УДК 004.49

ПЕРСПЕКТИВИ ВИКОРИСТАННЯ НЕЧІТКОГО ХЕШУВАННЯ В АНТИВІРУСНОМУ ЗАХИСТІ

Єремізін Олександр; Стьопочкіна Ірина
Фізико-технічний інститут «НТУУ «КПІ»

PERSPECTIVES OF FUZZY HASHING USE IN ANTIVIRUS PROTECTION

Yeremizin Oleksandr; Stopochkina Iryna
Institute of Physics and Technology «NTUU «KPI»

Анотація. **Запропоновано засоби підвищення ефективності використання функцій нечіткого хешування для завдань, пов'язаних з антивірусним захистом.**
Ключові слова: **Нечітке хешування, антивірусний захист, кластеризація шкідливого ПЗ, поведінковий аналіз**.
Summary: **The means to improve the use of fuzzy hashing functions for tasks associated with antivirus protection are proposed**.
Keywords: **Fuzzy hashing, virus protection, malware clustering, behavioral analysis**.

Вступ

Останнім часом значну увагу привертають функції нечітких обчислень, які традиційно використовувались для пошуку співпадінь у тексті. Перспективним є використання цих функцій в сфері аналізу шкідливого програмного забезпечення та антивірусного захисту. Розвитку набув напрямок алгоритмів нечіткого хешування [1], які мають низку переваг перед криптографічним хешуванням. В [2] розглянуто порівняльний аналіз функцій нечіткого хешування на прикладі ssdeep та більш нової sdhash, розглянуто властивості функцій, які можуть бути використані в завданнях форензики. Однак, використання алгоритмів нечіткого хешування в антивірусному захисті тільки починає розвиватись. Зокрема, в [3] запропоновано підхід до кластеризації зразків з використанням нечіткого хешування.

Згідно з дослідженнями, виконаними в [1] – [3], функціям нечіткого хешування притаманні деякі особливості, чи навіть недоліки, які треба враховувати при розв'язанні задач із їхнім використанням. Тому актуальним питанням залишається вироблення рекомендацій щодо застосування функцій нечіткого хешування в антивірусному програмному забезпеченні, які сприятимуть більш ефективному використанню властивостей цих функцій.

Постановка задачі

В даній роботі виконано порівняльний аналіз різних класів функцій нечіткого хешування, виявлення недоліків, переваг цих класів з точки зору використання в програмних продуктах антивірусного захисту. Реалізовано програмний модуль на основі алгоритму шматкового нечіткого хешування з контекстним пошуком. Виконано обчислювальний експеримент, який засвідчив про наявність деяких особливостей функцій нечіткого хешування. Із врахуванням цих особливостей запропоновано заходи та засоби щодо зменшення та усунення впливу недоліків відповідних функцій на успішне здійснення функцій пошуку та аналізу шкідливого програмного забезпечення.

Класи алгоритмів нечіткого хешування

Наразі можна виділити такі основні категорії алгоритмів нечіткого хешування:

1) шматкове хешування, що реагує на контекст (context-triggered piecewise hashing (CTPH)); приклади таких алгоритмів: ssdeep, FKSum, SimFD;

2) хешування, що базується на блоках (block-based hash); приклади таких алгоритмів: dcfldd, SimHash, sdhash, bbHash, mvHash-B.

Проблеми нечіткого хешування

Визначимо перелік проблем, які можуть впливати на ефективність застосування функцій нечіткого хешування для завдань антивірусного захисту та аналізу зразків шкідливого ПЗ:

- наявність колізій, коли CTPH-підписи двох різних файлів збігаються;
- аналіз маленьких файлів неможливий внаслідок того, що довжина будівельного блока перевищує довжину файла;
- великі файли можуть оброблятись дуже повільно із використанням деяких функцій (час порядку сотень секунд) (наприклад, bbhash обробляє файл порядку 10 Мб за сотні секунд, тоді як ssdeep - за десяті долі секунди);
- ряд алгоритмів нечіткого хешування здійснюють імовірнісний вибір блоків, тому для малих об'ємів даних дають некоректні результати;
- безпосереднє застосування функцій нечіткого хешування до порівняння даних, оброблених кодувальником, пакувальником, не рекомендується внаслідок низького відсотку правильних спрацювань.

Результатом аналізу файлів різного розміру будуть нечіткі підписи різної довжини. І коректне співставлення їх є додатковою задачею.

Вразливості алгоритмів нечіткого хешування

Виробники зловмисного коду можуть протидіяти успішному функціонуванню нечітких функцій хешування при виявленні присутності віруса в деякому файлі шляхом атак типу:

- протидії «чорним» спискам – попередньо застосовуючи різні методи кодування чи обфускації, зловмисник «маскує» шкідливе програмне забезпечення, роблячи його суттєво несхожим на зразок із чорного списку; несхожість тут мається на увазі не лише з точки зору криптографічного хеша, але й нечіткого хеша;
- симуляції елементів «білого» списку – використовуючи значення нечіткого хеша для файла з «білого списка» та роблячи хеш шкідливого файла схожим на нього (відповідним чином модифікуючи файл).

Для протидії атакам першого виду слід вдаватись перед етапом нечіткого аналізу до засобів декодування та деобфускації. Це особливо важливо при аналізі невідомих зразків шкідливого ПЗ, які вивчаються антивірусною компанією вперше.

Для протидії атакам симуляції "білого списку" рекомендується перевірка файлів нечіткими хеш-функціями різних класів, які мають різні принципи генерації підпису, щоб ускладнити можливість імітувати значення хеш-підпису файлу з легітимного списку.

Область застосування

Можна виділити такі основні напрямки використання алгоритмів нечіткого хешування: 1) кластеризація зразків шкідливого ПЗ, 2) класифікація зразків шкідливого ПЗ, 3) виявлення зараження (використання в складі продуктів антивірусного захисту), 4) динамічний аналіз (поведінковий) шкідливого ПЗ.

Слід зауважити, що при аналізі деяких типів файлів функції нечіткого хешування дають вищий відсоток правильних спрацювань (текстові файли, виконувані файли), а для деяких – більший відсоток хибних спрацювань (графічні файли, запаковані файли).

Підвищення ефективності використання функцій нечіткого хешування

При використанні алгоритмів нечіткого хешування в задачах 1) – 4) необхідно визначити характеристики використовуваних функцій нечіткого хешування, такі, як швидкодія та ступінь

коректності (другий показник зумовлений імовірнісним характером алгоритмів).

При інтеграції функцій нечіткого хешування в склад програмних продуктів, які виконують задачі 1), 2) слід надати перевагу тим алгоритмам, які мають більш низький відсоток хибних спрацювань (False Positives, False Negatives), але, можливо, не є настільки швидкодіючими, як їхні менш точні аналоги. Оскільки в умовах великої кількості зразків, що досліджуються антивірусними компаніями, швидкодія є важливою характеристикою, то наявний недолік недостатньої швидкодії можливо компенсувати а) реалізацією цих функцій на низькорівневих мовах програмування, б) розпаралелюванням потоків обробки зразків та виконанням розподілених обчислень, в) застосуванням методів оптимізації обчислень.

При використанні функцій нечіткого хешування в складі програмних продуктів, які виконують завдання 3), доцільно використовувати перевірку нечітким хешуванням після сигнатурного аналізу, при цьому треба зважати на наступні показники досліджуваного файлу: а) наявність додаткової обробки пакувальником, б) тип файлу, в) розмір файлу. Оскільки зловмисний код з одного й того ж самого сімейства вірусів не обов'язково демонструє високий ступінь схожості, то як критерій сигналізації про можливе зараження вірусами можна розцінювати навіть наявність невисоких відсотків співпадіння.

При використанні алгоритмів нечіткого хешування для задач 4) слід комбінувати їх із іншими алгоритмами, зокрема, алгоритмами машинного навчання. Використання нечіткого хешування може бути ефективним для визначення приналежності нових зразків зловмисного ПЗ до певного сімейства.

При дослідженнях, в яких не висуваються вимоги щодо швидкодії, але важливо одержати більш точну оцінку, можливо застосовувати серію досліджень за допомогою різних алгоритмів нечіткого хешування, а результат оцінки приймати зваженим середнім результатів реакції відповідних функцій. Ваги можуть формуватись на основі емпіричних результатів застосування цього виду алгоритмів для даного типу файлів. Іншим варіантом є серія застосувань одного й того самого алгоритму, але з різними показниками (зокрема, можна регулювати величини вхідних блоків), а результат оцінки відсотку співпадіння приймати середнім арифметичним від одержаних результатів. Це дозволить зменшити вплив випадковостей на результат оцінки співпадінь у файлах, які розглядаються.

Ключові параметри та схема застосування

Існує можливість регулювати деякі параметри в нечітких функціях, щоб досягти більш високого відсотку правильних спрацювань.

Серед таких параметрів можна вказати наступні: довжина блоків *l*, на які розбивається файл, поріг *t*, який визначає, чи буде заноситись індекс блоку до файлу із хеш-значенням (для алгоритмів, що базуються на блоках).

Слід зауважити, що довжина блоків впливає на такі аспекти:

- зменшення довжини *l* призводить до зменшення продуктивності алгоритму.
- збільшення довжини *l* призводить до зменшення довжини хеш-підпису, оскільки кожен символ підпису відповідає результатам аналізу кожної наступної підпослідовності довжиною *l* байт.

Основними складовими антивірусного продукту, який працює із використанням нечіткого хешування, можуть бути:

1. модуль сигнатурного аналізу;
2. модуль нечіткого аналізу;
3. модуль діагностики та порівняння результатів нечіткого аналізу;
4. модуль відновлення та лікування.

Бази даних, з якими може взаємодіяти такий продукт:

1. БД сигнатур шкідливого ПЗ;
2. БД нечітких підписів шкідливого ПЗ (їх може бути декілька – для кожної функції може бути своя БД, оскільки

принципи їх генерації для функцій різних класів відрізняються).

Тоді алгоритм застосування антивірусного продукту, який містить модуль перевірки нечіткою хеш- функцією, може складатись із частин.

1. *Підготовчий етап.*

1.1. Складання бази нечітких підписів для зразків зловмисного коду, що демонструють поліморфні властивості.

1.2. Інтеграція нечітких хеш-функцій із програмою-порівнювачем підписів, який може встановлювати входження елементів нечіткого підпису меншого файлу до підпису більшого файлу та виявлення номерів блоків, яким відповідають ідентичні підпослідовності підпису.

1.3. Задання масиву вагів w_{ik} - вагові коефіцієнти, які визначаються заздалегідь експертним методом, що визначають ступінь довіри для даної функції *i* для кожного типу файлів *k*.

2. *Робочий етап.*

2.1. Сигнатурний аналіз. Якщо загрозу виявлено, перехід на п. 2.5.

2.2. Застосування нечіткої функції з параметрами серій 1..n, порівнювач підписів визначає ділянки у вхідному файлі, що продемонстрували схожість з одним із зразків та надає інформацію про відсоток співпадінь *si*. Передача результатів до п. 2.3.

2.3. Обчислення середнього відсотку співпадінь у випадку серії застосувань однієї функції з різними параметрами як

$$s = \left(s_1 + s_2 + \ldots + s_n\right) / n \qquad (1)$$

або у випадку застосування різних функцій

$$s = \left(w_{1k} s_1 + w_{2k} s_2 + \ldots + w_{nk} s_n\right) / n, \qquad (2)$$

причому $w_{1k} + w_{2k} + \ldots + w_{nk} = 1.$

2.4. Якщо $s < s_{крит}$, де $s_{крит}$ – рівень безпеки (знаходиться емпірично), то завершення програми.

2.5. Завершення програми з повідомленням про інфікування, передача керування модулю лікування та відновлення.

Висновки

Аналіз властивостей функцій нечіткого хешування показав, що найбільш ефективно ці функції можуть використовуватись у сферах, де є прийнятним імовірнісний результат оцінки. Можна успішно використовувати ці функції в складі програмного забезпечення, яке призначене для кластеризації зразків шкідливого ПЗ. При використанні цих функцій безпосередньо для задач виявлення шкідливого ПЗ слід комбінувати його із традиційними засобами сигнатурного аналізу.

В роботі запропоновано алгоритм функціонування антивірусного програмного забезпечення, в складі якого використовуються функції нечіткого хешування.

Для реалізації обчислювального експерименту обрано функцію ssdeep, яка належить до класу функцій шматкового хешування і відрізняється швидкодією та широким спектром застосувань. Параметри цієї функції, які піддаються регулюванню – це Minblocksize та Numblockhashes (стандартні параметри розробників алгоритма). Однак, при застосуванні відповідних функцій в складі антивірусного продукту рекомендується застосовувати принаймні дві функції, що належать до різних класів. В даному випадку дію ssdeep можна доповнити, наприклад, sdhash, яка належить до класу блочного хешування.

Перспективою подальших досліджень є підвищення швидкодії функцій нечіткого хешування при обробці великих обсягів даних.

Література

[1] J. Kornblum *Identifying almost identical files using context triggered piecewise hashing.* Digital Investigation, vol. 3S, 2006, pp. 91–97. [Online]. Available: http://www.dfrws.org/2006/proceedings/12-Kornblum.pdf

[2] V. Roussev *An evaluation of forensic similarity hashes.* Digital investigation, vol.8, 2011, p 34-41[Online]. Available: https://www.dfrws.org/2011/proceedings/09-341.pdf

[3] *Experimental Study of Fuzzy hashing in Malware Clustering analysis/* Li P., Sundaramurthy S. C., Bardas A.G. and oth. [Electronic resource].-

[Access mode:] https://www.usenix.org/system/files/conference/cset15/cset15-li.pdf

References

[1] J. Kornblum *Identifying almost identical files using context triggered piecewise hashing.* Digital Investigation, vol. 3S, 2006, pp. 91–97. [Online]. Available: http://www.dfrws.org/2006/proceedings/12-Kornblum.pdf

[2] V Roussev. *An evaluation of forensic similarity hashes.* Digital investigation, vol.8, 2011, p 34-41[Online]. Available: https://www.dfrws.org/2011/proceedings/09-341.pdf

[4] *Experimental Study of Fuzzy hashing in Malware Clustering analysis/* Li P., Sundaramurthy S. C., Bardas A.G. and oth. [Electronic resource].- [Access mode:] https://www.usenix.org/system/files/conference/cset15/cset15-li.pdf

Реферат

Єремізін Олександр, Стьопочкіна Ірина

Перспективи використання нечіткого хешування в антивірусному захисті

У роботі виконано аналіз основних властивостей різних класів функцій нечіткого хешування. Окреслено перелік завдань антивірусного захисту та аналізу шкідливого програмного забезпечення, що можуть бути розв'язані із використанням цих функцій. Вказано основні недоліки, притаманні функціям цього класу, запропоновано заходи та засоби, які доповнюють дію нечіткого хешу та роблять його використання більш ефективним в складі антивірусного програмного забезпечення та програмного забезпечення для аналітики шкідливого програмного забезпечення. Запропоновано склад та алгоритм функціонування антивірусного продукту, який використовує в своєму складі функції нечіткого хешування. Розроблено програмний модуль, який виконує функції нечіткого хеш-аналізу та може бути інтегрований у склад антивірусного програмного забезпечення.

Еремизин Александр, Степочкина Ирина

Перспективы использования нечеткого хеширования в антивирусной защите

В работе выполнен анализ основных свойств разных классов функций нечеткого хеширования. Указан перечень задач антивирусной защиты и анализа вредоносного ПО, которые могут быть решены с использованием этих функций. Указаны основные недостатки, присущие функциям этого класса, предложены средства и меры, которые дополняют действие нечеткого хеша и делают его использование более эффективным в составе антивирусного ПО и ПО для аналитики вредоносного ПО. Предложен состав и алгоритм функционирования антивирусного продукта, который использует в своем составе функции нечеткого хеширования. Разработан программный модуль, который выполняет функции нечеткого хеш-анализа и может быть интегрирован в состав антивирусного программного обеспечения.

Yeremizin Oleksandr, Stopochkina Iryna

Perspectives of fuzzy hashing in antivirus protection

The analysis of the main properties of different classes of fuzzy hashing functions is performed. The tasks for anti-virus protection and analysis of malicious software that can be solved using these functions are outlined. The major disadvantages of fuzzy hashes are analyzed, the measures and tools that complement the fuzzy hash action and make its use more effective are proposed. The use of the fuzzy hashes as part of anti-virus software and malware analysis software is considered. A composition of antivirus product and functioning algorithm for antivirus product that uses in fuzzy hashing functions are proposed. A software module that performs the functions of the fuzzy hash analysis and can be integrated into the anti-virus software is developed.

Відомості про авторів

Стьопочкіна Ірина Валеріївна

Освіта: Вища (2001).

Місце роботи: Кафедра інформаційної безпеки, Фізико-технічний інститут НТУУ «КПІ», к.т.н. (2005).

Область знань: інформаційна безпека.

Наукові інтереси: математичне моделювання, інформаційна безпека.

Email: *Iryna.styopochkina@gmail.com*

Єремізін Олександр Сергійович

Освіта: Вища (2016).

Місце роботи: Кафедра інформаційної безпеки, Фізико-технічний інститут НТУУ «КПІ», к.т.н. (2005).

Область знань: інформаційна безпека.

Наукові інтереси: математичне моделювання, інформаційна безпека.