УДК 621.391 160164
Лисенко С.М., Гайбура Ю.О., Стецюк В.М.
Хмельницький національний університет

МЕТОД ВИЯВЛЕННЯ ТРОЯНСЬКИХ ПРОГРАМ НА ОСНОВІ АПАРАТУ НЕЧІТКОЇ КЛАСТЕРИЗАЦІЇ

Лисенко С.М., Гайбура Ю.О., Стецюк В.М. Метод виявлення троянських програм на основі апарату нечіткої кластеризації. В роботі запропоновано метод виявлення троянських програм на основі апарату нечіткої кластеризації. Метод дозволяє забезпечити реагування на нові троянські програми, забезпечуючи захист комп'ютерних систем від як відомого так і невідомого шкідливого програмного забезпечення. Робота системи виявлення нових троянських програм здійснюється на основі обробки зібраних системних подій в комп'ютерній системі множини ознак, які вказують на присутність троянських програм в комп'ютерній системі. Процес розмежування об'єктів дослідження здійснюється на основі залучення апарату нечіткої кластеризації з частковим навчанням. Проведені експерименти показали, що ефективність виявлення складає 95%, хибні спрацювання 6%.
Ключові слова: троянські програми, кібер-атаки, шкідливе програмне забезпечення, кластеризація, fuzzy c-means.

Лысенко С.М., Гайбура Ю.О., Стецюк В.М. Метод обнаружения троянских программ на основе аппарата нечёткой кластеризации. В работе предложено метод обнаружения троянских программ на основе аппарата нечёткой кластеризации. Метод позволяет обеспечить реагирование на новые троянские программы, обеспечивая защиту компьютерных систем от как известного, так и неизвестного вредоносного программного обеспечения. Работа системы обнаружения новых троянских программ осуществляется на основе обработки собранных системных событий в компьютерной системе множества признаков, которые указывают на присутствие троянских программ в компьютерной системе. Процесс разграничения объектов исследования осуществляется на основе привлечения аппарата нечёткой кластеризации с частичным обучением. Проведенные эксперименты показали, что эффективность обнаружения составляет 95%, ложные срабатывания 6%.
Ключевые слова: кибер-атаки, троянские программы, вредоносное программное обеспечение, кластеризация, fuzzy c-means.

Lysenko S.M., Haibura Y.O., Stetsiuk V.M. The method for Trojan programms detection based on the fuzzy clustering algorithm. An article presents the technique for Trojan progams detection based on the fuzzy clustering algorithm. The method allows providing a response to new Trojan programs, providing security of computer systems from both known and unknown malicious software. The detection of new Trojan programs is performed on base of the system events collection of a set of features that indicate the presence of Trojan programs in the computer system. The process of classification the research objects is implemented using the fuzzy clustering with partial supervision. Experiments showed that detection efficiency is up to 95%, false positives 6%.
Keywords: cyber-attacks, Trojan programs, malicious software, clustering, fuzzy c-means.

Вступ. В наш час число шкідливого програмного забезпечення (ШПЗ), яке здійснює кібер-атаки на комп'ютерні системи (КС) та мережі зростає з великою швидкістю. На сьогоднішній день спостерігаються численні кібер-атаки, кожна з яких відрізняється одна від одної мотивами, та використовує різні стратегії експлуатації комп'ютерні систем. Це робить виявлення та запобігання атак на системи вкрай складним процесом. Шкідливе програмне забезпечення використовується найбільш широко і його популярність в сфері безпеки зростає з кожним днем. Чільне місце серед множини ШПЗ займає клас троянських програм (ТП), який на сьогодні є найчисленнішим. Троянські програми подібні до будь-якої виконуваної програми. Вони маскуються під корисне ПЗ, що здійснює авторизовані дії користувача; однак дії, що вони виконують, надають зловмисникам доступ до експлуатації комп'ютерної системи. Троянська програма в основному надає віддалений доступ до КС, на якому він розгортається. Особливістю троянських програм те, що вони не можуть функціонувати без користувача КС, який надає початковий дозвіл, оскільки це виконуваний файл; його потрібно запустити в системі, щоб він почав працювати. Якщо користувач не запустив виконуваний файл в КС, то зловмисник не зможе отримати доступ до неї [3,9].

Методи виявлення шкідливого програмного забезпечення типу троянські програми поділяються на дві категорії. Метод, що найчастіше використовується для виявлення шкідливого програмного забезпечення в більшості антивірусних програм – технологія на основі сигнатур. Технологія на основі сигнатур неефективна для виявлення троянських програм через їх численну кількість модифікацій та появу нових механізмів проникнення КС [8].

Троянські програми не ідентичні, і досить важко відрізняти потенційно небезпечні файли від ліцензійних. Крім того, кожного дня розробляються різноманітні троянські програми, і їх

сигнатури відрізняються від вже існуючих. Як наслідок, цей метод є неефективним у виявленні троянських програм. Отримання сигнатур всіх троянських програм та постійне оновлення каталогу антивірусних сигнатур водночас і складна, і некерована задача. Більш ефективним методом являється динамічний моніторинг портів, регістрів та файлів конфігурації системи – евристичний підхід [10].

Троянська програма описується як простий виконуваний файл в операційній системі Windows. Однак він має деякі властивості, які дуже вирізняють його від звичайних виконуваних файлів. Можна використати ці властивості, щоб виявити присутність троянських програм. Спочатку ТП непомітно функціонують в системі до того, як активуються клієнтом або віддалено зловмисником. Тоді троянська програма починає виконувати роботу для експлуататора, наприклад, відправляє дані слухачеві, та надає віддалений доступ [14].

Пов'язані роботи. Проблемі виявлення троянських програм приділяється велика увага. Зокрема, в роботі [7] представлений метод виявлення троянських програм шляхом аналізу формату портативного виконуваного файлу, за допомогою якого можна отримати багато корисної інформації. Для того, щоб мати змогу працювати з інформацією, вилученої з портативного виконуваного файлу, метод передбачає побудову дерево рішень на основі алгоритму дерева рішень C5.0. Підхід має два етапи: отримання функції з портативного виконуваного файлу за допомогою фільтру портативних виконуваних атрибутів, та обробка отриманих ознак (за такими функціями) з подальшою побудовою класифікатора для ідентифікації троянських програм. Вихідним в цій роботі є застосування більш ефективного алгоритму C5.0 для побудови дерева рішень. Недоліком даного методу є висока обчислювальна складність у випадку багатомодульності ТП.

В [1] описано метод виявлення ТП з використанням бібліотек динамічних посилань для ідентифікації системних викликів, здійснених троянськими програмами. Розроблена система відслідковування процесів здійснює ідентифікацію шкідливого виконуваного файлу і визначає, чи це дійсно троянська програма, чи ні. Крім того, виконується спроба вивчити поведінку мережі після запуску троянської програми, використовуючи Wireshark. Недоліком даного підходу є нездатність до масштабування у випадку до появи нових ТП.

В роботі [6] представлено підхід до виявлення ТП, суть якого полягає в аналізі поліморфної природи ШПЗ, а також їх поведінки в КС. Метод застосовує евристичний підхід, який реалізується шляхом динамічного оновлення знань про zero-day загрози і побудову баз поведінок. Недоліком методу є необхідність в постійному оновленні баз нових поведінок ТП.

В роботі [12] представлено модель системи виявлення троянських програм, що заснована на аналізі їх поведінок. На відміну від [6], процес виявлення базується на абстрактному описі поведінок троянських програм, а також на визначені правил. Множина таких правил є основою бази даних характеристик поведінок, і евристичного аналізатора для аналізу ПЗ. Експерименти продемонстрували здатність методу до виявлення ТП в реальному часі. Недоліком методу є високий рівень хибних спрацювань.

В статті [15] розглядається проблема виявлення троянських програм і представляється метод виявлення троянських програм. За допомогою цього методу відслідковуються пакети, що передаються комп'ютером в режимі реального часу, ідентифікуються порти, через які передаються пакети. За відомим номером порту та інформацією, яка надається операційною системою, знаходиться процес, який відправляє пакети через порт, і відслідковується програмний файл, який створює процес. Таким чином, зв'язується порт, який використовується для відповідного процесу. Метод дозволяє виявити не лише відомі троянські програми, а й виявити нові ТП.

Вищевказані методи мають спільний недолік щодо високого рівня хибних спрацювань, що пояснюється тим, що ці методи мають евристичну природу.

З огляду на властивості троянських програм, їх різноманітність, нездатність відомих методів ефективно та своєчасно здійснювати їх виявлення зумовлює необхідність розроблення нових методів виявлення троянських програм.

Алгоритми нечіткої кластеризації як засіб виявлення шкідливого програмного забезпечення типу троянські програми. Одним з актуальних напрямків розвитку методів виявлення шкідливого програмного забезпечення сьогодні є залучення апарату інтелектуального аналізу даних.

Кластеризація - це метод інтелектуального аналізу даних, що призначений для виявлення об'єктів в деякому наборі даних шляхом визначення та кількісного визначення подібностей чи відмінностей між об'єктами. Нечітка кластеризація вводить поняття приналежності в розділ даних, оскільки приналежність може вказувати ступінь, в якій об'єкт належить кластеру і краще відображає розділ даних [5].

Одним з механізмів кластеризації, який застосовується до широкого спектру проблем, що пов'язані з аналізом функцій, кластеризацією та класифікацією є fuzzy c-means, заснований на теорії нечіткої кластеризації Руспіні [2, 13]. Цей алгоритм досліджується для оцінки відстаней між різними точками вхідних даних. Кластери формуються залежно від відстаней між точками даних, а кластерні центри формуються окремо для кожного кластера. Метод кластеризації дозволяє одному фрагменту даних належати до більш ніж двох кластерів, що є надзвичайно важливим фактором при виборі інструменту для кластеризації об'єктів, що мають значну схожість.

Таким чином, з огляду на властивості алгоритмів нечіткої кластеризації, вони можуть бути основою для виявлення ШПЗ типу ТП.

Постановка задачі. Отже, актуальною науково-практичною проблемою є виявлення шкідливого програмного забезпечення типу троянські програми, тому необхідно розробити новий метод виявлення троянських програм на основі апарату нечіткої кластеризації.

Метод виявлення троянських програм на основі апарату нечіткої кластеризації. Для виявлення троянських програм запропоновано новий метод, який базується на кластерному аналізі ознак, отриманих шляхом дослідження поведінки троянських програм. Метод використовує нечітку кластеризацію fuzzy c-means з частковим навчанням. Застосування нечіткої кластеризації пояснюється тим, що існують ситуації, коли існує невизначеність щодо приналежності ПЗ до класу шкідливого або корисного. Fuzzy c-means дозволяє мінімізувати таку невизначеність та забезпечити достатню точність результату кластеризації [4, 11].

У якості об'єктів кластеризації є множина векторів ознак, отриманих на основі аналізу поведінки ТП в КС.

Запропонований метод складається з двох етапів:

1. етап навчання системи виявлення троянських програм;
2. етап моніторингу та виявлення троянських програм.

Розглянемо кроки функціонування етапу навчання як складового метода виявлення троянських програм на основі алгоритму нечіткої кластеризації:

1) формування знань щодо поведінки троянських програм в КС на основі відомих ознак та їх проявів;
2) формування множини векторів ознак (поведінок), які описують поведінки троянської програми;
3) здійснення аналізу та поділ троянських програм на класи;
4) створення маркованих даних з отриманих векторів-ознак ШПЗ типу троянські програми з ціллю формування кластерів

Розглянемо кроки функціонування етапу моніторингу та виявлення троянських програм на основі алгоритму нечіткої кластеризації:

1) здійснення відслідковування системних подій в КС (моніторинг мережевого трафіку, відслідковування функціонування програмного забезпечення в операційній системі);
2) побудова вектору ознак, який формується на основі поведінки ПЗ в КС;
3) застосування алгоритму нечіткої кластеризації з частковим навчанням з метою виявлення троянських програм та віднесення їх до певного класу;
4) на основі одержаного результату кластеризації здійснення відповідних дій щодо підозрілого ПЗ.

Загальна схема функціонування методу виявлення троянських програм на основі алгоритму нечіткої кластеризації подана на рисунку 1.

Формування знань щодо поведінки троянських програм в КС на основі відомих ознак та їх проявів. Приймемо множину типів ШПЗ типу троянські програми множиною $B = \{b_1, b_2, ..., b_{16}\}$, де b_1 – Trojan Spy; b_2 - Trojan Dropper; b_3 - Trojan Banker; b_4 - Trojan Downloader; b_5 - PSW Trojans (Password-stealing Trojans); b_6 - Trojan SMS; b_7 - Trojan Ransom; b_8 - Trojan Proxy; b_9 - Trojan Notifier; b_{10} - Trojan Mailfinder; b_{11} - Trojan IM; b_{12} - Trojan GameThief; b_{13} - Trojan FakeAV; b_{14} – Flooder Trojan, b_{15} - Trojan DDoS; b_{16} - Trojan Clicker.

Приймемо множину функцій (API виклики), які застосовуються для функціонування ШПЗ типу троянські програми як $A = \{a_j\}_{j=1}^{N_A}$, де, наприклад, a_1 – функція, що витягує дескриптор модуля для вказаного модуля, a_2 – функція, що отримує адресу функції, що експортується, чи змінної з вказаної бібліотеки динамічних посилань, a_3 – функція, що конвертує рядок в цілочисельне значення, a_4 – функція, що створює потоковий об'єкт, який використовує дескриптор пам'яті для зберігання вмісту потоку, a_5 – функція, що знаходить перше входження підрядка всередині рядка, a_6 – функція, що записує форматовані дані у вказаний буфер, a_7 – функція, що ініціалізує для додатку використання функцій WinHTTP і повертає дескриптор WinHTTP-сесії, a_8 – функція, що отримує повний шлях до файлу, що містить вказаний модуль, a_9 – функція, що завантажує вказаний модуль в адресний простір процесу, що викликається, a_{10} – функція, що виділяє вказану кількість байтів з кучі, a_{11} – функція, що звільнює вказаний локальний об'єкт пам'яті та відміняє його дескриптор, a_{12} – функція, що завершує процес виклику та всі його потоки, N_A – загальна кількість, функцій, використовуваних ТП.

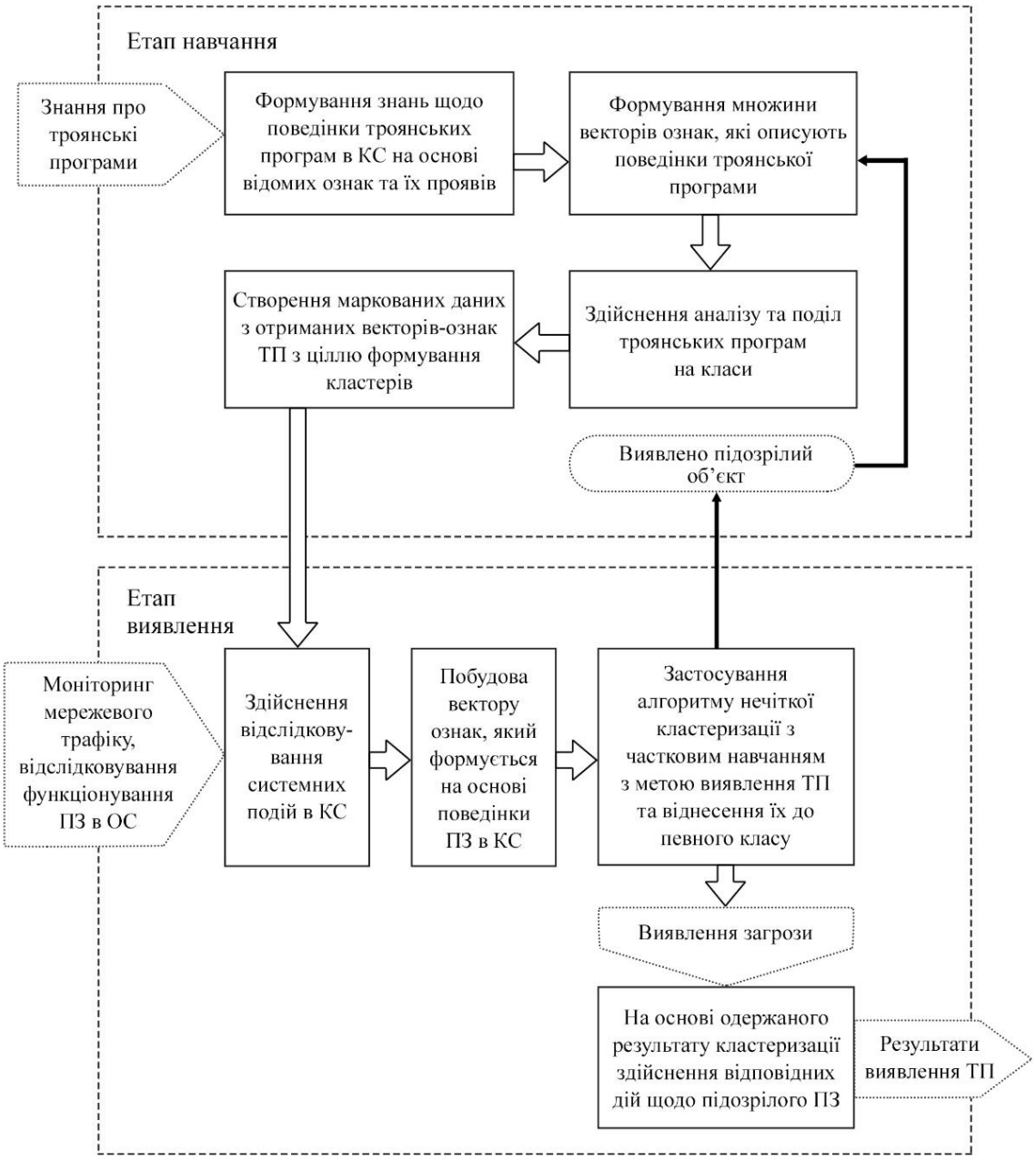

Рис. 1. Загальна схема функціонування методу виявлення троянських програм

Приймемо множину ознак присутності троянських програм в комп'ютерній системі множиною $F = \left\{ f_k b_j \right\}_{k=1,\, j=1}^{N_f}$, де N_f – кількість ознак. Під ознакою присутності ТП в КС вважатимемо сукупність API викликів, виконання яких призводить до деструктивних дій в системі, $a_i \in F$. Приклад, що демонструє перелік системних викликів, що формують поведінку троянської програми типу flooder (Trojan.Flooder.HSF [16]), подано в таблиці 1. Також приймемо множину комп'ютерних систем в мережі, що можуть бути інфікованими ШПЗ типу ТП як $H = \{h_i\}_{i=1}^{N_H}$, h_m, де N_H – кількість КС в мережі. Тоді функція інфікування КС троянською програмою *inf* запишемо таким чином: $\inf : h_m \times f_k \to b_i$.

Формування множини векторів ознак, які описують поведінки троянської програми та здійснення аналізу та поділ троянських програм на класи. Всі вищезгадані функції є основою для побудови множини векторів ознак (поведінок) $X = \{x_k\}_{k=1}^{N_x}$, де кожен вектор ознак x_k описує конкретну поведінку певної ТП, N_x - кількість ознак. Використовуючи отримані ознаки, які представлені як вектори функцій, будується множина правил, які описують поведінку ТП. Побудовані вектори ознак групуються за типами троянських програм, $X \to B$.

Таблиця 1. API виклики, здійснені троянською програмою Trojan.Flooder.HSF, виконання яких призводить до деструктивних дій в системі

Назва модуля	Назва фнккції
KERNEL32.DLL	GetModuleHandleA
KERNEL32.DLL	GetProcAddress
ntdll.dll	_wtoi
ole32.dll	CreateStreamOnHGIobal
SHLWAPI.dll	StrStrA
USER32.dll	wsprintfA
WINHTTP.dll	WinHttpOpen
KERNEL32.DLL	GetModuleFileNameW
KERNEL32.DLL	GetModuleHandleA
KERNEL32.DLL	LoadLibraryA
KERNEL32.DLL	LocalAlloc
KERNEL32.DLL	LocalFree
KERNEL32.DLL	GetModuleFileNameA
KERNEL32.DLL	ExitProcess

Створення маркованих даних з отриманих векторів-ознак ШПЗ типу троянські програми з ціллю формування кластерів. Позначимо *c* як кількість наперед визначених кластерів векторів ознак. Кожному кластеру відповідають троянські програми певного класу та один кластер відповідає корисному ПЗ. Приналежність вектору ознак x_k до і-того кластеру вказує на наявність чи відсутність троянської програми певного типу.

Для того, щоб побудувати прототип кластерів v_i, необхідно побудувати помарковані дані. Помарковані дані засновані на знанні ознак, які можуть вказувати на присутність ШПЗ типу троянські програми як набір векторів-ознак. Кожен вектор ознака x_k помаркованих даних належить одному з наперед визначених кластерів.

В роботі застосована нечітка кластеризація з частковим навчанням, яка заснована на мінімізації наступної цільової функції:

$$J_k = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{p} d_{ik}^{2} + a\sum_{i=1}^{c}\sum_{k=1}^{n} (u_{ik} - f_{ik} b_k)^{p} d_{ik}^{2}, \quad (1)$$

де N – загальна кількість векторів ознак, що підлягають кластеризації (марковані та немарковані вектори-ознак), u_{ik} - значення приналежності для k-того вектору-ознаки в і-тому кластері, f_{ik} - значення приналежності k-того помаркованого вектора-ознаки в і-тому кластері, d_{ik} - відстань між k-тою ознакою об'єкта та прототипом і-того кластеру, $b = [b_k]$ – булевий індикатор, що розрізняє помарковані та немарковані вектори-ознак:

$$b_k = \begin{cases} 1 & if\ \ sample\, x_k\ \ is\ labeled \\ 0 & otherwise \end{cases}. \quad (2)$$

Центроїд (прототип і-того кластеру) та матриця розбиття u_{ij} представлені рівнянням (3):

$$v_i = \frac{\sum_{k=1}^{N} ui_{ik}^{2} x_k}{\sum_{k=1}^{N} ui_{ik}^{2}}, u_{ij} = \frac{1}{1+\alpha}\left\{ \frac{1+\alpha(1-b_{oj}\sum_{l=1}^{c} f_{lj})}{\sum_{l=1}^{c}\left(\frac{d_{ij}}{d_{lj}}\right)^{2}} + \alpha f_{ij} b_{ij} \right\}, \quad (3)$$

де α визначає коефіцієнт масштабування для підтримки балансу між контрольованим та неконтрольованим компонентом в рамках механізму оптимізації [11].

В якості метрики відстані між k-тою ознакою та прототипом кластеру було застосовано відстань Махаланобіса:

$$d_{ik} = \|x_k - v_i\|^{T} A \|x_k - v_i\|, \quad (4)$$

де A – додатньо визначена матриця $R^n \times R^n$.

Здійснення відслідковування системних подій в КС (моніторинг мережевого трафіку, відслідковування функціонування програмного забезпечення в операційній системі). Для динамічного формування векторів поведінок програмного забезпечення в комп'ютерній системі здійснюємо відслідковування системних подій шляхом перехоплення API-викликів програм. API-функції подаємо у вигляді послідовності в тому порядку, в якому кожна функція викликається програмою. Зібрана інформація є вхідними даними для процесу кластеризації.

Побудова вектора ознак, який формується на основі поведінки ПЗ в КС та застосування алгоритму нечіткої кластеризації з частковим навчанням з метою виявлення троянських програм та віднесення їх до певного класу. Функції, що можуть вказувати на присутність ШПЗ типу троянські програми в комп'ютерній системі, витягуються з даних, зібраних на попередньому етапі, і повинні аналізуватись. Результатом аналізу є висновок про наявність чи відсутність атаки троянської програми. В якості засобу класифікації використовується кластеризація fuzzy c-means з частковим навчанням. Об'єктами кластеризації є вектори-ознак x_k, отримані при аналізі корисного навантаження та вихідного трафіку можливого зараження комп'ютерної системи.

Результатом кластеризації є значення приналежності u_{ik} векторів-ознак x_k для кожного кластеру i. Приналежність вектору ознак x_k в і-тому кластері присвоює тип атаки ШПЗ типу троянські програми.

Здійснення відповідних дій щодо підозрілого ПЗ. Якщо виявлено додаток, що відноситься до будь-якого з кластерів, які визначаються ШПЗ типу троянські програми, то такий додаток блокується. Про те, що виявлено ШПЗ типу троянські програми сигналізується користувачеві. Якщо ж додаток віднесено до кластеру, що не визначає ШПЗ типу троянські програми, то він продовжує далі своє виконання.

Експерименти. Для оцінки ефективності запропонованого методу було проведено ряд експериментів. З цією метою було згенеровано по 10 поведінок функціонування ТП кожного типу.

Кожен експеримент тривав 24 годин. Навчальна вибірка містила 15% промаркованих даних. Результати експериментальних досліджень подано в таблиці 2.

Проведені експерименти показали, що ефективність виявлення склала 95%, хибні спрацювання 6%.

Таким чином, запропонований метод продемонстрував можливість виявлення ШПЗ типу троянські програми з високою достовірністю.

Висновки. Запропоновано метод виявлення троянських програм на основі апарату нечіткої кластеризації. Метод дозволяє забезпечити реагування на нові троянські програми, забезпечуючи захист комп'ютерних систем від як відомих так і невідомого шкідливого програмного забезпечення.

Робота системи виявлення нових троянських програм здійснюється на основі обробки зібраних системних подій в комп'ютерній системі множини ознак, які вказують на присутність ТП.

Процес розмежування об'єктів дослідження здійснюється на основі залучення нечіткої кластеризації з частковим навчанням. Проведені експерименти показали, що ефективність виявлення складає 95%, хибні спрацювання 6%.

Таблиця 2. Результати експерименту: виявлені атаки та хибні спрацювання

Типи атак троянських програм	Число атак	Виявлені атаки, %	Хибні спрацювання, %
Trojan Spy	10	10	0
Trojan Dropper	10	9	1
Trojan Banker	10	10	0
Trojan Downloader	10	9	1
PSW Trojan	10	10	0
Trojan SMS	10	10	0
Trojan Ransom	10	10	0
Trojan Proxy	10	10	0
Trojan Notifier	10	9	1
Trojan Mailfinder	10	9	1
Trojan IM	10	10	0
Trojan GameThief	10	10	0
Trojan FakeAV	10	8	2
Flooder Trojan	10	9	1
Trojan DDoS	10	10	0
Trojan Clicker	10	9	1
Total	160	152 (95%)	8 (6%)

1. Abdelshakour A. Detection of Trojan horse by Analysis of System Behavior and Data Packets / A. Abdelshakour, V. K. Gudipati, V. Kumar, A. Vetwal, A. Adeniyi // Systems, Applications and Technology Conference (LISAT). – 2015. – p. 1-4.
2. Bezdek J. C. FCM: The Fuzzy c-means Clustering Algorithm / J. C. Bezdek, R. Ehrlich, W. Full // Computers & Geosciences Vol. 10. – 1984. – p. 191-196.
3. Gandotra E. Malware analysis and classification: A survey / E. Gandotra, D. Bansal, S. Sofat // Journal of Information Security, vol. 5. – 2014. - p. 56-64.
4. Ge L. An APT Trojans Detection Method for Cloud Computing Based on Memory Analysis and FCM / L. Ge, L. Wang, L. Xu // Information Science and Control Engineering (ICISCE), 2016 3rd International Conference. – 2016. – p. 179-180.
5. Giannella C. Spectral malware behavior clustering / C. Giannella, E. Bloedorn // Intelligence and Security Informatics (ISI), 2015 IEEE International Conference. – 2015. - p. 7.
6. Javed A. On the Approach of Static Feature Extraction in Trojans to Combat against Zero-day Threats / A. Javed, M. Akhlaq // IT Convergence and Security (ICITCS), 2014 International Conference on Security. – 2014. – p.1-6.
7. Li Y. Applied-information Technology with Trojan Horse Detection Method Based on C5.0 Decision Tree / Y. Li, Y. Liang, J. Liang // Applied Mechanics and Materials (Volume 540). – 2014. - p. 439.
8. Liu Y. Detecting Trojan horses based on system behavior using machine learning method / Y.Liu, L. Zhang, J.Liang, S.Qu, Z. Ni // Machine Learning and Cybernetics (ICMLC), 2010 International Conference. – 2010. - p. 855-856.
9. Liu Y. Neural Trojans / Y. Liu, Y. Xie, A. Srivastava // IEEE 35th International Conference on Computer Design. – 2017. - p. 45-46.
10. Mukhopadhyay I. Heuristic Intrusion Detection and Prevention System / I. Mukhopadhyay, K. Gupta, D.Sen, P. Gupta // Computing and Communication (IEMCON), 2015 International Conference and Workshop. – 2015. - p. 235-236.
11. Pedrycz W. Fuzzy Clustering with Partial Supervision / W. Pedrycz, J. Waletzky // IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 27. – 1997. – p. 787-791.
12. Qin J. A Trojan horse Detection Technology Based on Behavior Analysis / J. Qin, Q. Si, H. Yan, F. Yan // Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference. – 2010. - p. 2-4.
13. Ruspini E. H. A theory of fuzzy clustering / E. H. Ruspini // Decision and Control including the 16th Symposium on Adaptive Processes and A Special Symposium on Fuzzy Set Theory and Applications, 1977 IEEE Conference. – 1977. – p. 1378-1380.
14. Šťastná J. Exploring Malware Behaviour for Improvement of Malware Signatures / J. Šťastná, M. Tomášek // Scientific Conference on Informatics, 2015 IEEE 13th International. – 2015. - p. 276.
15. Wu N. A Novel Approach to Trojan Horse Detection by Process Tracing / N. Wu, Y. Qian, G. Chen // IEEE International Conference on Networking, Sensing and Control. – 2006. - p. 721-723.
16. Zula Z. Trojan.Flooder.HSF / Z. Zula // Threat Database. – 2015. – режим доступу: https://www.enigmasoftware.com/trojanflooderhsf-removal.