УДК 004.492.3

О.С. САВЕНКО, С.М. ЛИСЕНКО

Хмельницький національний університет, Україна

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ІНТЕЛЕКТУАЛЬНОГО ДІАГНОСТУВАННЯ ТРОЯНСЬКИХ ПРОГРАМ КОМП'ЮТЕРНИХ СИСТЕМ

Досліджено стан розробки шкідливого програмного забезпечення. Запропоновано концепцію здійснення діагностування троянських програм із застосуванням апарату нечіткої логіки та алгоритмів штучних імунних систем. Одержано модель процесу діагностування троянських програм, що дало можливість інтелектуалізувати процес діагностування троянських програм. Розроблено нову інформаційну технологію інтелектуального діагностування троянських програм комп'ютерних систем, відмінністю якої від відомих є те, що процес діагностування не потребує побудови баз вірусних сигнатур та дає змогу діагностувати нові невідомі троянські програми з більш високою достовірністю.

Ключові слова: *троянська програма, інформаційна технологія інтелектуального діагностування троянських програм комп'ютерних систем, нечіткий логічний висновок, алгоритм негативного відбору*

Вступ

Об'єднання комп'ютерних систем в локальні мережі та підключення їх до глобальної мережі Internet створює багато проблем, пов'язаних з їх функціюванням та використанням, однією з яких є комп'ютерні віруси. Їх наявність приводить до неправильного функціонування програмного та апаратного забезпечення. Аналіз ситуації щодо шкідливого програмного забезпечення (ПЗ) показує динамічне зростання чисельності класу вірусів – троянських програм (ТП), які здатні виконувати в комп'ютерній системі (КС) деструктивні дії.

Наявні факти викрадення конфіденційної інформації та здійснення деструктивних дій в КС, в якому встановлене антивірусне програмне забезпечення (АПЗ), свідчать про недоліки відомих інформаційних технологій (ІТ) діагностування троянських програм комп'ютерних систем (ТП КС).

Сучасні ІТ діагностування ТП КС орієнтовані на виявлення відомих ТП, та не повністю адаптовані до діагностування підозрілих об'єктів. Проведений аналіз сучасних інформаційних технологій діагностування троянських програм в комп'ютерних системах виявив недоліки їх роботи, а також показав неспроможність діагностування нових ТП КС з високою достовірністю та ефективністю [1].

1. Постановка задачі

Для підвищення достовірності діагностування невідомих ТП КС необхідно інтелектуалізувати процесу діагностування троянських програм за рахунок використання компонентів штучного інтелекту. Тому важливою є задача розробки нової інформаційної технологій інтелектуального діагностування ТП КС, яка б підвищила достовірність процесу діагностування нових троянських програм.

Для розв'язку поставленої задачі необхідно вирішити наступні під задачі:; розробити поведінкові моделі ТП із урахуванням їх функціонального навантаження; розробити модель процесу інтелектуального діагностування троянських програм КС; програмно реалізувати інформаційну технологію інтелектуального діагностування ТП КС у вигляді програмного забезпечення та впровадити їх у виробництво з метою підвищення достовірності та ефективності антивірусного діагностування комп'ютерних систем.

2. Модель процесу інтелектуального діагностування троянських програм комп'ютерних систем

Позначимо частини процесу інтелектуального діагностування ТП КС як Ω та Δ, де Ω – процес моніторингу, $\Omega = \{\Omega_1, \Omega_2, \Omega_3, \Omega_4, \Omega_5, \Omega_6\}$ а Δ – процес сканування КС, $\Delta = \{\Delta_1, \Delta_2, \Delta_3, \Delta_4\}$ (рис. 1, 2). Тоді Ω_1 – процедура відслідковування потоків, що здійснюються через системні порти КС; Ω_2 – процедура відслідковування потоку виконання системних функцій в КС; Ω_3 – процедура блокування виконання програмним об'єктом системних функцій або функцій ТП, підозрілість яких визначена на інших етапах процесу антивірусного діагностування; Ω_4 – виконання процедури фазифікації в ме-

жах системи нечіткого логічного висновку (НЛВ) визначення підозрілості функціонування програмних об'єктів в КС; Ω_5 – робота машини логічного висновку в межах системи НЛВ визначення підозрілості функціонування програмних об'єктів в КС; Ω_6 – виконання процедури дефазифікації в межах системи НЛВ визначення підозрілості функціонування програмних об'єктів в КС.

Для реалізації кожного етапу необхідні відповідні параметри, що представлені множиною векторів $\Phi_{mon} = \{\bar{\varphi}_1, \bar{\varphi}_2, \ldots, \bar{\varphi}_6\}$, де кожен вектор складається з множини параметрів відповідного етапу: $\bar{\Phi}_i = \{\varphi_1^i, \varphi_2^i, \ldots, \varphi_{n_i}^i\}$, де n_i - кількість параметрів і-того етапу, $i = \overline{1,6}$.

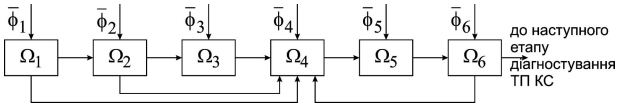

Рис. 1. Формалізована схема процесу моніторингу системних подій комп'ютерної системи

Процес сканування включає наступні етапи: Δ_1 – виконання формування набору файлів, що підлягають процедурі створення набору захищених бінарних послідовностей (так званого «свого»); Δ_2 – виконання генерації набору шаблонів файлів, відібраних на попередньому етапі та виконання кодування відібраних даних у визначеному форматі; Δ_3 – виконання генерації детекторів згідно обраного алгоритму; Δ_4 – виконання етапу сканування системи співставлення захищених бінарних послідовностей об'єктів антивірусного діагностування зі згенерованими на попередньому етапі детекторами.

Аналогічно процесу моніторингу, для реалізації кожного етапу процесу діагностування ТП КС для кожного етапу необхідні відповідні параметри, що представлені множиною векторів $\Phi_{scan} = \{\bar{\varphi}_7, \bar{\varphi}_8, \ldots, \bar{\varphi}_{10}\}$, де кожен вектор складається з множини параметрів відповідного етапу: $\bar{\Phi}_i = \{\varphi_1^i, \varphi_2^i, \ldots, \varphi_{n_i}^i\}$, де n_i - кількість параметрів і-того етапу, $i = \overline{1,4}$.

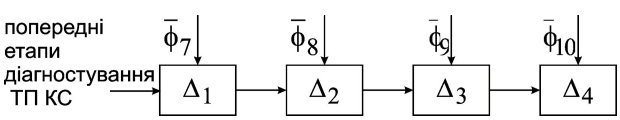

Рис. 2. Формалізована схема процесу сканування комп'ютерної системи

Формалізована схема процесу діагностування ТП КС включає дві основні частини: моніторинг системних подій в КС та сканування КС на виявлення факту підміни системних файлів троянськими версіями. З урахуванням зв'язків між частинами процесу діагностування, які полягають у здійсненні передачі даних від етапу Ω_6 до Δ_1, схему процесу діагностування ТП КС представимо на рис. 3.

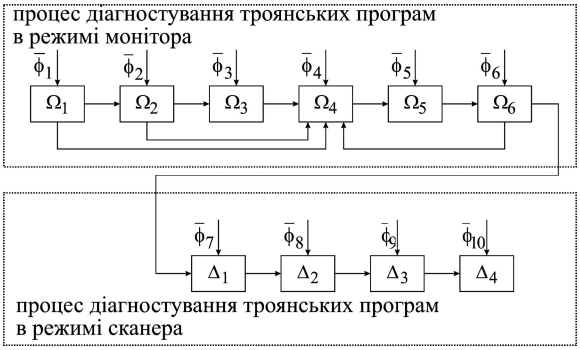

Рис. 3. Формалізована схема процесу діагностування троянських програм КС

Для формалізації виконання етапів антивірусного діагностування подамо модель процесу інтелектуального діагностування ТП КС з урахуванням параметрів, які використовують вищевказані етапи у вигляді [2]:

$$M_f = \langle E, S, R, V, L, H, S, D, \varepsilon \rangle, \qquad (1)$$

де для етапів Ω_1, Ω_2, Ω_3: E – множина об'єктів антивірусного діагностування $e_k \in E$, а саме множина файлів КС, причому $\Theta \in E$; аналогічно поведінковій моделі ТП поданій в [3] S є множиною станів $s \in S$ об'єкту діагностування, які відповідають життєвому циклу (ЖЦ) троянської програми; V – матриця відношень дій програмного об'єкту $m \in M$, які дозволяють здійснити потрапляння даного об'єкту через системні порти КС $p \in P$; L – матриця відношень дій програмного об'єкту $a \in A$ та системних бібліотек, що виконуються $b \in B$; R – результуюче число $R \in [0,1]$, яке свідчить про ступінь підозрілості досліджуваного програмного об'єкту для етапів, що продукується на етапах Ω_4, Ω_5, Ω_6; для етапів Δ_1, Δ_2, Δ_3, Δ_4: H – множина об'єктів $h \in H$, що підлягають процедурі сканування на предмет можливого факту їх підміни; S – множина бінарних послідовностей, згенерованих для формування набору захищених бінарних послідовностей $s \in S$; D – множина детекторів, згенерованих для сканування системи $d \in D$; відношення ε між об'єктами та станами, при чому для $\nu \in \Theta$ та $s \in S$, відношення $\nu \varepsilon s$ означає, що програмний об'єкт e перебуває в стані s, відношення $e \bar{\varepsilon} s$ означає, що програмний e не перебуває в стані s.

3. Інформаційна технологія інтелектуального діагностування троянських програм комп'ютерних систем

З метою усунення недоліків і вирішення наявних проблем відомих ІТ та підвищення ефективності антивірусного діагностування КС було розроблено ІТ діагностування ТП КС шляхом використання компонентів теорії штучного інтелекту, а саме апарату нечіткої логіки та штучних імунних систем.

ІТ інтелектуального діагностування ТП КС базується на моделі [2] та дозволяє здійснити висновок щодо можливої присутності ТП в КС як відомої, так і нової. ІТ також передбачає можливість виявлення факту підміни системних файлів троянськими версіями. Діагностування ТП здійснюємо за її ЖЦ, який включає етапи: потрапляння в КС, активізації та виконання закладених функцій. Інформаційна технологія використовує базу знань – базу поведінкових моделей ТП на її різних етапах ЖЦ. Діагностування ТП КС в режимі моніторингу здійснюємо за допомогою використання НЛВ, діагностування ТП КС в режимі сканування на предмет виявлення підміни файлів троянськими версіями здійснюємо з використанням алгоритму негативного відбору, який застосовується в штучних імунних системах (див. рис.4).

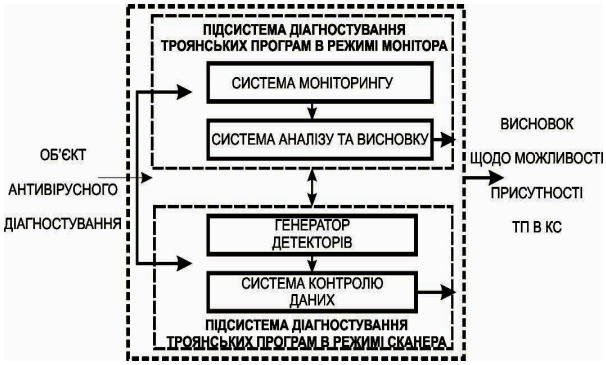

Рис. 4. Схема системи інтелектуального діагностування ТП КС

Процес визначення ступеня підозрілості об'єкта діагностування виконується відбувається шляхом здійснення НЛВ за допомогою задання множини ступенів підозрілості досліджуваного об'єкта [4].

Підсистема аналізу та висновку містить базу поведінок ТП на різних етапах її ЖЦ. У якості вхідних змінних НЛВ використовуються поведінки досліджуваного об'єкта. Лінгвістична змінна НЛВ описує схожість програмного 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.

Представимо процес здійснення висновку про наявність ТП в КС за результатами експертних оцінок впевненості в присутності ТП у вигляді множини $Y = \{y_1, y_2, ..., y_i, ..., y_n\}$, де y_i - ступінь підозрілості досліджуваного об'єкта. Для множини Y побудуємо множну функцій належності $M = \{\mu_{y_1}, \mu_{y_2}, ..., \mu_{y_i}, ..., \mu_{y_n}\}$, яка відображатиме ступінь впевненості експерта в істинності твердження стосовно ступеня підозрілості даного об'єкта. Подамо ступені впевненості експерта в 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 КС, встановимо 0,6.

Такий підхід дозволить з використанням алгоритму Мамдані здійснити визначення ступеня підозрілості об'єкта в системі шляхом виконання НЛВ.

Функцію належності μ_{y_i} можна обрати із класу кусково-лінійних, а саме трикутну або трапецеподібну функції належності на інтервалі [0,1].

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, яка відповідатиме її поведінці на певному етапі її роботи в КС. На дану шкалу будуть накладатися нормовані числові показники атомарних складових алгоритму потрапляння ТП. Згідно шкали задаємо кожному терму лінгвістичної змінної відрізок із універсуму.

Задамо вхідні змінні у вигляді лінгвістичних змінних та визначимо їх функції належності.

У якості вхідних змінних використаємо поведінки об'єкта на можливих етапах ЖЦ ТП: поведінка об'єкта на етапі потрапляння в КС; поведінка об'єкта на етапі його активізації; поведінка об'єкта на етапі виконання закладених інструкцій.

Формалізація оцінки схожості та підозрілості поведінки програмного об'єкта в системі здійснюється за допомогою лінгвістичної змінної, описаної кортежем виду: $\langle \beta_i, T, X, G, M \rangle$, де β_i – поведінка програмного об'єкта на певному етапі ЖЦ; T – базова терм-множина лінгвістичної змінної; $X_i = [0, n]$ – область визначення (універсум) нечітких змінних, які входять у визначення лінгвістичної змінної; G – процедура утворення нових термів за допомогою модифікаторів типу «дуже», «не», «більш-менш» тощо, M – семантична процедура

задання на проміжку $X_i = [0, n]$ нечітких змінних, а також відповідних нечітких множин для термів із $G(T)$ у відповідності з правилами модифікаторів «дуже», «не», «більш-менш».

В основі діагностування ТП використаємо нечітку модифіковану поведінкову модель ТП у вигляді множини перехідних станів ТП в ОС протягом її ЖЦ:

$$X = \{(X_П, X_A, X_Д); (R, V, L, Z)\}, \qquad (2)$$

де $X_П, X_A, X_Д$ – універсальні підмножини станів ТП на розглянутих етапах ЖЦ. Всі $x_i \in X$ на етапах ЖЦ представляються четвіркою нечітких параметрів R_{ij}, V_i, L_i, Z_i.

Для семантичного опису моделі (2) на початку задамо вхідні і вихідні лінгвістичні змінні (ЛЗ). ЛЗ вважаються заданими, якщо для них визначені базові терм-множини і функції належності. На розглянутих підмножинах лінгвістичною змінною визначимо як x, якій привласнюється ім'я: «Ступінь схожості». Для прийнятої ЛЗ визначаються нечіткі вхідні змінні (терм-множини) і їхні функції належності. Терм-множини приймемо наступним чином: «Мала схожість (М)», «Середня схожість (С)», «Велика схожість (В)», з областю визначення [0,1].

Значення вхідних змінних можуть бути отримані за допомогою будь-якого механізму моніторингу й статистичної обробки за участю експертів і повинні відображати рівень істинності (наближення) до лінгвістичної змінної.

Основою моделі (2) є матриця нечітких відношення M_Q, що формується на кожному етапі ЖЦ. Поняття нечітких відношення і нечітких множин є основою для побудови нечітких моделей складних систем. Нечітке відношення Q визначається як нечітка підмножина впорядкованих кортежів (елементів), заданих на універсумах: $X_П, X_A, X_Д$. Особливість побудови нечіткого відношення полягає в тому, що розглянуті причинні зв'язки не є однозначними, залежать від типу операційної системи (ОС), особливостей функціонального навантаження троянської програми ті інших факторів. Тому причинні зв'язки найбільше адекватно можуть бути представлені у вигляді бінарного нечіткого відношення, заданого на базисних множинах X,Y:

$$Q = \{< x_i, y_j >, \mu_Q(x_i, y_j)\} \qquad ,(3)$$

де $\mu_Q(x_i, y_j)$ – функція належності даного бінарного нечіткого відношення кількісно описує ступінь впевненості в причинно-наслідковому зв'язку x_i, y_j. Якщо відношення задане у вигляді матриці, то елемент матриці M_Q є відповідним значенням функції належності $\mu_Q(x_i, y_j)$ – даного відношення і має властивість R, що характеризує дане відношення. В (3) на етапі потрапляння матриця відношення M_Q записується у вигляді $V_П = |V_{mp}|$, у якій $m = \overline{1,k}$ – функції (механізми), які реалізують способи потрапляння ТП в КС через порти $p = \overline{1,h}$ мережних протоколів прикладного рівня.

На етапах активізації і виконання закладених функцій ТП використовуються матриці відношення $L_A = |L_{ab}|$ і $L_d = |L_{ab}|$ дій досліджуваного програмного об'єкта і структурних одиниць ОС, у яких $a = \overline{1,\sigma}$ - дії ТП, що завдають негативних впливів на структурні компоненти $b = \overline{1,\tau}$ ОС КС.

Четвертим параметром R в (3) є властивість досліджуваного програмного об’єкта, яка характеризує відношення в (3) і наділена деякою числовою характеристикою можливості інфікування КС, що визначається не тільки нечіткими відношеннями відповідних матриць, а також додатковим вектором $Z = \{z_k\}$, де $k = \overline{1,r}$ і z_k характеризують передбачувані деструктивні дії, що виконує ТП: зависання КС, втрата і спотворення вмісту файлів, порушення цілісності системного реєстру та ін. Кожному елементу вектора присвоюються нормовані пріоритетні ваги ε_k, що відображають ступінь їхньої небезпеки для КС користувача, при чому $\sum \varepsilon_k = 1$.

Властивість R представляється як вихідна лінгвістична змінна (y), якій присвоюється ім'я: «Підозрілість об’єкта». Функції належності вихідної змінної $\mu_R(y)$ формуються непрямим методом на основі оцінок експертів, що мають досить високу кваліфікацію і досвід роботи в області антивірусного діагностування, а також володіють численною статистикою щодо поведінки троянських програм.

Передумовою використання експертних оцінок є, як зазначалося, особливість побудови нечітких відношень. Так, наприклад, на етапі потрапляння ТП КС для кожного елемента матриці $V_П = |V_{mp}|$, може існувати множина різних пар (m_i, p_j) - варіантів потрапляння ТП КС із різним рівнем небезпеки. Конкретні шляхи (порти та дії, що дозволяють задіяти дані порти) потрапляння не однозначні й не можуть бути прогнозовані простими математичними засобами. Задача визначення $\mu_M(y)$ представмо як задачу ранжирування для кожного з механізмів (функцій) m_i матриці $V_П = |V_{mp}|$ портів потрап-

ляння p_j при заданих ознаках небезпеки z і вибору найбільш імовірного p_j при активізації певної функції m_i. Розв'язок даної задачі показано в [5].

Показник R обмежується граничним значенням Rp, обумовлений вимогою безпеки. Перевищення R над Rp приймається як сигнал для прийняття рішення щодо досліджуваного програмного об'єкту. Аналогічно розглядаються інші етапи ЖЦ ТП.

Для прийняття рішення щодо можливості присутності ТП КС використовується система НЛВ.

На етапі НЛВ використаємо нечітку базу знань і нечіткі операції, за допомогою яких реалізується висновок у вигляді нечіткої множини. База знань (бази правил і даних) містить множину нечітких продукційних правил (НПП), що зв'язують лінгвістичні змінні й функції належності. Фрагмент бази, у якій 27 правил, подано нижче:

П.1. If (potrapl is M) and (aktyviz is M) and (vykonan is M) then (pidozril is M) (1);

П.2. If (potrapl is M) and (aktyviz is M) and (vykonan is C) then (pidozril is M) (0,8);

П.3. If (potrapl is M) and (aktyviz is M) and (vykonan is B) then (pidozril is M) (0,8);

…….

П.27. If (potrapl is B) and (aktyviz is B and (vykonan is B) then (pidozril is B) (1).

У роботі використаємо стратегію на основі правила max-min композиції і нечіткої операції max-диз'юнкції для оцінки однакових висновків. Приведення до чіткості для прийняття остаточного рішення щодо можливості присутності ТП в КС виконуємо за допомогою методу центра тяжіння, при якому значення вихідної змінної дорівнює абсцисі центра ваги площі, обмеженої графіком функції належності підсумкової нечіткої множини $M_R(y)$. Обчислена числова характеристика досліджуваного програмного об'єкта інтерпретується як ступінь його підозрілості. Далі отримане число може нормуватися так, щоб результат мав межі [0,1]. Поріг підозрілості складає 0,6, що згідно з прийнятою стратегією безпеки потребує виконання блокування дій даного програмного об'єкту.

Згідно з моделлю (1) інтелектуальне діагностування в режимі сканера відбувається в чотири етапи, забезпечення виконання яких покладено на підсистему інтелектуального діагностування ТП КС в режимі сканера. Сканер включає в себе генератор детекторів та підсистему контролю даних. Метою сканування КС є виявлення факту підміни системних фалів троянськими версіями.

Підсистему інтелектуального діагностування ТП в режимі сканера розроблено з використанням алгоритму негативного відбору, який оперує поняттями "свій" та "чужий" [7]. Під поняттям "свого" будемо розуміти набір еталонних даних КС.

Для розмежування понять "свого" та "чужого" виконуємо генерацію множини захищених двійкових послідовностей S визначеної розмірності l для кожної операційної системи. Після формування даної множини здійснюємо генерацію кандидатів в набір детекторів R, які є двійковими послідовності відповідної довжини. Детектори генеруємо випадково, а потім перевіряємо на збіг із наборами двійкових послідовностей "свого", якщо виявляється збіг – кандидат відкидаємо; процес повторюємо доти, поки не згенеруємо задану кількість детекторів.

Розв'язок задачі з використанням ШІС вимагає вирішення питання представлення даних "свого"-"чужого" – задачі кодування, та вибору функцій афінності. У якості правила збігу, використовуємо правило правило неперервного збігу бітів (rbc), 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.

Задачею антивірусного сканування є контроль даних шляхом зіставлення детекторів з новими надходженнями в S. Виявлення збігу з детектором розглядаємо як факту підміни програмного об'єкту, що діагностується.

Здійснимо кодування двійкових послідовностей для ОС типу UNIX/GNU Linux та сімейства Microsoft Windows. У файловій системі ОС типу UNIX/GNU Linux інформація про файл зберігається в його індексному дескрипторі. Для кодування детектора вибрано лише ті атрибути файлу, зміна яких може вказувати на його підміну (троянську модифікацію). Про факт такої підміни свідчать операції з файлом: зміна стану типу файлу та прав доступу до даного файлу; зміна розміру файлу; зміна хешу; зміна власника файла; зміна групи-власника файла; зміна останнього часу модифікації.

Детектор для ОС типу Linux має вигляд:

$$D_i^L = \left\langle \begin{matrix} m_1...m_i...m_x, u_1...u_i...u_x, g_1...g_i...g_x, \\ s_1...s_i...s_x, t_1...t_i...t_x, h_1...h_i...h_y, C_1...C_i...C_z \end{matrix} \right\rangle \quad (4)$$

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

Для кодування детектора для файлових систем ОС типу MS Windows аналогічно вибрано лише ті

атрибути файлу, зміна яких може вказувати на факт підміни файлу або його троянську модифікацію. Про підміни можуть свідчити наступні операції з файлом: зміна розміру файлу; зміна хешу; зміна останнього часу модифікації; зміна параметрів-атрибутів файлу. Тоді детектор (антитіло), що генеруємо в ОС типу Windows, має вигляд:

$$D_i^W = \left\langle \begin{matrix} s_1...s_i...s_x, t_1...t_i...t_x, a_1...a_i...a_x, \\ h_1...h_i...h_y, C_1...C_i...C_z \end{matrix} \right\rangle, \qquad (5)$$

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

Отже, процес виявлення факту підміни системних файлів троянськими версіями включає наступні етапи: формування набору файлів, що підлягають процедурі створення "свого" шляхом відбору системних бібліотек операційної системи, виконуваних файлів системних служб та драйверів пристроїв КС, які можна вважати еталонними; згідно з типом ОС КС здійснення кодування даних "свого" та "чужого" у вигляді двійкових послідовностей; виконання генерації детекторів; на етапі антивірусного сканування КС виконання співставлення захищених двійкових послідовностей з детекторами; у випадку високої афінності з детектором виконання сповіщення про виявлення підміни та перевірка на підозрілість поведінки даного програмного об'єкту.

Виявлення факту підміни системних файлів троянськими версіями, яке покладено на антивірусний сканер, виконується із застосуванням алгоритму негативного відбору. В роботах [6, 7] проведені дослідження параметрів алгоритму, різних ймовірнісних оцінок для різних наборів вхідних параметрів алгоритму та його швидкодію. Результати дослідження показали високу ймовірнісні характеристики та можливість застосування даного алгоритму для здійснення діагностування ТП КС в режимі сканера.

Для можливості програмної реалізації розробленої ІТ були розроблені алгоритми функціонування підсистеми інтелектуального діагностування ТП КС в режимах монітора та сканера [6].

4. Програмні засоби інтелектуального діагностування ТП КС

Для ефективної організації процесу діагностування ТП було розроблено ПЗ, що реалізує ІТ інтелектуального діагностування ТП КС. ПЗ надає користувачу можливості виконати запуск антивірусного монітора, який відслідковуватиме системні події та має змогу виявляти підозрлі програмні обєкти в середовищі ОС, та виконати антивірусне сканування КС на предмет підміни системних файлів троянськими версіями.

При розробці ПЗ використовувалась мова програмування С++ із застосування середовища розробок програмних продуктів С++Builder 7.0 [8]. Також в програмній реалізації було використано модуль з пакету fuzzy logic toolbox, що входить до складу прикладного програмного забезпечення для вирішення технічних Matlab 7.0 [9].

ПЗ системи інтелектуального діагностування ТП КС (СІДТП) включає дві частини ПЗ: користувацьку і адміністраторську.

Користувацька частина передбачає можливість проведення антивірусного моніторингу системи, що діагностується. Користувач має змогу запустити антивірусний монітор чи відключити його. Користувацька частина ПЗ передбачає можливість проведення антивірусного сканування ПК за вимогою користувача для виявлення системних файлів, що можуть бути троянськими модифікаціями та виконувати деструктивні дії в ОС. Розроблена СПТП дає змогу здійснити оновлення антивірусних баз для забезпечення підвищення достовірності антивірусного діагностування КС. Адміністраторська частина передбачає можливість спеціалісту (чи спеціалістам) виконати формування (редагування) антивірусних баз, що використовуються в моніторі та сканері. Формування баз для монітора полягає в занесенні поведінкових моделей ТП на різних етапах їх ЖЦ. Антивірусні бази для сканера створюються шляхом вибірки файлів для сканування, генерації захищених двійкових послідовностей та детекторів.

Для демонстрації роботи ПЗ було програмно згенеравано 324 програмних об'єкти з функційним навантаженням ТП. Дані програми потенційно невідомі для антивірусних баз фірм-розробників антивірусного ПЗ. В своїй основі набір згенерованого ПЗ має такі властивості: 18 - класу rootki; 81 - класу BackDoor; 32 - класу Trojan-PSW; 21 - класу Trojan-Clicker; 85 – класу Trojan Downloader; 23 - класу Trojan-Dropper; 15 – класу Trojan-Proxy; 33 - класу Trojan-Spy; 16 - класу Trojan-Notifier. Чисельне співвідношення згенерованих програм відповідає процентному співвідношенню типів ТП, які найчастіше інфікують КС. З вказаного набору 8 програмних об'єктів виконували підміну системних файлів, але не виконували деструктивних дій в КС.

Інтерфейсні вікна результатів діагностування ТП КС із залученням розробленого ПЗ подані на рис. 5 та 6.

Результати роботи програмної реалізації СІДТП демонструють спроможність діагностування ТП із застосуванням компонентів штучного інтелекту. Так із згенерованого набору програмних об'єктів видно виявлення антивірусним монітором поведін-

ки, що занесені до бази і найбільше «схожі» до ТП класів backdoor, rootkit та ddos. Антивірусне сканування ПК виявило модифікацію системного файлу directx.sys.

Приймемо n_i, $i = \overline{1,s}$, $s \in N$ як кількість троянських програм і-того класу, k_i – кількість троянських програм, виявлених антивірусним засобом, тоді достовірність антивірусного діагностування p троянських програм буде:

$$p = \frac{\sum_{s=1}^{s} \alpha_i \cdot k_i}{\sum_{i=1}^{s} \alpha_i \cdot n_i}, \qquad (6)$$

де α_i – відсоток і-го класу від усіх троянських програм, $0 \le \alpha_i \le 1$.

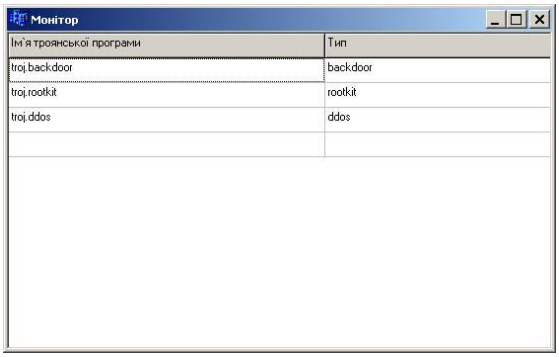

Рис. 5. Результати роботи антивірусного діагностування в режимі монітора

Розрахунки за формулою (6) показали приріст достовірності діагностування ТП КС на 5 – 15% у порівнянні з існуючими відомими АПЗ.

Висновок

В результаті проведених досліджень вперше запропоновано концепцію здійснення інтелектуального діагностування троянських програм, суть якої полягає у застосуванні апарату нечіткої логіки як засобу здійснення висновку щодо можливості присутності троянської програми в КС та використанні алгоритмів штучних імунних систем для виявлення факту підміни системних файлів троянськими версіями.

Дістав подальшого розвитку процес антивірусного діагностування КС, на основі вперше одержаної моделі процесу інтелектуального діагностування троянських програм, в основу якої покладено використання нечітку логіку та алгоритми штучних імунних систем, що дало можливість інтелектуалізувати процес діагностування троянських програм КС.

Розроблено нову інформаційну технологію інтелектуального діагностування ТП КС, яка не потребує побудови баз вірусних сигнатур та дає змогу діагностувати нові невідомі троянські програми. ІТ дала можливість підвищити достовірність діагностування невідомих троянських програм до 74%, що складає приріст достовірності у 5 – 15% у порівнянні з існуючими інформаційними технологіями діагностування троянських програм.

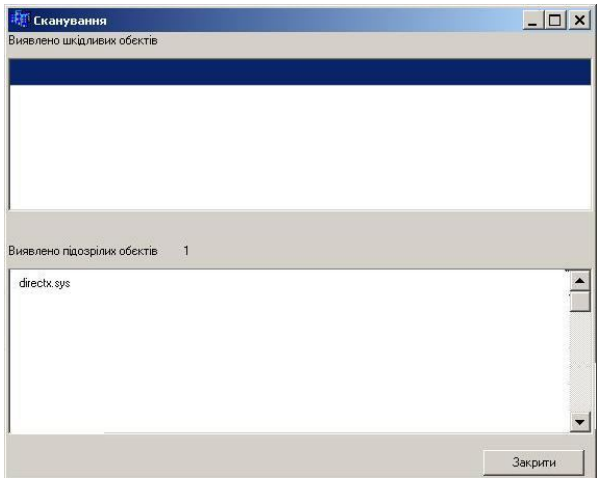

Рис. 6. Результати роботи антивірусного діагностування в режимі сканера

Література

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

2. Савенко О. Модель процесу пошуку троянських програм в персональному комп'ютері / О. Савенко, С. 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 міжнародної науково-практичної конференції [«Інтелектуальний аналіз інформації ІАІ-2008»], (Київ, 14-17 травня 2008 р.) / редкол.: С.В. Сирота [та ін]. – К. :Просвіта. – 2008. – С. 413-431.

5. Графов Р.П. Использование нечеткой логики для поиска троянских программных продуктов в вычислительных системах / Р.П. Графов, О.С. Савенко, С.М. Лисенко // Вісник Чернівецького національного університету, 2009. – № 6. – С. 85-91.

6. Савенко О.С. Алгоритми пошуку троянських програм в персональних комп'ютерах // Савен-

ко О.С., Лисенко С.М. Радіоелектронні і комп'ютерні системи, 2009. – № 5. – С.120-126.

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

8. Borland C++ Builder 6. Руководство разработчика / Джаррод Холингворт, Боб Сворт, Марк Кэшмэн, Поль Густавсон. – М.: Вильямс, 2004. – 976 с.

9. Леоненков А.В. Нечеткое моделирование в среде MATLAB в fuzzyTECH / А.В. Леоненков. – СПб.: БХВ – Петербург, 2005. – 736 с.

Поступила в редакцию 13.01.2010

Рецензент: доктор технічних наук, професор, завідувач кафедри системного програмування Хмельницького національного університету В.М. Локазюк, Хмельницький національний університет, Хмельницький, Україна.

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ИНТЕЛЛЕКТУАЛЬНОГО ДИАГНОСТИРОВАНИЯ ТРОЯНСКИХ ПРОГРАММ КОМПЬЮТЕРНЫХ СИСТЕМ

О.С. Савенко, С.М. Лысенко

Исследована ситуация относительно разработки вредоносного программного обеспечения. Предложена концепция осуществления диагностирования троянских программ с применением аппарата нечеткой логики и алгоритмов искусственных иммунных систем. Получена модель процесса диагностирования троянских программ, которое дало возможность интеллектуализировать процесс диагностирования троянских программ. Разработано новую информационную технологию интеллектуального диагностирования троянских программ компьютерных систем, которая не нуждается в построении баз вирусных сигнатур и дает возможность диагностировать новые неизвестные троянские программы с высокой достоверностью.

Ключевые слова: троянская программа, информационная технология, интеллектуальное диагностирование троянских программ компьютерных систем, нечеткий логический вывод, алгоритм негативного отбора.

THE INFORMATION TECHNOLOGY OF THE INTELLIGENT DIAGNOSIS OF THE TROJAN PROGRAMS OF COMPUTER SYSTEMS

O.S. Savenko, S.M. Lysenko

The situation of development of the malicious software is researched. Conception of realization of diagnosis of the trojan programs is offered by the use of fuzzy logic and algorithms of the artificial immune systems. The model of process of intelligent diagnosis of the trojan programs is proposed. New information technology of the intelligent diagnosis of the trojan programs of the computer systems is developed. New information does not need the construction of bases of viral signatures and is able to diagnose the new unknown trojan programs with high confidence.

Key words: trojan program, information technology, intelligent diagnosis of the trojan programs of computer systems, fuzzy inference, negative selection algorithm.

Савенко Олег Станіславович – канд. техн. наук, доцент, декан факультету комп'ютерних систем та програмування Хмельницького національного університету, Хмельницький, Україна, e-mail: kism@beta.tup.km.ua.

Лисенко Сергій Миколайович – асистент кафедри системного програмування Хмельницького національного університету, Хмельницький, Україна, e-mail: sirogyk@ukr.net, foros@datasvit.km.ua, kism@beta.tup.km.ua