УДК 004.67

О. С. САВЕНКО, А. О. НІЧЕПОРУК, С. М. ЛИСЕНКО

Хмельницький національний університет, Україна

МЕТОД ВИЯВЛЕННЯ ПОЛІМОРФНИХ ВІРУСІВ НА ОСНОВІ МОДИФІКОВАНИХ ЕМУЛЯТОРІВ

У роботі запропоновано метод виявлення поліморфних вірусів в корпоративній мережі на основі використання модифікованих емуляторів, які розміщенні на кожному хості у мережі, що дозволяє сформувати змінне віртуальне середовище для виконання підозрілої програми з метою отримання її зміненої копії. Формування оцінки схожості між підозрілою програмою та її копією після емуляції виконання здійснюється на основі модифікованої дистанції Дамерау-Левенштейна. Висновок про загрозу інфікування поліморфним вірусом здійснюється за допомогою системи нечіткого логічного висновку.

Ключові слова: *поліморфний вірус, модифіковані емулятори, розшифровувач,блок мутації.*

Вступ

Розповсюдження мережі Інтернет створює сприятливі умови існування шкідливих програм. Одними з найбільш розповсюджених шкідливих програм є комп'ютерні віруси, зокрема поліморфні віруси. За даними компанії DrWeb [1] у 2014 році файловий поліморфний вірус win32.Sector, що поширюється через P2P мережі та здійснює інфікування виконуваних файлів, а також завантажує інші шкідливі програми, інфікував близько 1,2 мільйони комп'ютерів. Інший поліморфний вірус win32.Virut у 2013 році, здійснюючи інфікування виконуваних файлів через змінні носії інформації, показав динаміку поширення, що склала більше одного відсотка у всьому світі. Складність виявлення такого типу шкідливого програмного забезпечення зумовлене використанням ними перетворень, що дозволяють створити копії, які при кожному новому інфікуванні будуть різні. Тому, актуальною є задача розробки метода виявлення нових файлових поліморфних вірусів та копій вже існуючих, що здійснюють інфікування .exe файлів.

1. Попередні дослідження

Для виявлення загрози інфікування більшість антивірусних засобів використовують сигнатурний аналіз [2]. Основна мета такого аналізу полягає у зіставлені та пошуку схожості визначеної сигнатури із набором команд вірусної програми [3]. У випадку поліморфних вірусів, пошук за сигнатурою є неефективним, оскільки копії, що створюються поліморфними вірусами, є різними при кожному новому інфікуванні.

Тому, для виявлення поліморфних вірусів, у ряді робіт [3-5] пропонується пошук сталих ознак. Цими ознаками можуть бути: граф потоку управління програми, послідовність викликів API функцій, структурна інформація виконуваного файлу.

У роботах [3,5] запропоновано методи виявлення поліморфних вірусів на основі використанням графу потоку управління програми. Пошук схожості між двома графами здійснюється на основі матриці, що отримана за допомогою алгоритму розфарбування графів [3]. Підхід, що передбачає формування вектору ознак з підгафів графу потоку управління та визначення схожості вірусних програм на основі Евклідової метрики відстаней представлений у роботі [5]. Проте, дані методи не враховують виклики системних функцій для ідентифікації поліморфних вірусів.

Для формування оцінки схожості поведінки поліморфних вірусів існують методи засновані на відслідковуванні API викликів. Ідея даних методів полягає у порівнянні отриманих API викликів з деяким шаблонним значенням або евристичний пошук підозрілих викликів у всій множині API функцій, що генеруються програмою [4]. Проте, у випадку використання вірусною програмою псевдо викликів API функцій (Fake API) дані методи нездатні ефективно здійснювати виявлення.

2. Метод виявлення поліморфних вірусів на основі модифікованих емуляторів

Складність виявлення поліморфних вірусів зумовлена використанням в їхній структурі, окрім розшифрувальника (decryption routine) та тіла вірусу (payload), блоку мутації (mutation engine).

Для отримання керування над інфікованою програмою, здійснюється заміна оригінальної точки входу у програму на точку входу, з якої розпочинається виконання вірусного коду. Оскільки вірусне тіло та блок мутації зашифровані, здійснюється завантаження у пам'ять коду розшифрувальника. Основним завданням розшифрувальника є пошук в інфікованому файлі основного тіла та блоку мутації та їх розшифрування, наприклад за допомогою побітної операції xor. Після завантаження вірусного тіла у пам'ять, здійснюється виконання функціонального навантаження вірусу та створення нової копії, що здійснює інфікування наступної програми. На рис. 1 представлено схему функціонування поліморфного вірусу.

Враховуючи особливості функціонування поліморфних вірусів було розроблено метод, який використовує емуляцію виконання підозрілого файлу на кожному хості в мережі. З метою пошуку схожості поліморфних розшифровувачів метод передбачає залучення модифікованих емуляторів.

Хости представляють собою мережні станції для обробки інформації, що поєднанні у локальну мережу. Основними функціями хостів є здійснення одноразової емуляції виконання невідомої програми та відправлення результатів на серверну частину.

Серверна частина слугує для опрацювання результатів виконання процесу емуляції, отриманих з хостів. З метою ускладнення процесу реверс інжинірингу та захисту даних від копірайту на рівні алгоритмів реалізації процес обфускації часто використовується у довірених додатках розробниками ПЗ. Тому, основним завданням серверної частини, є класифікація отриманих з хостів векторів ознак порівняння копій метаморфних вірусів.

Кожна програма, що надходить з мережі Internet на хост, перевіряється аналізатором підозрілості програми та відправляється на решту хостів в мережі. У випадку виявлення підозрілої поведінки, дана програма надходить на блок емуляції виконання для отримання зміненої версії цього ж файлу.

На стадії порівняння відбувається зіставлення програми до емуляції з цією ж програмою після виконання емуляції. Процес порівняння полягає в розбитті розшифровувачів двох версій програми на функціональні блоки та поблочне їх порівняння їх за допомогою метрики Дамерау-Левенштейна.

Отримані вектори з кожного хоста відправляються на серверну частину, де відбувається формування висновку про приналежність підозрілої програми до одного із рівнів поліморфних вірусів. На рис. 2 представлено узагальнену схему виявлення та класифікації поліморфних вірусів у мережі.

Приймемо P={suspicious, non-suspiciuous} – невідома програма, F_P – початковий зразок підозрілого коду, F_S – змінений зразок підозрілого коду коду, $\overline{S}$ – вектор ознак схожості F_P та F_S.

Кожна програма, що надходить в систему маркується як: P={suspicious, non-suspiciuous}.

Подамо вектор ознак, що визначає належність програми до одного з двох класів наступним чином:

$$\overline{U} = (M, Q, J, Y, L, N, H), \qquad (1)$$

де M – спроба програми отримати права адміністратора системи;

Q – спроба відкриття або закриття системного порту;

J – спроба видалення файлу;

Y – створення файлу або процесу;

L – перехоплення даних, що вводяться з клавіатури;

N – розсилка повідомлень в мережу;

H – створення або запис в системний реєстр.

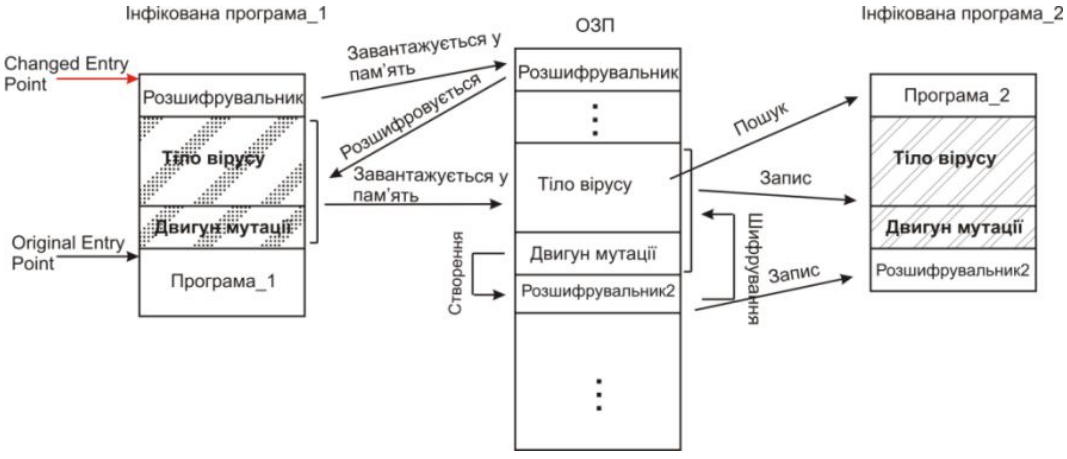

Рис. 1. Схема функціонування поліморфного вірусу

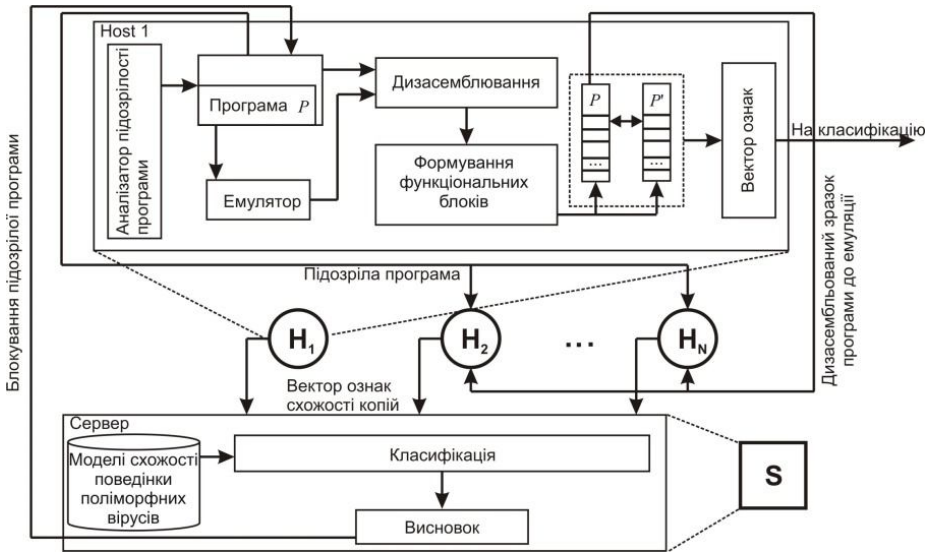

Рис. 2. Узагальнена схема виявлення та класифікації поліморфних вірусів у корпоративній мережі

Програму вважатимемо за підозрілу, якщо:

$$P = \text{suspicious}, \text{ if } \forall u \in \overline{U}, (u_i = 1 \wedge u_j = 1).$$

Для отримання зміненого зразку коду F_S, здійснюється емуляція виконання програми P.

Використання однотипного емулятора на всіх хостах мережі не дозволить з високим ступенем достовірності здійснити виявлення поліморфних достовірності здійснити виявлення поліморфних вірусів, оскільки для уникнення емуляції, деякі типи складних поліморфних вірусів здатні розпізнавати середовище у якому вони виконуються. Тому, на кожному хості створюються модифіковані емулятори.

Структура емулятора наступна: віртуальний процесор, визначає набори інструкцій, що доступні для роботи та включає в себе набори віртуальних регістрів; оперативна пам'ять та віртуальний стек; віртуальний мережний контролер; тип ОС та модуль евристики.

Оскільки виклик підпрограми розшифровувача для здійснення розшифрування відбувається перед запуском тіла вірусу, а шифрування – після завершення виконання вірусного тіла [1], то дана особливість дозволяє визначити розташування поліморфного розшифровувача та здійснити оцінку схожості копій одного і того ж вірусу.

Для формування поліморфних розшифровувачів використовуються техніки вставки, видалення та переміщення власних інструкцій, тому для пошуку схожості між функціональними блоками (ФБ) двох зразків коду F_P та F_S використовується дистанція Дамерау – Левенштейна. Для її пошуку використаємо алгоритм поліноміальної складності Вагнера-Фішера.

Розіб'ємо програму P на ФБ, тобто $P = \{B_1, B_2, ..., B_l\}$.

Програму до емуляції позначимо F_P, а програму після емуляції виконання – F_S.

Нехай ФБ B, що складається з множини опкодів, довжиною $|B| = m$ записується як $p_1, p_2, ..., p_m$, де p_i представляє і-й опкод p. Підмножина опкодів $x_i, x_{i+1}, ..., x_j$, ФБ B буде позначатись $B(i, j)$.

Вага перетворення опкода a в опкод b позначимо через $w(a,b)$. Таким чином, $w(a,b)$ – вага заміни одного опкоду на другий опкод, коли $a \neq b$, $w(b,a)$ – вага операції транспозиції, $w(a,\varepsilon)$ – вага видалення, а $w(\varepsilon,b)$ – вага вставки b. Нехай B_g та B_h – два ФБ, що складаються з послідовності опкодів (довжиною N та M відповідно), причому B_g ФБ програми F_P, позначимо $B_g^{F_P}$, а B_h – ФБ тієї ж програми після емуляції виконання F_S, позначимо $B_h^{F_S}$. Тоді відстань Дамерау – Левенштейна $dL(B_g^{F_P}, B_h^{F_S})$ визначимо:

$$dL(B_g^{F_P}, B_h^{F_S}) = OPT(N, M),$$

$$OPT = \begin{cases} 0, & i = 0,\ j = 0; \\ i, & j = 0,\ i > 0; \\ j & i = 0,\ j > 0; \\ \min \begin{cases} OPT(i, j-1) + w(a,\varepsilon), \\ OPT(i, -1j) + w(\varepsilon, b), \\ OPT(i, -1, j-1) + w(a,b), \\ OPT(i-2, j-2) + w(b,a), \end{cases} & j > 0,\ i > 0. \end{cases} \quad (2)$$

Після отримання відстані Дамерау-Левенштейна для двох блоків B_g та B_h, формується зважене усереднене значення параметру для всіх блоків коду. Для отримання зваженої усередненої оцінки параметрів використаємо показник центру розподілу зважене середнє арифметичне.

$$dL = \left\lfloor \frac{\sum_{i=1}^{n} dL_i * f_l}{\sum_{i=1}^{n} f_l} \right\rfloor \quad (3)$$

де dL_i – відстань Левенштейна для ФБ B_i, f_l – кількість ФБ з значенням dL_i

Для решти ознак (кількість операцій співпадіння, вставки, видалення, заміни, транспозиції) унормування відбувається аналогічно.

Таким чином, вектор ознак схожості копій метаморфних вірусів на основі метрики Дамерау-Левенштейна подамо наступним чином :

$$\bar{S} = \langle dL, T, D, I, R, M \rangle \quad (4)$$

де dL – відстань Дамерау – Левенштейна для функціонального блоку між програмами F_p та F_s ;

T – кількість необхідних операцій обміну опкодів для приведення блоку програму F_p у F_s ($F_p = F_s$);

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 функціональному блоці програми F_p та F_s .

Для формування висновку, отримані вектори ознак схожості надходять на серверну частину для їх класифікації.

Для здійснення класифікації розшифровувачів у поліморфному вірусі розроблений метод використовує систему нечіткого логічного висновку типу Сугено з шістьма входами та одним виходом. Кожен вхід має три гаусові функції приналежності, вихід – лінійну функцію належності.

Для проведення експериментів використано систему нейрон-нечіткого висновку (ANFIS). Для навчання моделі використано гібридний алгоритм.

3. Експерименти

Для здійснення експериментів множину поліморфних вірусів розділено на 4 класи, що відрізняються механізмами розшифрування основного тіла вірусу. П'ятий клас визначає множину довірених додатків. Ступені приналежності до кожного класу наведено у табл. 1.

Таблиця 1

Ступені приналежності об'єкту до одного із класів

Клас	Техніки обфускації	Ступінь приналежності
1	Декілька сталих розшифрувальника	0,21-0,4
2	Перестановка блоків	0,41-0,6
3	Генерація команд-сміття	0,61-0,8
4	Комбінована техніка	0,81-1
5	Довірені додатки	0-0,2

Для отримання різного середовища виконання шкідливого коду, у модифікованих мережних емуляторах змінювались наступні параметри: час виконання інструкцій, адреса початку емуляції, розмір віртуальної пам'яті та набори інструкцій.

В якості тестових даних, з сайту http://vxheaven.org/ вибрано такі зразки поліморфних вірусів: Tuareg.h, Polipos.a, Alman.b та Kido.a.

Результат нечіткого логічного висновку наведено на рис. 3.

Таблиця 2

Результат нечіткого логічного висновку

№ хоста	НЛВ	№ хоста	НЛВ	№ хоста	НЛВ
1	0,62	8	0,65	15	0,84
2	0,71	9	0,66	16	0,62
3	0,63	10	0,69	17	0,68
4	-	11	0,78	18	0,79
5	0,56	12	0,73	19	0,74
6	0,78	13	0,69	20	0,64
7	0,71	14	0,65		

У табл. 2 наведено ступінь приналежності невідомого об'єкту до одного з класів. Значення від 0,21 до 1 свідчить про інфікування системи поліморфним вірусом. Значення від 0 до 0,2 визначає приналежність до класу довірених додатків.

Висновок

Розроблений метод виявлення поліморфних вірусів на основі модифікованих емуляторів. Використання змінених параметрів емуляторів дозволяє здійснювати виявлення поліморфних вірусів, що використовують антиемуляційні техніки. Висновок про інфікування системи поліморфним вірусом здійснюється на основі системи нечіткого логічного висновку.

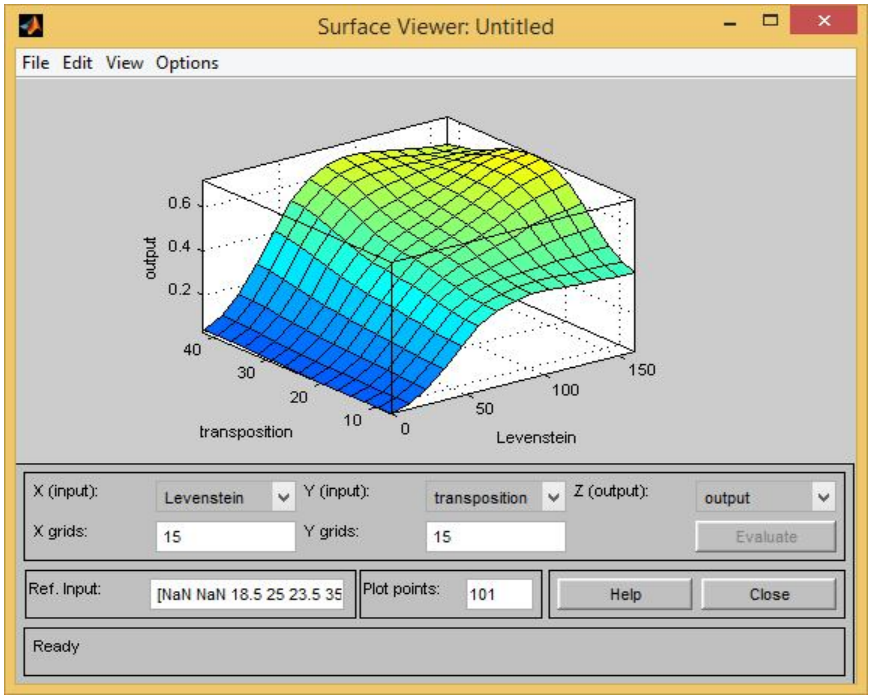

Рис. 3. Результати нечіткого логічного висновку

Література

1. Файловый вирус Win32.Sector заразил более миллиона компьютеров [Електронний ресурс] / Dr. Web Форум. – Режим доступу: http://news.drweb.ru/show/?i=5778 . – 21.05.2014.

2. Polymorphic Worm Detection Using Structural Information of Executables [Text] / C. Kruegel, E. Kirda, D. Mutz, W. Robertson, G. Vigna // Recent Advances in Intrusion Detection, ser. Lecture Notes in Computer Science, A. Valdes, D. Zamborini. – Springer, Heidelberg Dordrecht. – Seattle, 2005. – C. 207-226.

3. Kaspersky Lab core detection technologies: Comprehensive protection from threats of today and tomorrow [Text] / Kaspersky Lab White Paper. – 2009. – C. 2-22.

4. Cesare, S. Malware Variant Detection Using Similarity Search over Sets of Control Flow Graphs [Text] / S. Cesare, Y. Xiang // Trust, Security and Privacy in Computing and Communications– Changsha, 2011. – C. 181-189.

5. Malware detection based on mining API calls [Text] / A. Sami, B. Yadegari, H. Rahimi, N. Peiravian, S. Hashemi, A. Hamze // in Proc. of the 25th Annual ACM Symposium on Applied Computing (SAC '10). – Sierre, Switzerland, 2010. – C. 1020–1025.

References

1. Faylovyy virus Win32.Sector zarazil bolee milliona komp'yuterov. Available at: http://news.drweb.ru/show/?i=5778 (accessed 21.05.2014).

2. Kruegel, C., Kirda E., Mutz, D., Robertson, W., Vigna, G. Polymorphic Worm Detection Using Structural Information of Executables. *Recent Advances in Intrusion Detection*, Springer, Heidelberg, Dordrecht, Seattle, 2005, no. 1, pp. 207-226.

3. Kaspersky Lab core detection technologies: Comprehensive protection from threats of today and tomorrow, *Kaspersky Lab White Paper*, 2009, pp. 2-22.

4. Cesare, S., Xiang, Y. Malware Variant Detection Using Similarity Search over Sets of Control Flow Graphs, *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference*, Changsha, 2011, pp.181-189.

5. Sami, A., Yadegari, B., Rahimi, H., Peiravian, N., Hashemi, S., Hamze A. Malware detection based on mining API calls. *In Proc. of the 25th Annual ACM Symposium on Applied Computing (SAC '10)*, Sierre, Switzerland, 2010, pp. 1020–1025.

Надійшла до редакції 11.03.2016, розглянута на редколегії 14.04.2016

МЕТОД ОБНАРУЖЕНИЯ ПОЛИМОРФНЫХ ВИРУСОВ НА ОСНОВЕ МОДИФИЦИРОВАННЫХ ЭМУЛЯТОРОВ

О. С. Савенко, А. А. Ничепорук, С. Н. Лысенко

В работе предложен метод обнаружения полиморфных вирусов в корпоративной сети на основе использования модифицированных эмуляторов, которые размещены на каждом хосте в сети, что позволяет сформировать переменную виртуальную среду для выполнения подозрительной программы с целью получения ее измененной копии. Формирование оценки сходства между подозрительной программе и ее копией после эмуляции исполнения осуществляется на основе модифицированной дистанции Дамерау-Левенштейна. Вывод об угрозе инфицирования полиморфным вирусом осуществляется с помощью системы нечеткого логического вывода.

Ключевые слова: полиморфный вирус, модифицированные эмуляторы, расшифровщик, блок мутации.

METHODS FOR DETECTION OF POLYMORPHIC VIRUSES BASED ON MODIFIED EMULATORS

O. S. Savenko, A. O. Nicheporuk, S. M. Lysenko

The paper presents a method for detecting polymorphic viruses in corporate network using modified emulators that are placed on each host in the network, allowing create a variable virtual environment to perform suspicious programs to get its modified copy. Formation evaluation suspicious similarities between the program and its copy after emulating execution is based on the modified Damerau–Levenshtein distance. The conclusion of the threat of infection polymorphic virus by using fuzzy inference.

Key words: polymorphic virus, network emulators, decryption routine, mutation engine.

Савенко Олег Станіславович – канд. техн. наук, доц., Хмельницький національний університет, Хмельницький, Україна, e-mail: savenko_oleg_st@ukr.net.

Нічепорук Андрій Олександрович – аспірант, Хмельницький національний університет, Україна, e-mail: andrey.nicheporuk@gmail.com.

Лисенко Сергій Миколайович – канд. техн. наук, доц., Хмельницький національний університет, Хмельницький, Україна, e-mail: sirogyk@ukr.net.

Savenko Oleh Stanislavovych – PhD, Associate Professor, Khmelnytskyi National University, Ukraine, e-mail: savenko_oleg_st@ukr.net.

Nicheporuk Andriy Oleksandrovych – PhD Student, Khmelnytskyi National University, Ukraine, e-mail: andrey.nicheporuk@gmail.com.

Lysenko Serhiy Mykolayovych – PhD, Associate Professor, Khmelnytskyi National University, Ukraine, e-mail: sirogyk@ukr.net.