УДК 004.8.565.5 **Терейковський І. А.**

НЕЙРОМЕРЕЖЕВИЙ ПОВЕДІНКОВИЙ АНАЛІЗАТОР АНТИВІРУСНОЇ СИСТЕМИ

Стаття присвячена розробці нейромережевого поведінкового аналізатора антивірусної системи. Розроблено метод визначення вхідних параметрів нейронної мережі, який базується на застосуванні викликів потенційно небезпечних функцій операційної системи. Доведена доцільність використання нейронної мережі типу двошарового персептрону.

Ключові слова: антивірус, двошаровий персептрон, нейронна мережа, поведінковий аналізатор.

Вступ. Однією із основних тенденцій розвитку сучасних систем антивірусного захисту є акцентування уваги на проактивних методах виявлення шкідливого програмного забезпечення (ПЗ) [1, 5, 7]. Основою таких методів є використання поведінкових аналізаторів. В порівнянні з традиційними сигнатурними системами розпізнавання поведінковий аналізатор дозволяє виявляти шкідливі програми до їх внесення в антивірусні бази, що значно підвищує ефективність розпізнавання нових типів шкідливого ПЗ, кількість та небезпечність якого невпинно зростає. Хоча в більшості антивірусних систем задекларовано модуль проактивного захисту, однак їх широкому застосуванню заважає велика кількість хибних спрацювань як типу «хибна тривога», так і типу «пропуск цілі» [4]. Вказані причини визначають загальну проблематику даної статі, яка полягає у вдосконаленні поведінкових аналізаторів систем антивірусного захисту.

Аналіз останніх досліджень та постановка проблеми. Проактивний захист за допомогою поведінкових аналізаторів – технологія, в якій рішення про небезпечність ПЗ характер об'єкта, що перевіряється приймається на основі аналізу виконуваних ним операцій. Перші спроби використання поведінкових аналізаторів відомі ще з середини 90-х років. В таких аналізаторах рішення про заборону або дозвіл виконання програмою потенційно небезпечної дії визначалось користувачем. Це стало основною завадою їх широкому застосуванню, адже «підозрілі» дії характерні і великій кількості звичайних програми. Інтерес до поведінкових аналізаторів відновився після того, як стала зрозумілою неминучість зменшення достовірності сигнатурних методів розпізнавання комп´ютерних вірусів. Підвищити ефективність проактивного захисту пропонувалось шляхом застосування автоматизованої системи розпізнавання вірусів на основі аналізу послідовності дій підконтрольного ПЗ. Для цього практично у всіх антивірусних системах застосовано особливий програмний агент, інтегрований в операційну систему, що аналізує поведінку програм, виявляючи в ній ознаки вірусної поведінки – запис до реєстру, відкриття великої кількості мережевих з'єднань, запис на диск і модифікація важливих файлів, самовільний запуск додатків, блокування роботи тих чи інших утиліт і т. д. Визначені ознаки передаються в блок автоматизованого розпізнавання, методика роботи якого суттєво відрізняється в різних антивірусних системах. При цьому результати досліджень відомих антивірусних систем дозволяють стверджувати, що в основному використовуються наступні методи розпізнавання: шаблонів функціонування, віртуальних машин, довірених додатків, продукційних правил, еврістичних правил. Наведемо коротку характеристику означених методів.

Метод шаблонів функціонування, базується на співставленні дерева функціонування підконтрольного ПЗ з шаблонами поведінки звичайних програм та з шаблонами поведінки вірусів. Недоліки методу пов´язані як з складністю створення означених шаблонів, так і з складністю процесу співвіднесення. Даний метод в основному використовується для аналізу поведінки скриптів і макросів, оскільки відповідні віруси практично завжди виконують ряд однотипних дій.

Метод віртуальних машин, реалізований наприклад в деяких версіях антивірусу Eset Nod32, передбачає запуск програми в обмеженому середовищі віртуальної машини, яка

функціонує на підзахисному комп'ютері з наступним аналізом результатів роботи цієї програми. Якщо результати вказують на небезпеку, то приймається рішення про наявність вірусу (шкідливого ПЗ). В якості недоліків методу вказують його високу ресурсоємність, неможливість виявлення вірусів в реальному масштабі часу та можливість обходу системи розпізнавання. Також відомі факти обходу вірусом обмежень віртуальної машини.

Метод довірених додатків, що використовується наприклад у системі DefenseWall HIPS, поділяє всі додатки на довірені і не довірені. Не довірені додатки запускаються з обмеженими правами на модифікацію критичних системних параметрів у спеціально відведеній для них віртуальній зоні, що відокремлює їх від довірених процесів. Спроби виходу із віртуальної зони розцінюються як порушення. Ще одним прикладом подібного рішення може служити технологія російської компанії Protection Technology, під назвою інтелектуальне управління активністю. По своїй суті це спеціальний монітор призначений для контролю взаємодії між прикладним ПЗ і операційною системою. Даний монітор вбудовується в модулі операційної системи, фіксує всі системні виклики, що проходять через них, і в разі небезпеки блокує їх виконання. Однак для такого методу проактивного захисту важливо визначити не правила блокування, а виключення, щоб дати можливість коректно працювати тим програмам, які в процесі виконання свого виконання повинні звертатись до системних викликів. Тому, крім загальної політики контролю всіх програм, потрібні додаткові профілі для кожної програми окремо.

Метод продукційних правил базується на представлені знань про поведінку вірусу у вигляді конструкції «якщо-то». Разом з простотою та ефективністю даного методу відзначають неможливість формування відповідних правил для великої кількості шкідливого ПЗ.

Методи еврістичних правил, що використовуються в більшості антивірусів, практично не документовані. Однак практичний досвід дозволяє стверджувати, що в їх основі лежить набір окремих досить розрізнених правил, заданих експертами в галузі антивірусного захисту. При цьому навіть по рекламним заявкам, достовірність розпізнавання описаних методу не перевищує 60-70%.

Відповідно [5, 6], підвищити ефективність розпізнавання можливо за рахунок використання штучних нейронних мереж, які вже довели свою ефективність при вирішення задач розпізнавання в різноманітних галузях. Цим і визначається актуальність досліджень в галузі створення нейромережевих методів розпізнавання комп´ютерних вірусів поведінковими аналізаторами. Слід зазначити, що у відкритих джерелах інформації детального опису використання нейронної мережі в поведінковому аналізаторі не знайдено.

Таким чином *метою* даної роботи є визначення принципів створення нейромережевого аналізатора призначеного для розпізнавання комп´ютерних вірусів на основі аналізу функціонування ПЗ.

Основна частина. Загальний метод застосування нейронних мереж в засобах захисту інформації передбачає послідовне виконання наступних етапів: визначення виду та номенклатури вхідних та вихідних параметрів, розробка архітектури, навчання мережі, тестування мережі, уточнення параметрів архітектури [2, 6].

Очевидно, що номенклатура вхідних параметрів нейронної мережі повинна відображати реалізацію небезпечних подій в комп´ютерній системі, пов´язаних з функціонуванням підконтрольного ПЗ. Крім того, вказані події повинні піддаватись реєстрації у реальному масштабі часу. Відповідно [3, 6] такими подіями можуть бути виклики підконтрольним ПЗ системних функцій операційної системи. При цьому, базуючись на нормативному визначенні комп´ютерного вірусу для Windows-подібних операційних систем в першому наближенні можна виділити наступні категорії потенційно небезпечних функцій: управління розділами, управління файлами, роботи з реєстром, використання системної інформації, використання мережевих з´єднань, управління пам´яттю, використання сервісів, управління системою захисту об´єктів. Адже без

використання цих функцій комп´ютерний вірус втратить свою основну властивість – здатність до само розмноження та само розповсюдження. Крім того, без цих функцій потенційна шкода від вірусу також буде досить обмеженою. Для прикладу в табл. 1 фрагментарно наведено список та призначення потенційно небезпечних функцій управління розділами.

Таблиця 1

Потенційно небезпечні функції управління розділами

Ім´я функції	Призначення функції
DeleteVolumeMountPoint	Розмонтовує розділ від вказаної точки монтування розділу
lpszVolumeMountPoint	Адреса рядка, який вказує точку розмонтування розділу
FindFirstVolume	Повертає ім'я розділу на комп'ютері
FindFirstVolumeMountPoint	Повертає ім'я точки монтування розділу на зазначеному комп'ютері
FindFirstVolumeMountPoint	Відкриває дескриптор пошуку точок монтування та повертає інформацію про першу знайдену точку монтування на зазначеному розділі
FindNextVolume	Продовжує пошук розділів, розпочатий викликом функції FindFirstVolume
FindNextVolumeMountPoint	Продовжує пошук точок монтування розділу, розпочатий викликом функції FindFirstVolumeMountPoint
GetDriveType	Визначає тип дискового пристрою
GetLogicalDrives	Отримує бітову маску, що представляє доступні на поточний момент дискові пристрої
GetLogicalDriveStrings	Заповнює буфер рядками, які визначають дійсні пристрою в системі

По оціночним підрахункам для Win32 в номенклатурі вхідних параметрів слід врахувати приблизно 200-300 потенційно небезпечних функцій. Перехопити виклики цих функцій можливо методом зміни точки входу в таблицю імпорту, або методом зміни початкових байт самої функції. Очевидно, що вхідні параметри нейронної мережі, які відповідають реалізації виклику потенційно небезпечної функції будуть бінарними. Якщо функція викликається, то на відповідний вхід мережі подається 1 і 0 в протилежному випадку. Крім того, в номенклатурі вхідних параметрів доцільно відобразити послідовність викликів небезпечних функцій. Однак ця пропозиція потребує доопрацювання.

Виходячи з позицій практичного застосування та мінімізації структури нейронної мережі можна обмежитись одним виходом, величина якого буде сигналізувати про ймовірність розпізнавання вірусу.

Формування номенклатури вхідних та вихідних параметрів дозволяє перейти до визначення архітектури нейронної мережі. В якості основного застосуємо критерій максимізації обчислювальних потужностей мережі. Зазначимо, що на практиці обчислювальні потужності визначаються максимальною кількістю прикладів, яку може запам'ятати мережа для досягнення необхідної достовірності прийняття рішення. Також врахуємо наступні обмеження: мережа може бути попередньо навчена в лабораторних умовах на протязі тривалого терміну, в навчальних прикладах може бути відображений очікуваний вихід мережі, вхідні параметри в навчальних прикладах можуть бути зашумлені, кількість вхідних та вихідних параметрів принципово обмежена, вхідні та вихідні параметри мають числовий характер, розпізнавання повинно відбуватись в реальному масштабі часу, за рахунок оновлення вагових коефіцієнтів синаптичних зв´язків можна відмовитись від донавчання мережі в процесі експлуатації, поставлена задача розпізнавання вірусів відноситься до задач класифікації образів. Відповідно до результатів [6] серед класичних нейромережевих архітектур найбільш повно основному та обмежуючим критеріям вибору відповідає двошаровий персептрон, структура якого показана на рис.1.

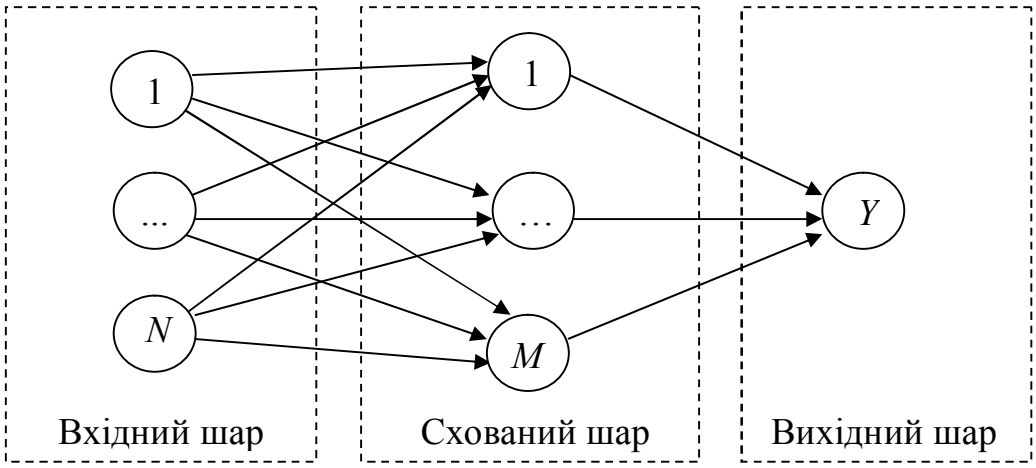

Рис. 1 Структура двошарового персептрону

N – кількість нейронів у вхідному шарі, M – кількість нейронів у схованому шарі.

Використовуючи критерій мінімізації помилки розпізнавання [2, 6] в першому наближені кількість нейронів в схованому шарі можна визначити так

$$M \approx (P/N)^{0,5}, \quad (1)$$

де P – кількість начальних прикладів.

Визначену кількість схованих нейронів слід уточнити, аналізуючи результати розпізнавання навчальної та тестової виборок сформованих з використанням баз даних антивірусних комплексів.

Висновки. Показано, що одним із найбільш перспективних шляхів вдосконалення сучасних антивірусних систем є застосування нейронних мереж для розпізнавання вірусної активності в поведінкових аналізаторах. Розроблено метод визначення вхідних параметрів нейронної мережі, який базується на застосуванні викликів потенційно небезпечних функцій операційної системи. Доведена доцільність використання нейронної мережі типу двошарового персептрону. Наведено методику розрахунку його параметрів.

ЛІТЕРАТУРА

1. Вилков А.С. Информационная безопасность персональных ЭВМ и мониторинг компьютерных сетей / А.С. Вилков. – М. : МИНИТ ФСБ России, 2005. – 210 с.
2. Каллан Р. Основные концепции нейронных сетей / Каллан Р. ; пер. с англ. А. Г. Сивака. – М. : Вильямс, 2003. – 288 с.
3. Корченко О. Г. Шкідливі програми та їх класифікація / О. Г. Корченко, К. П. Ануфрієнко // Защита информации : Сб. науч. трудов. – К. : НАУ, 2007. – С.26–32.
4. Огарок А. Виртуальные войны. Искусственный интеллект на защите от вирусов и программных закладок / А. Огарок, Д. Комашинский, Д. Школьников // Конфидент. – 2003. – №2 (50). – С. 64–69, 97.
5. Петров А. А. Определение оперативно-технических характеристик систем активной защиты информации / А. А. Петров // Захист інформації. – 2009. – № 1 – С. 73–75.
6. Терейковський І. Нейронні мережі в засобах захисту комп'ютерної інформації / І. Терейковський. – К. : ПолиграфКонсалтинг. – 2007. – 209 с.
7. Ян Гордон Компьютерные вирусы без секретов / Я. Гордон – М. : ИПРЖР, 2010. – 340 с.

Надійшла: 02.06.2012

Рецензент: д.т.н., проф. Юдін О.К.