УДК 004.492.3

О. С. Савенко, к. т. н., доц.;
С. М. Лисенко

ІНТЕЛЕКТУАЛЬНИЙ МЕТОД ТА АЛГОРИТМИ ПОШУКУ ТРОЯНСЬКИХ ПРОГРАМ В ПЕРСОНАЛЬНИХ КОМП'ЮТЕРАХ

Проведено аналіз основних методів пошуку троянських програм. Розроблено новий інтелектуальний метод пошуку троянських програм в персональних комп'ютерах з використанням нечіткого логічного висновку та алгоритмів штучних імунних систем.

Аналіз ситуації щодо шкідливого програмного забезпечення (ПЗ) показує динамічне зростання кількості вірусних програм, серед яких опосередковане місце займає клас вірусів — троянські програми (ТП), які здатні виконувати на персональному комп'ютері (ПК) деструктивні або шкідливі дії. Розробники троянських програм розробляють нові способи потрапляння на персональні комп'ютери, знаходять можливість маскування від сканерів антивірусного ПЗ та постійно вдосконалюють код троянських програм.

Розробники антивірусного ПЗ постійно проводять вдосконалення методів пошуку та виявлення троянських програм, оновлюють антивірусні бази, застосовують передові технології виявлення. Проте, реальні показники працездатності антивірусного ПЗ часто прикриваються маркетинговими матеріалами, а інформація про конкретні методи виявлення троянських програми замовчується через проблеми конкурентної переваги.

Наявні факти викрадення конфіденційної інформації та здійснення деструктивних дій в ПК, в якому встановлене антивірусне програмне забезпечення, свідчать про недоліки відомих методів виявлення ТП в ПК. Сучасні методи пошуку ТП в ПК орієнтовані на виявлення відомих ТП, та не повністю адаптовані до розпізнавання нових підозрілих об'єктів.

Найпоширенішими методами виявлення ТП в антивірусних засобах є сигнатурний аналіз, метод контрольних сум та евристичний аналіз [1].

Принцип роботи сигнатурного аналізу визначає границі його функціональності — можливість виявляти лише вже відомі віруси. Недоліком даного методу також є необхідність з'єднання з Інтернет для оновлення антивірусних баз. Відновлення пошкоджених файлів, виявлених за допомогою сигнатурного аналізу, неможливе, оскільки ТП часто є окремим об'єктом.

Метод контрольних сум має суттєві недоліки: можлива наявність однієї і тієї ж контрольної суми для різних файлів, відчутне використання системних ресурсів для знаходження контрольної суми файлів.

Евристичний сканер виконує порівняння зразків поведінки шкідливих програм із поведінкою досліджуваного об'єкту та нараховує «бали», якщо поведінка програми схожа на один із зразків. Сучасні евристичні аналізатори побудовані з використанням нейронних мереж. Проте в цьому методі присутній високий відсоток невірного спрацювання, тобто виникає ситуація, коли бали нараховуються не шкідливій програмі. Евристичний аналізатор працює досить повільно, використовує значні ресурси ПК при виконанні підозрілих програм у режимі віртуальної машини та може помилково позначати файл як троянську програму, в той час як це може бути цілком корисне програмне забезпечення чи його частина.

Враховуючи недосконалість існуючих методів пошуку ТП, необхідним є розроблення та впровадження нових методів шляхом використання технологій та алгоритмів інтелектуальної обробки інформації, які б підвищили достовірність та ефективність антивірусного діагностування персональних комп'ютерів.

Таким чином, **постає задача** розроблення нового інтелектуального методу пошуку троянських програм, згідно з яким процес їх виявлення та ідентифікації ґрунтується на аналізі їх поведінки в ПК, результат щодо можливості присутності троянської програми отримується шляхом здійснення

нечіткого логічного висновку, пошук троянських програми в ПК здійснюється на основі використання алгоритмів штучних імунних систем.

Для **розв'язання поставленої задачі** визначимо поняття троянської програми з урахуванням їх функційних особливостей, виділимо типи троянських програм за їх деструктивними діями в ПК.

Троянські програми визначають множину шкідливого вірусного програмного забезпечення, призначеного для виконання деструктивних дій в персональному комп'ютері. Троянські програми на відміну від класичних вірусних програм вирізняються відсутністю механізму створення власних копій; здатні до автономного подолання систем захисту комп'ютерних систем, з метою проникнення й зараження. У загальному випадку, ТП потрапляє в ПК разом з іншим вірусом, worm-вірусом у результаті необачних дій користувача або ж активних дій зловмисника.

Виділимо типи троянських програми згідно з діями, які вони виконують в персональному комп'ютері (див. табл. 1).

Таблиця 1

Типи троянських програм

Тип ТП	Опис
Backdoor	віддалене адміністрування у мережі; стеження за діями користувача в ПК без його відома; виконання закладених зловмисником дій
Trojan-PSW	викрадення інформації із ПК системної інформації, передача її на зазначеній в коді програми електронну адресу
Trojan-Clicker	організація несанкціонованих звертань до Internet-ресурсів.
Trojan Downloader	здійснення доставки інших шкідливих програм, запуск на виконання, або реєстрація як ТП в автозавантаження
Trojan-Dropper	здійснення таємної інсталяції інших вірусних програм на ПК або інших троянських програм
Trojan-Proxy	здійснення анонімного доступ до Internet і розсилка спаму
Trojan-Spy	здійснення електронного шпигування за діями користувача
ArcBomb	виконання деструктивних дій при спробі розархівування даних
Trojan-Notifier	повідомлення користувачеві про ураження ПК

Для усунення недоліків і вирішення наявних проблем відомих методів розроблено інтелектуальний метод пошуку троянських програм, який базується на моделі [2] та дозволяє здійснити висновок щодо можливої присутності троянської програми в персональному комп'ютері як відомої, так і нової. Метод також передбачає проведення аналізу аномалій в системі, що діагностується. Інтелектуальність методу визначають компоненти, що використовуються в процесі здійснення пошуку троянських програм. Під поняттям пошуку будемо розуміти процеси виявлення, ідентифікації та сканування файлів на предмет виявлення аномалій. Виявлення ТП здійснюємо за її життєвим циклом (ЖЦ), який включає етапи: потрапляння на ПК, активізації та виконання закладених функцій. Система виявлення використовує базу знань — базу поведінкових моделей ТП на її різних етапах ЖЦ [3]. Ідентифікацію ТП здійснюємо за допомогою використання нечіткого логічного висновку в межах підсистеми аналізу та висновку [4]. Сканування ПК на предмет підміни файлів троянськими версіями здійснюємо з використанням алгоритму негативного відбору, який застосовується в штучних імунних системах (ШІС).

Схема пошуку ТП в ПК (рис. 1) включає систему виявлення та ідентифікації (монітор) та систему аналізу аномалій (сканер).

Монітор включає в себе підсистему моніторингу та підсистему аналізу та висновку. Підсистема моніторингу відслідковує події в системі, реагує на програми, дії яких відповідають життєвому циклу ТП, виконує моніторинг виконання системних функцій, які дозволяють виконати потрапляння ТП на віддалений ПК, виконує блокування підозрілих функцій. Результати роботи даної підсистеми передаються на вхід підсистеми аналізу та висновку, яка за наявними правилами та алгоритмами робить висновок щодо можливості присутності ТП в ПК.

Сканер включає в себе генератор детекторів та підсистему контролю даних. Метою сканування ПК, що діагностується, є виявлення аномалії. Під аномалією будемо розуміти виявлення факту підміни системних фалів троянськими версіями [5].

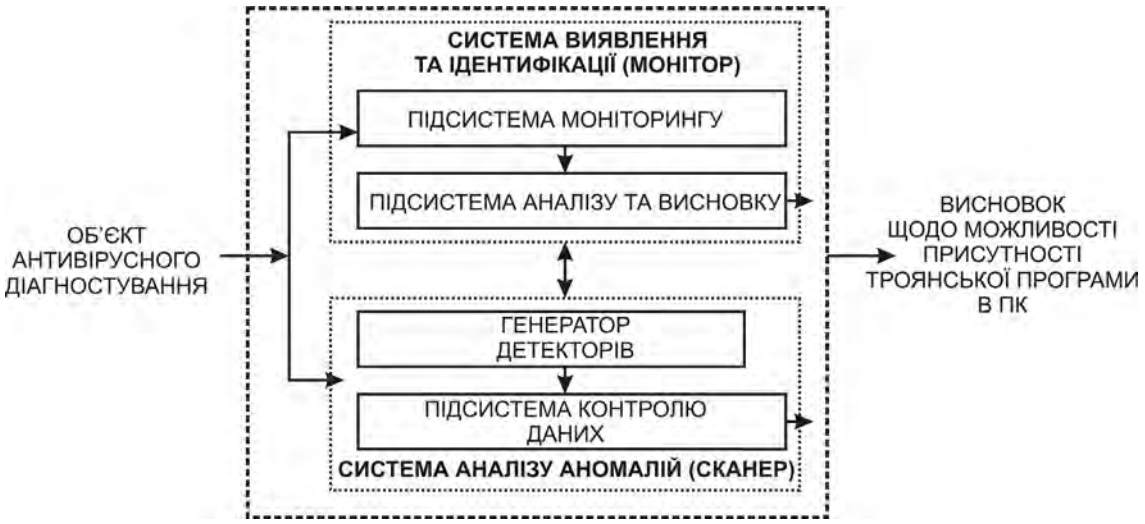

Рис. 1. Схема пошуку троянських програм в персональних комп'ютерах

Визначення ступеня підозрілості об'єкта

Визначення. Ступінь підозрілості об'єкта — числова характеристика по відношенню до об'єкта, яка відображає ступінь відповідності досліджуваного об'єкта троянській програмі.

Процес визначення ступеня підозрілості об'єкта діагностування виконується в межах системи виявлення та ідентифікації (рис. 2) і відбувається шляхом здійснення нечіткого логічного висновку (НЛВ) за допомогою задання множини ступенів підозрілості досліджуваного об'єкта.

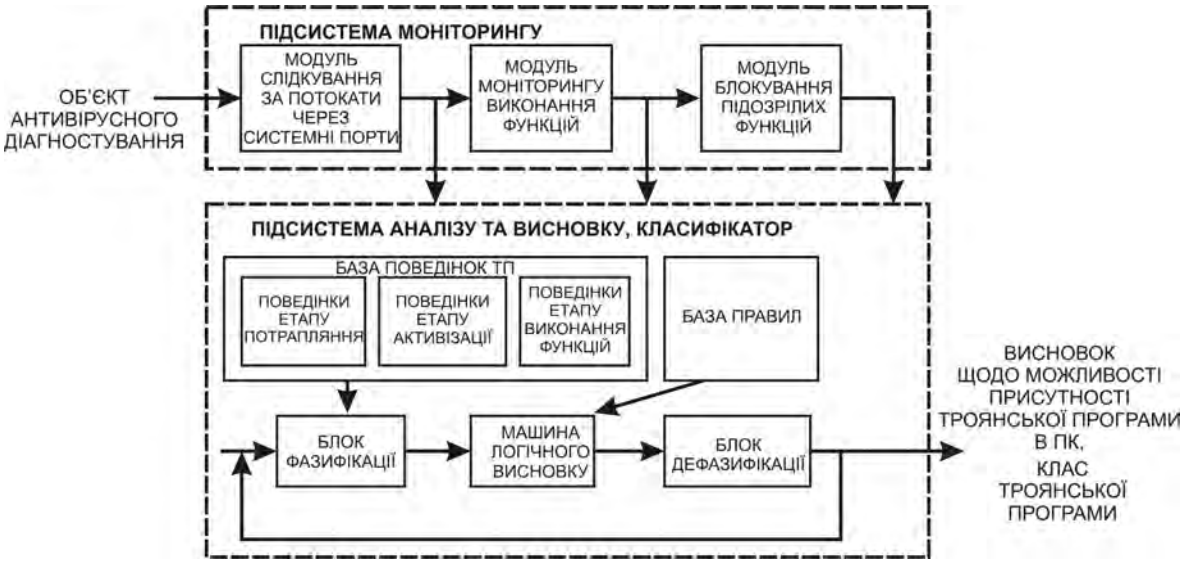

Рис. 2. Система виявлення та ідентифікації (монітор).

Підсистема аналізу та висновку містить базу поведінок ТП на різних етапах її ЖЦ [3, 6, 7]. У якості вхідних змінних НЛВ використовуються поведінки досліджуваного об'єкта. Лінгвістична змінна НЛВ описує схожість об'єкта на ТП. Універсумом нечітких змінних є шкала відповідності об'єкта із еталонною ТП, занесеною до бази поведінок. Результатом НЛВ є число, яке свідчить про ступінь підозрілості досліджуваного об'єкта [4].

Процес здійснення висновку про можливість присутності ТП в ПК виконуємо за результатами експертних оцінок впевненості в присутності ТП у вигляді множини ступенів підозрілості досліджуваного об'єкта. Для здійснення НЛВ обираємо функції належностей із класу кусково-лінійних, а саме трапецеподібні функції належності на інтервалі [0, 1].

Подамо ступені підозрілості об'єкта наступним чином: дуже не висока підозрілість — 0,05; не висока підозрілість — 0,2; більш-менш не висока підозрілість — 0,4; більш-менш висока підозрілість — 0,6; висока підозрілість — 0,8; дуже висока підозрілість — 0,95. Поріг підозрілості об'єкта, згідно з яким можна стверджувати про присутність ТП в ПК, встановимо 0,6.

Формалізацію оцінки схожості поведінки об’єкта на троянську програму здійснимо за допомогою лінгвістичної змінної, параметрами якої є: поведінка об’єкта на певному етапі життєвого циклу; базова терм-множина лінгвістичної змінної («дуже не схожа», «не схожа», «більш-менш не схожа», «більш-менш схожа», «схожа», «дуже схожа»); універсум нечітких змінних, які входять у визначення лінгвістичної змінної. Для визначеної терм-множини «поведінка об’єкта» на певному етапі її ЖЦ у якості оцінки відповідності досліджуваного об’єкта троянській програмі використаємо n-бальну шкалу, поділкою якої вважається атомарна складова алгоритму роботи троянської програми в ПК. Згідно шкали задаємо кожному терму лінгвістичної змінної відрізок із універсуму. Тоді, використовуючи алгоритм Мамдані, здійснюємо визначення ступеня підозрілості об’єкта в системі шляхом виконання НЛВ. Система Matlab 7.0 дозволила реалізувати систему нечіткого висновку для вирішення поставленої задачі (рис. 3).

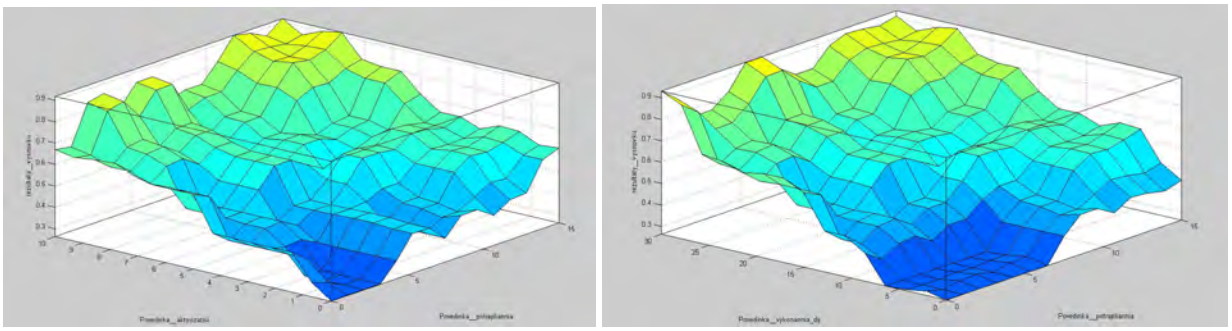

Рис. 3. Результат нечіткого логічного висновку

Алгоритм функціювання системи виявлення та ідентифікації ТП в ПК подано на рис. 4.

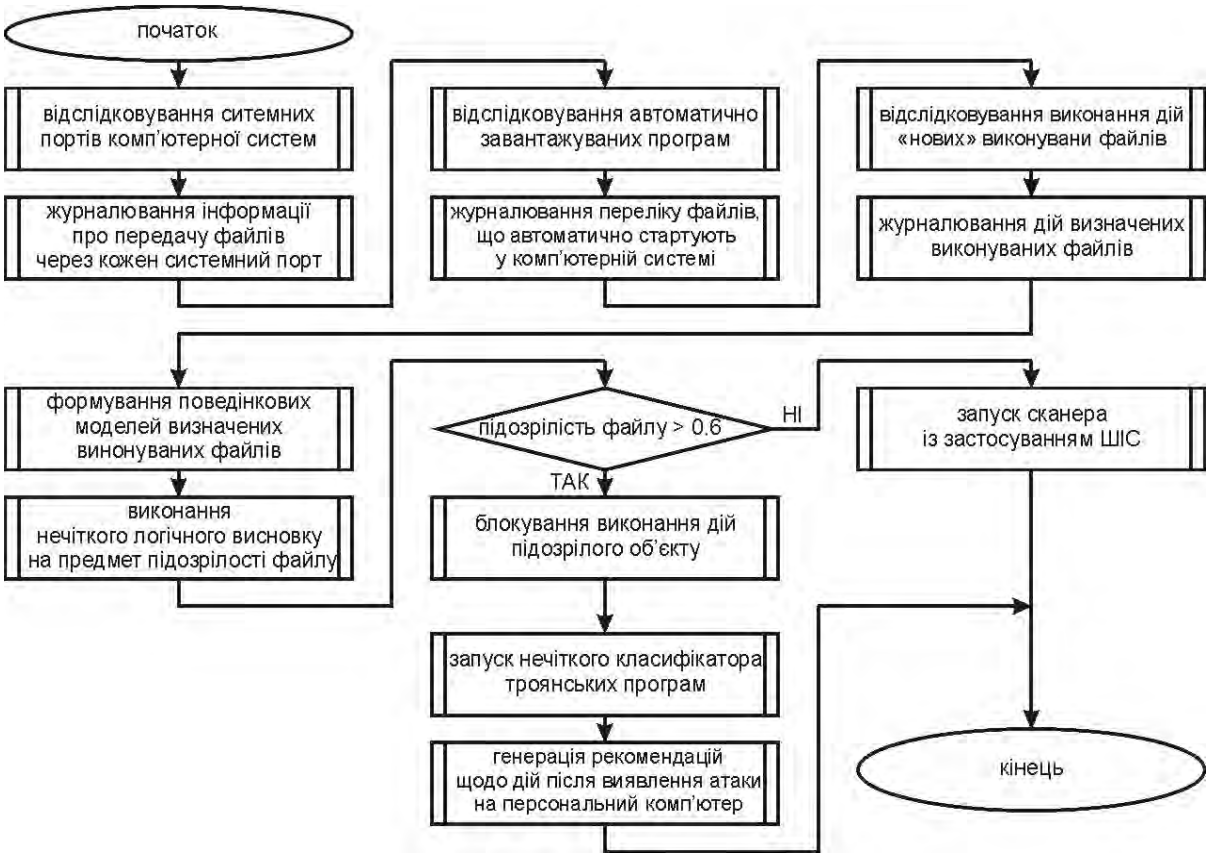

Рис. 4. Алгоритм функціювання системи виявлення та ідентифікації (монітора)

Виявлення факту підміни системних файлів

Систему аналізу аномалій (сканер) розробимо з використанням алгоритму негативного відбору, який оперує поняттями «свій» та «чужий» [5]. Під поняттям «свого» будемо розуміти набір еталонних даних системи, що контролюється.

Основою розмежування «свого» та «чужого» є генерація детекторів, які можуть виявити зміни в системі, що діагностується, а алгоритм, що використовується, — алгоритм негативного відбору. Складові алгоритму негативного відбору такі:

1. Визначений набір шаблонів нормального функціювання об'єкта діагностування як «свій», де у якості шаблону може виступати закодований певним чином рядок.

2. Згенерований набір детекторів.

3. Контроль даних шляхом постійного зіставлення детекторів з об'єктами в системі, що діагностується.

Кандидатів в набір детекторів генеруємо випадково, а потім перевіряємо на збіг із наборами шаблонів (рядків) «свого», якщо виявляється збіг — кандидат відкидаємо. Цей процес повторюємо доти, поки не згенеруємо необхідну кількість детекторів.

В розмежуванні «свого»-«чужого» необхідним є виявлення змін чи збігу в рядку з детектором (антитілом), при цьому така зміна відповідає зміні набору шаблонів нормального стану. Базу даних, що описує нормальний стан об'єкта діагностування, представимо у вигляді множини із S рядків довжини рівної l, складених з букв кінцевого алфавіту.

Розв'язання задачі з використанням ШІС вимагає вирішення питання зображення даних «свого»-«чужого» — задачі кодування, та вибору функцій афінності. Кодування детектора відбувається шляхом перетворення форми представлення даних із збереженням їх інформаційного змісту. Кодування детекторів виконуємо у бінарному вигляді. Набір детекторів R генеруємо так, щоб жоден з них не збігався з будь-яким із рядків множини S. Набори даних «свого» та «чужого» кодуємо однаково.

Основними поняттями алгоритму негативного відбору є правило збігу П, яке визначається за допомогою критеріїв афінності, і може бути подано у вигляді:

$$d\Pi x \leftrightarrow \text{міра збігу між } d \text{ та } x \text{ в заданих межах},$$

де d — детектор (антитіло) і x — шаблон (набір даних), міра збігу (міра афінності) — скалярна величина, що зображує близькість результату до оптимального значення.

В роботі використовуємо правило *rbc* (правило неперервного збігу бітів), яке є моделлю міри афінності в імунній системі. Згідно даного правила два рядки збігаються тоді і тільки тоді, коли вони ідентичні в r суміжних позиціях, де величина r вибирається у залежності від задачі.

Задача сканера полягає у постійному контролі даних шляхом неперервного зіставлення детекторів з новими надходженнями в *S*. Виявлення збігу з детектором розглядаємо як аномалію та виявлення факту підміни об'єкта, що діагностується.

Детектор (антитіло), що генеруємо в операційній системі типу Linux, має вигляд:

$$D_i^L = \left\langle m_1 \ldots m_i \ldots m_x, u_1 \ldots u_i \ldots u_x, g_1 \ldots g_i \ldots g_x, s_1 \ldots s_i \ldots s_x, t_1 \ldots t_i \ldots t_x, h_1 \ldots h_i \ldots h_y, C_1 \ldots C_i \ldots C_z \right\rangle, \quad (1)$$

де $m_1 \ldots m_i \ldots m_x$ — режим файлу (тип і права доступу); $u_1 \ldots u_i \ldots u_x$ — числовий ідентифікатор власника файлу, який показує власника файлу; $g_1 \ldots g_i \ldots g_x$ — числовий ідентифікатор групи власника файлу; $s_1 \ldots s_i \ldots s_x$ — розмір файлу; $t_1 \ldots t_i \ldots t_x$ — час останньої зміни файлу; $h_1 \ldots h_i \ldots h_y$ — створений хеш MD5 даного файлу; $C_1 .. C_i \ldots C_z$ — CRC даного файлу, при $i \in \overline{1,n}$, де n — кількість детекторів.

Детектор (антитіло), що генеруємо в операційній системі типу Windows, має вигляд:

$$D_i^W = \left\langle s_1 \ldots s_i \ldots s_x, t_1 \ldots t_i \ldots t_x, a_1 \ldots a_i \ldots a_x, h_1 \ldots h_i \ldots h_y, C_1 \ldots C_i \ldots C_z \right\rangle, \quad (2)$$

де $s_1 \ldots s_i \ldots s_x$ — розмір файлу; $t_1 \ldots t_i \ldots t_x$ — час останньої зміни файлу; $a_1 \ldots a_i \ldots a_x$ — атрибут файлу (параметри: лише читання, прихований, системний архівний); $h_1 \ldots h_i \ldots h_y$ — створений хеш MD5 даного файлу; $C_1 .. C_i \ldots C_z$ — CRC даного файлу, при $i = \overline{1,n}$, де n — кількість детекторів.

Таким чином, процес виявлення факту підміни системних файлів в межах роботи системи аналізу аномалій (сканера) включає такі етапи:

1. В системі, що діагностується, виконуємо формування набору файлів, що підлягають процедурі створення «свого». Здійснюємо відбір системних бібліотек операційної системи, виконуваних файлів системних служб та драйверів пристроїв ПК, які можна вважати еталонними.

2. Виконуємо генерацію набору шаблонів файлів, відібраних на попередньому кроці. Згідно з типом операційної системи персонального комп'ютера виконуємо кодування даних «свого» та «чужого» у вигляді визначеного рядка.

3. Виконуємо генерацію детекторів ТП в ПК. Набори детекторів генеруємо «жадібним» алгоритмом [5], у якому у порівнянні з лінійним не генеруються зайві детектори та час роботи якого зростає лінійно при збільшенні розміру «свого».

4. На етапі сканування системи виконуємо співставлення об'єктів антивірусного діагностування з детекторами.

5. У випадку високої афінності з детектором виконуємо сповіщення про виявлення аномалії та перевіряємо на підозрілість поведінку даного об'єкта.

Функціювання системи аналізу аномалій відбувається за алгоритмом, зображеним схемою на рис. 5.

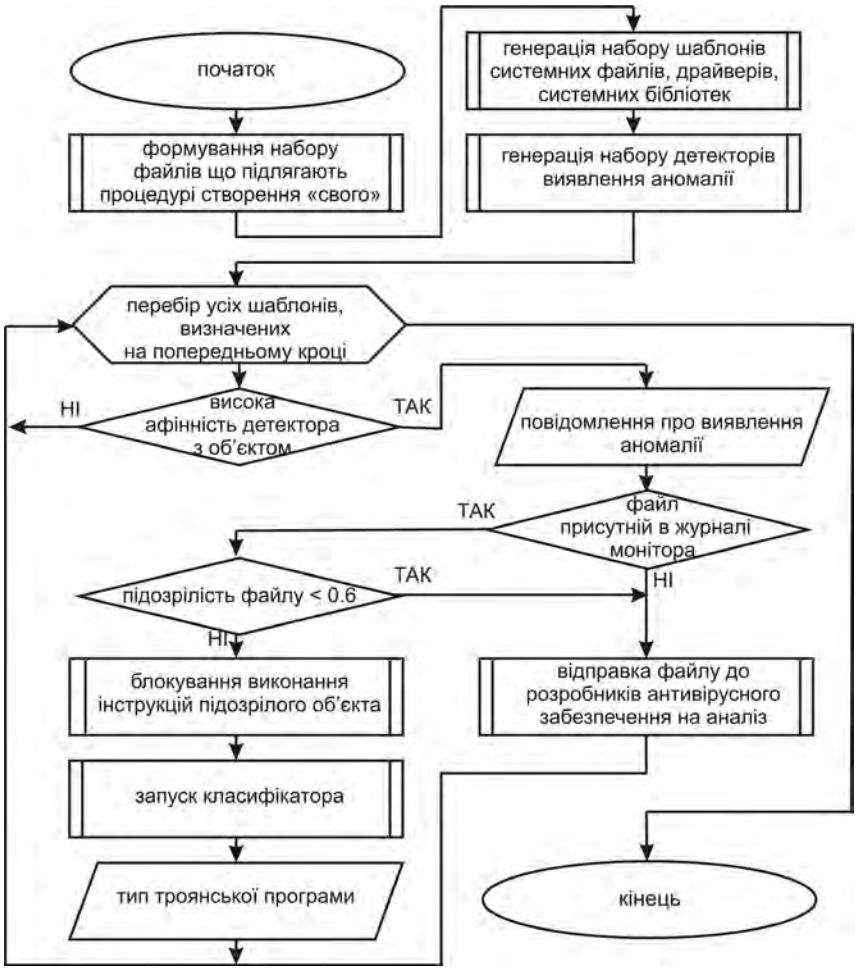

Рис. 5. Алгоритм функціювання система аналізу аномалій (сканера)

Проведене дослідження алгоритму негативного відбору [5] доводить ефективність розробленого методу стосовно його швидкодії та високої ймовірності виявлення аномалій, яка зростає експоненційно із зростанням числа копій N_t незалежно виконуваних алгоритмів (рис. 6, 7).

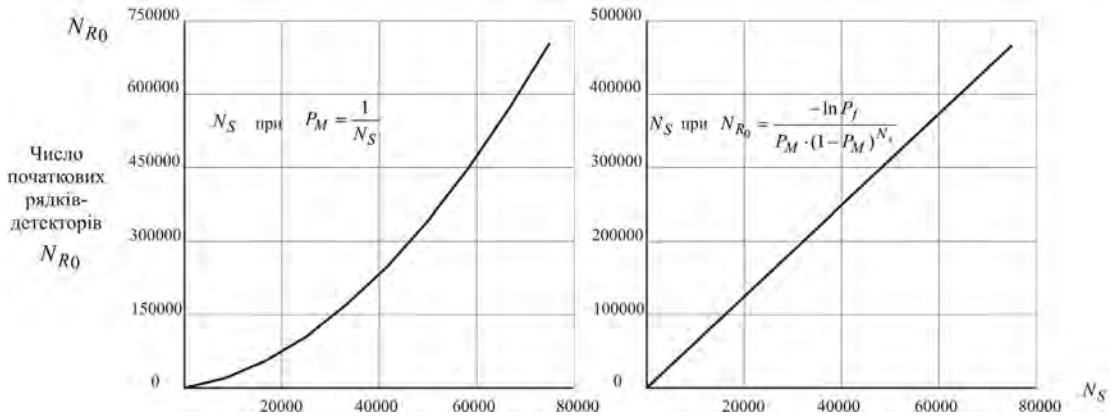

Рис. 6. Співвідношення числа рядків «свого» N_S та початкової кількості детекторів N_{R0}

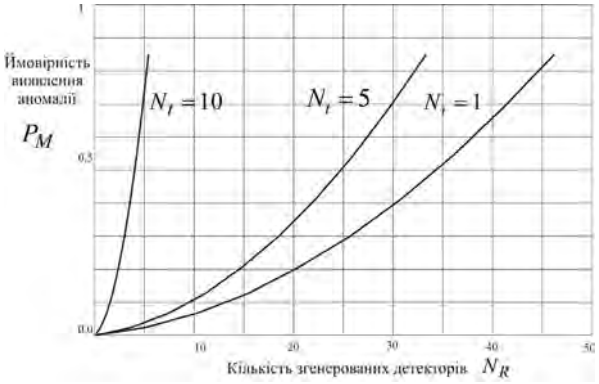

Рис. 7. Експоненційний ріст ймовірності виявлення при збільшенні N_t копій алгоритмів виявлення

Проведену порівняльну характеристику якісних показників відомих методів пошуку троянських програм у порівнянні із розробленим методом представлено в табл. 2.

Таблиця 2

Порівняльна характеристика якісних показників відомих методів пошуку ТП

Метод	Характеристика методу			
	виявлення нових ТП	висока швидкодія методу	невисокі вимоги до апаратних ресурсів	мала кількість помилкового спрацювання
сигнатурний	—	+	+	+
евристичний	+	—	—	—
контрольних сум	—	+	—	+
нейромережний	+	—	—	—
розроблений	+	+	+	+

Висновки

В результаті проведеного дослідження сучасних методів пошуку троянських програм виявлено наступні недоліки: невисока швидкодія, низька ймовірність ідентифікації нових троянських програм, вимогливість окремих методів до апаратного забезпечення.

Для усунення недоліків відомих методів пошуку троянських програм розроблено новий інтелектуальний метод пошуку троянських програм в персональних комп'ютерах шляхом використання алгоритмів нечіткої логіки та штучних імунних систем. Розроблений метод передбачає ідентифікацію троянських програм шляхом використання апарату нечіткої логіки. Сканування персонального комп'ютера на предмет наявності троянських програм згідно даного методу виконується шляхом використання алгоритму негативного відбору.

Результати дослідження розробленого методу показали якісну перевагу над відомими методами: можливість ідентифікації нових троянських програм при невисоких вимогах до потужностей ПК, швидке сканування ПК на предмет виявлення підміни системних файлів троянськими версіями.

Реалізація методу забезпечить здійснення пошуку та ідентифікації відомих та нових троянських програм в персональному комп’ютері, які не вимагатимуть побудови баз сигнатур, застосування евристичного аналізатора та використання контрольних сум.

СПИСОК ЛІТЕРАТУРИ

1. Савенко О. Дослідження методів антивірусного діагностування комп'ютерних мереж / Олег Савенко, Сергій Лисенко // Вісник Хмельницького національного університету. — 2007. — Том 2, № 2. — С. 120—126.

2. Савенко О. Модель процесу пошуку троянських програм в персональному комп’ютері / Олег Савенко, Сергій Лисенко // Радіоелектронні і комп’ютерні системи. — 2008. — № 7. — С. 87—92.

3. Савенко О. С. Поведінкова модель троянських програм / О. С. Савенко, С. М. Лисенко // Комп’ютерні науки та інформаційні технології (CSIT-2007): міжнар. наук.-техн. конф., 27—29 вересня 2007 р.: тези доповідей. — 2007. — С. 129—132.

4. Система пошуку троянських програм з використанням нечіткого логічного висновку: зб. наук. праць за матеріалами міжнародної наук.-практ. конф. «Інтелектуальний аналіз інформації ІАІ-2008», 14—17 травня 2008 р. /редкол.: С. В. Сирота [та ін.]. — К.: Просвіта. — 2008. — С. 413—431.

5. Савенко О. Розробка процесу виявлення троянських програм на основі використання штучних імунних систем / Олег Савенко, Сергій Лисенко // Вісник Хмельницького національного університету. — 2008. — № 5. — С. 183—188.

6. Використання поведінкової моделі троянських програм як засобу їх ідентифікації: зб. наук. праць міжвузівської наук.-практ. конф. «Прогресивні інформаційні технології в науці та освіті», 4—5 жовтня 2007 р. / відп. ред. О. М. Мельников. — Вінниця: Едельвейс, 2007. — С. 49—54.

7. Савенко О. С. Процес побудови бази поведінкових моделей троянських програм / Олег Савенко, Сергій Лисенко // Вісник Хмельницького національного університету. — 2008. — Том 1, № 4. — С. 191—196.

Рекомендована кафедрою автоматики та інформаційно-вимірювальної техніки

Надійшла до редакції 21.10.08
Рекомендована до друку 20.11.08

Савенко Олег Станіславович — декан факультету комп’ютерних систем та програмування; ***Лисенко Сергій Миколайович —*** асистент.

Кафедра системного програмування, Хмельницький національний університет