СЕКЦІЯ 4

СПЕЦІАЛІЗОВАНІ КОМП'ЮТЕРНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

УДК 681.51

І. КУЛЯС, В. СЛОБОДЯН, Ю. ЯКИМЕНКО, Р. ХОМЯК

Західноукраїнський національний університет

МЕТОД ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ З РІЗНИХ ДЖЕРЕЛ ДАНИХ В ОБ'ЄДНАНИЙ СЕМАНТИЧНИЙ ПРОСТІР ДЛЯ АНАЛІЗУ ШКІДЛИВИХ ФАЙЛІВ

Вступ. Щоб ускладнити спроби аналізувати шкідливе програмне забезпечення та створювати сигнатури для ідентифікації вірусів, новітні віруси все частіше використовують поліморфізм та метаморфізм. Це означає, що кількість варіацій у родинах шкідливих програм постійно зростає, що становить серйозну проблему для розробників антивірусних продуктів. Підходи, засновані на пошуку сигнатур у файлах, більше не є ефективними. Їх замінюють динамічним аналізом шкідливого коду в ізольованому середовищі [1], а також використанням різних евристичних методів виявлення [2].

Існує багато підходів до статичного аналізу виконуваних файлів. Найпопулярніші з них: аналіз *n*-грам байтів [3], класифікація типів файлів на основі аналізу частоти байтів [4], аналіз на основі n-грамів опкодов [5,6], аналіз на основі машин станів, що виявляють аномалії у коді [7], виявлення нових шкідливих файлів на основі атрибутів рядків [8, 9], класифікація на основі атрибутів, вилучених із структури файлу PE [10]. Зазначимо, що центральним етапом процесу виявлення шкідливого ПЗ є вибір представлення файлу PE, а також генерація та вибір інформативних ознак. На цьому етапі необхідно отримати максимально повне представлення простору ознак, що сприяє ефективній ідентифікації шкідливих виконуваних файлів.

Мета: Побудова моделі відображення інформації з різних джерел даних в об'єднаний семантичний простір.

1. Побудова базових представлень доменів

В першу чергу для навчання пропонованої моделі необхідно представити виконуваний файл та його поведінку в форматі, з яким зможе працювати нейромережевий кодер. Найбільш очевидний варіант - вектор дійсних чисел фіксованої довжини. Однак пропонована архітектура дозволяє працювати і з більш складними форматами (наприклад, символічні послідовності змінної довжини або двудольні графи з анотуванням вершин). Більш того, допускається побудова різних представлень для кодуючої та декодуючої частин моделі.

У наведених експериментах відображення виконуваного файлу в ознаковий простір представляє собою конкатенацію векторів, що описують різні групи характеристик файлу: структуру заголовка (кількість, розмір і відносний порядок секцій файлу та точки входу, права доступу і тип секцій), безліч статистик від байтової структури (гістограми частоти байт і машинних команд в різних сегментах секцій), а також опис безлічі зустрічаються в тілі файлу рядкових і числових констант.

Для побудови ознакового опису поведінкових логів будується словник найбільш інформативних подій та фрагментів рядкових аргументів (токенів). Для відібраного словника будуються векторні представлення за аналогією з підходом Word2Vec, після чого окремі представлення токенів консолідуються в представлення для подій, а отримані вектори, що описують події, консолідуються в єдине представлення поведінкового лога. Більш детальний опис процесу побудови ознакового опису не наводиться в даній роботі через комерційну таємницю, накладену на опис застосовуваних алгоритмів детектування шкідливих програм.

Слід зазначити, що однаковим векторам можуть відповідати кілька різних файлів або поведінкових логів. Це відбувається, наприклад, тому, що кількість біт, що використовуються в записі вектора ознак, значно менше, ніж кількість змінних біт в структурі типових виконуваних файлів. Таким чином, кожен вектор відповідає підмножині всіх можливих файлів або логів, що також є додатковим джерелом невизначеності при співвідношенні змісту файлів та їхньої поведінки.

2. Побудова моделі відображення та відновлення зі схованого простору

Введемо основні позначення для опису єдиної ймовірнісної моделі.

- X - простір представлень виконуваних файлів ($X \subseteq \mathbb{R}^{d_x}$);
- B - простір уявлень логів поведінки ($B \subseteq \mathbb{R}^{d_b}$);
- Z - загальний семантичний простір ($\mathbb{R}^{d_b}$);
- $\mathcal{P}(\ldots)$ - простір розподілів над відповідним векторним простором;
- $f_X[\theta_X]$: $\mathbb{R}^{d_x} \to \mathcal{P}\ (\mathbb{R}^d)$ - функція, що параметризується, що відображає подання файлу в розподіл над векторами загального семантичного простору;
- $f_b[\theta_B]$: $\mathbb{R}^{d_b} \to \mathcal{P}\ (\mathbb{R}^d)$ - параметризована функція, що відображає представлення поведінкового лога в розподіл над векторами загального семантичного простору.

Для зручності будемо вважати, що функції f_X і f_B завжди повертають нормальний розподіл з діагональною матрицею коваріації.

$$f_X(x) = p(z|x) = \mathcal{N}(z|\mu_x(x), \sum_x(x))$$
$$f_B(b) = p(z|b) = \mathcal{N}(z|\mu_b(x), \sum_b(x))$$

Вважаючи, що над векторами прихованого простору задане апріорне розподілення p(z)= $\mathcal{N}(0_d; I_d)$ ми отримуємо формулу умовного розподілу над виконуваними файлами при відомому семантичному векторі z:

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)} = \frac{f_X(x) \cdot p(x)}{p(z)}$$

Слід зазначити, що першочергово не відомо істинного апріорного розподілу p(x)p(x)p(x), однак, введення параметризованого розподілу $q_x[\phi_X]: z \to \mathcal{P}(x)$, що апроксимує розподіл $p(x|z)$, дозволяє отримати аналог варіаційної нижньої оцінки (Evidence Lower Bound - ELBO) на правдоподібність $\log p(x)$:

$$E_{p(x)} \times \log p(x) = E_{p(z)} KL(p(x|z)||q(x|z) +$$
$$E_{p(x)}\left[E_{p(x|z)} log q(x|z) - KL(p(x|z)||p(z)\right] \quad (1)$$

Причому, $KL(p(x|z)||q(x|z) \geq 0$, а $\left[E_{p(x|z)} log q(x|z) - KL(p(x|z)||p(z)\right] = ELBO^*$.

$$log\ p(x) \geq \int p(z|x) \log q(x|z)\, dz\ -\ KL\left(p(z|x)||p(z)\right) \quad (2)$$

Звернемо увагу на те, що на відміну від методу на основі варіаційного автокодеру, тут робиться припущення про те, що побудована генеративна частина моделі задає істинний розподіл над спостережуваними даними. Навпаки, припускається використання істинного розподілу на приховані змінні і будуємо апроксимацію q для генеративного процесу.

Така інверсія припущень зберігає всі теоретичні гарантії, закладені в оригінальний варіаційний автокодер, і абсолютно ніяк не впливає на процес навчання, але дозволяє уникнути грубого припущення про те, що чесно параметризовується процес генерації таких складних дискретних структур, як виконувані файли, або принаймні їх ознаковий опис.

У запропонованому підході апроксимація $q(x|z)$ повертає вектор одномірних параметризованих розподілів. При цьому різні компоненти загалом можуть бути згенеровані з різних розподілів і мати різну кількість параметрів. Наприклад, частина координат може бути описана гаусовим розподілом, частина завідомо додатніх координат - за допомогою лог-нормального або гамма-розподілу, а деякі бінарні ознаки можуть бути апроксимовані бернуллівською випадковою величиною. У експериментах, наведених у цій роботі, для простоти для всіх ознак використовується апроксимація за допомогою нормального розподілу з параметризованими першим і другим моментом.

Будуючи аналогічну оцінку правдоподібності для поведінкових логів, отримуються 4 навчальні нейронні мережеві моделі:

$$p_{[\theta_x]}(z|x);$$

$$p_{[\phi_x]}(x|z);$$

$$p_{[\theta_b]}(z|b);$$

$$p_{[\phi_b]}(b|z).$$

3. Навчання моделей відображення в загальний простір

Для того щоб відображення файлів і логів в загальному семантичному просторі були узгоджені, скористаємося методом максимізації правдоподібності для наявної навчальної вибірки з пов'язаних пар, що складаються з файлу та отриманого для нього на віртуальній машині поведінкового логу.

$$log\ p(X,B)\ =\ \sum_i \log p(x_i, b_i) \quad (3)$$

$$log\ p(x_i b_i) =\ log\ p(x_i) + log \int \frac{p(z|x_i)p(z|b_i)}{p(z)} dz + log\ p(b_i) \quad (4)$$

У формулі 4 для першого та останнього доданка можна використовувати нижню оцінку правдоподібності відповідно до формули 2. Вираз під інтегралом у другому доданку являє собою експоненту від квадратичної форми. Інтуїтивно, середній доданок відповідає за близькість умовних розподілів над семантичним простором для файлу та відповідного йому лога, а крайні доданки не дозволяють зійтися до тривіального результату, при якому розподіл над латентними змінними не залежить від вхідних даних.

Зазначимо, що описаний процес дозволяє використовувати під час навчання не лише пари узгоджених файлів і логів, але й колекції файлів, для яких не були отримані логи, а також логи, для яких відповідні файли з якихось причин недоступні. Для таких одиничних об'єктів виконується лише максимізація нижньої оцінки правдоподібності $p(x_i)$ або $p(b_i)$ разом із навчанням моделей для пов'язаних пар. Також, ця модель може бути узагальнена на довільну кількість типів даних, для кожного з яких навчається свій варіаційний.

Висновок. В роботі представлений метод відображення інформації з різних джерел даних в об'єднаний семантичний простір, який має великий потенціал як для вирішення широкого спектру завдань комп'ютерної безпеки, так і для завдань з інших галузей. Представлено навчання моделі який враховує не тільки пари узгоджених файлів і логів, але й колекції файлів, для яких не були отримані логи, а також логи, для яких відповідні файли з якихось причин недоступні.

Перелік використаних джерел.

1. Or-Mein O., Nissim N. Dynamic Malware Analysis in the Modern Era - A State of the Art Survey. Ben-Gurion University of the Negev, Beer-Sheva, Israel. ACM Comput. Surv., vol. 52, no. 5, Article 88, 2019. DOI: 10.1145/3329786.

2. Bazrafshan Z., Hashemi H. A survey on heuristic malware detection techniques. 2013 5th Conference on Information and Knowledge Technology (IKT). DOI: 10.1109/IKT.2013.6620049.

3. Tony Abou-Assaleh, Nick Cercone, Vlado Keselj, and Ray Sweidan. Detection of new malicious code using n-grams signatures In Proceedings of Second Annual Conference on Privacy, Security and Trust, pp. 193196, 2004.

4. W. Li, K. Wang, S. Stolfo, B. Herzog. Fileprints: Identifying file types by n-gram analysis. Proc. of the IEEE Workshop on Information Assurance and Security, 2005.

5. D. Bilar. Statistical structures: Fingerprinting malware for classification and analysis. In Blackhat, 2006.

6. I. Santos, F. Brezo, J. Nieves, Y. K. Penya, B. Sanz, C. Laorden, and P. G. Bringas. Opcodesequence-based malware detection, in Proc. 2nd Int. Symp. Eng. Secure Software and Syst. (ESSoS), Pisa, Italy, vol. LNCS 5965, pp. 3543, 2010.

7. R. Sekar, M. Bendre, D. Bollineni, and Bollineni, R. Needham and M. Abadi, Eds. A fast automaton-based method for detecting anomalous program behaviors, in Proc. 2001 IEEE Symp. Security and Privacy, IEEE Comput. Soc., Los Alamitos, CA, USA, 2001, pp. 144155.

8. Schultz, M., Eskin, E., Zadok, F., Stolfo. Data mining methods for detection of new malicious executables. In: Proceedings of the 22nd IEEE Symposium on Security and Privacy, 2001, 3849.

9. Yanfang Ye, Lifei Chen, Dingding Wang, Tao Li, Qingshan Jiang, Min Zhao. SBMDS: an interpretable string-based malware detection system using SVM ensemble with bagging, Journal in Computer Virology, vol. 5, no. 4, pp. 283293, 2009.

10. A. Shabtai, R. Moskovitch, Y. Elovici, C. Glezer. Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey, Information security technical report 14, 2009.