УДК 004.492.3

О.С. САВЕНКО, С.М. ЛИСЕНКО

Хмельницький національний університет, Україна

МОДЕЛЬ ПРОЦЕСУ ПОШУКУ ТРОЯНСЬКИХ ПРОГРАМ В КОМП'ЮТЕРНІЙ СИСТЕМІ

Досліджено типову структуру троянської програми, виділено та формалізовано її структурні складові, проаналізовано технології, що застосовуються при розробці троянських програм. На основі поведінкової моделі запропоновано модель процесу пошуку троянської програми в комп'ютерній системі.

троянська програма, антивірусне забезпечення, руткіт-технологія, поведінкова модель троянської програми, модель процесу пошуку троянської програми

Вступ

Згідно інформації розробників антивірусного програмного забезпечення (ПЗ) та аналітиків шкідливого (вірусного) програмного забезпечення [1] кількість вірусного ПЗ стрімко зростає. Зокрема, спостерігається зростання частки троянських програм (ТП) серед вірусних програм [2]. Аналіз шкідливого ПЗ показує, що його розвиток має певні чіткі тенденції [3]. Зокрема, спостерігається використання троянських програм у комерційних цілях. Розробники шкідливого ПЗ застосовують нові методи проникнення та приховування в системі троянських програм. При цьому троянські програми завдають деструктивні дії комп'ютерній системі (КС) [4-7]. Постає задача розробки нових моделей і підходів пошуку та знешкодження ТП в КС.

1. Постановка задачі

Аналіз методів пошуку та виявлення троянських програм показав, що відомі методи пошуку за сигнатурами, контрольними сумами, криптоаналізом, пошук з використанням евристичного аналізатора та емулятора мають недоліки [2, 3, 8–10]. Жоден із методів не враховує специфіку функціонування ТП та не адаптований до розпізнавання за їх життєвим циклом. Необхідно розробити таку модель процесу пошуку троянських програм, яка б дозволила усунути недоліки існуючих методів та здійснити пошук ТП в КС з урахуванням її поведінки [4, 5].

2. Типова структура троянської програми

Розроблення моделі неможливе без аналізу структури троянської програми. В результаті проведеного дослідження троянських програм на предмет їх структури було виділено п'ять модулів в складі троянської програми.

Основною складовою троянської програми є модуль її проникнення на віддалений комп'ютер. В структурі троянської програми також присутні ще чотири модулі (рис. 1). ТП, що потрапила на ПК, містить механізм реєстрації себе в системі. Велика група троянських програм містять модуль активізації.

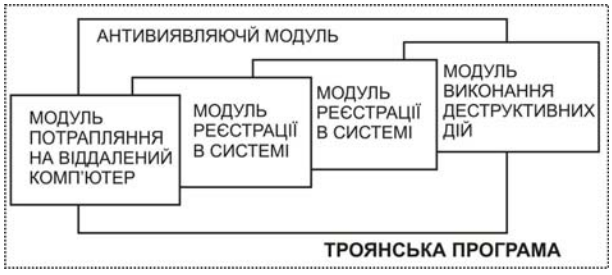

Рис. 1. Структура троянської програми

Кожна ТП має підпрограму деструктивних дій. Це може бути реалізація системи віддаленого керування ПК у мережі, викрадення інформації, інші

шкідливі дії. Особливо небезпечними є ТП, які реалізовані з використанням руткіт-технології [11], що дозволяє зловмиснику приховати сліди своєї діяльності та присутність самої шкідливої програми в системі шляхом заміни системних бібліотек, перехоплення й модифікації низькорівневих API функцій. Антивиявляючий модуль використовує руткіт-технологію.

3. Поведінкова модель троянської програми

Розглянемо поведінкову модель троянської програми [3], яка враховує її життєвий цикл, а саме стани: потрапляння ТП, активізація, виконання закладених функцій. Поведінкова модель троянської програми включає в себе наступні параметри:

1) система $\left(S, V_{MP}, L_{AB}, \longrightarrow\right)$, де S – певні стани ТП на протязі її життєвого циклу, $s \in S$, V_{MP} – матриця відношень механізмів та портів, де $m \in M$ – дії, які реалізують способи та методи потрапляння ТП на ПК, $p \in P$ – мережні протоколи прикладного рівня, через порти яких здійснюється потрапляння ТП на віддалений ПК; L_{AB} – матриця відношень дій ТП та структурних одиниць ОС [12], де $a \in A$ – дії ТП на віддаленому ПК, $b \in B$ – структурні одиниці операційної системи ураженого ПК, які зазнають негативних впливів від ТП та позначення $\longrightarrow$ використовується при формалізації процесів в розглядуваній предметній області, зокрема в [12,13], і є відношенням між трьома поняттями, а саме: якщо $s_i \xrightarrow{a} s_{i+1}$, то дія $a \in A$ спричинює перехід із стану s_i в стан s_{i+1};

2) множина об'єктів E та відношення ε між об'єктами та станами, де для $e \in E$ та $s \in S$, $e\varepsilon s$ означає, що об'єкт e перебуває в стані s; відношення $e\bar{\varepsilon}s$ означає, що об'єкт e не перебуває в стані s;

3) функція Aff, яка визначає дію між двома об'єктами e_i та e_j, множина $a \in Aff\left(e_i, e_j\right)$ є набором можливих дій у такий спосіб, що якщо дія $a \in Aff\left(e_i, e_j\right)$, тоді для усіх станів s з $e_j \varepsilon s$ дія a можлива лише якщо $e_i \varepsilon s$. $Aff\left(e_i, e_j\right)$ є множиною дій, які об'єкт e_j завдає об'єкту e_i;

4) шлях $s_i \xrightarrow{\omega} s_{i+1}$, де $\omega \in A^*$

$$\left(A^* = \left\{a_1, a_2, ..., a_{10}, m_1, m_2, ..., m_5\right\}\right),$$

об'єкт $\upsilon \in E$ з $\upsilon \varepsilon s_i$ та $\upsilon \varepsilon s_{i+1}$, але $\upsilon \bar{\varepsilon} s_0$, де s_0 – початковий стан не активізованої ТП, υ – троянська програма, що відтворюється в моделі.

Виходячи з вищеописаних параметрів, модель поведінки ТП представляється таким чином:

$$s_0 \xrightarrow{V_{MP}} s_1 \xrightarrow{L^1_{AB}} s_2 \xrightarrow{L^2_{AB}} s_3 .$$

4. Модель процесу пошуку троянської програми в комп'ютерній системі

На основі поведінкової моделі троянських програм будуємо модель процесу пошуку ТП в комп'ютерній системі.

Аналіз поведінки троянських програм [2–5] показав, що їх життєвий цикл може бути видозмінений або ж модифікуватися, сама ТП може містити антивиявляючі механізми. Тому при розробленні моделі необхідно враховувати можливі варіації в поведінці ТП при потраплянні на віддалений ПК, містити механізми протидії та виявлення руткіт-технологій. Підхід визначає способи пошуку (режим монітора та режим сканування) та підсистему аналізу і висновку.

В режимі монітора повинно постійно відбуватися відслідковування події в системі, реагувати на програми, дії яких відповідають життєвому циклу ТП. Підсистема слідкування за потоками через порти являє собою монітор проходження інформації через системні порти, а також прослуховування портів відбувається паралельно із моніторингом стану системних функцій, які дозволяють виконати потрапляння ТП на віддалений ПК. Результати такої перевірки передаються на підсистему моніторингу системних бібліотек, яка слідкує за станом бібліотек та модулів в операційній системі, відслідковує ство-

рення нових та модифікацію еталонних. Зібрана інформація передається на вхід підсистеми блокування виконання підозрілих операцій, яка не допускає виконання операцій в системі, які є шкідливими.

Підсистема в режимі сканування в межах моделі процесу пошуку ТП в КС має наступні компоненти: підсистема сканування завантажених файлів з мережі, підсистема сканування об'єктів, що автоматично стартують в системі, та підсистема сканування системних бібліотек на предмет їх перехоплення та вкорінення в них шкідливих фрагментів.

Кожна підсистема передає свої дані на вхід системи аналізу та висновку, яка за наявними правилами та алгоритмами робить результуючий висновок щодо наявності чи підозрілості ТП в КС.

Виходячи із задач кожної підсистеми, представимо модель процесу пошуку ТП в КС (рис. 1) таким чином:

$$M = \left\langle E, S, R, V_{MP}, L^1_{AB}, L^2_{AB}, N, T_{DG}, H_{FL}, \varepsilon, Aff, \longrightarrow \right\rangle$$

де E – множина досліджуваних об'єктів; S – множина станів $s \in S$ об'єкту пошуку на протязі його життєвого циклу; R – результуюче число $R \in [0..1]$, яке свідчить про ступінь підозрілості досліджуваного об'єкту; V_{MP} – матриця відношень дій $m \in M$, які дозволяють здійснити потрапляння об'єкту через системні порти $p \in P$; L^1_{AB} – матриця відношень дій об'єкту $a \in A$ на системні бібліотеки $b \in B$; N – множина завантажених з мережі файлів $n \in N$, які ймовірно можуть бути троянськими програмами; T_{DG} – матриця відношень об'єктів $d \in D$, що автоматично стартують в системі, та їх відповідних функцій них груп (бібліотек, служб, драйверів, сервісів, додатків) $g \in G$; H_{FL} – матриця відношень системних бібліотек $l \in L$ та функцій $f \in F$ закладених у відповідних бібліотеках; аналогічно до поведінкової моделі троянської програми $\longrightarrow$ є відношенням між трьома поняттями, а саме: якщо $s_i \xrightarrow{a} s_{i+1}$, то дія $a \in A$ спричинює перехід об'єкту із стану s_i в стан s_{i+1}; відношення ε між об'єктами та станами, де для $e \in E$ та $s \in S$, $e\varepsilon s$ означає, що об'єкт e перебуває в стані s; відношення $e\bar{\varepsilon}s$ означає, що об'єкт e не перебуває в стані s; функція Aff, яка визначає дію між двома об'єктами e_i та e_j множина $a \in Aff(e_i, e_j)$ є множиною дій, які об'єкт e_j завдає об'єкту , при цьому для усіх станів s з $e_j \varepsilon s$ та $e_i \varepsilon s$; $s_i \xrightarrow{\omega} s_{i+1}$ – шлях проходження об'єкту, $\omega \in A^*$:

$$A^* = \{a_1, a_2, ..., m_1, m_2, ..., d_1, d_2, ..., f_1, f_2, ...\},$$

об'єкт $\upsilon \in E$ з $\upsilon \varepsilon s_i$ та $\upsilon \varepsilon s_{i+1}$, але $\upsilon \bar{\varepsilon} s_0$, де s_0 – початковий стан не активізованої ТП, υ – ТП, яка шукається за допомогою даної моделі.

Отже, шлях пошуку троянської програми в комп'ютерній системі в режимі монітора, матиме вигляд:

$$s_0 \xrightarrow{V_{MP}} s_1 \xrightarrow{L^1_{AB}} s_2 \xrightarrow{L^2_{AB}} s_3 .$$

Шлях пошуку троянської вірусної програми в системі в режимі сканування має вигляд:

$$s_0 \xrightarrow{N} s_1 \xrightarrow{T_{DG}} s_2 \xrightarrow{H_{FL}} s_3 .$$

5. Приклад застосування запропонованої моделі

Розглянемо приклад троянської програми [15] Trojan-Spy.Win32.Delf.pg (троянська програма-шпигун, 26 кб, упакована UPX). Вона містить бібліотеку-перехоплювач із можливістю руткіт-перехоплення системних функцій.

У випадку запуску виконує інсталяцію, що зводиться до наступних дій:

1) створює в папці Program Files\Internet Explorer\ файл IEXPLORE.jmp;

2) створює в папці Program Files\Internet Explorer\ файл IEXPLORE.Dat;

3) створює ключ у реєстрі Software\Microsoft\Windows\CurrentVersion\Explorer\ShellExecuteHooks;

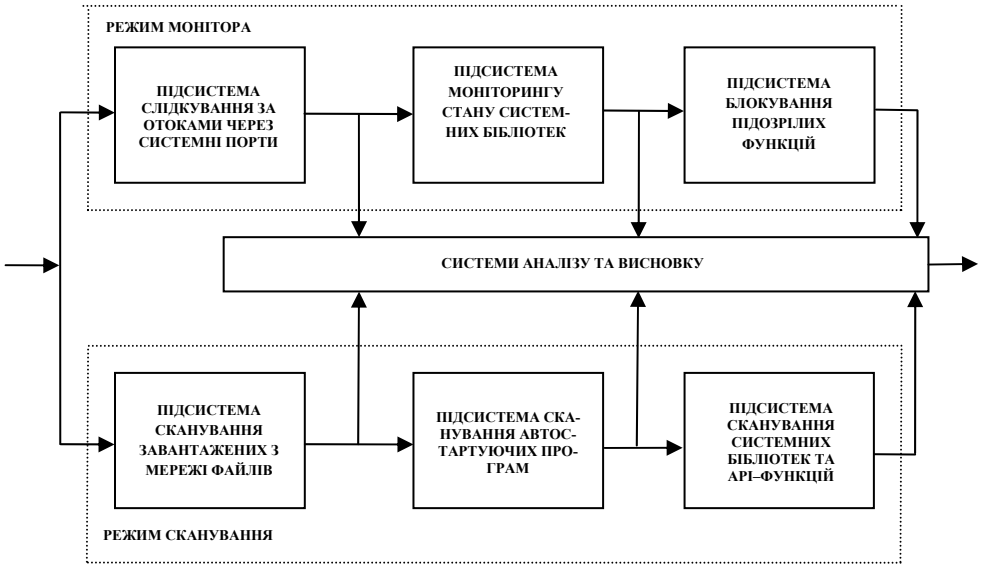

Рис. 2 Модель процесу пошуку троянської програми в комп'ютерній системі

4) створює ключ {FEB94F5A-69F3-4645-8C2B-9E71D270AF2E} ;

5) реєструє CLSID {FEB94F5A-69F3-4645-8C2B-9E71D270AF2E}, бібліотека з кодом класу - IEXPLORE.Dat;

6) виконує впровадження бібліотеки IEXPLORE. Dat у процеси, що запускаються;

7) виконує слідкування за активністю користувача шляхом запису зібраної інформації у файл.

Такий спосіб реєстрації призводить до впровадження бібліотеки IEXPLORE.Dat у процеси, що запускаються.

Проаналізуємо поведінку даної ТП на предмет етапів її життєвого циклу. Для цієї мети згрупуємо команди ТП таким чином, що кожен їх набір спричинюватиме перехід з одного етапу життєвого циклу ТП в інший: дії 1-5 – активізація, 6-7 – виконання закладених функцій. Варто відмітити, що в наявних інструкціях даної ТП немає команд, що реалізують потрапляння на віддалений ПК. Проте, якщо файл потрапить з мережі, то підсистема слідкування за потоками через системні порти обов'язково відмітить появу даного файлу і активізує ланцюг пошуку ТП. На основі результатів такого аналізу відтворимо модель пошуку ТП у двох режимах запропонованого підходу.

Приймемо m_1 – набір функцій роботи з електронною поштою; m_2 – набір функцій роботи з web-браузером; p_{20} – FTP протокол, p_{80} – http-пртокол; a_1 – створення та знищення файлів; a_2 – збір та відправка системної інформації та конфіденційної інформації, a_3 – ведення протоколу (логу) дій користувача з наступною відправкою звітів зловмиснику; a_4 – перехоплення системних функцій, a_5 – модифікація системного реєстру; b_1 – файлова система, b_2 – ядро файлової системи, b_3 – системні служби, b_4 – модулі системи; n – завантажений з мережі файл; d – об'єкт, що автоматично стартує в системі; g_1 – бібліотеки системи, g_2 – служби системи, g_3 – драйвери, g_4 – сервіси (модулі); f_1 – системна бібліотека 1, f_2 – системна бібліотека 2, f_3 – системна бібліотека 3, f_4 – системна бібліотека 4, f_5 – системна бібліотека 5; l_1 – функція перехоплення API-функцій в системі, l_2 – функція модифі-

кації процесів, l_3 – функція модифікації бібліотек, l_4 – функція завантаження бібліотек в адресний простір.

Режим монітора. Було визначено, що $S=\{s_0,s_1,s_2,s_3\}$ – стани життєвого циклу ТП, де s_0 – фактично не є станом життєвого циклу, але необхідний для ймовірного переходу в перший із станів життєвого циклу ТП: потрапляння ТП на ПК. Як було відмічено, ТП не має явної функції, що реалізує механізм потрапляння на віддалений ПК і об'єкт потрапив на ПК через зовнішні носії (флеш-накопичувачі, CD-ROM, floppy тощо), тому $s_0 \notin S$, значення матриці відношень $V_{MP}=0$. За умови потрапляння ТП через, наприклад, веб-браузер, підсистема слідкування за потоками через системні порти активізує наступні підсистеми пошуку. За таких умов стане можливим перехід об'єкту із стану s_0 в стан s_1 отримавши наступне значення матриці відношень V_{MP}:

$$V_{MP}(m_1)=\{p_{80},p_{3128}\}.$$

Дії 1-5 здійснюють активізацію об'єкту і спричинюють його перехід із стану s_1 в стан s_2 і визначають такі значення матриці відношень L^1_{AB}:

$$L^1_{AB}(a_1)=\{b_1\},\ L^1_{AB}(a_5)=\{b_4\};$$

дії 6 та 7 спричинюють перехід об'єкту діагностування в останній етап життєвого циклу ТП – виконання закладених функцій, тобто відбувається перехід із стану s_2 в стан s_3. Значення матриці відношень L^2_{AB}:

$$L^2_{AB}(a_4)=\{b_1\},\ L^2_{AB}(a_2)=\{b_4\}.$$

Набір дій, які завдає об'єкт $\upsilon \in E$ об'єктам b_1,b_4:

$$m_2,a_1,a_2,a_4,a_5 \in Aff(b_1,b_4,\upsilon),$$

$\upsilon \varepsilon s$, тому можливий перехід об'єкта υ із одного стану в інший.

Об'єкт $\upsilon \in E$ з $\upsilon \varepsilon s_i$ ($\upsilon \bar{\varepsilon} s_0$) та $\upsilon \varepsilon s_{i+1}$. Об'єкт υ проходить шлях:

$$s_0 \xrightarrow{V_{MP}} s_1 \xrightarrow{L^1_{AB}} s_2 \xrightarrow{L^2_{AB}} s_3.$$

Система висновку та аналізу при наявних правилах та алгоритмах пошуку визначить досліджуваний об'єкт як троянську програму.

Режим сканування. $S=\{s_0,s_1,s_2,s_3\}$ – стани життєвого циклу ТП, де s_0 – фактично не є станом життєвого циклу, але необхідний для ймовірного переходу в перший із станів життєвого циклу ТП: потрапляння ТП на ПК. ТП не має явного механізму потрапляння на ПК, тому $s_0 \notin S$, $N=\{0\}$.

Дії 1-5 здійснюють активізацію об'єкту і спричинюють його перехід із стану s_1 в стан s_2 і визначають такі значення матриці відношень T_{DG}:

$$T_{DG}(d)=\{g_1\};$$

дії 6 та 7 спричинюють перехід об'єкту діагностування в останній етап життєвого циклу ТП – виконання закладених функцій, тобто відбувається перехід із стану s_2 в стан s_3. Значення матриці відношень H_{FL}:

$$H_{FL}(l_1)=\{f_1,f_2\},\ H_{FL}(l_4)=\{f_1,f_5\}.$$

Дії, які завдає об'єкт $\upsilon \in E$ об'єктам f_1,f_2,f_5:

$$l_1,l_4 \in Aff(f_1,f_5,\upsilon),$$

$\upsilon \varepsilon s$, тому можливий перехід об'єкта υ із одного стану в інший.

Об'єкт $\upsilon \in E$ з $\upsilon \varepsilon s_i$ ($\upsilon \bar{\varepsilon} s_0$) та $\upsilon \varepsilon s_{i+1}$. Об'єкт υ проходить шлях:

$$s_0 \xrightarrow{N} s_1 \xrightarrow{T_{DG}} s_2 \xrightarrow{R_{FL}} s_3.$$

Система висновку та аналізу при наявних правилах та алгоритмах пошуку визначить досліджуваний об'єкт як троянську програму.

Висновок

Аналіз структури та принципів функціонування ТП дав можливість виділити та узагальнити найбільш типові їх складові. На основі отриманих результатів запропоновано модель процесу пошуку ТП в КС. Запропонована модель є основою методу, реалізація якого дозволить враховуватиме поведінкову модель троянської програми.

В подальшому з метою практичної реалізації процесу пошуку необхідно буде деталізувати алгоритми та правила ідентифікації троянських програм в системі аналізу та висновку.

Література

1. ДСТУ 3396.2-97 Державний стандарт України. Захист інформації. Технічний захист інформації. Терміни та визначення.

2. Зайцев О. Rootkits, SpyWare/AdWare, Keyloggers & BackDoors. Обнаружение и защита. – СПб.: БХВ-Петербург, 2006. – 304 с.

3. Erbschloe M. Trojans, worms and spyware. A Computer Security Professional's Guide to Malicious Code. – Burlington: Butterworth–Heinemann USA, 2005. – 212 p.

4. Савенко О.С., Лисенко С.М. Поведінкова модель троянських програм // Тези наукових доповідей міжнародної науково-технічної конференції "Комп'ютерні науки та інформаційні технології" (CSIT-2007). – Львів: Українські технології, 2007. – С. 129-132.

5. Савенко О.С., Лисенко С.М. Процес виявлення троянських програм з використання поведінкової модель троянських програм на основі матриць відношень // Вимірювальна та обчислювальна техніка в технологічних процесах. – Хмельницький, 2007. – № 1. – С. 69-74.

6. Локазюк В.М. Надійність, помилки і тестування програмного забезпечення комп'ютерних пристроїв та систем: Навчальний посібник. – Хмельницький: ТУП, 2003. – 74 с.

7. Локазюк В.М., Савченко Ю. Г. Надійність, контроль, діагностика і модернізація ПК: Посібник. – К.: Видавничий центр "Академія", 2004 – 376 с.

8. Савенко О.С., Лисенко С.М. Дослідження методів антивірусного діагностування комп'ютерних мереж // Вісник Хмельницького національного університету. – 2007. – № 2, т. 2. – С. 120-126.

9. Локазюк В.М., Савенко О.С., Лисенко С.М. Використання поведінкової моделі троянських програм як засобу їх ідентифікації // Прогресивні інформаційні технології в науці та освіті: Збірник наукових праць міжвузівської науково-практичної конференції. – Вінниця: Едельвейс, 2007. – С. 49-54.

10. Локазюк В.М., Савенко О.С. Метод антивірусного комбінованого діагностування персональних ком-п'ютерів // Вісник Технологічного університету Поділля. – Хмельницький: ХНУ, 1999. – № 2. – С. 46-48.

11. RootKit – принципы и механизмы работы [Електронний ресурс]. – Режим доступу: http://www.z-oleg.com/ secur/articles/rootkit.php.

12. Столлингс Вильям. Операционные системы. – М.: Вильямс, 2002. – 848 с.

13. Matt Webster and Grant Malcolm Classification of Computer Viruses Using the Theory of Affordances // 2nd International Workshop on the Theory of Computer Viruses Nancy - France - 10 and 11 may 2007.

14. Matt Webster and Grant Malcolm. Reproducer classification using the theory of affordances. In Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife 2007), pp. 115-122, IEEE Press, 2007.

15. Троян-шпион Trojan-Spy.Win32.Delf.pg [Електронний ресурс]. – Режим доступу: http://www.z-oleg.com/ secur/virlist/vir1164.php.

Надійшла до редакції 29.01.2008

Рецензент: д-р техн. наук, проф. О.В. Поморова, Хмельницький національний університет, Хмельницький.