеріїв ефективності для розробленої РБС було віднесено такі: витрати часу на здійснення процесу виявлення хостом; витрати часу на здійснення процесу виявлення всією РБС; оперативність у прийнятті рішень; ресурсоспоживання; достовірність виявлення за часом, який витрачався на обробку помилок першого, другого та третього роду. Для цих критеріїв було встановлено аналітичні залежності, за якими отримано числові значення. Після унормування отриманих результатів ефективності окремо для кожного критерію було знайдено їх добуток, який виразив ефективність РБС у локальних комп'ютерних мережах за формулою (12) наступним чином:

$$E = p_{d,norm} \cdot k_{e,norm} \cdot r_{e,norm}, \quad (12)$$

де $p_{d,norm}$ – унормований показник достовірності виявлення за часом; $k_{e,norm}$ – унормований показник ефективності витрат часу; $r_{e,norm}$ – унормований показник ефективності ресурсоспоживання.

Загальна ефективність роботи розробленої РБС у ЛКМ, з використанням емуляторів процесора із зміненими налаштуваннями, складає: $E \approx 0{,}91 \cdot 0{,}97 \cdot 0{,}94 \approx 0{,}83$.

Отже, результати дослідження достовірності розробленої РБС у ЛКМ показують, що застосування розробленого програмного забезпечення дозволяє підвищити рівень достовірності виявлення на 5–12 % порівняно з існуючими антивірусними програмними засобами та досягти зниження рівня помилок першого роду до 5 %.

ВИСНОВКИ

У дисертаційній роботі вирішено актуальну науково-технічну проблему – здійснено розвиток теорії і практики створення розподілених систем виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах для покращення ефективності його виявлення. Вирішення даної проблеми має важливе значення в усіх галузях, де активно застосовуються локальні комп'ютерні мережі. При цьому отримано такі основні наукові й практичні результати:

1. Проведено аналіз сучасних мережних систем виявлення зловмисного програмного забезпечення, який показав невисоку ефективність таких систем, що зумовлено низькою достовірністю виявлення нового та існуючого зловмисного програмного забезпечення, через використання ними комбінації технологій

приховування своєї присутності та поширення. В якості напряму дослідження проблеми було вибрано розподілені системи виявлення в локальних комп'ютерних мережах.

2. Розроблена удосконалена модель архітектури розподіленої системи виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах базується на комплексному врахуванні вимог розподіленості, децентралізованості, багаторівневості та самоорганізованості, що дозволяє створювати на її основі розподілені системи та їх компоненти, які функціонуватимуть автономно і самостійно прийматимуть рішення про наявність зловмисного програмного забезпечення та нарощення своїх функціональних можливостей.

3. Розроблена модель архітектури типових компонентів розподіленої багаторівневої системи виявлення зловмисного програмного забезпечення на основі структур Кріпке з представленням компонентів через стани, в яких вони можуть перебувати під час функціонування. Вона дозволила враховувати перебування програмних модулів у різних станах і стала основою для визначення стану безпеки всієї розподіленої системи та її компонентів. Модель її архітектури дозволяє здійснювати збільшення кількості рівнів системи без її зміни. Основою архітектури РБС виступають програмні модулі з однаковими архітектурами, але при цьому кожен з них може самостійно приймати рішення на основі різних даних, зібраних з різних КС локальної мережі.

4. Розроблений метод взаємодії компонентів розподіленої багаторівневої системи виявлення зловмисного програмного забезпечення на основі підтримки її цілісності та визначення порядку передачі знань між її компонентами і використання встановлених аналітичних залежностей між рівнями безпеки програмних модулів та рівнем безпеки всієї розподіленої багаторівневої системи, дозволяє системі автономно змінювати свою архітектуру та функції без втручання користувача, а також визначати стратегію своєї подальшої роботи. Метод є основою для розробки зв'язуючої частини програмного забезпечення, яка організовує взаємодію компонентів розподіленої багаторівневої системи виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах.

5. Розроблені алгебраїчні системи та алгебри, з введеними операціями на множині ЗПЗ, стали основою для створення поведінкових сигнатур ЗПЗ з метою їх формалізованого представлення в системах виявлення. Особливістю розроблених алгебраїчних систем є структуризація ЗПЗ за типами, яка дозволяє здійснювати їх розподіл і віднесення до підмножин на основі характеристичних властивостей ЗПЗ для проведення ідентифікації та класифікації. Формалізовані властивості ЗПЗ відображені в розроблених алгебрах і заданих моделях. Удосконалені моделі типів зловмисного програмного забезпечення представлені їх алгебрами поведінки стали основою створення базису поведінкових сигнатур, враховують особливості їх функціонування в локальних комп'ютерних мережах і дозволяють здійснити класифікацію за типами поведінки.

6. Розроблена еталонна модель бот-мереж, в основі якої її типові компоненти, задані відповідними функціями, відображають типові дії мережного зловмисного програмного забезпечення в локальних комп'ютерних мережах. Вона базується на основі компонентів трьох рівнів, що дало можливість представити відомі бот-мережі,

які були виявлені і певним чином класифіковані за характерними ознаками. Розроблені для компонентів еталонної моделі та моделей типових бот-мереж числові характеристики стали основою для застосування методу виявлення бот-мереж у локальних комп'ютерних мережах.

7. Розроблено метод виявлення бот-мереж у локальних комп'ютерних мережах, суть якого полягає в здійсненні активного моніторингу системних подій та узгодженій взаємодії компонентів розподіленої системи при прийнятті рішення, надав змогу створювати засоби, які здатні інтегруватись у розподілену систему та класифікувати бот-мережі за їх поведінковими сигнатурами, що формуються закладеними в їх компоненти функціями.

8. Розроблено метод виявлення файлового зловмисного програмного забезпечення в локальних комп'ютерних мережах, який полягає в поєднанні роботи програмних агентів, що здійснюють виявлення зловмисного програмного забезпечення в окремих комп'ютерних системах, відповідно до імплементованих в них методів: динамічного формування поведінкової сигнатури шляхом відстеження викликів прикладного програмного інтерфейсу; знаходження поліморфного та метаморфного програмного коду; сканування виконуваних програм шляхом створення для них автономних процесів та відповідних програмних агентів у розподіленій системі. Це дозволило покращити аналіз і підвищити достовірність виявлення зловмисного програмного забезпечення.

9. Розроблено метод виявлення файлового зловмисного програмного забезпечення, який базується на основі динамічного формування поведінкової сигнатури шляхом відстеження API-викликів. Він може бути використаний для виявлення інших видів вірусних програм, зокрема, нових версій існуючих вірусів. Метод включає формування сигнатури вірусної програми на основі трасування API-викликів, що дозволяє здійснити виявлення вірусної програми, яка представлена розробленою поведінковою сигнатурою з бази сигнатур. Поведінкова сигнатура включає критичні API-виклики за групами зловмисної активності та відображає частоту їх входження, а також характер взаємодії критичних API-функцій вірусної програми та описує взаємозв'язок між критичними API-функціями. Це надає можливість розмежувати вірусні програми від корисних застосунків не тільки за наявністю критичних API-викликів, але й за їх взаємодією між собою. Для здійснення виявлення використовується класифікація.

10. Для файлового ЗПЗ, яке використовує техніки заплутування свого коду, розроблено метод виявлення поліморфних та метаморфних вірусів на основі аналізу функцій обфускації. Особливістю методу є аналіз програмного об'єкта та його модифікованих версій, отриманих від різних ПМ РБС, і подальший аналіз на основі пошуку еквівалентних функціональних блоків. Це дозволяє здійснити більш детальний аналіз коду програмного об'єкта на наявність поліморфних та метаморфних вірусів.

11. Розроблено програмне забезпечення РБС Distributed Multilevel System, яке реалізує запропоновані теоретичні основи розподілених систем та підтверджує можливість їх практичного створення. Воно дозволяє здійснити перевірку достовірності виявлення ЗПЗ на основі запропонованих рішень, проведенням експериментів, порівняно з існуючими мережними АПЗ.

12. Впровадження розподіленої багаторівневої системи Distributed Multilevel System виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах здійснено в Державному підприємстві «Новатор» (відділ автоматизованих систем управління), товаристві з обмеженою відповідальністю «ITT telecommunication company», товаристві з обмеженою відповідальністю «ЮКС++», софтовій компанії CYPRESS і дозволило підвищити достовірність виявлення нового та існуючого ЗПЗ порівняно з відомим мережними антивірусними засобами на 5–12 % та досягти зниження рівня помилок першого роду до 5 %.

Результати дисертаційної роботи можуть бути запропоновані для використання науковими організаціями і підприємствами, які займаються розробкою та впровадженням мережних антивірусних засобів з метою покращення ефективності їх функціонування.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковані основні наукові результати дисертації

1. Савенко О. С. Метод антивірусного діагностування персональних комп'ютерів на основі матриць інцидентності / О. С. Савенко, В. М. Джулій, В. М. Стецюк // Вісник Технологічного університету Поділля. Технічні науки. – 2000. – № 3. – С. 136–139. – Розроблено представлення файлового ЗПЗ на основі матриць інцидентності та метод виявлення файлового ЗПЗ.

2. Савенко О. С. Методика генерації матриць інцидентності для використання в антивірусних програмних засобах / О. С. Савенко // Вісник Технологічного університету Поділля. Технічні науки. – 2003. – № 3, т. 2. – С. 48–56.

3. Савенко О. С. Генерація моделей комп'ютерних вірусних програм в системі оцінки достовірності результатів роботи антивірусних засобів / О. С. Савенко, С. В. Мостовий // Вісник Хмельницького національного університету. Технічні науки. – 2005. – № 4, т. 1. – С. 198–200. – Розроблено систему представлення файлового ЗПЗ на основі матриць та механізм їх отримання.

4. Савенко О. С. Дослідження методів антивірусного діагностування комп'ютерних мереж / О. С. Савенко, С. М. Лисенко // Вісник Хмельницького національного університету. Технічні науки. – 2007. – № 2, т. 2. – С. 120–126. – Розроблено типову структуру мережного ЗПЗ.

5. Савенко О. С. Дослідження та аналіз блокування процесів в комп'ютерній системі / О. С. Савенко, Ю. П. Кльоц, С. В. Мостовий // Вісник Хмельницького національного університету. Технічні науки. – 2007. – № 3, т. 1. – С. 248–251. – Здійснено аналіз та формалізовано представлення станів процесів.

6. Савенко О. С. Життєвий цикл процесів комп'ютерної системи / О. С. Савенко, С. В. Мостовий // Радіоелектронні і комп'ютерні системи. – 2010. – № 7 (48). – С. 35–38. – Здійснено формалізоване представлення станів процесів.

7. Локазюк В. М. Модель прогнозування стану взаємоблокування процесів комп'ютерної системи / В. М. Локазюк, О. С. Савенко, С. В. Мостовий // Вісник Вінницького політехнічного інституту. – 2011. – № 4. – С. 130–133. – Розроблено представлення граничних станів процесів.

8. Савенко О. С. Діагностування комп'ютерних систем на наявність шкідливого

програмного забезпечення на основі антивірусної мультиагентної системи / О. С. Савенко, С. М. Лисенко, А. Ф. Крищук // Вісник Національного університету «Львівська політехніка». – 2011. – № 5. – С. 99–105. – Розроблено базові характеристики компонентів розподілених систем виявлення ЗПЗ.

9. Савенко О. С. Адаптивна інформаційна технологія виявлення троянських програм в комп'ютерних системах / О. С. Савенко, С. М. Лисенко // Комп'ютинг. – 2011. – Т. 10, № 3. – С. 85–92. – Розроблено концепцію поділу файлового ЗПЗ на класи за їх поведінкою.

10. Берников А. Р. Поиск вредоносных программ в распределенных тренажерах с использованием технологии нечеткой логики / А. Р. Берников, Р. П. Графов, С. Н. Лысенко, О. С. Савенко // Информационные технологии. *(РИНЦ)* – 2011. – № 10. – С. 42–47. – Розроблено методологію вирішення проблеми виявлення файлового ЗПЗ у розподілених системах на прикладі розробки методу виявлення троянських програм.

11. Савенко О. С. Прогнозування потрапляння процесів у стан взаємоблокування / О. С. Савенко, С. В. Мостовий // Вісник Вінницького політехнічного інституту. – 2013. – № 2. – С. 81–86. – Здійснено визначення граничного стану для групи процесів, які наближаються до стану взаємоблокування.

12. Нічепорук А. О. Моделі життєвого циклу поліморфних вірусів / А. О. Нічепорук, О. С. Савенко // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2013. – № 11. – С. 64–71. – Розроблено поділ файлового ЗПЗ на класи за їх поведінкою.

13. Савенко О. С. Моделі рівнів поліморфних комп'ютерних вірусів / О. С. Савенко, С. М. Лисенко, А. О. Нічепорук // Вісник Вінницького політехнічного інституту *(Index Copernicus)*. – 2015. – № 2. – С. 75–83. – Розроблено деталізоване множиною дій представлення файлового ЗПЗ, яке здійснює заплутування програмного коду, на основі застосування його моделей.

14. Савенко О. С. Програмне забезпечення інформаційної технології моделювання поширення вірусних кодів в гетерогенних мережах / О. С. Савенко // Вісник Хмельницького національного університету. Технічні науки *(Index Copernicus)*. – 2017. – № 1. – С. 144–148.

15. Савенко О. С. Распределенная многоуровневая сетевая система обнаружения метаморфных вирусов в локальных компьютерных сетях / О. С. Савенко // Вестник Брестского государственного технического университета (физика, математика, информатика). – 2017. – № 5 (107). – С. 40–44.

16. Савенко О. С. Критерії класифікації методів виявлення шкідливого програмного забезпечення / О. С. Савенко // Вісник Хмельницького національного університету. Технічні науки. – 2018. – № 1. – С. 23–27.

17. Савенко О. С. Модель та архітектура розподіленої багаторівневої системи виявлення шкідливого програмного забезпечення в локальних комп'ютерних мережах / О. С. Савенко // Вісник Хмельницького національного університету. Технічні науки. – 2018. – № 2. – С. 153–163.

18. Савенко О. С. Формалізація шкідливого програмного забезпечення в локальних комп'ютерних мережах / О. С. Савенко // Вісник Хмельницького національного університету. Технічні науки. – 2018. – № 3. – С. 145–154.

19. Савенко О. С. Архітектура багаторівневої програмної системи виявлення шкідливого програмного забезпечення в локальних комп'ютерних мережах / О. С. Савенко, В. І. Грибинчук, М. О. Кульчицький // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2018. – № 30–31. – С. 132–140. – Розроблено архітектуру РБС та її представлення UML-діаграмами.

20. Савенко О. С. Архітектура розподіленої багаторівневої системи виявлення шкідливого програмного забезпечення в локальних комп'ютерних мережах / О. С. Савенко // Вчені записки Таврійського національного університету. Технічні науки. – 2018. – Т. 29 (68), № 2. – С. 172–181.

21. Савенко О. С. Формування сигнатури поведінки програми на основі трасування API викликів / О. С. Савенко, А. О. Нічепорук, А. А. Нічепорук, Ю. О. Нічепорук // Електротехнічні та комп'ютерні системи. – 2018. – № 29 (105). – С. 67–77. – Розроблено метод отримання поведінкової сигнатури на основі трасування API-викликів.

Праці, які засвідчують апробацію матеріалів дисертації

22. Savenko O. Multi-agent based approach of botnet detection in computer systems / O. Savenko, S. Lysenko, A. Kryschuk // Communications in Computer and Information Science, ISSN: 1865-0929 *(Scopus, Web of Science*). – 2012. – Vol. 291. – Pp. 171–180.

23. Pomorova O. Multi-Agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic / O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk // Communications in Computer and Information Science, ISSN: 1865–0929 *(Scopus, Web of Science)*. – 2013. – Vol. 370. – Pp. 146–156.

24. Pomorova O. A Technique for detection of bots which are using polymorphic code / O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, A. Nicheporuk // Communications in Computer and Information Science, ISSN: 1865–0929 *(Scopus, Web of Science)*. – 2014. – Vol. 431. – Pp. 265–276.

25. Savenko O. Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search / O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko // CEUR-WS, ISSN: 1613–0073 *(Scopus)*. – 2017. – Vol. 1844. – Pp. 555–569.

26. Lysenko S. Information technology for botnets detection based on their behaviour in the corporate area network / S. Lysenko, O. Savenko, K. Bobrovnikova, A. Kryshchuk, B. Savenko // Communications in Computer and Information Science, ISSN: 1865–0929 *(Scopus, Web of Science)*. – 2017. – Vol. 718. – Pp. 166–181.

27. Markowsky G. The technique for metamorphic viruses' detection based on its obfuscation features analysis / G. Markowsky, O. Savenko, S. Lysenko, A. Nicheporuk // CEUR-WS, ISSN: 1613–0073 *(Scopus)*. – 2018. – Vol. 2104. – Pp. 680–687.

28. Markowsky G. Distributed Malware Detection System Based on Decentralized Architecture in Local Area Networks / G. Markowsky, O. Savenko, A. Sachenko // Advances in Intelligent Systems and Computing, ISSN: 2194–5357 *(Scopus)*. – 2019. – Vol. 871. – Pp. 582–598.

29. Savenko O. The Technique for Computer Systems Trojan Diagnosis in the Monitor Mode / O. Savenko, S. Lysenko // Proceedings of the 6-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Prague (Czech Republic), September 15–17, 2011 *(Scopus)*. – Prague,

2011. – Pp. 770–774.

30. Savenko O. Botnet detection technique for corporate area network / O. Savenko, S. Lysenko, A. Kryshchuk, Y. Klots / Proceedings of the 7-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Berlin (Germany), September 12–14, 2013 *(Scopus)*. – Berlin, 2013. – Pp. 363–368.

31. Lysenko S. DNS-based Anti-evasion Technique for Botnets Detection / S. Lysenko, O. Pomorova, O. Savenko, A. Kryshchuk and K. Bobrovnikova // Proceedings of the 8-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Warsaw (Poland), September 24–26, 2015 *(Scopus, Web of Science)*. – Warsaw, 2015. – Pp. 453–458.

32. Pomorova O. A Technique for the Botnet Detection Based on DNS-Traffic Analysis / O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, K. Bobrovnikova // Proceedings of the 22-nd International Conference Computer Networks, Brunów (Poland), June 16–19, 2015 *(Scopus, Web of Science)*. – Brunów, 2015. – Vol. 522. – Pp. 127–138.

33. Pomorova O. Anti-evasion Technique for the Botnets Detection Based on the Passive DNS Monitoring and Active DNS Probing / O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, K. Bobrovnikova // Proceedings of the 23-nd International Conference Computer Networks, Brunów (Poland), June 14–17, 2016 *(Scopus, Web of Science)*. – Brunów, 2016 – Vol. 608. – Pp. 83–95.

34. Savenko O. Approach for the Unknown Metamorphic Virus Detection / O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko // Proceedings of the 8-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Bucharest (Romania), September 21–23, 2017 *(Scopus, Web of Science)*. – Bucharest, 2017. – Pp. 71–76.

35. Markowsky G. Distributed System for Detecting the Malware in LAN / G. Markowsky, O. Savenko, A. Sachenko // Proceedings of the 2018 IEEE 13th International Scientific and Technical Conference on Computer Science and Information Technologies, CSIT’2018, Lviv (Ukraine), September 11–14, 2018 *(Scopus, Web of Science)*. – Lviv, 2018. – Pp. 306–309.

36. Графов Р. П. Забезпечення надійності розподільних мереж на основі динамічної структурної реконфігурації / Р. П. Графов, О. С. Савенко // Інформаційні технології та комп’ютерна інженерія. – 2005. – №3. – С. 241–246.

37. Савенко О. С. Метод виявлення бот-мереж розподіленими системами на основі самоорганізації / О. С. Савенко // Штучний інтелект. – 2018. – № 4 (82). – С. 58–72.

38. Savenko O. Intelligent method of the search of the trojan program in computer systems / O. Savenko, S. Lysenko // Праці IV міжнародної конференції «Сучасні комп'ютерні системи та мережі: розробка та використання», (ACSN'2009), (Львів, 17–19 грудня 2009). – Львів, 2009. – С. 154–158.

39. Savenko O. Software for computer systems trojans diagnosing as a safety-case tool / O. Savenko, S. Lysenko // Proceedings of the 1-st International Workshop on Critical infrastructure safety and security, CrlSS-DESSERT'11, Kirovograd, May 11–13, 2011. – Kirovograd, 2011. – Pp. 353–361.

40. Савенко О. С. Дослідження антивірусних технологій діагностування

комп'ютерних систем на наявність шкідливого програмного забезпечення / О. С. Савенко, С. М. Лисенко, А. Ф. Крищук // Праці XII міжнародної науково-практичної конференції «Сучасні інформаційні та електронні технології», (СІЕТ-2011), (Одеса, 27–31 травня 2011). – Одеса, 2011. – С. 165–166.

41. Savenko O. Interoperability of distributed multiple system for malware detection based on components levels of safety / O. Savenko // Проблеми інформаційних технологій. – 2018. – № 24. – С. 78–92.

42. Савенко О. С. Функційна модель бота як складової ботнет-мережі / О. С. Савенко, С. М. Лисенко, А. Ф. Крищук // Тези доповідей І Міжнародної науково-технічної конференції «Захист інформації і безпека інформаційних систем», (Львів, 2012). – Львів, 2012. – С. 123–124.

43. Савенко О. С. Розподілена апаратно-програмна система та методи захисту інформації в комп'ютерних системах локальних мереж / О. С. Савенко // Наукові праці Чорноморського національного університету ім. П. Могили. Комп'ютерні технології. – 2018. – Т. 320. Вип. 308. – С. 72–75.

44. Савенко О. С. Метод взаємодії компонентів розподіленої системи виявлення зловмисного програмного забезпечення в локальних обчислювальних мережах / О. С. Савенко, А. О. Нічепорук // Збірник тез доповідей XIV Міжнародної конференції «Контроль і управління в складних системах», (КУСС-2018), (Вінниця, 15–17 жовтня, 2018). – Вінниця, 2018. – С. 37.

45. Савенко О. С. Формалізоване структурування шкідливого програмного забезпечення на основі алгебраїчних систем / О. С. Савенко // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2018. – №1. – С. 67–72.

46. Савенко О. С. Розподілена система виявлення зловмисного програмного забезпечення та метод взаємодії її компонент в локальних мережах / О. С. Савенко // Матеріали доповідей V Міжнародної науково-практичної конференції «Інформаційні технології та взаємодії», (Київ, КНУ ім. Т. Шевченка, 20–21 листопада 2018). – Київ, 2018. – С. 308–309.

Публікації, які додатково відображають наукові результати дисертації

47. Пат. на корисну модель 108238 Україна, МПК G06F 21/55 Мультиагентний спосіб локалізації бот-мереж у корпоративних комп'ютерних мережах / О. В. Поморова, О. С. Савенко, А. Ф. Крищук, С. М. Лисенко, К. Ю. Бобровнікова, А. О. Нічепорук; заявник і патентовласник Хмельницький національний університет. – № u201600127; заявл. 04.01.2016; опубл. 11.07.2016, Бюл. № 13/2016.

48. Пат. на корисну модель 118456 Україна, МПК G06F 21/55 Спосіб виявлення метаморфних вірусів на основі статистичних метрик для визначення еквівалентних функціональних програмних блоків / О. С. Савенко, С. М. Лисенко, К. Ю. Бобровнікова, А. О. Нічепорук, Б. О. Савенко; заявник і патентовласник Хмельницький національний університет. – № u201701743; заявл. 23.02.2017; опубл. 10.08.2017, Бюл. № 15/2017.

49. Пат. на корисну модель 118663 Україна, МПК G06F 21/55 Спосіб ідентифікації бот-мереж у корпоративних комп'ютерних мережах на основі аналізу DNS-трафіку / О. С. Савенко, С. М. Лисенко, К. Ю. Бобровнікова, А. О. Нічепорук, Б. О. Савенко; заявник і патентовласник Хмельницький національний університет. –

№ u201612041; заявл. 28.11.2016; опубл. 28.08.2017, Бюл. № 16/2017.

50. А. с. 80223 Україна. Комп'ютерна програма пошуку та визначення еквівалентних функціональних блоків у виконуваних файлах для ідентифікації ознак метаморфних вірусів в локальних комп'ютерних мережах / А. О. Нічепорук, О. С. Савенко, С. М. Лисенко. 2018.

51. А. с. 80445 Україна. Розподілена комп'ютерна програма для виявлення зловмисного програмного забезпечення в локальних обчислювальних мережах на основі аналізу поведінкових сигнатур / О. С. Савенко. 2018.

АНОТАЦІЯ

Савенко О. С. Теорія та практика створення розподілених систем виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах. – *На правах рукопису.*

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 05.13.05 – Комп'ютерні системи та компоненти. – Національний університет «Львівська політехніка» Міністерства освіти і науки України, Львів, 2019.

Дисертація присвячена вирішенню актуальної науково-технічної проблеми розроблення теорії і практики створення розподілених систем виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах для покращення ефективності його виявлення.

В роботі розроблено удосконалену модель архітектури розподіленої системи виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах та модель архітектури її типових компонентів на основі структур Кріпке, а також метод взаємодії компонентів розподіленої багаторівневої системи виявлення зловмисного програмного забезпечення для підтримки її цілісності та визначення порядку передачі знань між її компонентами.

Розроблені алгебраїчні системи та алгебри з введеними операціями на множині ЗПЗ для створення поведінкових сигнатур ЗПЗ з метою їх формалізованого представлення в системах виявлення. Розроблено метод виявлення бот-мереж у локальних комп'ютерних мережах, суть якого полягає в здійсненні активного моніторингу системних подій та узгодженій взаємодії компонентів розподіленої системи при прийнятті рішення. Розроблено метод виявлення файлового зловмисного програмного забезпечення в локальних комп'ютерних мережах, який полягає в поєднанні роботи програмних агентів, що здійснюють виявлення зловмисного програмного забезпечення в окремих комп'ютерних системах.

Ключові слова: розподілені системи, зловмисне програмне забезпечення, засоби захисту інформації, самоорганізовані системи, децентралізовані системи, методи виявлення.

АННОТАЦИЯ

Савенко О. С. Теория и практика создания распределенных систем обнаружения вредоносных программ в локальных компьютерных сетях. – *На*

правах рукописи.

Диссертация на соискание ученой степени доктора технических наук по специальности 05.13.05 – Компьютерные системы и компоненты. – Национальный университет «Львивська политэхника» Министерства образования и науки Украины, Львив, 2019.

Диссертация посвящена решению актуальной научно-технической проблемы разработки теории и практики создания распределенных систем обнаружения вредоносных программ в локальных компьютерных сетях для улучшения эффективности обнаружения.

В работе разработаны усовершенствованная модель архитектуры распределенной системы обнаружения вредоносных программ в локальных компьютерных сетях и модель архитектуры ее типовых компонентов на основе структур Крипке, а также метод взаимодействия компонентов распределенной многоуровневой системы обнаружения вредоносных программ для поддержания ее целостности и определения порядка передачи знаний между ее компонентами.

Разработанны алгебраические системы и алгебры, с введенными операциями на множестве вредоносных программ, для создания поведенческих сигнатур вредоносных программ с целью их формализованного представления в системах обнаружения. Разработан метод выявления бот-сетей в локальных компьютерных сетях, суть которого заключается в осуществлении активного мониторинга системных событий и согласованного взаимодействия компонентов распределенной системы при принятии решения. Разработан метод выявления файловых вредоносных программ в локальных компьютерных сетях, который заключается в совмещении работы программных агентов, осуществляющих выявление вредоносных программ в отдельных компьютерных системах.

Ключевые слова: распределенные системы, вредоносное программное обеспечение, средства защиты информации, самоорганизующиеся системы, децентрализованные системы, методы обнаружения.

ABSTRACT

Savenko O. S. The Theory and Practice of Creating Distributed Malware Detection Systems on Local Area Networks. – *On the rights of the Manuscript.*

The thesis for the degree of Doctor of technical sciences, specialty 05.13.05 – Computer Systems and Components. – Lviv Polytechnic National University, Ministry of Education and Science of Ukraine, Lviv, 2019.

The dissertation is devoted to the solution of the actual scientific and technical problem of development of the theory and practice of creation of the distributed systems of detection of malware in local computer networks in order to increase its reliability of detection. Addressing this is important in all areas where LANs are being used extensively.

In the work the advanced model of architecture of the distributed system of detection of malware in local computer networks is developed, based on complex consideration of the requirements of distribution, decentralization, multilevel and self-organization, and the model of architecture of its typical components on the basis of the Strengths components with representation of components which they may be in operation. It allowed to take into

account the presence of software modules in different states and became the basis for determining the security status of the whole distributed system and its components. A method of interaction between components of a distributed multilevel malware detection system was developed on the basis of maintaining its integrity and determining the order of knowledge transfer between its components and using established analytical dependencies between the security levels of software modules and the security level of the whole distributed multilevel system. The method is the basis for the development of a linking piece of software that organizes the interaction of the components of a distributed multilevel malware detection system on local computer networks.

Algebraic systems and algebras have been developed with the introduction of multiple malware operations, which became the basis for creating behavioral signatures of malware for their formalized representation in detection systems.

The method of discovery of botnets in local computer networks was developed, the essence of which is to carry out active monitoring of system events and coordinated interaction of components of the distributed system when making a decision, made it possible to create tools that are able to integrate into the distributed system and to classify botnets for them behavioral signatures formed by the functions embedded in their components.

The method of detecting malware on local computer networks has been developed, which consists in combining the work of software agents that detect malware in individual computer systems, according to the methods implemented in them: dynamic formation of behavioral signatures by tracking calls by example software interface, finding polymorphic and metamorphic program code, scanning executable programs by creating them autonomously these processes and related software agents in a distributed system.

A method for detecting malware is based on the dynamic formation of behavioral signatures by tracking API-calls. It can be used to detect other types of virus programs, including new versions of existing viruses. The method involves the formation of a signature of a viral program based on the trace of API-calls, which allows to detect a viral program represented by a developed behavioral signature from the signature base. The behavioral signature includes critical API-calls by malicious activity groups and reflects the frequency of their occurrence, as well as the nature of the interaction of the critical API-features of the viral program and describes the relationship between the critical API-functions. This makes it possible to differentiate virus programs from useful applications not only in the presence of critical API challenges, but also in their interaction with each other. Classification is used to detect this.

For the file-based API that uses entanglement techniques, a method for detecting polymorphic and metamorphic viruses has been developed based on an analysis of obfuscation functions. The peculiarity of the method is the analysis of the software object and its modified versions, obtained from different modules, and further analysis based on the search for equivalent functional blocks. This allows a more detailed analysis of the software object code for the presence of polymorphic and metamorphic viruses.

Key words: distributed systems, malware, information security, self-organized systems, decentralized systems, detection methods.

Підписано до друку 05.09.2019. Формат 148*210. Офсетний друк.
Умовн. друк. арк. 1,9. Обл.-вид. арк. – 2,0
Наклад 100 прим. Зам. №384, 2019.

Віддруковано з готового оригінал - макета ПП Ковальський В.В.
29016, м. Хмельницький, вул. Свободи, 53, тел.: (0382) 79-56-27
Свідоцтво суб'єкта видавничої справи
ДК № 2827