УДК 621.391
DOI https://doi.org/10.32782/2663-5941/2024.1.1/39

Нічепорук А.О.
Хмельницький національний університет

Бармак О.В.
Хмельницький національний університет

Манзюк Е.А.
Хмельницький національний університет

Продеус М.С.
Хмельницький національний університет

МЕТОД ВИЯВЛЕННЯ МЕТАМОРФНИХ ВІРУСІВ РОЗПОДІЛЕНИМИ СИСТЕМАМИ НА ОСНОВІ ПОРІВНЯННЯ ЕКВІВАЛЕНТНИХ ФУНКЦІОНАЛЬНИХ БЛОКІВ

Представлена стаття присвячена проблемі виявлення зловмисного програмного забезпечення, зокрема метаморфним вірусам. Складність виявлення та ідентифікації такого типу зловмисного програмного забезпечення зумовлена використанням ними технік переміщення та переписування власного коду при поширенні. Кожна нова копія, що створюється метаморфним вірусом відрізняється від вже існуючих. Зазначена особливість такого типу вірусів дозволяє нівелювати використання сигнатурного аналізу, що лежить в основі більшості сучасних антивірусних засобів. Для вирішення цієї проблеми запропоновано метод виявлення метаморфних вірусів розподіленими системами на основі порівняння еквівалентних функціональних блоків. Представлений метод ґрунтується на отриманні характеристичних ознак, за якими можна ідентифікувати метаморфні віруси. Цими ознаками є кількісні показники, що визначають схожість зразків метаморфних вірусів між собою за дистанцію Дамерау-Левенштейна, кількістю операцій вставки, видалення, перестановки та співпадіння опкодів, а також за поведінкою програмою. Вихідними даним для отримання кількісних ознак є дизасембльовані лістинги операційних кодів (опкодів): підозрілої програми та її зміненої версії, що сформована в захищеному віртуальному середовищі. Формування логічної ознаки (поведінки) здійснюється на основі опрацювання послідовності API викликів функцій, що здійснює програма в процесі власного виконання. Для проведення виявлення метаморфних вірусів залучено систему нечіткого логічного висновку. Проведено експериментальні дослідження по визначенню оптимальної метрики подібності, що залучається до визначення еквівалентних функціональних блоків, а також порогу подібності функціональних блоків. В результаті проведеного експерименту ефективність виявлення метаморфних вірусів NGVCK склала 94%, а рівень хибних спрацювання 4% при порозі подібності функціональних блоків на рівні 0,6.

Ключові слова: *зловмисне програмне забезпечння, метаморфний вірус, NGVCK.*

Постановка проблеми. В умовах постійного розвитку технологій та зростання віртуальних загроз, виявлення зловмисного програмного забезпечення (ЗПЗ) стає актуальним завданням для забезпечення безпеки інформаційних систем. Зловмисне програмне забезпечення включає в себе різноманітні загрози, такі як віруси, троянські програми, worms та інші форми шкідливого коду, які можуть завдати серйозної шкоди конфіденційності, цілісності та доступності інформації. Актуальність цієї проблематики визначається зростанням кількості та складності атак, які спрямовані на користувачів, компанії та урядові установи. Зловмисне програмне забезпечення може використовуватися для крадіжки особистих даних, розповсюдження шахрайської інформації, атак на критичну інфраструктуру та інші злочинні дії.

У цьому контексті, особливу увагу слід приділити проблематиці виявлення метаморфних вірусів. Головною особливістю, що відрізняє метаморфний вірус від іншого типу ЗПЗ є використання обфускації [1]. Застосування методів обфускації дозволяє видозмінити синтаксис програмного коду, проте залишити семантичну складову роботи вірусного алгоритму. Ця особливість робить створення сигнатур для вірусного коду неможливим, що створює умови для появи нових форм шкідливого програмного забезпечення.

Також важливим фактором є доступність метаморфних генераторів, які дозволяють використовувати методи обфускації без спеціальних знань у цій області.

Таким чином враховуючи ці загрози, розробка ефективних методів виявлення ЗПЗ й, зокрема, метаморфних вірусів стає стратегічно важливою задачею для забезпечення цифрової безпеки в різних сферах.

Аналіз останніх досліджень і публікацій. Для виявлення метаморфного зловмисного програмного забезпечення відомі підходи відрізняються набором ознак, за якими здійснюється віднесення досліджуваного зразка до одного із класів – ЗПЗ або довірених додатків [2-6]. Ці ознаки можуть включати як статичні так і динамічні атрибути, такі як опкоди (кодові інструкції), структуру графу потоку керування (control flow graph), API виклики, поведінку виконуваних файлів (мережеву активність, зовнішнє середовище виконання, таке як наприклад, системний реєстр) або їх комбінації. Таким чином відомі методи можна класифікувати відповідно до способу отримання цих характеристик, що дозволяє розділити їх на статичні, динамічні та комбіновані методи виявлення [7].

При статичному аналізі ЗПЗ здійснюється дослідження зразка без його фактичного виконання. Під час статичного аналізу аналізуються вихідний код або виконуваний файл, а також його структура, без виконання фактичних операцій.

Динамічний аналіз виконується шляхом спостереження за діями програми під час її роботи в реальному або безпечному середовищі. За допомогою динамічного аналізу можна ідентифікувати поведінку, функціональність та індикатори програми, які дозволяють визначити, чи є програма злоякісною чи ні. Процес динамічного аналізу складається з трьох етапів: налаштування та дезінфекція середовища, виконання шкідливої програми та моніторинг та логування поведінки шкідливого програмного забезпечення.

Ще однією категорію за якою можна класифікувати відомі методи виявлення метаморфного ЗПЗ є методами опрацювання цих ознак, що включають в себе такі методи як дерева рішень, нейронні мережі, генетичні алгоритми, приховані марківські моделі, тощо.

Проте, разом із досить високою достовірністю виявлення, сучасні методи характеризуються значним рівнем хибнопозитивних та хибнонегативних спрацювань, що зменшує загальну точність виявлення метаморфних вірусів, і як наслідок задача розробки нових методів виявлення є досить важливим завданням.

Метою статті є аналіз методів виявлення метаморфних вірусів розподіленими системами на основі порівняння еквівалентних функціональних блоків.

Виклад основного матеріалу досліджень. Для виявлення метаморфних вірусів запропоновано метод, що складається із наступних кроків: підготовка даних, локалізація місця пошуку, пошук еквівалентних функціональних блоків, уточнення вибору еквівалентних функціональних блоків та класифікація. Розглянемо детальніше кожен крок методу. Схему функціонування методу виявлення метаморфних вірусів зображено на рис. 1.

Підготовка даних. Для виявлення метаморфних вірусів, як й іншого ЗПЗ, основним етапом є виокремлення характеристичних ознак, опрацювання яких засобами машинного навчання дозволяє віднести досліджувану програму до класу корисних програм чи метаморфних вірусів.

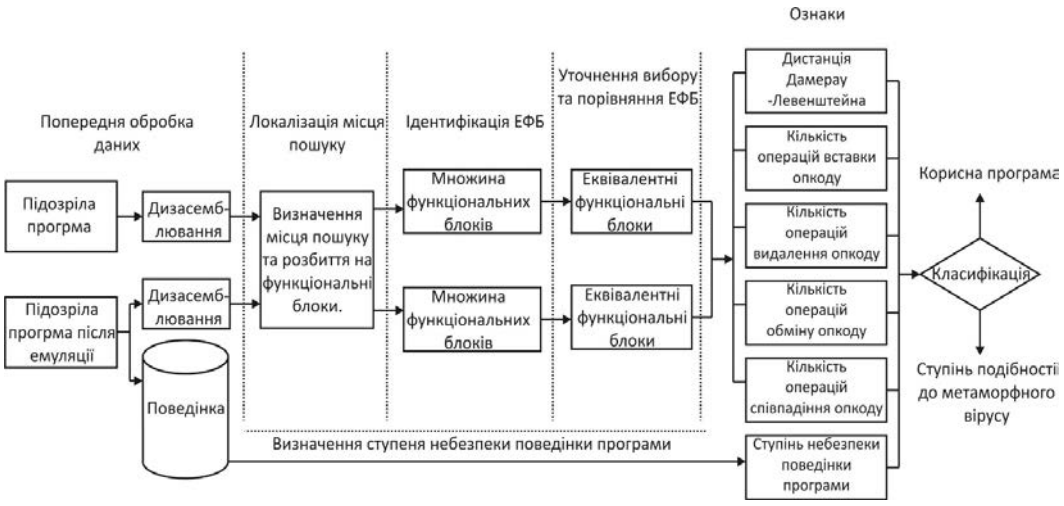

Рис. 1. Схема функціонування методу виявлення метаморфних вірусів

У даній роботі розділимо ознаки для ідентифікації метаморфних вірусів на дві групи: кількісні ознаки та логічна ознака. Кількісні ознаки дозволяють оцінити на скільки відрізняються між собою дві версії метаморфного вірусу, в той час як логічна ознака визначає поведінку підозрілої програми. Комбінація зазначених ознак дозволяє розмежувати ситуацію при якій відмічається видозмінення коду програми, попри відсутність шкідливої активності. Така ситуація може виникнути, наприклад, коли корисні програми застосовують обфускацію до власного коду, переслідуючи при цьому ідею захисту прав інтелектуальної власності, а не механізм уникнення розпізнання антивірусною програмою.

Вихідними даним для отримання кількісних ознак є дизасембльовані лістинги операційних кодів (опкодів): підозрілої програми та її зміненої версії. Формування логічної ознаки (поведінки) здійснюється на основі опрацювання послідовності API викликів функцій, що здійснює програма в процесі власного виконання.

З метою створення зміненої версії підозрілої програми та формування її поведінки, здійснюється запуск підозрілої програми в середовищі модифікованого емулятора. Під поняттям «модифікований емулятор» розуміється захищене середовище для емуляції апаратного забезпечення комп'ютерної системи, параметри та налаштування якого змінюються при кожному новому запуску, з метою уникнення розпізнання вірусною програмою власного виконання у віртуальному середовищі [8].

Таким чином, в результаті виконання кроку підготовки даних буде отримано два лістинги опкодів (для підозрілої програми та її зміненої версії) та лістинг API викликів, що представляє поведінку підозрілої програми.

Локалізація місця пошуку. При вирішені задачі виокремлення характеристичних ознак для ідентифікації метаморфних вірусів важливим етапом є визначення місця пошуку всередині програми. Оскільки, складовими одиницями у структурі виконуваних файлів PE EXE є секції, то й опрацювання даних буде здійснюється тільки у межах визначених секцій, а не у всьому виконуваному файлі. Ці обмеження місця пошуку зумовлено значними часовими та ресурсними витратами.

Визначимо поняття «секція для порівняння» як секція в структурі підозрілої програми або її зміненої версії, що використовуватиметься для виокремлення кількісних характеристичних ознак. Після визначення точки входу у програму, здійснюється пошук секції для порівняння. Секція буде маркуватись як «секція для порівняння», якщо її ім'я не належить до стандартних імен секцій, вона володіє атрибутом виконання, або якщо в секції міститься команда довгого переходу в останню секцію. Слід зазначити, що для формування кількісних ознак, локалізація місця пошуку проводиться як для підозрілої програми, так й до її зміненої версії.

Пошук еквівалентних функціональних блоків. Після визначення місця пошуку характеристичних ознак у підозрілій програми та її зміненій версії наступний етап передбачає розбиття лістингів опкодів на функціональні блоки (ФБ), з метою пошуку еквівалентних частин коду – еквівалентних функціональних блоків. Порівняння еквівалентних функціональних блоків дозволить отримати кількісні ознаки, що надасть змогу оцінити на скільки дві версій метаморфного вірусу відрізняються між собою (або схожі між собою). Отриманні ознаки будуть покладені в основу вектора ознак схожості зразка коду до метаморфного вірусу.

В запропонованому методі ФБ *FB* буде називатись максимальна послідовність дизасембльованих інструкцій $\{I_1, I_2, ..I_m\}$, що характеризується наступним властивостями: потік керування обов'язково заходить в блок через першу інструкцію; всередині блоку не може бути інструкції безумовного або умовного переходу (інструкції виклику підпрограми допускаються), всі інструкції в блоці виконуються послідовно; в кінці блоку присутня принаймні одна інструкція умовного або безумовного переходу.

З метою спрощення процесу аналізу та обробки операнди інструкцій не враховуються.

Тоді позначимо програму до емуляції через F_P, а після емуляції – F_S. Після виконання процесу дизасемблювання, з використанням інтерактивного дизасемблера IDA Pro, отримаємо дві множини функціональних блоків: $FB^{F_P} = \left\{ fb_1^{F_P}, fb_2^{F_P}, \ldots, fb_m^{F_P} \right\}$ та $FB^{F_S} = \left\{ fb_1^{F_S}, fb_2^{F_S}, \ldots, fb_n^{F_S} \right\}$. Тоді для пошуку еквівалентних ФБ використаємо статистичну метрику Term Frequency – Inverse Document Frequency, яка застосовуватиметься до кожного окремого функціонального блоку для програм F_P та F_S:

$$S_{FB} = \frac{n_i}{\sum_k n_i} * \log\left(\frac{N + 1.0}{n_j}\right) \qquad (1)$$

де n_i – кількість входжень *i*-го опкоду у функціональний блок;

$k = \overline{1, k_a}$ – кількість опкодів у функціональному блоці, де k_a – загальна кількість асемблерних інструкцій;

N – загальна кількість функціональних блоків, причому $N_{F_P} \neq N_{F_P}$;

n_j – кількість функціональних блоків в якому присутній *i*-й опкод.

Результатом виконання етапу обчислення статистичної оцінки присутності опкоду у ФБ для програми до емуляції F_P та програми після емуляції F_S є матриці, рядки яких визначають ФБ програми, а стовпці – опкоди, що присутні в функціональному блоці. Кожна комірка матриці визначає оцінку появи *i*-го опкода в *j*-му функціональному блоці.

Після отримання ФБ наступним кроком здійснюється визначення еквівалентних функціональних блоків. Для цього обчислюється оцінка схожості двох функціональних блоків з програми F_P та програми F_S. Для реалізації цього етапу використовуються метрики відстані.

Якщо значення оцінки схожості двох ФБ менше порогового значення δ, тобто $E\left(FB_i^{F_P}, FB_j^{F_S}\right) \le \delta$, то виконується повторне обчислення оцінки схожості для ФБ із програми $FB_i^{F_P}$ та наступного ФБ, що слідує за блоком $FB_j^{F_S}$. Зазначені вище дії повторюються поки значення оцінки схожості буде менше або рівне порогового значення. Значення δ визначаєть експериментальним чином.

Уточнення вибору еквівалентних функціональних блоків. Якщо за результатами попереднього етапу виявиться так, що одному ФБ з F_P відповідатимуть декілька ФБ із програми F_S, тоді слід додатково виконати процес уточнення вибору еквівалентних ФБ.

Задля уточнення вибору еквівалентних ФБ визначимо ймовірність слідування операційних кодів у ФБ. Для цього для кожного еквівалентного ФБ $eFB_1^{F_S}, eFB_2^{F_S}, \ldots, eFB_n^{F_S}$ та блоку $FB_i^{F_P}$ сформуємо матрицю ймовірності слідування операційних кодів. Кожна комірка матриці скрадатиметься із відношення кількості появи пари опкодів до загальної кількості опокодів у рядку. Після отримання матриць еквівалентних ФБ для програми до та після емуляції здійснюється їх порівняння та вибір мінімальної оцінки схожості:

$$R = \frac{1}{N^2}\left(\sum_{i,j=1}^{N-1}\left|a_{i,j} - b_{i,j}\right|^2\right), \qquad (2)$$

де $a_{i,j}$ – комірка матриці для еквівалентного ФБ eFB^{F_P}, $b_{i,j}$ – комірка матриці для еквівалентного ФБ eFB^{F_S}, N – загальна кількість опокодів для пар блоків.

Порівння ЕФБ, формування вектора ознак та класифікація. Після отримання пар еквівалентних функціональних блоків, наступним етапом є їх порівняння із використанням метрики Дамерау-Левенштейна. В результаті порівняння пар ЕФБ здійснюється формування вектора ознак схожості зразка коду до метаморфного вірусу:

$$\overline{V_m} = \begin{matrix} L_{mod}(\mathrm{E}), L_{med}(E), X_{mod}(E), X_{med}(E), D_{mod}(E), \\ D_{med}(E), I_{mod}(E), I_{med}(E), M_{mod}(E), M_{med}(E), Y \end{matrix}, \qquad (3)$$

де $\mathrm{E} = \{\varepsilon_i\}_{i=1}^{n}$ пари еквівалентних функціональних блоків (ПЕФБ) між програмами до та після емуляції; n – загальна кількість ЕФБ; L_{mod} – модальне значення метрики Дамерау-Левенштейна між ПЕФБ ε_i програм до та після емуляції; L_{med} – медіанне значення метрики Дамерау-Левенштейна між ПЕФБ ε_i програм до та після емуляції; X_{mod} – модальне значення кількості необхідних операцій обміну опкодів між ПЕФБ для ε_i; X_{med} – медіанне значення кількості необхідних операцій обміну опкодів між ПЕФБ для ε_i; D_{mod} – модальне значення кількості необхідних операцій видалення операційних кодів між ПЕФБ для ε_i; D_{med} – медіанне значення кількості необхідних операцій видалення опкодів між ПЕФБ для ε_i; I_{mod} – модальне значення кількості необхідних операцій вставки опкодів між ПЕФБ для ε_i; I_{med} – медіанне значення кількості необхідних операцій вставки опкодів між ПЕФБ для ε_i; M_{mod} – модальне значення кількості співпадінь опкодів між ПЕФБ для ε_i; M_{med} – медіанне значення кількості співпадінь опкодів між ПЕФБ для ε_i; Y – ступінь небезпеки поведінки програми.

З метою оцінки ступеня небезпеки поведінки проводиться порівняння її поведінки із визначеним набором шкідливих поведінкових шаблонів. Якщо формується відповідність між діями підозрілої програми та одним із шкідливих шаблонів, то властивість вектора схожості для метаморфних вірусів приймає значення підозрілості (Low, Medium або High).

Для класифікації сформованого вектора ознак залучено систему нечіткого логічного висновку на основі алгоритму Мамдані [9]. Вхідними лінгвістичними виступатимуть ознаки із сформованого вектора ознак (3). В якості вихідної лінгвістичної змінної визначено ступінь подібності до метаморфного вірусу. Кожна вхідна та вихідна лінгвістична змінна задана терм-множиною: Low, Medium та High. В якості функцій приналежності для входів було обрано трапецієподібну, для виходів – трикутну.

Експериментальні дослідження. При визначені кількісних ознак ключовим питанням є вибір метрик за якими буде здійснюватись визначення ЕФБ. Фактично рішення, за яким ФБ будуть вважатись еквівалентними приймається на основі того, наскільки ці блоки є схожими між собою в метричному просторі. Тому, було проведено дослідження впливу метрик подібності, а також

значення порогу подібності ФБ на загальну ефективність виявлення метаморфних вірусів.

Для проведення експерименту, в якості тестових даних, було згенерувало 210 зразків метаморфних вірусів. Тестові зразки були сформовані за допомогою метаморфного генератора NGVCK. Весь процес дослідження проводився в середовищі модифікованого емулятора на основі Qemu. В якості метрик подібності було вибрано наступні метрики: евклідова метрика (m^1), квадрат евклідової метрики (m^2), махнетенська метрика (m^3), метрика Чебишева (m^4) та метрика Мінковського (m^5).

Таблиця 1

Залежність ефективності виявлення метаморфних вірусів NGVCK від метрики та значення порогу подібності ФБ

Поріг подібності двох ФБ	m1	m2	m3	m4	m5
δ=0,5	0,87	0,83	0,87	0,78	0,81
δ=0,6	0,88	0,86	0,88	0,83	0,86
δ=0,7	0,92	0,86	0,82	0,79	0,82
δ=0,5	0,87	0,91	0,81	0,91	0,81
δ=0,6	0,89	0,92	0,84	0,88	0,82
δ=0,7	0,84	0,86	0,84	0,87	0,83
δ=0,5	0,84	0,81	0,78	0,91	0,78
δ=0,6	0,82	0,84	0,86	0,94	0,83
δ=0,7	0,73	0,82	0,83	0,88	0,82
δ=0,5	0,85	0,82	0,84	0,91	0,86
δ=0,6	0,84	0,86	0,88	0,9	0,87
δ=0,7	0,86	0,81	0,79	0,87	0,82
δ=0,5	0,74	0,82	0,85	0,84	0,75
δ=0,6	0,75	0,82	0,87	0,81	0,82
δ=0,7	0,72	0,74	0,79	0,75	0,8

В результаті проведеного експерименту ефективність виявлення метаморфних вірусів NGVCK склала 94% (табл. 1). Для досягнення такого результату на етапі вибору еквівалентних функціональних блоків залучено манхетенську метрику зі значенням порогу подібності $\delta = 0.6$. При такому значені ефективності виявлення хибні спрацювання склали близько 4%. Стосовно залежності кількості еквівалентних функціональних блоків від значення порогу подібності, то слід відмітити, що при зменшені значення порогу збільшувалась кількість ЕФБ. Теоретично встановлення мінімального значення порогу подібності призведе до ситуації, при якій одному функціональному блоку підозрілої програми будуть еквівалентними всі функціональні блоки її зміненої версії, що є невірним.

Висновки. В роботі запропоновано метод виявлення метаморфних вірусів, що заснований на виділені характеристичних ознак для ідентифікації метаморфних вірусів. Цими ознаками є кількісні показники, що визначають схожість зразків метаморфних вірусів між собою за дистанцію Дамерау-Левенштейна, кількістю операцій вставки, видалення, перестановки та співпадіння опкодів, а також за поведінкою. Вихідними даним для отримання кількісних ознак є дизасембльовані лістинги опкодів: підозрілої програми та її зміненої версії, що сформована в захищеному віртуальному середовищі. Формування логічної ознаки здійснюється на основі опрацювання послідовності API викликів функцій, що здійснює програма в процесі власного виконання.

Проведено експерименти по визначенню оптимальної метрики подібності, що залучається до визначення еквівалентних функціональних блоків. В результаті проведеного експерименту ефективність виявлення метаморфних вірусів NGVCK склала 94%, а рівень хибних спрацювання 4% при порозі подібності функціональних блоків на рівні 0,6.

Список літератури:

1. Savenko O., Lysenko S., Nicheporuk A., Savenko B. Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search *CEUR-WS*. 2017. 1844. Pp. 555–569.

2. Jha A.K., Vaish A., Patil, S. A Novel Framework for Metamorphic Malware Detection. *SN Computer Science*. 2023. 4, 10. https://doi.org/10.1007/s42979-022-01433-1

3. Sahay S.K., Sharma A., Rathore H. Evolution of Malware and Its Detection Techniques. *Information and Communication Technology for Sustainable Development. Advances in Intelligent Systems and Computing*, 2019. 933. https://doi.org/10.1007/978-981-13-7166-0_14

4. Alsmadi T., Alqudah N. A Survey on malware detection techniques, *2021 International Conference on Information Technology (ICIT)*, Amman, Jordan, 2021, Pp. 371–376, doi: https://doi.org/10.1109/ICIT52682.2021.9491765.

5. Mohammed A. F., Marhusin M. F., Sulaiman R., Instrumenting API Hooking for a Realtime Dynamic Analysis, *2019 International Conference on Cybersecurity*, Negeri Sembilan, Malaysia, 2019, Pp. 49-52, doi: https://doi.org/10.1109/ICoCSec47621.2019.8971017.

6. Verma A. K., Sharma S. K., Malware Detection Approaches using Machine Learning Techniques-Strategic Survey, *2021 3rd International Conference on Advances in Computing, Communication Control and Networking*, Greater Noida, India, 2021, Pp. 1958-1962, doi: https://doi.org/10.1109/ICAC3N53548.2021.9725369.

7. Yang Y., Li Z., Wang H., Xu C., Ma X., Towards effective metamorphic testing by algorithm stability for linear classification programs. Journal of Systems and Software. 2021. 180. 111012.

8. Нічепорук А.О., Нічепорук Ю.О., Савенко Б.О., Стецюк М.В. Використання модифікованих емуляторів для виявлення метаморфних вірусів в корпоративній мережі. *Вісник Хмельницького національного університету: Серія «Технічні науки».* Хмельницький, 2017, № 2. С. 199–207.

9. Нічепорук А.О. Використання нечіткої класифікації для виявлення метаморфних вірусів в корпоративній мережі *Вісник Хмельницького національного університету: Серія «Технічні науки».* Хмельницький. 2016. № 4. С. 128–132.

Nicheporuk A.O., Barmal O.V., Manziuk E.A., Prodeus M.S. THE METHOD OF DETECTION OF METAMORPHIC VIRUSES BY DISTRIBUTED SYSTEMS BASED ON COMPARISON OF EQUIVALENT FUNCTIONAL BLOCKS

The presented article is devoted to the problem of detecting malicious software, in particular metamorphic viruses. The difficulty of detecting and identifying this type of malicious software is due to their use of techniques of substitution and rewriting their own code during propagation. Each new copy created by the metamorphic virus is different from the existing ones. This feature of this type of virus allows to eliminate the use of signature analysis, which is the basis of most modern antiviruses. To solve this problem, a method of detection of metamorphic viruses by distributed systems based on comparison of equivalent functional blocks is proposed. The presented method is based on obtaining characteristic features that can be used to identify metamorphic viruses. These features are quantitative indicators that determine the similarity of samples of metamorphic viruses to each other by the Damerau-Levenshtein distance, the number of operations of insertion, deletion, permutation and coincidence of opcodes, as well as by the behavior of the program. The source data for obtaining quantitative features are the disassembled listings of operational codes (opcodes): the suspicious program and its modified version, formed in a protected virtual environment. The formation of a logical feature (behavior) is carried out on the basis of the processing of the sequence of API calls of functions carried out by the program in the process of its own execution. A fuzzy logic inference system is used to detect metamorphic viruses. Experimental studies have been carried out to determine the optimal metric of similarity, which is involved in the definition of equivalent functional blocks, as well as the threshold of similarity of functional blocks. As a result of the conducted experiment, the detection efficiency of NGVCK metamorphic viruses was 94%, and the level of false positives was 4% at a threshold of similarity of functional blocks at the level of 0.6.

Key words: *malware, metamorphic virus, NGVCK.*