УДК 004.056 (043.2)

С.Ю. Гавриленко, І.В. Шевердін, Т.М. Шипова

Національний технічний університет «ХПІ», Харків

ВДОСКОНАЛЕНА МЕТОДОЛОГІЯ ПРОЕКТУВАННЯ СИСТЕМ АНТИВІРУСНОГО ЗАХИСТУ

У статті розроблені шаблони побудови антивірусної системи та описано шляхи вирішення проблематики антивірусного захисту. Забезпечено цілісність антивірусних модулів системи та можливість масштабування кінцевого продукту. Запропоновано принципово новий підхід у побудові антивірусів базуючись на використанні гіпервізору для компонентної агрегації. Вирішено питання відтворення інформації після руйнівної дії вірусної атаки. Запропоновано новий метод мапоорієнтованого опису комп'ютерних загроз. Описано один з методів побудови мап комп'ютерної загрози.

Ключеві слова: *комп'ютерні системи, гіпервізор, захист інформації, методологія проектування систем антивірусного захисту.*

Вступ

Постановка завдання та аналіз літератури. У наш час питання захисту особистих даних та інформації є дуже важливим, тому, що в сучасному світі існує дуже багато шкідливого програмного забезпечення, яке створюється спеціально для крадіжки особистих даних, або просто шпигує за користувачем, що в свою чергу може призвести до негативних наслідків. На даний момент розвитку людства є проблема інформаційної безпеки, як для нашої країни так і для багатьох інших країн світу, що породжує попит на нові методи перешкоджань.

Методи і принципи захисту теоретично не мають особливого значення, головне щоб вони були направлені на боротьбу зі шкідливими програмами. Але на практиці це є набагато складніше. Практично будь-яка антивірусна програма об'єднує в різних пропорціях всі технології і методи захисту від вірусів, створені до сьогоднішнього дня.

Аналіз літератури [1 – 9] показав, що антивірусні програми – це програми, основним завданням яких є захист від вірусів, або точніше, від шкідливого програмного забезпечення.

З усіх методів антивірусного захисту можна виділити дві основні групи:

1) сигнатурні методи – точні методи виявлення вірусів, засновані на порівнянні файлу з відомими зразками вірусів (сигнатурами);

2) евристичні методи – інтелектуальні методи виявлення, які дозволяють з певною вірогідністю припустити, що файл є зараженим.

Використовуючи данні методи разом можливо досягти великого проценту виявлення вірусів. Данні методи є основою сучасного антивірусного програмного забезпечення.

Суттєвою проблемою на даному розвитку антивірусного програмного забезпечення є можливість пошкодження модулів антивірусного захисту, що руйнує захисну функцію антивірусу та можливість пошкодження інформації новою вірусною атакою без відтворення її у майбутньому. Для вирішення даних проблем необхідно використовувати принципово новий підхід у проектуванні антивірусів, а саме виділити антивірус з операційної системи, тому що він має майже однаковий пріоритет з іншим програмним забезпеченням. Проблема виділення антивірусу або підвищення пріоритету у системі є дуже важлива для побудови комплексного антивірусного рішення.

Основна частина

У роботі пропонується виділити антивірус від операційної системи шляхом відтворення гіпервізору та використання антивірусного модуля у модулі виконання гіпервізору. Використання гіпервізору надає можливість рішення питання цілісності даних, а саме можливість збереження та відтворення даних у режимі виконання операційної системи за алгоритмами антивірусного модуля.

Даний підхід дозволяє абстрагувати операційну систему та антивірус, що гарантує цілісність антивірусного модуля захисту та аналізу, а також надає можливість масштабування всієї системи за рахунок інтерфейсної побудови, можливо кардинально змінювати структуру системи без зміни архітектури.

Також велика перевага гіпервізору, це апаратна віртуалізація. Зокрема сервіс, що приймає запити від зовнішніх ресурсів, є одним з класичних джерел для різного роду уразливостей в безпеці системи. Найчастіше несанкціонований доступ до операційної системи здійснюється саме через експлуатацію уразливостей деякого сервісу. Якщо піддався атаці сервіс поміщений у віртуальний контейнер, наслідки злому можуть бути значно менш руйнівними, ніж у випадку, коли сервіс виконується безпосередньо в операційній системі. В останньому випадку операційна система може бути зламана через один з сервісів, а під удар підставляються все виконуються сервіси.

У разі віртуалізованого сервісу, зловмисник може отримати доступ тільки в рамках віртуального контейнера, так що навіть якщо в системі паралельно виконуються інші сервіси (в інших віртуальних контейнерах), то їх роботі даний інцидент загрожувати не буде. Спроби зломщика вичерпати всі фізичні ресурси призведуть тільки до вичерпання ресурсів даного контейнера. Таким чином, зломщик не зможе заволодіти віддалений доступом, вичерпавши всі доступні ресурси. Крім того, зсередини контейнера у зломщика немає можливості перезаписати завантажувач і отримати який-небудь прямий контроль над обладнанням. При подібному інциденті спроектований мною модуль захисту може просто цілком знищити скомпрометований віртуальний контекст і відновити його і дані з резервної копії, без перезавантаження системи і не зупиняючи роботу інших виконуються на ньому сервісів.

Віртуалізація сервісів дає можливість простого і беззбиткового впровадження, переміщення і виведення з ладу будь-якого з сервісів без шкоди загальній інфраструктурі і без простою. Ще одна неочевидна, але важлива вигода використання гіпервізору, яка надається приміщенням сервісів у віртуальні контейнери походить від того, що віртуальні контексти, з одного боку, самодостатні, а з іншого – відокремлені від устаткування. Завдяки цьому, модуль управління має можливість маніпулювати віртуальними контейнерами як цілісними об'єктами, у тому числі переносити з одного носія на інший. Так, для більшості технологій віртуалізації є можливість «заморозки» контейнера в поточному стані, з можливістю подальшої розморожування, в тому числі на іншій машині, при тому що виконуються всередині контейнера процеси навіть не помітять що сталися зміни. Такі можливості дозволяють навіть в процесі роботи мігрувати віртуалізовані сервіси, наприклад, за різними вузлам мережі, гнучко перерозподіляючи обчислювальну навантаження, у тому числі серед вузлів кластера. Крім того, можна знищити віртуальний контейнер, який став непотрібним.

Для побудови даної антивірусної системи доцільно вирішити наступні задачі проектування:

1) розробка антивірусної операційної системи, котра мітить операційну систему та відтворює віртуальні апаратні засоби для неї;

2) розробка нейронного блоку, котрий аналізує стан оболонки, відтворює події та передає їх на блоки прийняття рішень та аналізу;

3) розробка блоку запитів користувача, котрий відстежує всі дій користувача, для того щоб виключити можливість потенційної помилки користувача або можливості дії програмного забезпечення яке використовується користувачем;

4) розробка блоку формування клонів, котрий будує зріз апаратної пам’яті для подальшого аналізу;

5) розробка блоку відгалуження, котрий формує окремі гілки пам’яті за зазначеною подією та відновлює інформацію чи об’єднує її;

6) розробка блоку віртуалізації, котрий відтворює поведінку вірусу та складає алгоритм вирішення проблеми та відновлення даних;

7) розробка блоку хмарного обчислення, котрий збирає всю інформацію та аналізує її для побудови сигнатурних баз.

В системі присутній блок віртуалізації обладнання, він здійснює повний контроль та перевірку стану системі і у разі атаки може відхилити запит, або відтворити попередній стан даних, також захистити антивірусну систему від несанкціонованого видалення чи пошкодженню компонентів. Системний блок статистичного аналізу роботи веде безперебійний моніторинг всіх подій та порядку вирішення проблем, що дозволяє побудувати алгоритми та сигнатурні бази на основі котрих можливо припинити розширення вірусів у інших користувачів системи. Завдяки хмарному блоку є можливість об’єднання всіх систем у єдину мережу обробки та операційної віртуалізації для збору та тестування клонів оперативної пам’яті потенційно небезпечних сервісів або програмного забезпечення. У системі існує два основних блоки аналізу подій у оболонці:

1) нейронний блок;

2) блок запитів користувача.

Нейронний блок існує для моніторингу всіх запитів від сервісів та програмного забезпечення. У нейронній мережі можливо представити виконувальні файли як набір карт команд. Нехай набір всіх функцій мови Assembler описується алфавітом P з p символів, p_t – t-й символ даного алфавіту, є певною функцією мови Assembler. Нехай A_{lyz} – граф потоку керування аналізованої програми, складається з n вершин, a_q – q-та вершина в A_{lyz}, q =1…q, S_{ig} – граф потоку керування сигнатури поведінки, складається з m вершин, s_g – будь-яка конкретна вершина графа сигнатури поведінки, g = 1…m , n < m. Нехай a_q, s_g містять k і *l* стандартних функцій мови Assembler, що входять в кожен конкретний список функцій вершини; p, n, m, k, *l* ∈ N, і нехай гомоморфізм розглядається як функція

$$\Psi : S_{ig} \rightarrow A_{lyz},$$

така, що якщо в графі S_{ig} існує шлях, який з'єднує вершини s_i і s_j, то і в графі A_{lyz} існує шлях , котрий з'єднує вершини $\Psi(s_i)$ і $\Psi(s_j)$. Тоді слова wa_q і ws_g визначені як:

$$wa_q = \bigcup_{p(v)\in P(A), v=\overline{1,k}} p(v);$$

$$wa_g = \bigcup_{p(v)\in P(S), v=\overline{1,\ell}} p(v).$$

Нехай p_{sydr} – ймовірність існування гомоморфізму Ψ графа S_{ig} в граф A_{lyz}.

Задаємо f (p_{sydr}, m) як

$$f(p_{sydr}, m) = [cros(wa_q, ws_g)] –$$

функція «схожості» слів wa_q і ws_g, що приймає значення «0» або «1». Якщо DLD – функція, яка обчислюється на підставі алгоритму Дамерау-Левенштейна для всіх вершин графа потоку управління аналізованої програми і графа потоку управління поведінкової сигнатури, то нехай

$$cros(wa_q, ws_g) = DLD(wa_q \cap ws_g).$$

В разі, якщо існує f(p*sydr*, m) вершин $a_q \in A_{lyz}$, для яких існує $s_g \in S_{ig}$, таких, що $cros(wa_q, ws_g) = 1$, то приймається рішення про перевірку існування гомоморфізму Ψ графа S_{ig} в граф A_{lyz}. Всі виявлені вершини a_q з $cros(wa_q, ws_g)=1$ в A_{lyz} позначаються. На підставі алгоритму «пошуку простого шляху» перевіряється, чи є шлях в A_{lyz}, що дозволяє з'єднати вершини a_q графа A_{lyz} з $cros(wa_q, ws_g) = 1$ в тій же послідовності, що і вершини s_g в графі S_{ig}. У разі існування подібного шляху приймається рішення, що код шкідливий [2].

Блок запитів користувача потрібен для відстеження всіх дій користувача, це необхідно для того щоб виключити можливість потенційної помилки користувача або можливості дії програмного забезпечення яке встановлюється користувачем.

Висновки

Запропонована система, дозволяє оцінити ризики та забезпечити відновлення інформації при атаки на користувацьку операційну систему, що може гарантувати безпеку цілісності даних.

Практичне значення розроблюваних методів та засобів є важливим в даному направлені, тому що існує велика кількість загроз втрати інформації. Сучасні антивіруси, як правило реагують на зараження досить швидко однак врятувати втрачену інформацію вже не можливо.

Таким чином, робота має велику актуальність, тому що на даний момент, не існує антивірусної системи, котра містила у собі операційну систему. Даний підхід дозволяє повністю контролювати всі дії та при необхідності блокувати їх, також можливо у режимі використання змінювати операційні файли або будувати апаратні клони чи копії апаратної пам'яті всіх пристроїв.

Особистий внесок в роботу полягає в розробці системи, котра буде містити у собі всю низку сучасних алгоритмічних апаратів для вирішення апаратно-програмних загроз та побудові антивірусної системи.

Список літератури

1. Кнут Е.Д. Мистецтво програмування. Т. 1. Основні алгоритми. – М.: Вид. дім «Вільямс», 2000. – 832 с.

2. Туманов Ю.М. Виявлення шкідливих сценаріїв JavaScript на основі поведінкових сигнатур / Ю.М. Туманов // Безпека інформаційних технологій. – 2009. – № 4. – С. 63-65.

3. Левенштейн В.І. Двійкові коди з виправленням випадінь, вставок і заміщень символів / В.І. Левенштейн // Доповіді АН СРСР. – 1965. – Т. 163, вип. 4. – С. 845-848.

4. Опис Microsoft Azure [Електронний ресурс]. - Режим доступу: https://azure.microsoft.com/ru-ru/overview/what-is-azure/.

5. Касперський К. Записки дослідника комп'ютерних вірусів / К. Касперський. - СПб.: Пітер, 2006. – 316 с.

6. Шелухін О.І. Виявлення вторгнень в комп'ютерні мережі / О.І. Шелухін, Д.Ж. Сакалема, А.С. Філінова. – М.: Гаряча лінія-Телеком, 2013. – 220 с.

7. Семенов. С.Г. Захист даних в комп'ютеризованих керуючих системах (монографія) / С.Г. Семенов, В.В. Давидов, С.Ю. Гавриленко «LAP LAMBERT ACADEMIC PUBLISHING» Німеччина, 2014. – 236 с.

8. Порошин С.М. Розробка і дослідження математичної моделі комп'ютеризованої інформаційно-вимірювальної системи, що управляє критичного застосування з урахуванням фактора зовнішніх впливів / С.М. Порошин, С.Г. Семенов // Системи обробки інформації. – Х.: ХУ ПС, 2013. – Вип. 2 (110). – С. 208-210.

9. *Лукацкий А.В. Выявление атак / А.В. Лукацкий. – СПб.: BXB-Петербург, 2001. – 624 с.*

Надійшла до редколегії 18.05.2016

Рецензент: д-р техн. наук, с.н.с. С.Г. Семенов, Національний технічний університет «ХПІ», Харків.

УСОВЕРШЕНСТВОВАННАЯ МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ СИСТЕМ АНТИВИРУСНОЙ ЗАЩИТЫ

С.Ю. Гавриленко, И.В. Шевердин, Т.Н. Шипова

В статье разработаны шаблоны построения антивирусной системы и описаны пути решения проблематики антивирусной защиты. Обеспечена целостность антивирусных модулей системы и возможность масштабирования конечного продукта. Предложен принципиально новый подход в построении антивирусов, основываясь на использовании гипервизора для компонентной агрегации. Решен вопрос воспроизведения информации после разрушительного действия вирусной атаки. Предложен новый метод картоориентированного описания компьютерных угроз. Описан один из методов построения карт компьютерной угрозы.

Ключевые слова: *компьютерные системы, гипервизор, защита информации, методология проектирования систем антивирусной защиты.*

IMPROVED DESIGN METHODOLOGY ANTIVIRUS PROTECTION SYSTEMS

S.Yu. Gavrylenko, I.V. Sheverdin, T.M. Shipova

The article templates designed building antivirus system and describes solutions to problems of anti-virus protection. Ensuring the integrity of antivirus modules and scalability of the final product. A new approach to building antivirus based on the use of hypervisor component for aggregation. The issue of reproduction of information after the ravages of a virus attack. A new method of describing oriented to the map computer threats. We describe a method of constructing maps of computer threats.

Keywords: *computer systems, hypervisor, information security, systems design methodology antivirus protection.*