інструментарію; здійснення аналізу ітерацій циклів в реалізації фазера для пошуку аномалій по порушенню строків завершення; кластеризації місць виникнення переривань від таймерів або пристроїв з урахуванням пріоритетів, здатних в більшій мірі впливати на поведінку системи.

Перелік посилань

1. ARINC Industry Activities. ARINC 653P1. Airlines Electronic Engineering Committee (AEEC). 2015-08-21. 285c.

2. IEEE Std 1003.1-2017. Portable Applications Standards Committee System Services Working Group. IEEE Standard for Information Technology Portable Operating System Interface (POSIX). Base Specifications, издание7. Введён 2017. 3951c.

3. ISO 17356-3:2005. ISO/TC 22/SC 31 Data communication. Road vehicles Open interface for embedded automotive applications Part 3: OSEK/VDX Operating System (OS). Введён 2005-11. 61c.

4. Ken Sakamura, TRON ASSOCIATION. μITRON4.0 Specification. Версия 4.03.00. Введён 2007. URL: https://www.tron.org/wp-content/themes/ dp-magjam/pdf/specifications/en_US/WG024-S001-04.03.00_en.pdf (дата звернення: 31.05.19).

Алгоритми побудови та функціонування нейромережевої штучної імунної системи для виявлення шкідливих програм

Мазурок М.В.
Науковий керівник: ктн. доц. Джулій В.М.
Хмельницький національний університет

Розглянемо процеси генерації, навчання, відбору та функціонування імунних детекторів на основі нейронних мереж. Генерується початкова популяція імунних детекторів, кожен з яких являє собою штучну нейронну мережу. Представимо нейромережевий імунний детектор у вигляді чорного ящика, який має n входів і два виходи (рисунок 1).

Вихідні значення детектора формуються після подачі всіх образів на нього відповідно до наступного виразу:

$$Z_1 = \begin{cases} 1, \textit{якщо чистий файл} \\ 0, \textit{інакше} \end{cases}$$

$$Z_2 = \begin{cases} 1, \textit{якщо вірус} \\ 0, \textit{інакше} \end{cases}$$

Рисунок 1 - Нейромережевий імунний детектор

Для коректного функціонування нейромережеві імунні детектори (НІД) повинні пройти процес навчання. Навчальна вибірка формується з чистих файлів (клас чистих програм) і шкідливих програм (клас шкідливих програм). Присутність вірусу або його сигнатури при навчанні дозволяє навченим імунним детекторам знаходити різницю між чистими файлами і комп'ютерними вірусами [1].

Очевидно, що чим більше різноманітних файлів присутні в навчальній вибірці, тим різноманітніше будуть імунні детектори.

Бажано також мати представників всіх типів шкідливих програм - хробаки, троянські програми, макровіруси і т. д. Однак це необов'язкова умова, тому що шкідливі програми структурно (по набору команд) відрізняються від неінфікованих файлів, так як мають на увазі деструктивні функції, що впливає на рішення імунного детектора при скануванні файлу.

Нейронна мережа навчається шляхом навчання з учителем, тобто ми вказуємо штучній нейронній мережі, де дані з чистих файлів, а де – з шкідливих програм [2].

Нехай T – множина чистих файлів, а F – множина шкідливих файлів. З них випадковим чином формується множина вхідних образів для навчання i-го детектора.

$$X_i = \begin{bmatrix} X_i^1 \\ X_i^2 \\ \dots \\ X_i^L \end{bmatrix} = \begin{bmatrix} X_{i1}^1 & X_{i2}^1 & \dots & X_{in}^1 \\ X_{i1}^2 & X_{i2}^2 & \dots & X_{in}^2 \\ \dots & \dots & & \\ X_{i1}^L & X_{i2}^L & \dots & X_{in}^L \end{bmatrix}$$

де L – розмірність навчальної вибірки. Відповідно, множина еталонних образів виглядає наступним чином:

$$l_i = \begin{bmatrix} l_i^1 \\ l_i^2 \\ \dots \\ l_i^L \end{bmatrix} = \begin{bmatrix} l_{i1}^1 & l_{i2}^1 \\ l_{i1}^2 & l_{i2}^2 \\ \dots & \\ l_{i1}^L & l_{i2}^L \end{bmatrix}$$

Еталонні вихідні значення для i-го детектора формуються так:

$$l_{i1}^k = \begin{cases} 1, \textit{ якщо } X_i^k \in T \\ 0. \end{cases}$$

$$l_{i2}^k = \begin{cases} 1, \textit{ якщо } X_i^k \in F \\ 0. \end{cases}$$

Навчання кожного детектора здійснюється з метою мінімізації сумарної квадратичної помилки детектора. Сумарна квадратична помилка i-го детектора визначається наступним чином:

$$E_i = \frac{1}{2} \sum_{k=1}^{L} \sum_{j=1}^{2} (Z_{ij}^k - l_{ij}^k)^2$$

де Z_{ij}^k – значення j-го виходу i-го детектора при подачі на вхід його k-го образу.

Величина сумарної квадратичної помилки характеризує пристосованість детектора до виявлення шкідливих файлів. Чим менше її значення, тим більше пристосованість детектора [3].

Тому величину сумарної квадратичної помилки можна використовувати для відбору кращих детекторів.

Набір навчених нейронних мереж утворює популяцію імунних детекторів, які циркулюють в комп'ютерній системі і виробляють виявлення шкідливих програм. Наявність різноманітних файлів для навчання і елемента випадковості у формуванні вхідних векторів дає можливість отримати велику кількість різних за своєю структурою імунних детекторів.

В процесі сканування невідомого файлу нейронна мережа ідентифікує невідомий образ, в результаті чого імунний детектор приймає рішення про належності файла до класу шкідливих програм або до класу чистих файлів. Загальний алгоритм функціонування нейромережевої імунної системи, можна представити у вигляді наступної послідовності:

1. Генерація початкової популяції імунних детекторів, кожен з яких являє собою штучну нейронну мережу з випадковими синаптичними зв'язками:

$$D = \{D_i, i = \overline{1, r}\}$$

де D_i i-й нейромережевий імунний детектор, r – загальна кількість детекторів.

2. Навчання сформованих імунних нейромережевих детекторів. Навчальна вибірка формується випадковим чином із сукупності чистих файлів (як правило, це різноманітні системні утиліти операційної системи) і з сукупності шкідливих програм або їх сигнатура [4].

3. Відбір (селекція) нейромережевих імунних детекторів на тестовій вибірці. На даній ітерації знищуються ті детектори, які виявилися нездатні до навчання, і детектори, в роботі яких спостерігаються різні недоліки (наприклад, помилкові спрацьовування). Для цього кожен детектор перевіряється на тестовій вибірці. В результаті для кожного детектора визначається значення квадратичної помилки E_i.

Селекція детектора проводиться наступним чином:

$$D_i = \begin{cases} 0, \textit{ якщо } E_i \neq 0, \\ D_i, \textit{ інакше.} \end{cases}$$

де 0 – операція знищення детектора.

4. Кожен детектор наділяється часом життя і випадковим чином вибирає файл для сканування з сукупності файлів, які він не перевіряв.

5. Сканування кожним детектором обраного файлу, в результаті якого визначаються вихідні значення детекторів $Z_{i1}, Z_{i2}, i = 1, r.$

6. Якщо i-й детектор не виявив вірус в сканованому файлі, тобто $Z_{i1} = 1$ та $Z_{i2} = 0$, то він вибирає наступний файл для сканування. Якщо час життя i-го детектора закінчився, то він знищується, замість нього генерується новий
детектор.

7. Якщо i-й детектор виявив вірус в сканованому файлі, тобто $Z_{i1} = 0$ та $Z_{i2} = 1$, то подається сигнал про виявлення шкідливого файлу і здійснюються операції клонування і мутації відповідного детектора.

Операція мутації полягає в додатковому навчанні детекторів-клонів на виявленому шкідливому файлі. Так створюється сукупність детекторів, налаштованих на виявлену шкідливу програму.

8. Відбір клонованих детекторів, які є найбільш пристосованими до виявлення шкідливої програми. Якщо $E_{ij} < E_i$, то детектор пройшов відбір.

Тут E_{ij} – сумарна квадратична помилка j-го клону i-го детектора, яка обчислюється на шкідливому файлі.

9. Детектори-клони здійснюють сканування файлового простору комп'ютерної системи до тих пір, поки не відбудеться знищення всіх проявів шкідливої програми.

10. Формування детекторів імунної пам'яті.

На цій ітерації визначаються нейромережеві імунні детектори, що показали найкращі результати при виявленні присутнього в комп'ютерній системі вірусу. Детектор імунної пам'яті знаходяться в системі досить тривалий час і забезпечують захист від повторного зараження [5].

Особливістю запропонованого алгоритму є те, що кожен нейромережевий імунний детектор є повністю самостійним об'єктом (автономним агентом), тобто сам вибирає собі область сканування. Для цього він отримує список файлів, що зберігаються в просторі пам'яті, і випадковим чином вибирає файл зі списку для його перевірки. Після перевірки одного файлу детектор переходить до наступного файлу, також обраному випадковим чином з існуючого списоку. Сканування файлів нейромережевим імунним детектором триває до тих пір, поки детектор не виявляє шкідливу програму, або до закінчення часу, відведеного для функціонування даного детектора.

Перелік посилань

1. MITRE corporation. Інтернет, режим доступу http://mitre.org
2. CERT. Інтернет, режим доступу http://www.cert.org/, вільний.
3. Stuxnet. Інтернет. режим доступу http://wikipedia.org/wiki/Stuxnet
4. Rootkit.Win32.Stuxnet Інтернет, режим доступу http://www.securelist.com/ru/descriptions/15071647/Rootkit.Win32.Stuxnet.a
5. Вопросы защиты SCADA-систем в 2019 году. Інтернет, режим доступу http://www.securitylab.ru/news/435484.php

Розробка автоматизованої системи «Електронна залікова книжка EZCUSPU»

Максименко А. Г., Максименко Я. А.
Науковий керівник – к. ф.-м. н., Болілий В. О.
Центральноукраїнський державний педагогічний університет
імені Володимира Винниченка

Дані про користувачів та всі дані з начального-виховного процесу зберігаються в БД (базі даних) «EZcuspu» - рисунок 1 (реалізовано за допомогою системи управління базами даних MySQL).