https://doi.org/10.31891/2219-9365-2023-76-18
УДК 621.391 160164

НІЧЕПОРУК Андрій
Хмельницький національний університет
https://orcid.org/0000-0002-7230-9475
e-mail: andrey.nicheporuk@gmail.com
БАРМАК Олександр
Хмельницький національний університет
https://orcid.org/0000-0003-0739-9678
e-mail: alexander.barmak@gmail.com
МАНЗЮК Едуард
Хмельницький національний університет
https://orcid.org/0000-0002-7310-2126
e-mail: edemasu@outlook.com
НІЧЕПОРУК Анастасія
Хмельницький національний університет
https://orcid.org/0000-0001-5366-5792
e-mail: eldess06@gmail.com
ДАНЧУК Сергій
Хмельницький національний університет
https://orcid.org/0009-0003-4510-0363
e-mail: sergey.danchuk.p@gmail.com

ІНФОРМАЦІЙНА СИСТЕМА ВИЯВЛЕННЯ ТА ІДЕНТИФІКАЦІЇ МЕТАМОРФНИХ ВІРУСІВ У ЛОКАЛЬНІЙ КОМП'ЮТЕРНІЙ МЕРЕЖІ

В роботі запропоновано інформаційну систему виявлення та ідентифікації метаморфних вірусів у локальній комп'ютерній мережі. В основу представленої інформаційної системи закладено два методи, а саме метод виявлення метаморфних вірусів на основі аналізу поведінки програми з використанням модифікованих емуляторів в локальній мережі та метод ідентифікації на основі пошуку та порівняння еквівалентних функціональних блоків між програмами. В основу обох методів закладено концепцію порівняння копій метаморфних вірусів, результатом якого є визначення набору ознак, що використовується для виявлення метаморфних вірусів. Особливістю запропонованої інформаційної системи є те, що у випадку недостатнього прояву шкідливої поведінки та підвищення рівня достовірності для виявлення метаморфного зловмисного програмного забезпечення залучаються інші хости мережі.

Ключові слова: метаморфне зловмисне програмне забезпечення, обфускація, модифіковані емулятори

NICHEPORUK Andrii, BARMAK Oleksandr, MANZIUK Eduard,
NICHEPORUK Anastasiia, DANCHUK Serhii
Khmelnytskyi National University

INFORMATION SYSTEM FOR METAMORPHIC MALWARE DETECTION AND IDENTIFICATION IN LOCAL NETWORK

The paper proposes an information system for detection and identification of metamorphic viruses in a local computer network. The presented information system is based on two methods, namely the method of detecting metamorphic viruses based on the analysis of program behavior using modified emulators in the local network and the identification method based on the search and comparison of equivalent functional blocks between programs. Both methods are based on the concept of comparing copies of metamorphic viruses, the result of which is the definition of a set of features used to detect metamorphic viruses.

The functioning of the system involves sending a suspicious program in a protected container to other hosts in the network in order to run it in modified emulators and manifest malicious activity. In order to create a variable execution environment for modified emulators, a number of parameters and settings that change on each computer system in a local computer network are proposed. A system of fuzzy logical inference is used to form a conclusion about the similarity of a suspicious program to a metamorphic virus. Thus, a feature of the proposed information system is that in the case of insufficient manifestation of malicious behavior and increasing the level of reliability to detect metamorphic malware, other network hosts are involved. A number of experiments were conducted to assess the accuracy of detection and the level of false positives in the detection of metamorphic viruses. Metamorphic generators NGVCK, VCL32, G2 and MetaPHOR were used to create a test sample of metamorphic malware. According to the results of the experiments, the highest level of detection accuracy was recorded for metamorphic viruses created using the G2 generator, which is 0.97. The lowest level of detection accuracy was for the NGVCK-generated metamorphic malware, which is 0.8671. At the same time, it should be noted that the lowest level of errors of the 1st kind was recorded for the metamorphic malware VCL32 (0.0587), however, for the rest of the metamorphic samples, this indicator did not exceed the value of 0.0641.

Keywords: metamorphic malware, obfuscation, modified emulators.

ISSN 2219-9365

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

З швидким розвитком та поширенням інформаційних систем у всіх сферах сучасного життя різко постає проблема протидії кіберзлочинності. Однією із таких кіберзагроз є поширення метаморфних вірусів. Основною особливістю таких вірусів є їх здатність постійно змінювати власну структуру та поведінку, що робить їх надзвичайно складними для виявлення традиційними сигнатурними методами. Ще однією перешкодою до виявлення метаморфних вірусів є застосування ними антивідлагоджувальних та антиемуляційних технологій, що ускладнює тим самим виявлення із застосуванням емуляції виконання – основним методом виявлення такого типу зловмисного програмного забезпечення (ЗПЗ). Тому розробка нових методів виявлення метаморфних вірусів є актуальною задачею.

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 [1-5]. Ці ознаки можуть включати як статичні так і динамічні атрибути, такі як опкоди (кодові інструкції), структуру графу потоку керування, 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 [6]. Проведений огляд відомих методів показав, що сучасні методи характеризуються досить високою достовірністю виявлення, проте основним недоліком розглянутих методів та систем є їх схильність до хибнопозитивних та хибнонегативних результатів.

Інформаційна система виявлення та ідентифікації метаморфних вірусів у локальній комп'ютерній мережі

Розглянемо процес виявлення метаморфних вірусів у локальній комп'ютерній мережі. Використання мережі продиктоване наявністю, окрім обфускаційних технік, антиемуляційних засобів, що перешкоджають здійсненню процесу емуляції виконання – одного із головних методів виявлення метаморфних вірусів [7], що, в свою чергу, призводить до низької ефективності виявлення. Тому, здійснення виявлення метаморфних вірусів, які застосовують антиемуляційні технології, засобами однієї комп'ютерної системи є не завжди можливим, в зв'язку з чим розглядається саме локальна комп'ютерна мережа.

Процес виявлення метаморфних вірусів реалізовано у формі інформаційної системи (ІС), яка складається двох методів. У запропонованій ІС функціонування методу виявлення метаморфних вірусів на основі аналізу поведінки програми з використанням модифікованих емуляторів в локальній мережі включає метод ідентифікації на основі пошуку та порівняння еквівалентних функціональних блоків між програмами [8, 9]. В основу обох методів закладено концепцію порівняння копій метаморфних вірусів, результатом якого є визначення набору ознак, що використовується для виявлення метаморфних вірусів. Структурну схему запропонованої ІС наведено на рис. 1.

В результаті функціонування методу ідентифікації на основі пошуку та порівняння еквівалентних функціональних блоків між програмами отримується набір характеристичних ознак для ідентифікації метаморфних вірусів. Цими ознаками є кількісні показники, що визначають схожість зразків метаморфних вірусів між собою за дистанцією Дамерау-Левенштейна, кількістю операцій вставки, видалення, перестановки та співпадіння операційних кодів, а також логічною ознакою – поведінкою підозрілої програми. Вихідними даним для отримання кількісних ознак є дизасембльовані лістинги операційних кодів: підозрілої програми та її зміненої версії, що сформована в захищеному віртуальному середовищі модифікованого емулятора. Формування логічної ознаки (поведінки) здійснюється на основі опрацювання послідовності API викликів функцій, що здійснює програма в процесі власного виконання. Для отримання кількісних ознак дизасембльовані лістинги операційних кодів розбиваються на функціональні блоки, з подальшим визначення еквівалентних функціональних блоків. Отримані характеристичні ознаки покладені в основу вектора ознак схожості зразка коду до метаморфного вірусу.

Функціонування методу виявлення метаморфних вірусів в локальній мережі з використанням модифікованих емуляторів передбачає розсилання підозрілої програми в захищеному контейнері на інші хости в мережі з метою її запуску в модифікованих емуляторах та прояву зловмисної активності. З метою формування змінного середовища виконання для модифікованих емуляторів запропоновано ряд параметрів та налаштувань, що змінюються на кожній комп'ютерній системі в локальній комп'ютерній мережі. Для формування висновку про схожість підозрілої програми до метаморфного вірусу використовується система нечіткого логічного висновку.

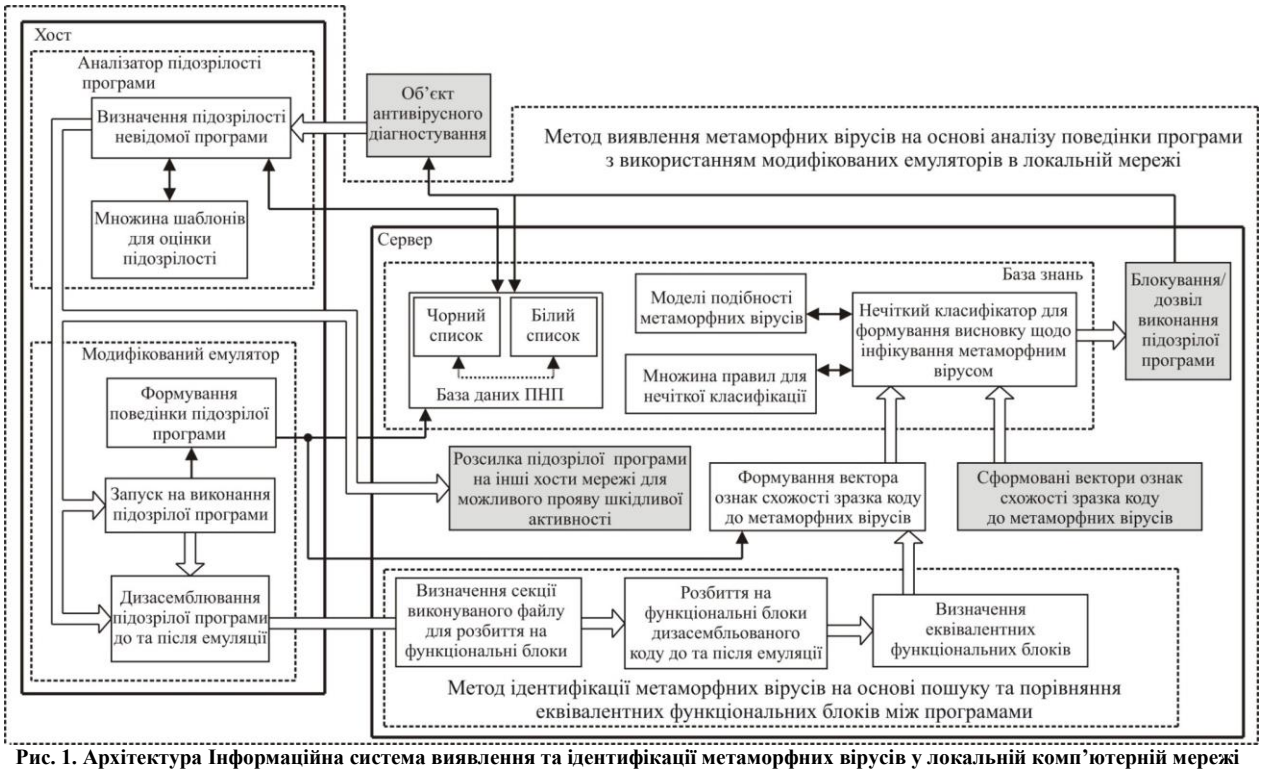

Рис. 1. Архітектура Інформаційна система виявлення та ідентифікації метаморфних вірусів у локальній комп'ютерній мережі

Розглянемо основні кроки функціонування запропонованої ІС виявлення метаморфних вірусів:

1. Перевірка підозрілості кожної нової програми на хості засобами *аналізатора підозрілості програми*. Перевірка здійснюється на основі евристичних правил, в основу яких покладено API виклики, що здійснює програма. Якщо в процесі перевірки ця програма буде визначена аналізатором підозрілості як *suspicious,* то здійснюється запит до сервера на предмет наявності поведінкової сигнатури для даної програми; на основі наявної на сервері бази потенційно небезпечних поведінок (ПНП), здійснюється пошук відповідної поведінки для підозрілої програми. Якщо відповідна поведінка присутня в чорному списку, тоді підозріла програма блокується; у разі наявності відповідної поведінки в білому списку – підозріла програма продовжує власне виконання; відсутність підозрілої поведінки в базі ПНП вимагає подальшого аналізу підозрілої програми.

2. Запуск на виконання програми, визначеної як підозріла, в середовищі модифікованого емулятора, який присутній на кожному хості. Виконання дизасемблювання підозрілої програми та отримання зразка коду F_p; здійснення емуляції виконання підозрілої програми та формування зміненого зразка коду F_s, його дизасемблювання, на основі відстеження API викликів формування поведінки підозрілої програми. Відправлення на сервер для формування висновку щодо присутності метаморного вірусу на хості зразків коду до та після емуляції (лістинги опкодів), підозрілої програми та її поведінки (лістинг API викликів) .

3. Опрацювання сервером отриманих результатів з хоста: розбиття отриманих з хоста зразків коду до та після емуляції на функціональні блоки, визначення еквівалентних функціональних блоків для зразків коду F_p та F_s; на основі попарного порівняння еквівалентних функціональних блоків програми до та після емуляції та формування векторів ознак схожості копій метаморфних вірусів для пар еквівалентних функціональних блоків; формування результату про ступінь подібності підозрілої програми до метаморфного вірусу на основі аналізу обфускації коду та поведінки з використанням нечіткого класифікатора. Якщо ступінь подібності до метаморфного вірусу має значення *High* то здійснюється блокування підозрілої програми на хості та додавання підозрілої поведінки до чорного списку бази ПНП. Якщо ступінь подібності до метаморфного вірусу підозрілої програми отримав значення *Low* або *Medium,* то здійснюється розсилання підозрілої програми в захищеному контейнері на інші хости в мережі з метою їх запуску в модифікованих емуляторах.

4. Збір сервером інформації з хостів щодо поведінки, попередньо розісланої підозрілої програми: зразків коду до та після емуляції та поведінки. Здійснення нечітким класифікатором висновку щодо схожості підозрілої програми на метаморфних вірус. Якщо бодай на одному з хостів рівень схожості

підозрілої програми на метаморфний вірус *High*, то здійснюється блокування підозрілої програми. Якщо рівень схожості – *Low* або *Medium*, то надається дозвіл на виконання для цієї програми.

Експериментальні дослідження оцінки достовірності виявлення метаморфних вірусів запропонованою ІС

Для визначення ефективності запропонованого методу було проведено ряд експериментів. Для цього було залучено університетську мережу. Вона складається з 4 робочих станцій. На кожній станції було встановлено модифіковані емулятори на основі Qemu [10]. В якості дизасемблера в модифікованому емуляторі було використано IDA Pro. З метою отримання змінених версій метаморфних вірусів було використано метаморфні генератори чотирьох типів: NGVCK, VCL32, G2 та MetaPHOR. Всі метаморфні версії, що створювались за допомогою зазначених генераторів були скомпільовані з опціями anti-debugging та anti-emulation. Експерименти передбачали визначення рівня достовірності виявлення метаморфних вірусів та рівня помилок першого типу (False Positives). З цією метою було згенеровано по 50 копій метаморфних вірусів кожного типу (200 зразків). Реалізація системи нечіткого логічного висновку передбачала залучення системи, представленій у [11]. Рівень достовірності виявлення було визначено наступним чином:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

де, *TP* – кількість вірно виявлених метаморфних вірусів, *FN* – кількість хибно класифікованих метаморфних вірусів, *TN* – кількість вірно ідентифікованих корисних програм, *FP* – є кількість корисних програм неправильно класифікованих як метаморфних вірус.

За результатами проведених експериментів можна зробити висновок, що найвищий рівень достовірності виявлення зафіксовано для метаморфних вірусів, створених за допомогою генератора G2, і який складає 0,97 (таблиця 1). Найнижчий рівень достовірності виявлення було зафіксовано для метаморфного ЗПЗ, створеного за допомогою NGVCK – 0,8671. Разом із тим, слід відзначити, що найнижчий рівень помилок 1-роду зафіксовано для метаморфного ЗПЗ VCL32 (0,0587), проте для решти метаморфних зразків, цей показник не перевищував значення 0,0641.

Таблиця 1

Оцінка достовірності виявлення метаморфних вірусів запропонованою інформаційною системою

Вид метаморфного ЗПЗ	Кількість зразків	Достовірність виявлення	Помилки 1-го типу
NGVCK	50	0.8671	0.0641
VCL32	50	0.8905	0.0587
G2	50	0.9752	0.0094
MetaPHOR	50	0.9157	0.0598

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

В роботі запропоновано інформаційну систему виявлення та ідентифікації метаморфних вірусів у локальній комп'ютерній мережі. В основу представленої інформаційної системи закладено два методи, а саме метод виявлення метаморфних вірусів на основі аналізу поведінки програми з використанням модифікованих емуляторів в локальній мережі та метод ідентифікації на основі пошуку та порівняння еквівалентних функціональних блоків між програмами. Обидва методи ґрунтуються на концепції порівняння копій метаморфних вірусів, результатом якого є визначення набору ознак, що використовується для виявлення метаморфних вірусів. Для оцінки достовірності виявлення та рівня хибних спрацювань проведено ряд експериментів. Створення тестової вибірки метаморфного зловмисного програмного забезпечення передбачало залучення метаморфних генераторів NGVCK, VCL32, G2 та MetaPHOR. За результатами проведених експериментів найвищий рівень достовірності виявлення зафіксовано для метаморфних вірусів, створених за допомогою генератора G2, і який складає 0,97. Найнижчий рівень достовірності виявлення було зафіксовано для метаморфного ЗПЗ, створеного за допомогою NGVCK – 0,8671. Разом із тим, слід відзначити, що найнижчий рівень помилок 1-роду зафіксовано для метаморфного ЗПЗ VCL32 (0,0587), проте для решти метаморфних зразків, цей показник не перевищував значення 0,0641.

Література

1. Jha A. K. A Novel Framework for Metamorphic Malware Detection / A. K. Jha, A. Vaish, S. Patil // SN Computer Science. – Vol. 4, 10 (2023). doi: 10.1007/s42979-022-01433-1

2. Sahay S. K. Evolution of Malware and Its Detection Techniques. / S. K. Sahay, A. Sharma, H. Rathore // Information and Communication Technology for Sustainable Development. Advances in Intelligent Systems and Computing, vol 933, 2020, Springer, Singapore. doi: 10.1007/978-981-13-7166-0_14

3. Campion M. Learning metamorphic malware signatures from samples / M. Campion, M. Dalla Preda, R. Giacobazzi // Journal of Computer Virology and Hacking Techniques, 17 (2021), pp. 167-183.

4. Salah M. Instrumenting API Hooking for a Realtime Dynamic Analysis / M. Salah, M. F. Marhusin, R. Sulaiman // *Proceedings of 2019 International Conference on Cybersecurity (ICoCSec)*, Nilai, Negeri Sembilan, Malaysia, 2019, pp. 49-52

5. Verma A. K. Malware Detection Approaches using Machine Learning Techniques- Strategic Survey / A. K. Verma, S. Sharma // *Proceedings of 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, 2021, pp. 1958-1962. doi: 10.1109/ICAC3N53548.2021.9725369

6. Yang Y. Towards effective metamorphic testing by algorithm stability for linear classification programs / Y. Yang, Z. Li, H. Wang, C. Xu, X. Ma // Journal of Systems and Software. – 180 (2021). doi: 10.1016/j.jss.2021.111012.

7. Pomorova O. Metamorphic Viruses Detection Technique based on the the Modified Emulators / O. Pomorova, O. Savenko, S. Lysenko, A. Nicheporuk // CEUR Workshop Proceedings. – 1614 (2016). – pp. 375-383.

8. Savenko O. Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search / O. Savenko, S. Lysenko, A. Nicheporuk et al // CEUR Workshop Proceedings. – 1844 (2017). – pp. 555-569.

9. Savenko O. Approach for the Unknown Metamorphic Virus Detection / O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko // *Proceedings of 9-th IEEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Bucharest, Romania, 2017, pp. 453-458. doi: 10.1109/IDAACS.2017.8095052

10. Qemu. Open source processor emulator [online] Available: http://wiki.qemu.org/Main_Page

11. Нічепорук А.О. Використання нечіткої класифікації для виявлення метаморфних вірусів в корпоративній мережі / А.О. Нічепорук // Вісник Хмельницького національного університету. – 2016. – № 4. – С.128-132

References

1. Jha A. K. A Novel Framework for Metamorphic Malware Detection / A. K. Jha, A. Vaish, S. Patil // SN Computer Science. – Vol. 4, 10 (2023). doi: 10.1007/s42979-022-01433-1

2. Sahay S. K. Evolution of Malware and Its Detection Techniques. / S. K. Sahay, A. Sharma, H. Rathore // Information and Communication Technology for Sustainable Development. Advances in Intelligent Systems and Computing, vol 933, 2020, Springer, Singapore. doi: 10.1007/978-981-13-7166-0_14

3. Campion M. Learning metamorphic malware signatures from samples / M. Campion, M. Dalla Preda, R. Giacobazzi // Journal of Computer Virology and Hacking Techniques, 17 (2021), pp. 167-183.

4. Salah M. Instrumenting API Hooking for a Realtime Dynamic Analysis / M. Salah, M. F. Marhusin, R. Sulaiman // Proceedings of 2019 International Conference on Cybersecurity (ICoCSec), Nilai, Negeri Sembilan, Malaysia, 2019, pp. 49-52

5. Verma A. K. Malware Detection Approaches using Machine Learning Techniques- Strategic Survey / A. K. Verma, S. Sharma // Proceedings of 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2021, pp. 1958-1962. doi: 10.1109/ICAC3N53548.2021.9725369

6. Yang Y. Towards effective metamorphic testing by algorithm stability for linear classification programs / Y. Yang, Z. Li, H. Wang, C. Xu, X. Ma // Journal of Systems and Software. – 180 (2021). doi: 10.1016/j.jss.2021.111012.

7. Pomorova O. Metamorphic Viruses Detection Technique based on the the Modified Emulators / O. Pomorova, O. Savenko, S. Lysenko, A. Nicheporuk // CEUR Workshop Proceedings. – 1614 (2016). – pp. 375-383.

8. Savenko O. Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search / O. Savenko, S. Lysenko, A. Nicheporuk et al // CEUR Workshop Proceedings. – 1844 (2017). – pp. 555-569.

9. Savenko O. Approach for the Unknown Metamorphic Virus Detection / O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko // Proceedings of 9-th IEEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Bucharest, Romania, 2017, pp. 453-458. doi: 10.1109/IDAACS.2017.8095052

10. Qemu. Open source processor emulator [online] Available: http://wiki.qemu.org/Main_Page

11. Nicheporuk A. O. Vykorystannia nechitkoi klasyfikatsii dlia vyiavlennia metamorfnykh virusiv v korporatyvnii merezhi / A. O. Nicheporuk //Herald of Khmelnytskyi National University . – Vol. 4 (2016). – C.128-132