



МІНІСТЕРСТВО
ЕКОНОМІЧНОГО
РОЗВИТКУ І ТОРГІВЛІ
УКРАЇНИ

УКРАЇНА

(19) UA

(11) 118456

(13) U

(51) МПК

G06F 21/55 (2013.01)

(12) ОПИС ДО ПАТЕНТУ НА КОРИСНУ МОДЕЛЬ

(21) Номер заявки: **u 2017 01743**

(22) Дата подання заявки: **23.02.2017**

(24) Дата, з якої є чинними
права на корисну
модель: **10.08.2017**

(46) Публікація відомостей
про видачу патенту: **10.08.2017, Бюл.№ 15**

(72) Винахідник(и):

**Савенко Олег Станіславович (UA),
Лисенко Сергій Миколайович (UA),
Бобровнікова Кіра Юліївна (UA),
Нічепорук Андрій Олександрович (UA),
Савенко Богдан Олегович (UA)**

(73) Власник(и):

**ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ,
вул. Інститутська, 11, м. Хмельницький,
29016 (UA)**

(54) СПОСІБ ВИЯВЛЕННЯ МЕТАМОРФНИХ ВІРУСІВ НА ОСНОВІ СТАТИСТИЧНИХ МЕТРИК ДЛЯ ВИЗНАЧЕННЯ ЕКВІВАЛЕНТНИХ ФУНКЦІОНАЛЬНИХ ПРОГРАМНИХ БЛОКІВ

(57) Реферат:

Спосіб виявлення метаморфних комп'ютерних вірусів в комп'ютерних системах на основі використання статистичних метрик для визначення еквівалентних функціональних програмних блоків. Використовують уточнення вибору функціональних блоків за рахунок критерію вибору мінімальної оцінки схожості, процес вибору функціональних блоків для порівняння складається з двох етапів, де на першому етапі визначають еквівалентні функціональні блоки на основі обчислення статистичної оцінки появи інструкцій у блоці, а на другому етапі здійснюють уточнення еквівалентних блоків та вибір найбільш придатного блока, який і буде використаний для формування оцінки схожості між програмою до емуляції та програмою після емуляції. При цьому основні кроки способу вимагають визначення еквівалентних функціональних блоків, здійснення уточнення визначення еквівалентних функціональних блоків, формування вектора ознак схожості копій метаморфних вірусів та отримання висновку щодо інфікування системи метаморфним вірусом.

UA 118456 U

Корисна модель належить до інформаційної безпеки і може використовуватись для виявлення метаморфних комп'ютерних вірусів в комп'ютерних системах на основі використання статистичних метрик для визначення еквівалентних функціональних програмних блоків.

Задача виявлення метаморфних комп'ютерних вірусів фактично зводиться до задачі визначення еквівалентності двох програм [1, 2] або їхніх частин. Через проблеми, пов'язані з організацією великої кількості порівнянь та обчислень, обчислювальна складність таких способів є дуже великою і не може бути використана для практичної реалізації. Крім того, метаморфні коди створюють з використанням різноманітних методів обфускації, зокрема і перемішування частин програм, що дозволяє зловмисникам отримувати дуже багато різних копій однієї і тієї ж програми. Фактично синтаксис таких копій програми різний, а семантика є однаковою. Дослідження семантики програм потребує також наявності еталонів всеможливих варіантів обфускації коду для порівняння. А множина таких еталонів динамічно поповнюється щоденно, що ускладнює застосування на практиці програмних засобів виявлення метаморфного коду розроблених на основі такого підходу. Тому, на практиці все частіше використовують способи, які не передбачають глибокої деталізації коду досліджуваних програм, а за основу беруть певний рівень узагальнення, що дозволяє зменшити обчислювальну складність при реалізації таких способів і досліджувати наявність нового метаморфного коду з певною точністю, на відміну від способів [1, 2], в яких ставиться за мету отримання точної відповіді при заданих початкових даних можливих варіантів обфускації. До таких наближених способів належать способи, представлені в [3-7].

Авторами роботи [3] запропоновано спосіб, що оснований на динамічному формуванні траси виконання інструкцій та побудови графу операційних кодів (опкодів). Для перетворення графа у ланцюг Маркова, зв'язки між вершинами позначаються перехідними ймовірностями, на основі даних, зібраних у процесі формування траси виконання інструкцій. Матриця схожості між двома графами формується на основі використання комбінації ядер графа. Для здійснення класифікації матриці схожості використовується метод опорних векторів. Даний спосіб показав ефективність виявлення на рівні 96,41 %, включаючи поліморфні віруси, що значно перевищує відомі антивірусні засоби, проте автори роботи розглядали метаморфні віруси, як підмножину поліморфних вірусів і не виділяли окремо їх особливості.

В роботі [4] представлено спосіб виявлення метаморфних вірусів на основі пошуку схожості графів опкодів. Для побудови орієнтованого графа опкодів використовується множина дизасембльованих інструкцій, де вершинами виступають опкоди, а ребрами - переходи між інструкціями. Кожне ребро графа містить ймовірність переходу з інструкції А до інструкції В з урахуванням кількості подібних переходів. Класифікація здійснюється на основі використання SVM (Support Vector Machine - метод опорних векторів), де прямо здійснюється порівняння графів опкодів та формується оцінка схожості на основі HMM (Hidden Markov Model - прихована марківська модель) класифікації. Однак, збільшення розміру файла призводить до збільшення кількості опкодів, і відповідно, збільшує розмір графа, що призводить до вирішення NP (NP - клас задач для яких не знайдено поліноміальні алгоритми розв'язку) повної задачі ізоморфізму графів.

У роботі [5] виявлення здійснюється на основі побудови гістограми частот інструкції. З цієї метою використано найпоширеніші інструкції асемблерного коду, на основі дослідження частоти появи опкодів у вірусних програмах. Після отримання лістингу інструкцій виконується їх нормалізація, що визначається як співвідношення і-тої інструкції до загальної кількості інструкцій та формування гістограм. Відповідні гістограми порівнюються з використанням Евклідової відстані з формуванням відповідної матриці відстаней для кожної гістограми. Задача класифікації метаморфного вірусу аналогічно вирішується за допомогою методу опорних векторів.

Інший статистичний спосіб, представлений у роботі [6] для визначення схожості метаморфних вірусів використовує 11 статистичних метрик (TF-IDF, TF-IDF-CF, CPD тощо) і згідно з ним аналізують їх з використанням тесту McNemar (тест в статистиці, дозволяє визначити, чи є модифікація змінної значущою або випадковою для досліджуваної групи індивідів.), що оснований на хі-квадрат розподілі випадкової величини. Для здійснення класифікації використано алгоритми J48, Random Forest та AdaBoostM1 (алгоритми машинного навчання для здійснення класифікації).

Однак, розроблені способи [3-6] на основі статистичної оцінки інструкцій, є неефективним для метаморфних вірусів, що використовують техніку перемішування блоків коду, оскільки частота появи інструкцій в змінній версії метаморфного вірусу не зміниться. Їх реалізація не дозволяє з достатнім ступенем достовірності виявляти метаморфні віруси.

Аналіз відомих способів виявлення метаморфних вірусів показав необхідність розробки нових способів з метою підвищення достовірності та ефективності виявлення метаморфних вірусів.

Найбільш близькими до заявленого способу можна вважати мультиагентний спосіб локалізації бот-мереж у корпоративних комп'ютерних мережах [7-9], який дозволяє здійснювати локалізацію бот-мереж з метаморфними кодами на основі їх моделей [8], а також способи описані в [5, 6] на основі пошуку схожості графів опкодів та на основі побудови гістограми частот інструкції.

У відомих рішеннях є недостатньо висока достовірність виявлення нових метаморфних вірусів.

Задачею корисної моделі є підвищення достовірності виявлення нових метаморфних вірусів у комп'ютерних системах на основі використання статистичних метрик.

Поставлена задача вирішується тим, що для підвищення достовірності та ефективності виявлення нових метаморфних вірусів пропонується здійснювати вибір блоків для порівняння за рахунок критерію вибору мінімальної оцінки схожості, що дозволить зменшити рівень хибних спрацювань та підвищить достовірність виявлення.

Процес вибору функціональних блоків для порівняння складається з двох етапів. Перший етап визначає еквівалентні функціональні блоки на основі обчислення статистичної оцінки появи інструкцій у блоці. На другому етапі здійснюється уточнення еквівалентних блоків та вибір найбільш придатного блока, який і буде використаний для формування оцінки схожості між програмою до емуляції F_p та програмою після емуляції F_s .

В розробленому способі функціональним блоком FB назвемо максимальну послідовність дизасембльованих інструкцій $\{i_1, i_2, \dots, i_m, i_j\}$, що характеризується наступним властивостями:

- потік керування обов'язково заходить в блок через першу інструкцію;
- всередині блока не може бути інструкції безумовного або умовного переходу (інструкції виклику підпрограми допускають), всі інструкції в блоці виконуються послідовно;
- в кінці блока присутня принаймні одна інструкція умовного або безумовного переходу.

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

Представимо програму F у вигляді орієнтованого графа. Нехай, V - множина функціональних блоків програми F , тобто $V = \{FB_1, FB_2, \dots, FB_n\}$ і $E \subseteq V \times V \times \{\text{True}, \text{False}\}$ є переходом потоку керування між блоками, де True та False визначають умови переходу, тоді $G = \{V, E\}$ буде орієнтованим графом, де вершинами виступають функціональні блоки, а ребрами - зв'язки між блоками в потоці керування програмою.

Здійснимо визначення еквівалентних функціональних блоків. З метою уникнення виявлення антивірусними засобами метаморфні віруси використовують широкий спектр технологій, зокрема вставка команд сміття, переміщення блоків, використання еквівалентних інструкцій та регістрів [1-8]. Використання зазначених технологій дозволяє створювати метаморфні копії, що однакові за функціоналом, проте різні в реалізації (таблиця). Але, як видно з таблиці, частина команд залишаються однаковими. Це зумовлено тим, що при створенні нової копії, метаморфний вірус виконує дизасемблювання власного коду та заново компілює код, попередньо визначивши можливі місця видозмінення коду (за допомогою методів обфускації) і, з використанням закладеного у його тіло генератора, формує можливий набір альтернативних інструкцій або команд-сміття, що вставляються у відповідне місце (причому ця множина є скінченною). Тому, нова копія метаморфного вірусу буде складатись з множини змінених команд та з деякої множини команд, що не змінились. При цьому кожного разу ці місця вставок будуть змінюватись від копії до копії, що накладає обмеження на використання сигнатурного методу.

Еквівалентними функціональними блоками програм A та B будемо називати функціональний блок FB програми A , якому відповідають один або більше функціональних блоків з програми B , що виконують однакові функції, і до яких застосовані методи обфускації програмного коду (можливо їх комбінації).

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: $FB^{F_p} = \{fb_1^{F_p}, fb_2^{F_p}, \dots, fb_m^{F_p}\}$ та $FB^{F_s} = \{fb_1^{F_s}, fb_2^{F_s}, \dots, fb_n^{F_s}\}$. Тоді, для пошуку еквівалентних функціональних блоків використаємо статистичну метрику Term Frequency-Inverse Document Frequency, яка застосовуватиметься до кожного окремого функціонального блока для програм F_p та F_s :

$$s_{FB} = \frac{n_i}{\sum_{i=0}^k n_i} * \log \left(\frac{N+1,0}{n_j} \right), (1)$$

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;

5 N - загальна кількість функціональних блоків, причому $N_{F_p} \neq N_{F_s}$;

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 $M(FB^{F_p})$ та $M(FB^{F_s})$, рядки яких визначають функціональні блоки програми, а стовпці - опкоди, що присутні в функціональному блоці. Кожна комірка матриці визначає оцінку появи i -го опкода в j -му функціональному блоці (Фіг. 1).

10 З метою визначення еквівалентних функціональних блоків, на наступному кроці обчислюється оцінка схожості двох функціональних блоків з програми F_p та програми F_s . З цією метою використаємо квадрат Евклідової метрики:

$$15 \quad E(FB_i^{F_p}, FB_j^{F_s}) = \sum_{i=0, j=0}^k (s_i - s_j)^2, (2)$$

де s_i - оцінка появи опкодів в i -тому блоці програми F_p , s_j - оцінка появи опкодів в j -тому блоці програми F_s .

Якщо значення оцінки схожості двох функціональних блоків менше визначеного порогового значення δ , тобто $E(FB_i^{F_p}, FB_j^{F_s}) \leq \delta$, тоді виконується повторне обчислення оцінки схожості для

20 функціонального блока з програми $FB_i^{F_p}$ та наступного блока, що іде за блоком $FB_j^{F_s}$, тобто $E(FB_i^{F_p}, FB_j^{F_s} + FB_{j+1}^{F_s})$. Зазначені вище дії повторюються, поки значення оцінки схожості буде меншим або рівним пороговому значенню. Величина порогового значення δ визначається експериментальним чином.

Після виконання даного етапу може виникнути ситуація, коли одному функціональному блоку з програми F_p відповідатимуть декілька функціональних блоків з програми F_s (Фіг. 2). Це пояснюється тим, що в метаморфному вірусі застосована технологія розбиття на блоки власного коду.

30 Приклад схематичного представлення розміщення в двовимірній площині еквівалентних ФБ програми до та після емуляції подано на Фіг. 2. Так одному блоку з програми до емуляції, можуть відповідати, наприклад, 5 еквівалентних ФБ з програми після емуляції. Таким чином, з метою усунення невизначеності, необхідно здійснити уточнення еквівалентних функціональних блоків.

Здійснимо уточнення визначення еквівалентних функціональних блоків. Задача уточнення визначення еквівалентних функціональних блоків зводиться до вибору одного з них, що отримані на попередньому кроці. З цією метою вибиратимемо мінімальне значення схожості серед множини еквівалентних функціональних блоків:

$$FB_i^{F_p} \equiv \min(eFB_1^{F_s}, eFB_2^{F_s}, \dots, eFB_n^{F_s}), (3)$$

де $eFB_1^{F_s}, eFB_2^{F_s}, \dots, eFB_n^{F_s}$ - еквівалентні функціональні блоки відносно до блока $FB_i^{F_s}$.

40 З метою уточнення еквівалентних функціональних блоків визначимо імовірність наступності опкодів в функціональному блоці. Для цього для кожного еквівалентного функціонального блока $eFB_1^{F_s}, eFB_2^{F_s}, \dots, eFB_n^{F_s}$ та блока $FB_i^{F_p}$ побудуємо матрицю імовірності наступності опкодів. Кожна комірка матриці буде складатись з відношення кількості появи пари опкодів до загальної кількості опкодів в рядку. Псевдоалгоритм заповнення комірок матриці імовірності наступності подано нижче:

45 for each cells in row begin
if $o_i \rightarrow o_{i+1}$ then

$$\text{occur}(o_i, o_{i+1}) = \text{occur}(o_i, o_{i+1}) + 1;$$

$$M_i^{\text{probabilities}} = \frac{\text{occur}(o_i, o_{i+1})}{\sum_{i=1}^{\text{row}} o_i}.$$

Наприклад, якщо функціональний блок заданий наступною послідовністю опкодів: mov, push, lea, pop, mov, push, push, push, call, mov, тоді матриця ймовірності наступності опкодів буде мати вигляд як подано Фіг. 3.

Останнім кроком визначення еквівалентних функціональних блоків є порівняння матриць ймовірності наступності опкодів для програми до та після емуляції (4) та вибір мінімальної оцінки схожості.

$$R = \frac{1}{N^2} \left(\sum_{i,j=1}^{N-1} |a_{i,j} - b_{i,j}| \right)^2, \quad (4)$$

де $a_{i,j}$ - комірка матриці для функціонального блока F_p , $b_{i,j}$ - комірка матриці для функціонального блока F_s , N - загальна кількість опкодів для пар блоків.

Отримана оцінка для пар блоків дозволяє визначити еквівалентні функціональні блоки з більшою ймовірністю.

Формуємо вектор ознак схожості копій метаморфних вірусів. Визначивши еквівалентні функціональні блоки програми після емуляції, що відповідають функціональним блокам програми до емуляції, виконуємо порівняння їх за допомогою метрики Дамерау-Левенштейна. Це дозволяє отримати множину векторів ознак схожості для пар еквівалентних функціональних блоків, які можна представити у вигляді кортежа:

$$\bar{S} = \langle L, T, D, I, R, M \rangle, \quad (5)$$

де L - відстань Дамерау-Левенштейна для функціонального блока між програмами F_p та F_s ;

T - кількість необхідних операцій обміну опкодів для приведення блока програми;

D - кількість необхідних операцій видалення опкоду;

I - кількість необхідних операцій вставки опкоду;

R - кількість необхідних операцій заміни відповідних опкодів;

M - кількість співпадінь між опкодами в функціональних блоках програми F_p та F_s .

Для пошуку дистанції Дамерау-Левенштейна використано алгоритм поліноміальної складності Вагнера-Фішера, який дав змогу сформулювати найкоротший ланцюг перетворення, для приведення множини опкодів програми після емуляції у множину опкодів програми до емуляції.

Вибір дистанції Дамерау-Левенштейна як метрики для порівняння, зумовлено використанням в її основі основних операцій, що застосовуються метаморфними вірусами для видозмінення власної копії: вставки, видалення, заміни та перестановки опкодів. Слід зауважити, що в процесі розбиття програми метаморфного вірусу на еквівалентні функціональні блоки та їх порівняння не виконуються операція "очищення" метаморфного коду від обфускованих перетворень, а весь метаморфний код представлений в початковому вигляді [7, 8].

Формування висновку щодо інфікування системи метаморфним вірусом. Для формування висновку щодо інфікування системи метаморфним вірусом, отримані вектори ознак схожості підлягають класифікації шляхом залучення системи нечіткого логічного висновку.

Кожен невідомий об'єкт може бути віднесений до одного з трьох класів метаморфних вірусів або до класу довірених додатків.

Система нечіткого логічного висновку оперує вхідними та вихідними лінгвістичними змінними. Як вхідні лінгвістичні змінні приймемо: "ступінь схожості підозрілої програми з її копією за дистанцією Левенштейна" (L), "ступінь схожості підозрілої програми з її копією за кількістю операцій вставки" (I), "ступінь схожості підозрілої програми з її копією за кількістю операцій видалення" (D), "ступінь схожості підозрілої програми з її копією за кількістю операцій заміни" (R), "ступінь схожості підозрілої програми з її копією за кількістю операцій перестановки" (T) та "ступінь схожості підозрілої програми з її копією за кількістю операцій співпадіння" (M). Вихідною лінгвістичною змінною приймемо ступінь схожості з класом метаморфних вірусів. Для кожної лінгвістичної змінної задано терм-множину Low, Medium та High. Для визначення приналежності

метаморфного вірусу до одного із класів в системі задіяно 38 правил. Як функції приналежності для входів було вибрано трапецієвидну, для виходу - трикутну.

Таким чином, розроблений спосіб передбачає пошук відповідності між функціональними блоками в метаморфних копіях на основі статистичних метрик та складається з двох етапів. Перший етап визначає еквівалентні функціональні блоки на основі обчислення статистичної оцінки появи інструкцій у блоці. На другому етапі здійснюється уточнення еквівалентних блоків та вибір найбільш придатного блока, що буде використаний для формування векторів ознак схожості копій нових метаморфних вірусів. Спосіб здійснює класифікацію векторів ознак зі залученням нечіткої логіки.

Розроблений спосіб виявлення метаморфних комп'ютерних вірусів в комп'ютерних системах на основі використання статистичних метрик для визначення еквівалентних функціональних програмних блоків дозволяє зменшити кількість хибних спрацювань і підвищити достовірність виявлення, бо на відміну від відомих способів використовує уточнення вибору функціональних блоків за рахунок критерію вибору мінімальної оцінки схожості.

Таблица

Оригінальний код	Вставка команд сміття	Переміщення блоків	Заміна інструкцій
call 0h	call 0h	call 0h	call 0h
pop ebx	pop ebx	pop ebx	pop ebx
lea ecx, [ebx+42h]	lea ecx, [ebx+42h]	jmp S2	lea ecx, [ebx+42h]
push ecx	pop	S3: push eax	sub esp, 03h
push eax	xor ax, ax	push eax	sidt [esp-02h]
push eax	push ecx	jmp S4	sidt [esp-02h]
sidt [esp-05h]	push eax	add ebx, 1Ch	add [esp], 1Ch
pop ebx	inc eax	jmp S6	mov ebx, [esp]
add ebx, 1Ch	push eax	S2: lea ecx, [ebx+42h]	inc esp
cli	dec [esp-0h]	push ecx	cli
mov ebp, [ebx]	dec eax	jmp S3	mov ebp, [ebx]
	sidt [esp-02h]	S4: pop ebx	
	pop ebx	cli	
	add ebx, 1Ch	jmp S5	
	cli	S5: mov ebp, [ebx]	
	mov ebp, [ebx]		

Джерела інформації:

1. Podlovchenko, R.I., Kuzyurin, N.N., Shcherbina V.S., Zakharov V.A.: Using algebraic models of programs for detecting metamorphic malwares. Journal of Mathematical Sciences, Vol. 172 (5), pp. 740-750 (2011).

2. Подловченко Р.И., Захаров В.А. Полиномиальный по сложности алгоритм, распознающий коммутативную эквивалентность схем программ // Доклады РАН, 1998, т. 362, № 6, с. 744-747.

3. Anderson, B., Quist, D., Neil, J., Storlie C., Lane, T.: Graph-based malware detection using dynamic analysis. Journal in Computer Virology, 7, pp. 247-258 (2011).

4. Runwal, N., Low, R.M., Stamp, M.: Opcode Graph Similarity and Metamorphic Detection. Journal in Computer Virology, 8, pp. 37-52 (2012).

5. Nagaraju, A.: Metamorphic malware detection using base malware identification approach. Journal Security and Communication Networks, 7, pp. 1719-1733 (2014).

6. Kuriakose, J., Vinod, P.: Unknown Metamorphic Malware Detection: Modelling with Fewer Relevant Features and Robust Feature Selection Techniques, IAENG International Journal of Computer Science, Vol. 42(2), p. 139-151 (2015).

7. Pomorova, O., Savenko, O., Lysenko, S., Nicheporuk, A.: Metamorphic Viruses Detection Technique Based on the Modified Emulators. ICT in Education, Research and Industrial Applications, Integration, Harmonization and Knowledge Transfer, Vol. 1614, Kyiv, June 2016. - PP. 375-383 (2016).

8. Oksana Pomorova. A Technique for Detection of Bots Which Are Using Polymorphic Code / Oksana Pomorova, Oleg Savenko, Sergii Lysenko, Andrii Kryshchuk, and Andrii Nicheporuk // Computer Networks Communications in Computer and Information Science, Computer Networks 21th International Conference, CN 2014, Lwowek Slaski, Poland, June 23-27, 2014. Proceedings, vol. 431, p. 265-276, ISBN: 978-3-319-07940-0.

9. Мультіагентний спосіб локалізації бот-мереж у корпоративних комп'ютерних мережах [Текст]: пат. 108238 Україна: МПК G06F 21/55 (2013.01) / О.В. Поморова, О.С. Савенко, А.Ф. Кришук, С.М. Лисенко, К.Ю. Бобровнікова, А.О. Нічепорук; заявник та власник Хмельницький національний університет. - № u201600127; заявл. 04.01.16; опубл. 11.07.16, Бюл. № 13.

ФОРМУЛА КОРИСНОЇ МОДЕЛІ

Спосіб виявлення метаморфних комп'ютерних вірусів в комп'ютерних системах на основі використання статистичних метрик для визначення еквівалентних функціональних програмних блоків, який **відрізняється** тим, що використовує уточнення вибору функціональних блоків за рахунок критерію вибору мінімальної оцінки схожості, процес вибору функціональних блоків для порівняння складається з двох етапів, де на першому етапі визначають еквівалентні функціональні блоки на основі обчислення статистичної оцінки появи інструкцій у блоці, а на другому етапі здійснюють уточнення еквівалентних блоків та вибір найбільш придатного блока, який і буде використаний для формування оцінки схожості між програмою до емуляції F_p та програмою після емуляції F_s , при цьому основні кроки способу вимагають визначення еквівалентних функціональних блоків з використанням формули (1):

$$s_{FB} = \frac{n_i}{\sum_{i=0}^k n_i} * \log \left(\frac{N+1,0}{n_j} \right),$$

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

$k = \overline{1, k_a}$ - кількість операційних кодів (опкодів) у функціональному блоці, де

k_a - загальна кількість асемблерних інструкцій;

N - загальна кількість функціональних блоків, причому $N_{F_p} \neq N_{F_s}$;

n_j - кількість функціональних блоків, в якій присутній i -й опкод і результатом виконання етапу

обчислення статистичної оцінки присутності опкоду у ФБ для програми до емуляції F_p та

програми після емуляції F_s є матриці $M(FB^{F_p})$ та $M(FB^{F_s})$, рядки яких визначають функціональні

блоки програми, а стовпці - опкоди, що присутні в функціональному блоці, причому кожна комірка матриці визначає оцінку появи i -го опкода в j -му функціональному блоці, а визначення еквівалентних функціональних блоків, на наступному кроці обчислюється оцінка схожості двох

функціональних блоків з програми F_p та програми F_s за формулою Евклідової метрики і якщо

значення оцінки схожості двох функціональних блоків менше визначеного порогового значення δ , тобто $E(FB_i^{F_p}, FB_j^{F_s}) \leq \delta$, тоді виконують повторне обчислення оцінки схожості для

функціонального блока з програми $FB_i^{F_p}$ та наступного блока, що іде за блоком $FB_j^{F_s}$, тобто

$E(FB_i^{F_p}, FB_j^{F_s} + FB_{j+1}^{F_s})$ і ці дії повторюються, поки значення оцінки схожості стане меншим або

рівним пороговому значенню, а далі здійснюють перехід до другого кроку способу, де

- здійснюють уточнення визначення еквівалентних функціональних блоків і після чого порівнюють матриці еквівалентних функціональних блоків і після чого порівнюють матриці імовірності наступності опкодів для програми до та після емуляції та здійснюють вибір мінімальної оцінки схожості і формують вектор ознак схожості копій метаморфних вірусів та висновок щодо інфікування комп'ютерної системи метаморфним вірусом.
- 5

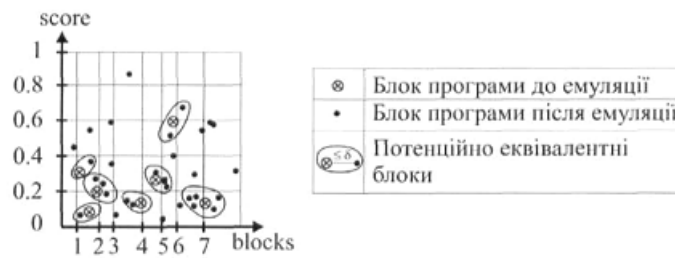
$$M(FB^{Fp}) = \begin{matrix} & i_1 & i_2 & \dots & i_k \\ \begin{matrix} FB_1^{Fp} \\ FB_2^{Fp} \\ \dots \\ FB_m^{Fp} \end{matrix} & \begin{matrix} s_{11} & s_{12} & \dots & s_{1k} \\ s_{21} & s_{22} & \dots & s_{2k} \\ \dots & \dots & \dots & \dots \\ s_{m1} & s_{m2} & \dots & s_{mk} \end{matrix} \end{matrix}$$

a)

$$M(FB^{Fs}) = \begin{matrix} & i_1 & i_2 & \dots & i_g \\ \begin{matrix} FB_1^{Fs} \\ FB_2^{Fs} \\ \dots \\ FB_n^{Fs} \end{matrix} & \begin{matrix} s_{11} & s_{12} & \dots & s_{1g} \\ s_{21} & s_{22} & \dots & s_{2g} \\ \dots & \dots & \dots & \dots \\ s_{n1} & s_{n2} & \dots & s_{ng} \end{matrix} \end{matrix}$$

б)

Фіг.1



Фіг.2

	mov	push	lea	pop	Call
mov	0	0	0	1	1
push	2	2	0	0	0
lea	0	1	0	0	0
pop	0	0	1	0	0
call	0	1	0	0	0



	mov	push	lea	pop	call
mov	0	0	0	1/2	1/2
push	2/4	2/4	0	0	0
lea	0	1	0	0	0
pop	0	0	1	0	0
call	0	1	0	0	0

Fig.3