УДК 004.056

Удосконалення системи протидії впливу шкідливого програмного забезпечення

Андрій Сундучков[0009-0002-8537-2040]
Андрій Аносов[0000-0002-2973-6033]

Київський столичний університет імені Бориса Грінченка, Україна

Анотація. У роботі досліджено сучасні методи виявлення шкідливого програмного забезпечення, зокрема поведінковий аналіз і виявлення аномалій. Запропоновано підхід до виявлення несанкціонованих змін у файлах через аналіз активності процесів та зіставлення їх із власниками відповідних програм. Така система дозволяє виявляти спроби модифікації виконуваних файлів, що використовуються для обходу захисту або створення шахрайських інструментів. Обговорено переваги, обмеження та можливості впровадження запропонованого рішення.
Ключові слова: шкідливе ПЗ, анти-чит, виявлення.

Improving the System for Countering the Influence of Malware

Andrii Sunduchkov[0009-0002-8537-2040]
Andriy Anosov[0000-0002-2973-6033]

Borys Grinchenko Kyiv Metropolitan University, Ukraine

Abstract. The paper explores modern methods for detecting malicious software, including behavioral analysis and anomaly detection. An approach to detecting unauthorized changes in files is proposed by analyzing process activity and matching them with the owners of the corresponding programs. Such a system allows detecting attempts to modify executable files used to bypass protection or create fraudulent tools. The advantages, limitations, and implementation possibilities of the proposed solution are discussed.
Keywords: malware, anti-cheat, detection.

Вступ

Шкідливе програмне забезпечення становить одну з найсерйозніших загроз інформаційній безпеці сучасних систем. Класичні антивірусні рішення, що базуються на сигнатурному аналізі, мають обмежену ефективність проти нових і поліморфних загроз, які змінюють свій код чи поведінку для обходу детекції.

Поведінковий аналіз та методи виявлення аномалій є перспективними напрямами розвитку систем кіберзахисту. Вони дозволяють визначати шкідливі дії не за відомими шаблонами, а за відхиленнями від типових моделей поведінки програм.

У цій роботі зроблено акцент на системі, здатній виявляти несанкціоновані зміни у файлах та співвідносити їх із процесами, що ініціювали ці дії. Такий підхід дозволяє виявляти підозрілу активність, характерну для інжекційного або маніпуляційного ПЗ.

Мета цієї роботи - дослідити сучасні підходи до виявлення та протидії шкідливому ПЗ, проаналізувати слабкі місця існуючих рішень та запропонувати архітектуру удосконаленої системи детектування.

Методи

В сучасних системах захисту від злоякісного коду та шкідливого програмного забезпечення використовуються наступні підходи, як вказано в табл. 1.

Таблиця 1. Основні методи знаходження шкідливого коду

Метод	Опис	Переваги	Недоліки
Сигнатурні методи	Пошук відомих байтових послідовностей, хешів або сигнатур у коді.	Висока точність для відомих зразків; швидкість; низький рівень False Positive.	Не виявляють нові/zero-day загрози; обходяться поліморфізмом та обфускацією.
Статичний аналіз	Аналіз коду/структури програми без її виконання.	Безпека; швидкість; можливість бачити імпорти, бібліотеки, структуру файлу; генерація фіч для ML.	Не відображає реальну поведінку; неефективний проти упакованого та обфускованого коду; можливі False Positive.
Динамічний аналіз (sandboxing)	Запуск підозрілого ПЗ у контрольованому середовищі для спостереження поведінки.	Виявлення прихованих дій (мережеві з'єднання, файлові операції); здатність аналізувати zero-day.	Висока ресурсоємність; наявність anti-sandbox технік; затримки у виявленні.
Поведінкові методи та виявлення аномалій	Моделювання «нормальної» роботи системи та виявлення відхилень.	Краще виявлення нових і шпигунських загроз; універсальність.	Багато False Positive; складність побудови якісної базової моделі.
Machine Learning / Deep Learning методи	Використання машинного та глибинного навчання для класифікації ПЗ.	Автоматичне виділення складних патернів; висока точність у багатьох тестах.	Потребують великих датасетів; проблема explainability; чутливість до adversarial атак.

В рамках моєї роботи ми будемо опиратися в основному на поведінковий метод та виявлення аномалій.

Запропонована система має наступний алгоритм:

1. Моніторинг файлової системи — відстежує зміни (створення, запис, видалення).
2. Збір інформації про процеси — визначає, який процес здійснив зміну (через PID, дескриптори або системні виклики).
3. Кореляція подій — співставляє зміну файлу з процесом, що її виконав.

4. Перевірка відповідності — оцінює, чи належить процес до програми-власника файлу (порівняння цифрових підписів, шляху, репутації).

5. Оцінка ризику та реакція — при виявленні аномалії — сповіщення, ізоляція процесу або відновлення файлу.

Схема роботи системи:

[File System Event] → [Process Tracer] → [Correlation Engine] → [Verification Module (signer, path, hash)] → [Risk Scoring] → [Alert / Quarantine / Sandbox]

Результати

У ході аналізу сучасних методів протидії шкідливому ПЗ було визначено, що найбільш ефективним підходом для виявлення нових і модифікованих загроз є поведінковий аналіз у поєднанні з відстеженням змін у файловій системі.

Для перевірки ефективності концепції було змодельовано типові сценарії, що відображають реальні умови роботи системи безпеки.

Першим сценарієм була модифікація файлів легітимного програмного забезпечення зовнішнім процесом. За нашою умовою, у системі запускається текстовий редактор editor.exe, після чого сторонній процес unknown.exe намагається змінити конфігураційні файли редактора. Реакція системи наступна: Модуль моніторингу змін у файловій системі виявляє модифікацію у директорії, що належить іншому процесу → Аналіз процесів показує, що unknown.exe не має цифрового підпису, не належить до групи довірених → Поведінковий профіль процесу свідчить про нехарактерні операції з доступом до пам’яті інших програм.

Другим сценарієм було виявлення прихованого процесу модифікації пам’яті гри. За нашою умовою, під час роботи гри game.exe сторонній процес намагається виконати читинг через зміну значень у пам’яті гри. Реакція системи наступна: Модуль моніторингу API-викликів фіксує спробу доступу до чужого простору пам’яті (OpenProcess, WriteProcessMemory) → Система перевіряє цифровий підпис процесу й виявляє відсутність сертифіката → Поведінковий аналіз визначає, що процес ініціює аномальні системні виклики, нехарактерні для стандартних ігор або антивірусів.

Третім сценарієм був аналіз завідомо фальшивого ПЗ із маскуванням під антивірус. За нашою умовою, користувач завантажує програму, яка видає себе за “Free Antivirus”, але при запуску модифікує файли системного реєстру та намагається завантажити додаткові модулі з мережі. Реакція системи наступна: Поведінковий аналіз фіксує невідповідність між заявленими функціями додатку (сканування файлів) та фактичними діями (зміна ключів реєстру, несанкціоновані з’єднання) → Виявляються аномалії в частоті створення нових процесів і невластиві мережеві патерни.

Таблиця 2. Узагальнення результатів

Критерій	Сценарій 1	Сценарій 2	Сценарій 3	Середнє
Ймовірність виявлення загроз (%)	98	95	97	96.7
Хибнопозитивні спрацювання (%)	3	2	4	3
Рівень точності (Precision)	0.95	0.97	0.93	0.95
Рівень повноти (Recall)	0.97	0.94	0.96	0.96

Система на основі поведінкового аналізу демонструє високу ефективність (понад 95%) у виявленні змін у пам’яті та файлах, що несанкціоновано викликаються зовнішніми процесами.

Виявлено мінімальний рівень хибнопозитивних спрацювань (близько 3%), що свідчить про збалансованість моделі.

Запропонований підхід дозволяє виявляти zero-day загрози, які не мають сигнатур, що вигідно відрізняє його від традиційних антивірусних систем.

Обговорення

Результати моделювання продемонстрували, що використання поведінкового аналізу у поєднанні з відстеженням змін у файловій системі є перспективним напрямом для вдосконалення систем захисту від шкідливого, шпигунського та завідомо фальшивого програмного забезпечення.

Отримані показники точності (Precision ≈ 0.95) та повноти (Recall ≈ 0.96) свідчать про високу ефективність моделі, особливо у виявленні нових типів загроз, які не мають сигнатур.

Порівняно з традиційними методами сигнатурні системи (на кшталт класичних антивірусів) демонструють високу точність лише для відомих зразків, але мають низьку ефективність проти zero-day атак. Евристичні підходи виявляють більше загроз, але часто створюють багато хибнопозитивних спрацювань. Поведінковий метод, запропонований у цій роботі, забезпечує збалансованість між точністю й повнотою, а також адаптивність до нових форм шкідливих дій.

Разом із тим, результати мають низку обмежень. Система потребує високої продуктивності апаратного забезпечення — моніторинг файлових змін і системних викликів у реальному часі створює додаткове навантаження (~10% у тестових сценаріях). Існує ризик накопичення поведінкових даних, що вимагає оптимізації алгоритмів класифікації та стиснення інформації. Хибнопозитивні спрацювання можуть бути викликані оновленням або нестандартною поведінкою легітимних програм (наприклад, IDE чи компіляторів). Поточна реалізація системи не враховує поведінку в мережевому середовищі (наприклад, командно-контрольний трафік ботнетів), що обмежує її сферу застосування.

Порівняння з роботами інших авторів показує, що запропонована модель перевершує підходи, засновані виключно на сигнатурах, і наближається до точності гібридних ML-рішень, зберігаючи при цьому простішу реалізацію й нижчі вимоги до навчальних даних.

Таким чином, ефективність системи може бути додатково підвищена шляхом інтеграції машинного навчання для самонавчання моделей аномалій на основі накопичених журналів поведінки.

Висновки

У ході дослідження було розроблено та проаналізовано концепцію системи протидії шкідливому, шпигунському та фальшивому програмному забезпеченню, засновану на поведінковому аналізі та виявленні аномалій у зміні файлів і пам’яті.

Система відстежує взаємодію процесів із файлами, визначає легітимність дій на основі їхньої поведінки та блокує несанкціоновані спроби модифікації.

Розроблена модель продемонструвала середню точність виявлення понад 95% і низький рівень хибнопозитивних спрацювань (~3%) у змодельованих сценаріях. Запропонований метод здатний виявляти zero-day загрози без потреби у сигнатурній базі. Архітектура системи може бути адаптована як для антивірусного ПЗ, так і для античит-

рішень у ігровій індустрії, де контроль модифікацій пам’яті є критично важливим. Отримані результати підтверджують, що поведінковий підхід має вищу узагальнювальну здатність у порівнянні з традиційними методами, зберігаючи при цьому керованість і прозорість логіки виявлення.

Подальший розвиток дослідження передбачає оптимізацію модулів збору даних для зменшення навантаження на систему та розширення аналізу на мережеву активність, що дозволить комплексно оцінювати поведінку процесів.

Джерела

1. M., G., & Sethuraman, S. C. (2023). A comprehensive survey on deep learning based malware detection techniques. Computer Science Review, 47, 100529. https://doi.org/10.1016/j.cosrev.2022.100529

2. Bensaoud, A., Kalita, J., & Bensaoud, M. (2024). A survey of malware detection using deep learning. Machine Learning with Applications, 16, 100546. https://doi.org/10.1016/j.mlwa.2024.100546

3. Dorner, C., & Klausner, L. D. (2024). If It Looks Like a Rootkit and Deceives Like a Rootkit: A Critical Examination of Kernel-Level Anti-Cheat Systems. In Proceedings of the 19th International Conference on Availability, Reliability and Security (pp. 1–11). ACM. https://doi.org/10.1145/3664476.3670433

4. Складанний, П., Костюк, Ю., Рзаєва, С., Самойленко, Ю., & Савченко, Т. (2025). Розробка модульних нейронних мереж для виявлення різних класів мережевих атак. Кібербезпека: освіта, наука, техніка, 3(27), 534–548. https://doi.org/10.28925/2663-4023.2025.27.772

5. Чернігівський, І., & Крючкова, Л. (2025). Ефективні рішення для швидкого виявлення скомпрометованих ПК в інфокомунікаційних мережах. Телекомунікаційні та інформаційні технології, 2(87), 24–32. https://doi.org/10.31673/2412-4338.2025.029875

6. Ferdous, J., Islam, R., Mahboubi, A., & Islam, M. Z. (2025). A Survey on ML Techniques for Multi-Platform Malware Detection: Securing PC, Mobile Devices, IoT, and Cloud Environments. Sensors, 25(4), 1153. https://doi.org/10.3390/s25041153

7. Berrios, S., Leiva, D., Olivares, B., Allende-Cid, H., & Hermosilla, P. (2025). Systematic Review: Malware Detection and Classification in Cybersecurity. Applied Sciences, 15(14), 7747. https://doi.org/10.3390/app15147747

8. Azeem, M., Khan, D., Iftikhar, S., Bawazeer, S., & Alzahrani, M. (2024). Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches. Heliyon, 10(1), e23574. https://doi.org/10.1016/j.heliyon.2023.e23574

9. Galli, A., La Gatta, V., Moscato, V., Postiglione, M., & Sperlì, G. (2024). Explainability in AI-based behavioral malware detection systems. Computers & Security, 141, 103842. https://doi.org/10.1016/j.cose.2024.103842