DOI 10.31891/2307-5732-2019-271-2-114-121
УДК 004.491.42

О. С. САВЕНКО
Хмельницький національний університет

МЕРЕЖНИЙ МЕТОД ВИЯВЛЕННЯ ФАЙЛОВОГО ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В КОМП'ЮТЕРНИХ СИСТЕМАХ ЛОКАЛЬНИХ МЕРЕЖ

В роботі розроблено мережний метод виявлення файлового зловмисного програмного забезпечення. Він базується на двох методах, які здійснюють побудову поведінкової сигнатури та її подальший аналіз на наявність файлового зловмисного програмного забезпечення і розбиття на блоки виконуваної програми та дослідження її на наявність поліморфного та метаморфного коду вірусу. Мережний метод є основою для організації функціонування програмних модулів розподіленої багаторівневої системи і дозволяє організувати її роботу та її компонент, зокрема здійснювати вилучення ймовірно інфікованих програмних модулів з розподіленої багаторівневої системи, встановлення відношення до файлового зловмисного програмного забезпечення на основі обміну і обробки знань, сканування виконуваних файлів створенням для них окремих процесів. Для здійснення детальнішого аналізу програмного коду на основі мережного методу до процесу виявлення залучаються інші програмні модулі.

Ключові слова: зловмисне програмне забезпечення, розподілені системи, комп'ютерні системи, локальна комп'ютерна мережа, мережний метод, поліморфний вірус, метаморфний вірус.

O. S. SAVENKO
Khmelnytsky National University

NETWORK METHOD OF DETECTION OF COMPUTER VIRUSES IN THE LOCAL NETWORK

The network malware detection method is developed in this work. It is based on two methods for constructing a behavioral signature and its subsequent analysis of the existence of file malware and breaking down the blocks of the executable program and examining it for the presence of a polymorphic and metamorphic virus code. The network method is the basis for organizing the functioning of the distributed module multi-level system software modules and allows it to organize its work and its component, in particular, to seize potentially infected software modules from a distributed multi-level system, to establish a relation to file malware based on the exchange and processing of knowledge, scan executable files creating separate processes for them. To implement a more detailed analysis of the code based on the network method to the detection process involved other software modules. The developed methods for detecting malicious software exploits behavioral signatures, and a representation of behavioral signatures is used to represent them in the RBS, which is pre-filled with sample types of file malware. Behavioral signatures are formed on the basis of incidence distribution and malicious file malware matrices, which are specified by the functions of API functions calls.

Keywords: Malware, Distributed Systems, Computer Systems, Local Computer Network, Network Method, Polymorphic Virus, Metamorphic Virus.

Вступ. Постановка задачі

Використання зловмисного програмного забезпечення (ЗПЗ) щоденно зростає, створюючи при цьому проблеми користувачам комп'ютерних систем (КС). Новими сферами його застосування для отримання вигоди і переваг, враховуючи попередню фінансову, стали військова і політична. Військові доктрини багатьох країн світу включають розвиток військових кіберпідрозділів, які розвивають і використовують ЗПЗ для нанесення матеріальної шкоди інфраструктурі інших країн. Застосування руйнуючих технологій ЗПЗ дозволяє здійснювати віддалені атаки, не потребуючи перебування поряд з об'єктом атаки [1–3]. Особливої уваги заслуговує захист від таких атак підприємств і організацій реального сектору економіки, атаки на які протягом останніх років суттєво зросли і завдають їм значних збитків. Проведення успішних атак в різних секторах країни може паралізувати на тривалий час її основну інфраструктуру та фінансовий сектор. Крім того, поява нових фінансових інструментів, зокрема альтернативних валют, мотивує зловмисників до подальшого технологічного розвитку ЗПЗ для отримання вигоди. При цьому вони починають залучати обчислювальні потужності не тільки власні, а і інших користувачів, комп'ютерні системи яких приєднані до глобальної мережі. Тому, проблема виявлення ЗПЗ для організацій та підприємств залишається актуальною.

Тенденції розвитку технологій створення і поширення зловмисного програмного забезпечення демонструють активне розширення технічних можливостей таких засобів. Сучасне зловмисне програмне забезпечення представляє собою складні багатофункційні програмні системи та комплекси, які побудовані з використанням ефективних методів створення програмних засобів та методів поширення зловмисного коду.

Виявлення зловмисного програмного забезпечення здійснюється за допомогою різноманітних засобів. Ефективність та достовірність виявлення суттєво залежать, зокрема, і від архітектури таких засобів, а також їх позиціонування та місця розміщення в комп'ютерних системах, зокрема і локальних мережах. Дослідження відомих антивірусних методів та засобів вказують, що реалізація нових принципів, моделей та методів виявлення конкретних типів зловмисного програмного забезпечення шляхом створення відповідних систем потребує подальшого розвитку. Перспективним напрямом досліджень є створення розподілених систем виявлення ЗПЗ [4] для використання в локальних мережах організацій та підприємств. Такі засоби виявлення мають розподілену архітектуру і тоді підвищення достовірності виявлення ЗПЗ можливе за

рахунок залучення груп КС локальної мережі. Важливим критерієм ефективності таких розподілених систем виявлення є наявність в них реалізованих методів, що дозволяють здійснювати виявлення нового ЗПЗ.

В сучасних системах виявлення ЗПЗ застосовуються методи та засоби, що використовують знання про його функціонування та поведінку, які представляються поведінковими сигнатурами. Дослідження функціонування ЗПЗ в різних КС локальної мережі розподіленими системами виявлення дає змогу здійснити порівняння отриманих поведінкових сигнатур ЗПЗ з різних КС з метою його виявлення. Важливим при цьому є вибір архітектури таких розподілених систем, яка дозволить ефективно організувати процес виявлення в локальній мережі в порівнянні з виявленням в окремій КС.

Розглянемо файлове зловмисне програмне забезпечення, яке характеризується тим, що його поширення першочергово має відбуватись в окремій комп'ютерній системі. Файлове ЗПЗ, що міститься у файлах виконуваних програм, після запуску виконуваних програм отримує можливість для пошуку програмних файлів для подальшого свого поширення, для виконання деструктивних дій та отримання контролю над КС з метою приховування своєї присутності. Переміщення файлового ЗПЗ в інші КС мережі може бути здійснено переважно користувачами через порушення політик безпеки. Наявність однакового за функціоналом файлового ЗПЗ в різних КС локальної мережі організації чи підприємства потребує для його виявлення наявності таких методів його виявлення, які б дозволяли враховувати результати моніторингу і сканування різних КС локальної мережі і приймати рішення про наявність ЗПЗ. Тому, перспективною задачею є розробка методів виявлення файлового ЗПЗ з врахуванням особливостей їх використання саме в розподілених системах.

Попередні роботи

Фахівцями з розробки методів виявлення ЗПЗ розроблено системи виявлення, в яких реалізовано виконання відповідних методів.

В роботі [5] запропоновано систему виявлення кібератак на основі залучення нейромережних імунних детекторів. Розроблена система складається з двох частин. Перша реалізована апаратно й працює постійно в режимі реального часу. Друга частина представлена програмним забезпеченням на виділеному комп'ютері, який використовується для аналізу поточних атак та створення відповідних засобів захисту. Прийняття рішення про можливий вплив ЗПЗ здійснюється із залученням системи нейромережних детекторів, в основу якої закладено алгоритм Мамдані.

В роботі [6] представлено інтелектуальну адаптивну систему виявлення ЗПЗ на основі інтеграції штучних імунних систем та штучних нейронних мереж. Така система працює за основними принципами штучної імунної системи, де імунні детектори представляють нейронну мережу і виявляють зловмисний шаблон за допомогою аналізу структури виконуваного коду. Запропонована система володіє функціями самоадаптивності та самонавчання та дозволяє здійснювати виявлення як відомих, так і нових видів ЗПЗ.

Авторами роботи [7] запропоновано систему ідентифікації та класифікації для мережних кібератак. Для реалізації системи запропоновано використання комбінація різних методів штучного машинного навчання, а саме нейронних мереж, імунної системи, нейрофізичних класифікаторів та метод опорних векторів. Відмінною особливістю запропонованої системи є багаторівневий аналіз мережевого трафіку, що дає можливість виявляти атаки методом підпису та комбінувати набір адаптивних детекторів на основі методів машинного навчання.

У роботі [8] запропоновано гібридну систему виявлення рідковживаних кібератак на основі використання штучних імунних систем та нечіткої кластеризації (FC-ANN). Спочатку FC-ANN розподіляє навчальні дані на декілька підгруп з використанням методів нечіткої кластеризації. Для отримання остаточних результатів, визначається оцінка для кожного елементу з сформованих підгруп та виконується їх поєднання з використанням штучної імунної системи.

Проведений аналіз показав, що для виявлення ЗПЗ відомі системи, представлені в [5–8] та інші, здійснюють аналіз мережного трафіку, файлів аудиту, пакетів, що передаються по мережі, перевіряють конфігурацію відкритих мережевих сервісів, сканують файли виконуваних програм, перевіряють завантажувальний сектор жорсткого диску, сканують оперативний запам'ятовуючий пристрій, здійснюють динамічне трасування процесів. Для встановлення факту порушення роботи КС, відомі системи використовуються різні методи машинного навчання, а саме нейронні мережі, штучні імунні системи, метод опорних векторів, Байєсові мережі, нечітку кластеризацію. Проте, основним недоліком представлених систем є переважно їх хост-орієнтований підхід до виявлення ЗПЗ.

Для того, щоб ефективно застосовувати методи та засоби виявлення зловмисного програмного забезпечення необхідно розробити метод, виконання якого б враховувало мережний підхід.

Основна частина

Розроблена розподілена багаторівнева система (РБС) виявлення ЗПЗ, яка представлена в [4], дозволяє здійснювати виявлення ЗПЗ, як мережного так і файлового. Для виявлення файлового ЗПЗ розроблено методи, використання яких можливе в окремих КС і які реалізовано в РБС. Для використання цих методів в локальній мережі розроблено мережний метод виявлення файлового ЗПЗ, який інтегрує розроблені методи та використовує організацію розподіленої системи для підвищення достовірності виявлення.

Мережний метод виявлення файлового ЗПЗ в комп'ютерних системах локальних мереж базується на основі здійснення узгодження програмних модулів РБС при прийнятті рішення та використанні

розроблених методів виявлення файлового ЗПЗ і складається з таких основних кроків:

1. Здійснення сканування виконуваних файлів із створенням окремих процесів для кожного досліджуваного виконуваного файлу. Всі процеси створюються запуском одного компоненту програмного модуля (ПМ) в КС, який на основі закладеного функціоналу для сканування приймає рішення про потребу застосування методів виявлення.

2. Здійснення збору даних моніторингу після виявлення певних ймовірно зловмисних проявів в КС у вектор.

3. Формування вектору ознак ймовірно підозрілих дій для зібраних даних, компонентами якого є API функції.

4. Застосування методу [9] виявлення файлового зловмисного програмного забезпечення на основі динамічного формування поведінкової сигнатури шляхом відстеження викликів API функцій.

5. Прийняття рішення про місце обробки вектору ймовірно зловмисних дій.

6. Якщо аналіз завантаженості ресурсів КС показав невеликий відсоток завантаженості, тоді здійснити обробку в цій КС, інакше надіслати в іншу визначену ПМ КС.

7. Аналіз результатів кроку 6.

7.1. Якщо встановлено віднесення такого вектору до файлового ЗПЗ, тоді застосування методу виявлення файлового ЗПЗ на основі аналізу функцій обфускації [10] для перевірки на наявність поліморфного та метаморфного вірусу.

7.2. Якщо перевірка встановила, що досліджуваний вектор не містить зловмисного навантаження, тоді здійснити зупинку дослідження процесу, на основі якого він був сформований.

7.3. Якщо встановлено, що досліджуваний вектор містить зловмисне навантаження, тоді здійснити зупинку відповідного процесу та почати дослідження файлів з таким же іменем в інших КС.

7.4. Здійснення пошуку і дослідження на основі отриманих відомостей аналогічних файлів в інших КС мережі, де встановлена РБС її програмними модулями. Сканування файлів в одній КС викликає такі ж сканування в інших ПМ, оскільки однакові файли, якщо це організація чи підприємство, міститимуться в різних КС. Тоді, якщо це так, то здійснювати порівняння їх поведінкових сигнатур.

8. Обробка варіантів подій із залученням решти ПМ РБС. На основі варіантів подій з табл. 1 виокремлення варіантів, в яких можливе відключення зловмисним програмним забезпеченням ПМ РБС, і встановлення такої події для оцінки іншими компонентами РБС. В цьому випадку здійснення вилучення ПМ з РБС.

8.1. Для варіантів 1-256 задіяти стратегію 1 на основі прийняття рішення рештою ПМ та здійснити вилучення ПМ з РБС.

8.2. Для варіантів 257-320 задіяти стратегію 2 на основі прийняття рішення рештою ПМ РБС.

8.3. Для варіантів 321-340 задіяти стратегію 3 на основі прийняття рішення рештою ПМ.

8.4. Для варіантів 341-484 задіяти стратегію 4 на основі прийняття рішення всіма ПМ з РБС та обміну інформацією між ПМ.

8.5. Для варіантів 485-502 задіяти стратегію 5 на основі прийняття рішення всіма ПМ з РБС та обміну інформацією між ПМ.

8.6. Для варіантів 503-512 задіяти стратегію 6 на основі прийняття рішення ПМ та обміну інформацією з рештою ПМ РБС.

9. Обчислення значення ймовірностей в станах ПМ і вимога для інших ПМ здійснити обчислення ймовірності бути ураженої для всієї РБС. Цей крок здійснюється позапланово через дослідження наявного зловмисного прояву в одній з КС.

10. Здійснення оптимізації вектору, що додається в базу зловмисних дій та атак, за генетичним алгоритмом.

11. Формування ймовірностей перебування в станах для надсилання іншим ПМ для визначення стану РБС.

Таким чином, згідно розробленого мережного методу виявлення файлового ЗПЗ програмні модулі РБС дозволяють здійснити вилучення ймовірно уражених ПМ з РБС, встановлення відношення до файлового ЗПЗ на основі обміну і обробки знань, сканування виконуваних файлів створенням для них окремих процесів. ПМ на основі такого мережного методу виявлення файлового ЗПЗ приймають рішення про зміну структури РБС та визначають уражену КС. Завдяки використанню РБС між її ПМ здійснюється обмін інформацією про зловмисні процеси в КС для додаткової перевірки та динамічно залучаються обчислювальні ресурси інших КС для виконання певних завдань інших компонентів системи. Це дозволяє оперативно виконувати задачі виявлення зловмисного програмного забезпечення. РБС надає можливість аналізувати додатково відмічені процеси та здійснювати виявлення ЗПЗ в тих КС локальної мережі, в яких воно не було виявлено в процесі дослідження або було пропущено, за рахунок інформації про зловмисний процес з іншої КС.

Для виконання кроку 8 проаналізуємо можливі варіанти подій в КС за присутності в ньому ПМ РБС та ЗПЗ. На основі їх дослідження визначимо стратегії поведінки ПМ в процесі появи таких подій. Розгляд можливих варіантів проведемо з врахуванням можливої наявності, також, мережного ЗПЗ, зокрема бот-мереж. Це необхідно враховувати, щоб здійснювати обробку повної множини варіантів для введених подій.

Таблиця 1

Фрагмент можливих стратегій для подій в КС локальної мережі

№ з/п	Встановлення в КС до (1)/після(0) встановлення ПМ або ПЗ вузла бот-мережі (Б)		Стартовий контроль в КС: так(1) / ні(1)		Атака з цієї КС: так(1) / ні(1)	Атака з цієї КС на КС цієї ж мережі: так(1) / ні(1)	Атака на КС: так(1) / ні(1)	Атака з КС цієї ж мережі: так(1) / ні(1)	Наявність файлового ЗПЗ, встановленого до(1) / після(0) встановлення ПМ	Стратегії
	ПМ	Б	ПМ	Б						
1	1	0	0	1	1	1	1	1	1	1
2	1	0	0	1	1	1	1	1	0	1
…	…	…	…	…	…	…	…	…	…	…
512	0	0	0	1	0	0	0	0	0	6

Розглянемо випадок, коли в КС вже створено вузол бот-мережі, тобто завантажено зловмисне програмне забезпечення вузла бот-мережі, отримано контроль над певним програмним забезпеченням КС. Якщо вузол бот-мережі міститься в КС, тоді при її запуску стартове зловмисне навантаження повинно проявити себе в одному з процесів, які створюються прописаними виконуваними програмами в файлах автозапуску. Інакше, воно не зможе активуватись в КС при кожному її ввімкнені і, як наслідок, буде вилучено з часом з бот-мережі. Або воно очікуватиме запуску користувачем певних програм чи програми, що теж може відбутись або не відбутись. Тому, висуваємо гіпотезу, що запуск програмного забезпечення вузла бот-мережі здійснюється після ввімкнення КС. В процесі отримання контролю над КС бот-мережа чи на перших етапах свого функціонування вузол бот-мережі здійснить призупинення роботи відомих антивірусних засобів та перейде в режим їх імітації їх роботи.

Варіантів встановлення ПМ РБС може бути два: 1) під час нового встановлення програмного забезпечення КС встановлюється програмне забезпечення ПМ; 2) програмне забезпечення ПМ встановлюється у вже функціонуючу КС. При першому варіанті КС може бути новою і тому потребує встановлення ПЗ або вже використовуваною раніше і потребує повного переустановлення ПЗ. Але навіть, якщо вона є новою, то як правило містить операційну систему і до включення її до складу локальної мережі вже міститиме певне визначене ПЗ. Таким чином, вірогідність отримати зловмисний програмне забезпечення при першому варіанті дуже мала, але теоретично можлива. При другому варіанті ймовірність наявності ЗПЗ в КС більша, ніж в першому.

Якщо припустити, що ЗПЗ в першому варіанті не було, тоді ПМ РБС встановлюється першою і отримує доступ для контролю над всією КС. Цей контроль передбачає основні такі дії: запуск першою, звірку виконуваних програм з файлу автозапуску, розміщення резидентної програми в пам’яті, моніторинг API викликів і збір їх у вектори. Програмне забезпечення вузла бот-мережі може потрапити в КС двома способами: користувач, мережа. Тоді, таке ЗПЗ буде обов’язково пробувати прописати себе для можливості активації при наступному ввімкнені КС, створити можливості для розміщення своєї резидентної частини в пам’яті. При спробі прописатись в ПМ РБС буде виявлено через здійснення самоконтролю при запуску, який закладено в функціонал ПМ, якщо ЗПЗ допустить запуск наступних після нього команд. Якщо ж ЗПЗ не допустить запуск команд ПМ, тоді він не відзвітується перед рештою ПМ РБС і буде ними вилучений з РБС, що дозволить блокувати таку КС. При другому варіанті якщо ЗПЗ не було, а з’явилось, тоді та ж стратегія, що і для першого розглянутого варіанту. А якщо ЗПЗ вже було, тоді в процесі функціонування КС виникне конфлікт між зловмисним програмним забезпеченням вузла бот-мережі і ПМ, суть якого полягатиме в змаганні за перший запуск та доступ до оперативної пам’яті, облік для ПМ і блокування відомих АЗ. Певний час вони можуть функціонувати. В будь-якому з варіантів ПМ виступить як об’єкт для атаки в якості приманки. Але не все ЗПЗ, а особливо бот-мережі, розроблено на основі стратегії змагання за повний контроль над КС чи перший запуск. Для приховування своєї присутності стратегія перебування в КС може передбачати, наприклад, тільки прописування в одному з файлів, які містяться у файлі автозапуску. Тоді, виявити таке ЗПЗ можна лише за його проявами, важливих з яких для вузла бот-мережі, є необхідність підтримки зв’язку з своїм контролюючим центром. Тобто, для цього випадку вузол бот-мережі не проявляє активність, а очікує команди і потім запускає повний пакет програмного забезпечення необхідний для її виконання. Така стратегія дозволяє перебувати в КС тривалий час без виявлення. Але її та інші стратегії, закладені в механізм функціонування бот-мережі, можуть порушити інші події, які викликані сторонніми проявами. Наприклад, розглянемо перший варіант подій з табл.1, тобто коли отримано команду здійснити атаку на іншу КС чи ресурс, а в цей час подібна атака відбувається на цю ж КС. Така подія може відбуватись тільки лише коли атака на КС ведеться не з цієї ж бот-мережі, вузол якої міститься в КС, а іншим зловмисним програмним забезпеченням. В цьому випадку ПМ РБС переходить до стану 7 [4], який активується через надмірне звернення до портів та викликом функцій орієнтованих на встановлення мережних з’єднань. ПМ визначає IP адресу, куди націлена інтенсивна відправка пакетів і здійснює збільшення інтервалу надсилання пакетів через блокування відповідних пакетів. Проведення атаки на цю ж

КС, яка є в локальній мережі, тому не може містити ресурсів для атаки, може здійснюватись тільки іншим зловмисником для встановлення контролю над нею. Ця атака може відбуватись з іншої КС цієї ж мережі, тоді ПМ встановлює КС, про яку повідомляє решті ПМ РБС для здійснення її дослідження, або з-за меж локальної мережі. В будь-якому з випадків відбивання атаки здійснюватиметься відповідними засобами. Але частина атакуючих дій може бути успішною, тоді в КС будуть одночасно присутні ПМ, ПЗ вузла бот-мережі та нове ЗПЗ. При їх функціонуванні виникне конфлікт, який проявиться в спробі здійснення контролю над КС, що буде впливати на її нормальний порядок функціонування. В цьому випадку ПМ при звірці інформації про розміщені в КС файли виявить поточні зміни і почне дослідження підозрілих файлів.

У другому варіанті подій якщо в КС міститься вузол бот-мережі і з неї здійснюється атака, тоді ПМ порівнює зростаючу інтенсивність викликів мережних з'єднань та здійснює їх затримки методом призупинення та виявляє процес, який їх генерує. Цей варіант можливий при повному контролі ПМ в КС і другорядній ролі вузла бот-мережі, якщо ж інакше, то тобі вузол бот-мережі змагався б за ресурси КС з ПМ і цим виявив би себе.

В третьому варіанті подій розглядаємо можливість проникнення в КС з малою ймовірністю. При цьому ПМ аналізуватиме інтенсивність звернення до портів та надходження пакетів, а також подальшого розміщенням файлів, що надійшли і фіксування місця їх розміщення. В подальшому ПМ розміщує такі файли в свій реєстр для спостереження за ними протягом певного часу.

В четвертому варіанті подій ПМ здійснює змагання за ресурси з ПЗ вузла бот-мережі. І аналогічно, як у першому варіанті, може бути дві варіації: ПЗ вузла бот-мережі прагнутиме встановити повний контроль над КС або згідно своєї стратегії функціонування приховуватиме свою присутність до отримання команди на проведення атаки.

П'ятий варіант подій включає результати першого з врахуванням того, що додатково наявне в КС файлове ЗПЗ. Його присутність ускладнюватиме функціонування КС, бо воно перебуватиме в оперативній пам'яті, здійснюватиме своє поширення шукаючи об'єкти для втілення. Характерною особливістю в цьому варіанті буде завантаженість ресурсів і сповільнення роботи КС. ПМ відмічатиме зміни в файлах, які прописані в її реєстрі для цієї КС. Основним місцем, де зіткнуться ПМ, ПЗ вузла бот-мережі та файлове ЗПЗ буде оперативна пам'ять. Атака на цю КС підсилюватиме ускладнення роботи КС, що аналогічно до першого варіанту призведе до тривалої обробки результатів моніторингу ПМ в КС і повідомлення про події іншим ПМ РБС.

Аналогічно будуються інші стратегії розвитку подій для варіантів, коли не передбачається наявність ПЗ вузла бот-мережі, тобто при наявності лише файлового ЗПЗ.

Стратегії ПМ в різних варіантах подій:

1) КС контролюється бот-мережею, ПМ в КС заблоковано вузлом бот-мережі, КС виконує поставлені користувачем задачі;

2) КС контролюється бот-мережею, ПМ в КС заблоковано вузлом бот-мережі, КС функціонує з тривалими перебоями та затримками у виконанні запитів користувача;

3) КС контролюється бот-мережею, ПМ в КС заблоковано вузлом бот-мережі, КС функціонує з тривалими перебоями та затримками у виконанні запитів користувача, файлове ЗПЗ в КС здійснює своє поширення;

4) КС контролюється ПМ, вузол бот-мережі в КС досліджується ПМ, КС виконує поставлені користувачем задачі;

5) КС контролюється ПМ, вузол бот-мережі та файлове ЗПЗ в КС досліджуються ПМ, КС виконує поставлені користувачем задачі;

6) КС контролюється ПМ, який здійснює моніторинг і обробку подій.

Проаналізуємо логіку взаємодії програмного модуля РБС та ПЗ вузла бот-мережі в КС для отримання стратегій. Задамо стадії функціонування варіантів подій та можливих стратегій їх розвитку часовою діаграмою та виділимо в ній повторювані фрагменти для здійснення оптимізації при прийнятті рішення ПМ РБС. Шаблони можливих варіантів подій в КС та їх варіації формують одну з шести стратегій. Стратегії для подій в КС локальної мережі представлено в табл.1.

Ймовірності в станах ПМ РБС залежать від варіантів подій, які відбуватимуться в КС, пов'язаних з присутністю ЗПЗ, та визначаються за формулами 1:

$$p_{1,j} = \begin{cases} 0, ПМ\ знаходиться\ тільки\ в\ стані\ 1\ або\ стані\ 1\ і\ 8\ одночасно \\ \frac{s}{100}, ПМ\ перейшов\ до\ стану\ s\ і\ продовжує\ бути\ в\ стані\ 1 \end{cases}, \quad (1)$$

$$p_{s,j} = \frac{\sum_{c=1}^{7} p_{s,j,c}}{7 * s},$$

де j -й програмний модуль РБС, s – номер стану (для станів 2–7).

Для подій, коли декілька станів одночасно працюють розрахунок проводиться з врахуванням виконання умов.

Розроблений мережний метод виявлення файлового ЗПЗ базується на двох методах, представлених

детальніше в [9, 10], які здійснюють побудову поведінкової сигнатури та її подальший аналіз на наявність файлового ЗПЗ і розбиття на блоки виконуваної програми та дослідження її на наявність поліморфного та метаморфного вірусу. Отримання результату підвищення достовірності виявлення згідно мережного методу досягається залученням до процесу програмних модулів РБС, що дозволяє здійснювати детальніший аналіз програмного коду.

Експерименти

Для порівняння розроблених методів і РБС [4] було проведено експеримент в КС локальної мережі для перевірки достовірності виявлення мережним методом файлового ЗПЗ. Результати роботи системи було збережено у файли-журнали.

Для визначення ефективності розробленої системи було проведено ряд експериментів. Для цього було залучено мережу, що складалась з 20 комп'ютерних систем. Кожна комп'ютерна система була обладнана віртуальним середовищем на основі Qemu, яке задіювалось ПМ розробленої системи для дослідження поведінок ймовірно зловмисних програм і отримання викликів API функцій. Дослідження виконуваних програм здійснювалось на трьох етапах їх функціонування: потрапляння в КС, активізації та виконання закладених функцій. Кожен ПМ використовував базу поведінкових моделей файлового ЗПЗ на її різних етапах функціонування. Для розрахунку достовірності виявлення файлового ЗПЗ був проведений наступний експеримент з різними типами файлового ЗПЗ: файлові віруси, поліморфні віруси, метаморфні віруси, троянські програми. Було згенеровано 600 програмних об'єктів з функціональним навантаженням чотирьох розглядуваних типів файлового ЗПЗ по 150 кожного. Для отримання тестових зразків поліморфних та метаморфних вірусів було використано генератори NGVCK, PS-MPC, VCL32 та G2. Всі метаморфні варіації вірусів, які створювались за допомогою цих генераторів були скомпільовані з опціями anti-debugging та anti-emulation. Кожна із згенерованих варіацій метаморфних вірусів застосовувала основні техніки заплутування коду: вставку сміттєвих команд, використання еквівалентних інструкцій та переміщування блоків інструкцій.

Сигнатури згенерованих вірусів відсутні в базах сигнатур АПЗ та розробленої системи. Всі програмні об'єкти було поділено на групи для задання способу їх потрапляння в КС, щоб врахувати усі можливі шляхи проникнення в КС:

1) програмні об'єкти скопійовані на жорсткий диск кожної КС;
2) програмні об'єкти завантажені на флеш-носії і підключені до кожної КС;
3) програмні об'єкти завантажені на попередньо створений web-сайт;
4) програмні об'єкти архівовані та відправлені на попередньо створені електронні адреси;
5) програмні об'єкти завантажені на попередньо створений ftp-сервери всіх КС.

Запуск на виконання згенерованих програмних об'єктів був здійснений спеціальною програмою, яка встановлена в кожну КС і запустила по одному програмному об'єкту із ЗПЗ в кожній КС одночасно. Потім все розпочиналось знову і вибирався інший програмний об'єкт. Запуск корисних програм в усіх КС не виконувався. Після ввімкнення всіх КС завантажилась ОС та всі програми, які потрібні прописані для автоматичного запуску. Всі КС містили однакове апаратне та програмне забезпечення.

Результати проведеного експерименту та оцінки достовірності виявлення ЗПЗ розподіленою системою Distributed Multilevel System [11], в якій реалізовано методи, представлено в табл. 2. Крім того, за результатами проведено експерименту було також встановлено кількість ПМ, які залучались для дослідження протягом всього експерименту, та кількість ПМ, які були заблоковані рештою ПМ розробленої системи, під час виявлення. Це підтверджує застосування решти компонент розподіленої системи в процесі виявлення ЗПЗ окремим ПМ.

Таблиця 2

Результати експерименту для файлового ЗПЗ

Програмні об'єкти з наявним в них ЗПЗ		Кількість програм, виявлених як підозрілі	Відсоток виявлення, %	Кількість ПМ, які залучались для дослідження протягом всього експерименту	Кількість ПМ, які були заблоковані рештою ПМ розробленої системи, під час виявлення
Файлові віруси	150	146	97,3%	0	2
Поліморфні віруси	150	134	89,3%	57	7
Метаморфні віруси	150	138	92,3%	24	3
Троянські програми	150	128	85,3%	2	5
Всього	600	546	90,9	20,75	5,7

Для проведення порівняльного аналізу було обрано наступні відомі антивірусні засоби: ESET Smart Security (версія 10.1.204.0), Avast (версія 17.5.2303), Comodo Antivirus (версія 8.2.0.4674), Kaspersky (версія 17.0.0.61), McAfee Internet Security (версія 10.1.0), Dr.Web (версія 11.0), Microsoft Security Essentials (версія 4.11.15063.446), Avira Antivirus (версія 10.0). Результати роботи розробленої розподіленої системи Distributed Multilevel System виявлення ЗПЗ представлено діаграмою на рис. 1.

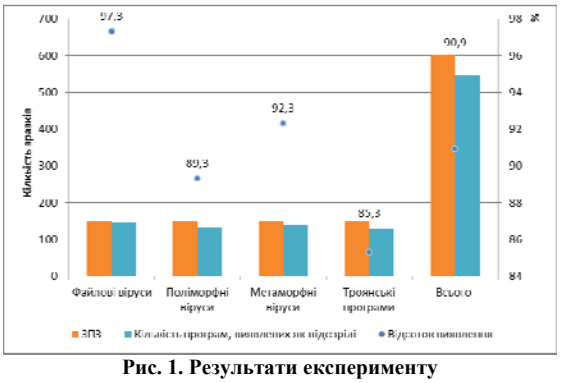

Рис. 1. Результати експерименту

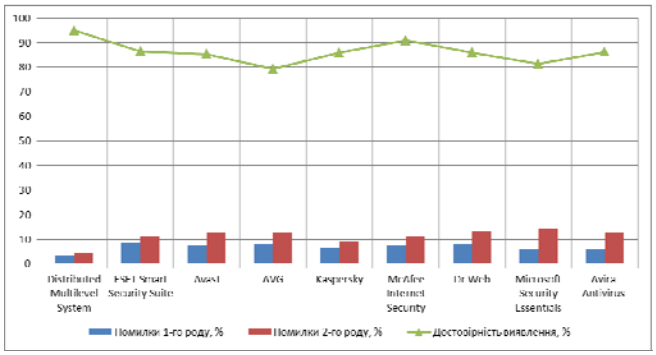

Рис. 2. Порівняльний аналіз розробленої системи Distributed Multilevel System з існуючими АПЗ (достовірність виявлення, %)

Порівняльний аналіз розробленої системи Distributed Multilevel System з існуючими АПЗ стосовно помилок першого роду (хибні спрацювання) та помилок другого роду (хибно-негативні результати) зображено діаграмою на рис. 2 та рис. 3.

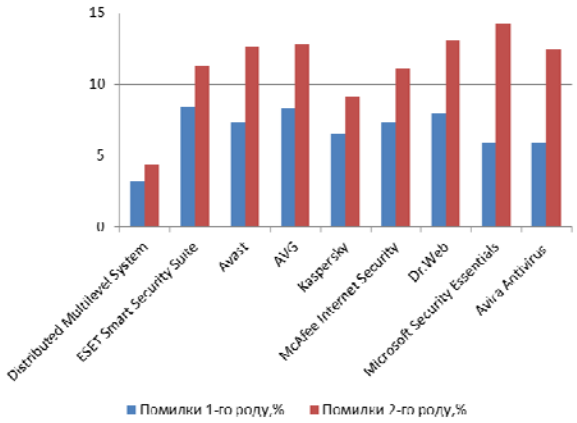

Рис. 3. Порівняльний аналіз розробленої системи Distributed Multilevel System з існуючими АПЗ: помилки першого роду (хибні спрацювання), %, та помилки другого роду (хибно-негативні результати), %

Результати експериментальних досліджень з використанням розробленої системи Distributed Multilevel System підтверджують вірність наукових положень розроблених методів та ефективність архітектури розподіленої багаторівневої системи, оскільки її впровадження підвищує достовірність виявлення на 5-12% в мережному представленні в порівнянні з хостовим, та на 2-4% у порівнянні з існуючими мережними АПЗ виявлення файлового ЗПЗ.

Таким чином, результати використання розробленої РБС Distributed Multilevel System та її підсистем на основі розробленого мережного методу для виявлення певних типів файлового ЗПЗ в КС локальних мереж є достатніми в порівнянні з аналогічними мережними АПЗ і підтверджують ефективність запропонованих теоретичних моделей та розроблених методів.

Висновки

Розроблені методи виявлення файлового ЗПЗ використовують поведінкові сигнатури і для їх представлення в РБС використовується база поведінкових сигнатур, яка попередньо наповнена зразками типів ЗПЗ. Поведінкові сигнатури формуються на основі матриць інцидентності поширення та деструктивних дій файлового ЗПЗ, які уточнюються послідовностями викликів API функцій.

Мережний метод виявлення файлового ЗПЗ є основою для організації функціонування програмних модулів РБС і дозволяє здійснити вилучення ймовірно уражених ПМ з РБС, встановлення відношення до файлового ЗПЗ на основі обміну і обробки знань, сканування виконуваних файлів створенням для них окремих процесів. Він базується на двох методах, які здійснюють побудову поведінкової сигнатури та її подальший аналіз на наявність файлового ЗПЗ і розбиття на блоки виконуваної програми та дослідження її на наявність поліморфного та метаморфного вірусу. Для здійснення детальнішого аналізу програмного коду на основі мережного методу до процесу виявлення залучаються інші ПМ РБС.

Напрямком подальших досліджень є розробка нових методів для виявлення ЗПЗ з метою розширення можливостей розподіленої системи для виявлення.

Література

1. Українська правда. Затримали хакерів, які "чистили" банківські рахунки і переводили кошти в криптовалюту [Електронний ресурс]. – Режим доступу : https://www.pravda.com.ua

/news/2019/01/10/7203471/ (дата звернення 25.03.2019). – Назва з екрану.

2. Українська правда. Найвідомішим у Darknet ресурсом заправляли українці – Кіберполіція [Електронний ресурс]. – Режим доступу : https://www.pravda.com.ua/news/2019/01/28/7205116/ (дата звернення 25.03.2019). – Назва з екрану.

3. Фокус. Хакеры через WordPress пытались атаковать сайт ЦИК, СБУ [Электронный ресурс]. – Режим доступа : https://focus.ua/ukraine/422005-xakery-cherez-wordpress-pytalis-atakovat-sajt-cik--sbu.html (дата обращения 25.03.2019). – Название с экрана.

4. Markowsky G. Distributed Malware Detection System Based on Decentralized Architecture in Local Area Networks / G. Markowsky, O. Savenko, A. Sachenko // Advances in Intelligent Systems and Computing. – 2019. – Vol. 871. – P. 582–598.

5. Komar M. High performance adaptive system for cyber attacks detection / M. Komar, V. Kochan, L. Dubchak, A. Sachenko, V. Golovko, S. Bezobrazov, I. Romanets // Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. – Bucharest (Romania), 21-23 September, 2017. – Vol. 2. – P. 853–858.

6. Golovko V. Neural Network Artificial Immune System for Malicious Code Detection / V. Golovko, S. Bezobrazov// Brest State Technical University. – 2015. – P. 1–7.

7. Branitskiy A. Hybridization of computational intelligence methods for attack detection in computer networks / A. Branitskiy, I. Kotenko // Journal of Computational Science. – 2017. – No. 23. – P. 145–156.

8. Wang G. A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering / G. Wang, J. Hao, J. Ma // Huang L. Expert Systems with Applications // An International Journal. – 2010. – Vol. 37. – Issue 9. – P. 6225–6232.

9. Савенко О.С. Формування сигнатури поведінки програм на основі трасування API викликів / О.С. Савенко, А.О. Нічепорук, А.А. Нічепорук, Ю.О. Нічепорук // Електротехнічні та комп'ютерні системи. – 2018. – № 29(105). – С. 67–77.

10. Markowsky G. The technique for metamorphic viruses' detection based on its obfuscation features analysis / G. Markowsky, O. Savenko, S. Lysenko, A. Nicheporuk // CEUR-WS – 2018. – Vol. 2104. – P. 680–687.

11. Савенко О.С. Архітектура багаторівневої програмної системи виявлення шкідливого програмного забезпечення в локальних комп'ютерних мережах / О.С. Савенко, В.І. Грибинчук, М.О. Кульчицький // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2018. – № 30-31. – С. 132–140.

References

1. Ukrainska pravda. Zatrymaly khakeriv, yaki "chystyly" bankivski rakhunky i perevodyly koshty v kryptovaliutu [Elektronnyi resurs]. – Rezhym dostupu : https://www.pravda.com.ua /news/2019/01/10/7203471/ (data zvernennia 25.03.2019). – Nazva z ekranu.

2. Ukrainska pravda. Naividomishym u Darknet resursom zapravlialy ukraintsi – Kiberpolitsiia [Elektronnyi resurs]. – Rezhym dostupu : https://www.pravda.com.ua/news/2019/01/28/7205116/ (data zvernennia 25.03.2019). – Nazva z ekranu.

3. Fokus. Hakery cherez WordPress pytalis atakovat sajt CIK, SBU [Elektronnyj resurs]. – Rezhim dostupa : https://focus.ua/ukraine/422005-xakery-cherez-wordpress-pytalis-atakovat-sajt-cik--sbu.html (data ob¬rasheniya 25.03.2019). – Nazvanie s ekrana.

4. Markowsky G. Distributed Malware Detection System Based on Decentralized Architecture in Local Area Networks / G. Markowsky, O. Savenko, A. Sachenko // Advances in Intelligent Systems and Computing. – 2019. – Vol. 871. – P. 582–598.

5. Komar M. High performance adaptive system for cyber attacks detection / M. Komar, V. Kochan, L. Dubchak, A. Sachenko, V. Golovko, S. Bezobrazov, I. Romanets // Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. – Bucharest (Romania), 21-23 September, 2017. – Vol. 2. – P. 853–858.

6. Golovko V. Neural Network Artificial Immune System for Malicious Code Detection / V. Golovko, S. Bezobrazov// Brest State Technical University. – 2015. – P. 1–7.

7. Branitskiy A. Hybridization of computational intelligence methods for attack detection in computer networks / A. Branitskiy, I. Kotenko // Journal of Computational Science. – 2017. – No. 23. – P. 145–156.

8. Wang G. A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering / G. Wang, J. Hao, J. Ma // Huang L. Expert Systems with Applications // An International Journal. – 2010. – Vol. 37. – Issue 9. – P. 6225–6232.

9. Savenko O.S. Formuvannia syhnatury povedinky prohram na osnovi trasuvannia API vyklykiv / O.S. Savenko, A.O. Nicheporuk, A.A. Nicheporuk, Yu.O. Nicheporuk // Elektrotekhnichni ta kompiuterni systemy. – 2018. – № 29(105). – S. 67–77.

10. Markowsky G. The technique for metamorphic viruses detection based on its obfuscation features analysis / G. Markowsky, O. Savenko, S. Lysenko, A. Nicheporuk // CEUR-WS – 2018. – Vol. 2104. – P. 680–687.

11. Savenko O.S. Arkhitektura bahatorivnevoi prohramnoi systemy vyiavlennia shkidlyvoho prohramnoho zabezpechennia v lokalnykh kompiuternykh merezhakh / O.S. Savenko, V.I. Hrybynchuk, M.O. Kulchytskyi // Kompiuterno-intehrovani tekhnolohii: osvita, nauka, vyrobnytstvo. – 2018. – № 30-31. – S. 132–140.

Рецензія/Peer review : 27.3.2019 р. Надрукована/Printed :11.4.2019 р.
Рецензент: д.т.н., проф. Говорущенко Т.О.