УДК 004.491.42

Савенко О.С.
Хмельницький національний університет

ВИЯВЛЕННЯ БОТ-МЕРЕЖ РОЗПОДІЛЕНИМИ СИСТЕМАМИ НА ОСНОВІ КЛАСИФІКАЦІЇ

У статті представлено типи бот-мереж на основі їх характерних особливостей архітектури, функційного призначення елементів та з урахуванням архітектури засобів виявлення. Це представлення формалізоване до компонент векторів і на його основі здійснено створення зразків для підкласів та класів, які використовуються в баєсівському класифікаторі компонентів бот-мереж. Розроблений метод виявлення бот-мереж на основі класифікатора містить два рівні. Перший (хостовий) рівень при виявленні використовує баєсів класифікатор, що досліджує вектор викликів API функцій. На основі отриманого результату на другому (мережному) рівні здійснюється оцінка ймовірності наявності бот-мереж всією розподіленою системою.

Ключові слова: *розподілена система, зловмисне програмне забезпечення, бот-мережі, мережні антивіруси, наївний баєсів класифікатор, класифікація.*

Постановка проблеми. Тенденції розвитку технологій створення і поширення зловмисного програмного забезпечення (ЗПЗ) демонструють активне розширення технічних можливостей таких засобів. Це досягається шляхом зацікавленості в ньому дедалі більшої кількості користувачів. Основними мотиваційними чинниками, що спонукають до його створення, є бажання досягти фінансової вигоди та політичної переваги. Одним з актуальних напрямів розвитку зловмисного програмного забезпечення є розробка бот-мереж, які надають змогу зловмисникам отримувати віддалений доступ до комп'ютерів користувачів. Завдана використанням такого зловмисного програмного забезпечення шкода стрімко зростає. Тому дослідження бот-мереж із метою розробки нових методів і засобів їх виявлення є актуальною проблемою.

Аналіз останніх досліджень і публікацій. Нині до проблеми виявлення бот-мереж прикута увага як комерційних антивірусних засобів, так і науковців, що ставлять за мету розробку принципово нових підходів та методів до виявлення бот-мереж, зокрема нових, які раніше були невідомими.

Оскільки бот-мережі є керованим розподіленим програмним забезпеченням, його виявлення доцільно проводити розподіленими засобами. До таких засобів належать мережні антивірусні засоби. Мережне антивірусне програмне забезпечення переважно використовується разом із засобами антивірусного захисту вузлів мережі (робочих станцій і серверів) як другий рівень захисту для підвищення достовірності виявлення і блокування зловмисного програмного забезпечення. Якщо на першому рівні захисту атаку не буде виявлено, тоді зберігається ймовірність його виявлення та блокування на другому рівні захисту.

Відомі реалізації таких засобів переважно мають єдиний центр керування з певним рівнем централізації. До таких засобів зараховують: ESET® Endpoint Security для Windows, яка реалізована у вигляді системи захисту кінцевих точок у корпоративних мережах [1], «Dr.Web CureNet!» [2], Symantec Endpoint Protection [3], «Malwarebytes Endpoint Security» [4], «Cisco® Network Admission Control (NAC)» [5]. В антивірусі Kaspersky Administration Kit [6] реалізовано принцип автономної роботи і прийняття рішень без участі адміністратора, якому видаються лише повідомлення про критичні ситуації. Проте, як і у випадку з рештою мережних антивірусних засобів, принцип його функціонування базується на централізованому способі організації взаємодії компонентів системи. Відомі засоби виявлення бот-мереж побудовані на основі методів, які недостатньо враховують всі стадії функціонування бот-мереж та можливі їх структурні особливості, що призводить до зниження достовірності виявлення.

Проаналізуємо відомі методи виявлення бот-мереж. В статті «Botnet evolution: Network traffic indicators» [7] автори здійснюють аналіз та розробку методів виявлення бот-мереж з обов'язковим врахування можливих архітектур бот-мереж. Розробка методів без врахування принципових особливостей архітектур бот-мереж дає змогу зловмисникам обходити засоби, які використовують типові представлення.

Система, яка базується на необхідності врахування особливостей побудови та структури бот-мережі, представлена в роботі «Experiments With Simulation Of Botnets And Defense Agent Teams» [8]. В основу її роботи покладено принцип моделювання агентами архітектур бот-мереж із різними механізмами їх функціонування.

Методи виявлення бот-мереж, представлені в роботах певних науковців [9–13], базуються на аналізі трафіку. Процес виявлення бот-мереж передбачає порівняння отриманих результатів аналізу трафіку з шаблонами бази аномалій.

Недоліком представлених методів є необхідність постійного розбору трафіку та виділення важливих характеристичних ознак, які можуть змінюватись зловмисниками. При цьому не враховано архітектуру бот-мережі і блокування пакетів у подальшому не гарантує їх повторення.

У роботах науковців [14–16] представлено методи виявлення відомих ботів на основі використання сигнатурного аналізу. Вони передбачають контроль кожного пакету, і порівняння їх із попередньо налаштованими сигнатурами і шаблонами атаки, що містяться в базі даних. Спільним недоліком цих методів є необхідність оновлення шаблонів, що впливає на ефективність виявлення нових бот-мереж або їх вузлів, які не представлені у базі сигнатур.

В статті «Multi-agent based approach of botnet detection in computer systems» [17] представлено метод локалізації бот-мереж у корпоративних комп'ютерних мережах, який включає використання мультиагентної системи для обміну інформацією між групами агентів із метою визначення рівня присутності бот-мереж. Розроблений метод базується на централізації та здійсненні перепідключення хостів як засіб прояву присутності бот-мереж.

Автори в статті «Attractiveness Study of Honeypots and Honeynets in Internet Threat Detection» [18] розробили систему приманок для провокування проявів зловмисного програмного забезпечення, яка розміщена в розподіленій системі. Для виявлення нових бот-мереж вона потребуватиме її постійного доповнення.

Відомі методи та засоби не забезпечують високого рівня достовірності у процесі виявлення нових бот-мереж. Це пояснюється застосуванням зловмисниками великої кількості різних технологій та методів приховування наявності та поширення бот-мереж у комп'ютерних системах, а також відставанням розроблених відомих методів їх виявлення у зв'язку з появою нових, раніше не відомих можливостей функціонування бот-мереж. Крім того, мережні антивірусні засоби побудовані переважно як жорстко централізовані, що також використовується зловмисниками для атаки на КС, в яких, ймовірно, знаходиться їх центр.

Тому розробка нових методів і засобів, які б протидіяли бот-мережам і враховували перспективні можливості їх створення, функціонування та архітектуру засобів виявлення, є актуальною проблемою сьогодення.

У статтях «Архітектура розподіленої багаторівневої системи виявлення шкідливого програмного забезпечення в локальних комп'ютерних мережах» та «Distributed Malware Detection System Based on Decentralized Architecture in Local Area Networks» [19; 20] представлена архітектура розподіленої системи виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах, яка є децентралізованою і дає змогу здійснювати її наповнення різними функціоналами виявлення. Розподілена система належить до реагуючих систем, яка постійно здійснює моніторинг запущених процесів та виконуваних програм у комп'ютерних системах мережі. Об'єктами для дослідження зі сторони системи є перевірка наявного програмного забезпечення та запущених процесів у комп'ютерних системах локальної мережі на присутність зловмисного програмного забезпечення.

Виклад основного матеріалу дослідження. Методи та засоби виявлення бот-мереж характеризуються за місцем, в якому здійснюється дослідження програмного забезпечення на наявність зловмисних компонент, та його локацією в межах комп'ютерних систем, які можуть досліджуватись локально чи бути в певному середовищі, що поєднуватиме їх у локальну мережу. Також важливою є орієнтація на час дослідження. Зокрема, виявлення може здійснюватись у момент атаки, після атаки чи після тривалого спостереження. Ці особливості впливають на розробку методів і засобів виявлення бот-мереж. Характерною особливістю бот-мереж, яку треба враховувати під час розробки ефективних методів їх виявлення, є час, який витрачається на наповнення бот-мережі новими вузлами. Водночас чинні вузли, контролюючи свої комп'ютерні системи, перебувають у стадії очікування команд від зловмисника, проводять певні роботи з перевірки своєї цілісності, використовуються зловмисником задля нарощування бот-мережі новими вузлами. Ці особливості стадій функціонування вузлів бот-мереж треба враховувати у процесі розробки методів їх

виявлення, які орієнтовані на тривалий активний моніторинг подій у комп'ютерних системах. Це особливо актуально, коли зловмисне програмне забезпечення проникає в комп'ютерні системи, обминаючи засоби його виявлення, які орієнтовані на атаки чи автономну роботу тільки в окремих комп'ютерних системах.

Враховуючи, що бот-мережі створюються у вигляді розподілених систем, це надає їм перевагу над окремими хостами мережі. Таким чином, ефективні методи та засоби теж мають орієнтуватись на використання їх у мережах, тобто бути розподіленими. Організувати роботу таких розподілених засобів можна в локальних мережах організацій та підприємств. Однією з переваг дослідження виявлення бот-мереж саме в локальних мережах є наявність відомостей про встановлене там програмне забезпечення адміністратором мережі.

Таким чином, розробку методів та засобів виявлення бот-мереж у локальних мережах комп'ютерних систем орієнтуватимемо на їх використання для організацій та підприємств на етапі наповнення бот-мереж новими вузлами.

Розробка методу виявлення бот-мереж потребує здійснення аналізу їх архітектури та можливостей. Нині немає загальноприйнятого чіткого стандарту для бот-мереж стосовно їх структури, класифікації, можливих зловмисних дій. Тому при розробці методів їх виявлення необхідним є здійснення представлення об'єкта дослідження формалізованими загальноприйнятими засобами та його функційних можливостей.

Бот-мережа являє собою комплекс розподіленого програмного забезпечення, який охоплює велику кількість вузлів і містить рівень зв'язуючого програмного забезпечення. За своєю структурою в бот-мережах виділяють вузли, які належать до керування мережею і підтримки її цілісності, та вузли, які є кінцевими і з яких здійснюється виконання зловмисних дій. Керування бот-мережею здійснюється зловмисником через командно-контролюючий центр безпосередньо або через інші проміжні віддалені контролюючі центри. Виділимо такі складники бот-мережі: командно-керуючий центр, контролюючі центри, базові елементи мережі (боти). Зображення узагальненої структури бот-мережі представлено на рис. 1. Командно-контролюючі центри бот-мережі позначимо їх підмножинами E_{1,i_1}, де $i_1 = 1,2,\ldots,n_1$, n_1 – кількість командно-керуючих центрів бот-мережі. Кількість таких центрів може бути різною. Контролюючі центри бот-мережі позначимо підмножинами E_{2,i_2}, де $i_2 = 1,2,\ldots,n_2$, n_2 – кількість контролюючих центрів бот-мережі. Базові елементи бот-мережі позначимо підмножинами E_{3,i_3}, де $i_3 = 1,2,\ldots,n_3$, n_3 – кількість базових елементів мережі.

Вважатимемо, що побудована бот-мережа може тільки розширюватись і при цьому змінювати конфігурування, а зміна її структури здійснюється виключно адміністратором за потреби, а не динамічно за заданими алгоритмами. Бот-мережі можуть мати різну архітектуру залежно від топології і зв'язку елементів: мультисерверну, ієрархічну, випадкову (peer-to-peer), гібридну. Представлена на рис. 1 архітектура є узагальненою і покриває ці топології.

Представимо цілісну бот-мережу, як об'єднання її складових частин, так:

$$E = \bigcup_{i_1=1}^{n_1} E_{1,i_1} \bigcup_{i_2=1}^{n_2} E_{2,i_2} \bigcup_{i_3=1}^{n_3} E_{3,i_3}, \quad (1)$$

де E – множина складових частин бот-мережі. Елементами підмножин $E_{w,i}$ є функції f_{i_1,i_2,i_3}. У різних елементах підмножин $E_{w,i}$ можуть бути

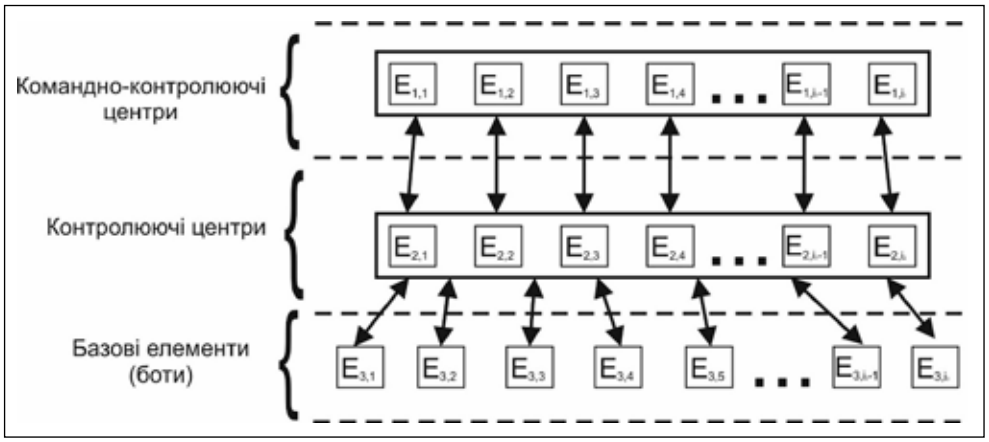

Рис. 1. Структура розподіленої керованої бот-мережі

однакові функції. Функційне навантаження кожної з виокремлених за призначенням функцій залежить від типу операційних систем та їх API функцій відповідно. Представимо функції, що формують функціонування бот-мереж, через API виклики. Для цього виділимо особливі дії, які описуватимуться відповідними API функціями та можуть бути зараховані до зловмисних дій бот-мережі. Такими діями можуть бути: сканування портів, розсилання спаму, завантаження файлів, перегляд директорій, переміщення файлів або директорій, створення перенаправлюваних портів, відкриття файлів, знищення процесу, виконання файлів, запис даних про натиснені клавіші, створення і знищення директорій, перегляд директорій, здійснення атак, створення хибного проксі-сервера тощо.

Для представлення цих дій, які реалізовані у відповідних елементах, введемо вектори зловмисних дій та атак $v_{z,f_{i_1,i_2,i_3}}$, відповідні їм вектори кількості входжень API функцій $v_{API,x,f_{i_1,i_2,i_3},m_u,l}$ та вектори, ймовірно, зловмисних дій $v_{p,e_s}=\left(v_{p,e_s,1},v_{p,e_s,2},\ldots,v_{p,e_s,n_{v_p}}\right)$, відповідні їм вектори кількості входжень API функцій $v_{p,K,e_s}=\left(v_{p,K,e_s,1},v_{p,K,e_s,2},\ldots,v_{p,K,e_s,n_{v_p}}\right)$, де z – позначення зловмисної дії та атаки, x – номер операційної системи, p – позначення ймовірно зловмисної дії, m_u – кількість компонент вектора, l – кількість варіацій, e_s – номер вектора, e – число, що відповідає номеру вектора, який сформовано з певної кількості або всіх API функцій досліджуваного процесу, s – номер варіанта послідовності API функцій досліджуваного процесу вектора для e процесу, n_{v_p} – кількість компонент вектора v_{p,e_s}. Компонентами вектора v_{p,e_s} є номери API функцій, які отримуються із загальної нумерації для всіх таких функцій і функцій для бот-мереж. Згідно з класифікатором необхідно встановити належність вектора v_{p,e_s} до одного з класів бот-мереж.

На основі загальної структури (рис. 1) бот-мережі, яка наповнена функціями і їх представленнями через вектори зловмисних дій та атак, задамо еталонну модель бот-мережі і зарахуємо її до класу ‘0’. Відомі типи бот-мереж, які класифіковано, можуть мати менше функційних можливостей, але при цьому обов’язково певні їх функції проявлятимуться частіше, на основі чого і було їх класифіковано. Відомі типи бот-мереж зарахуємо до класів ‘1’–‘6’ [17]. З появою нових типів бот-мереж кількість класів може збільшуватись. Векторів зловмисних дій атак для однієї і тієї самої функції може бути кілька, що означає варіацію представлення. Такі вектори будуть містити набори API функцій, які найчастіше викликатимуться для виконання певних визначених функцій і, відповідно, характеризуватимуть класи бот-мереж. Враховуючи, що у вектор зловмисних дій та атак будуть збиратись API функції певної функції вузла бот-мережі, вони не можуть характеризувати і порівнюватись з усім класом. Заради цього в кожному класі виділимо підкласи, які відповідатимуть відокремленим елементам.

Для зарахування отриманого вектора ймовірно зловмисних дій до зловмисного ПЗ або корисного здійснимо класифікацію за введеними класами типів бот-мереж. Враховуючи показники швидкості, які необхідно забезпечити задля отримання результатів програмним модулем (ПМ) розподіленої багаторівневої системи (РБС) [19; 20], виберемо наївний баєсівський класифікатор.

Нехай

$$V_p=\left\{v_{p,1},v_{p,2},\ldots,v_{p,n_{V_p}}\right\}$$

– вибірка, сформована на основі значень API функцій для векторів типу v_{p,e_s}, А – гіпотеза про належність значень V_p до одного з класів K_l бот-мереж, де l=0,1,…,6. Для вирішення задачі класифікації необхідно визначити ймовірність того, що вибірка V_p належить класу K_l, враховуючи знання про опис атрибутів V_p. З цією метою, використовуючи теорему Баєса, визначимо апостеріорну ймовірність P(A | V_p), тобто ймовірність того, що значення А залежить від певних атрибутів вибірки V_p.

Класи бот-мереж K_l визначені і представлені сукупністю пар двох векторів $v_{z,f_{i_1,i_2,i_3}}$; $v_{API,x,f_{i_1,i_2,i_3},m_u,l}$. Вибірка V_p належить до класу K_l з найвищою апостеріорною ймовірністю тоді і тільки тоді, коли виконується умова:

$$P\left(K_{l_1}\mid V_p\right)>P\left(K_{l_2}\mid V_p\right), \quad (3)$$

для всіх l_1 та l_2 таких, що $0\le l_1\le 6, 0\le l_2\le 6$, $l_1\neq l_2$. Таким чином, здійснюємо пошук класу, який максимізує ймовірність $P(K_l \mid V_p)$. Тоді такий клас $P(K_l \mid V_p)$ буде класом із максимальною апостеріорною гіпотезою. Для зарахування вектора до певного класу знайдемо добуток ймовірностей тих API функцій, які увійшли у вектор ймовірно підозрілих дій. Використовувана мультиномінальна генеративна модель враховує кількість повторень API функцій і не враховує відсутність певних API функцій. З метою здійснення пошуку в підкласах класу у застосуємо формулу 4:

$$P\left(K_{y,g}\mid v_{p,e_s}\right)=\frac{\prod_{j=1}^{m_{u,y}}\frac{v_{API,x,w,f_{i_1,i_2,i_3},g,l}}{l_{m_{u,y}}}*\frac{l_{m_{u,y}}}{\sum_{g=1}^{m_{u,y}}l_{g,y}}}{\prod_{g=1}^{n_{v_p}}\frac{v_{p,K,e_S,w^{+1}}}{n_{v_p}+1}}, \quad (4)$$

Задля проведення навчання необхідно визначити ймовірності $P\left(v_{z,z_v,f_{i_1,i_2,i_3}}\mid K_{y,g}\right)$. Обчислимо

оптимальні оцінки ймовірностей того, що певна API функція зустрінеться в кожному класі чи підкласі, згладивши результат за схемою Лапласа задля уникнення проблеми «нульової частоти»:

$$P\left(v_{z,d,f_{i_1,i_2,i_3}} \mid K_{y,g}\right) = \frac{1+\sum_{b=1}^{m_{u,y}} t_{g,y}\, v_{API,x,b,f_{i_1,i_2,i_3},g,l}\, P\left(K_{y,g} \mid v_{z,b,f_{i_1,i_2,i_3}}\right)}{n_{m_u}+\sum_{d=1}^{n_{m_u}} \sum_{b=1}^{m_{u,y}} t_{g,y}\, v_{API,x,d,f_{i_1,i_2,i_3},g,l}\, P\left(K_{y,g} \mid v_{z,b,f_{i_1,i_2,i_3}}\right)}, \quad (5)$$

Навчання баєсівського класифікатора подамо у вигляді такої послідовності кроків:

1) визначимо підкласи, що складаються з одного представлення API функцій, та розрахуємо ймовірності для кожної API функції, що входять до заданого класу та підкласу; зафіксуємо отриманні результати як первинні.

2) для кожної відомої наступної варіації компоненти функції, що представлена API функціями, здійснимо класифікацію для класів і підкласів; якщо отримане представлення зараховане вірно до вказаного підкласу, тоді додаємо його марковані елементи до тих, що вже є у підкласі, як окремий зразок; якщо отриманий результат не зараховує його до потрібного підкласу, тоді треба зарахувати його до його підкласу, але при цьому зробити порівняння з іншими значеннями класів; результатом порівняння буде відхилення, тобто різниця, ймовірностей від початкових; якщо результуюча ймовірність суттєво (більше встановленого порогового значення) відрізняється від первинної ймовірності підкласу чи класу, тоді створюємо окремий підклас цього підкласу класу; фіксуємо для кожного кроку навчання отримані ймовірності для кожної API функції, підкласів та класів; для тих підкласів і класів, де розбіжність із первинними становить більше порогового значення, здійснюємо створення нового підкласу в його підкласі; отримані за кілька ітерацій всі ймовірності усереднюємо та фіксуємо як відповідні ймовірності для використання в подальших розрахунках;

3) після завершення основного етапу навчання та доповнення баєсівського класифікатора новими даними здійснимо перевірку відхилень для додатково сформованих підкласів певних підкласів і встановимо розбіжність між їхніми середніми значеннями ймовірностей; якщо розбіжність становить менше порогового значення, тоді доповнюємо підклас даними додаткового підкласу, його вилучаємо та перераховуємо всі ймовірності і їхні середні значення;

4) визначаємо різниці середніх ймовірностей за кроками навчання та ймовірностей, отриманих за розрахунком класифікатора; якщо для певних підкласів різниці становлять більше порогового значення, тоді продовжуємо для них навчання додаванням додаткових даних і повторення кроків 1–3.

Вектор v_{p,e_s} може бути не зарахований до жодного класу і підкласу із заданих, тобто дослідження його компонент встановило, що серед них немає ймовірно зловмисних дій. Але встановлення такого факту базується на застосуванні не тільки пошуку максимального значення ймовірності, обчисленої за теоремою Баєса, але і відповідність цієї ймовірності пороговим значенням класів і підкласів, визначення яких здійснюють у процесі навчання класифікатора. Це пов'язано з тим, що виконуваний процес, який відображено вектором v_{p,e_s}, може не належати до зловмисного програмного забезпечення, тобто належати класу корисного ПЗ. В такому разі виконуваний процес далі не розглядається.

Здійснення самонавчання проводимо за схемою навчання (кроки 1–4). При аналізі вектора ймовірно зловмисних дій заносимо почергово його дані до кожного класу та здійснюємо розрахунки. Якщо відхилення отриманих ймовірностей перебувають у межах порогових значень для одного з підкласів, тоді після завершення класифікації включаємо його нові дані до його підкласу та робимо новий розрахунок для всього класифікатора і його середніх значень відхилень ймовірностей.

Після додавання нового елементу до класифікатору ПМ в одній із КС здійснюється розсилання решті ПМ розподіленої системи інформації про цю подію та самого контейнера з новим елементом для доповнення класифікатора. Таким чином, знання, отримані одним ПМ, передаються іншим компонентам системи для використання.

Рішення про місце здійснення обробки сформованого вектора ймовірно зловмисних дій визначає ПМ, в якому було зібрано ці дані. Якщо аналіз завантаження основних ресурсів показує високий рівень завантаженість, тоді він направляє запит іншому ПМ на можливість обробки. Після отримання підтвердження ПМ КС надсилає відповідний контейнер із вектором. Результати обробки в цьому разі надходять тому ПМ, який ставив задачу обробки, якщо ймовірність наявності зловмисного коду не встановлена. Якщо ж класифікатор програмного модуля обробника встановив наявність зловмисного коду, тоді він виконує наступну послідовність дій: додає до свого класифікатора, розсилає відповідний контейнер із новими знаннями (інформація про новий елемент для класифікатора, дані про виконуваний процес, на основі якого сформовано такий вектор) решті

ПМ та повідомлення тому ПМ про наявність у КС процесу, який виконує зловмисні команди.

Метод виявлення бот-мереж у комп'ютерних системах локальних мереж на основі залучення розподілених систем [19; 20] складається з таких основних кроків:

1) проведення активного моніторингу виконання команд у КС (починаючи з першої API функції кожного процесу, що буде виконуватись після запуску КС) та отримання даних про активні процеси та мережні пакети;

2) здійснення збору даних моніторингу після виявлення певних ймовірно зловмисних проявів у КС у вектор;

3) формування вектора ознак ймовірно підозрілих дій для зібраних даних, компонентами якого є API функції;

4) прийняття рішення про місце обробки вектора ймовірно зловмисних дій;

5) якщо аналіз завантаженості ресурсів КС показав низький рівень завантаженості, тоді треба здійснити обробку в цій КС, інакше надіслати в іншу визначену ПМ КС;

6) здійснення класифікації вектора ймовірно зловмисних дій та аналіз її результатів.

6.1) якщо встановлено зарахування такого вектора до певного підкласу класу бот-мереж, тоді додавання цієї інформації до класифікаторів всіх ПМ;

6.2) якщо встановлено віднесення такого вектора до декількох підкласів класів бот-мереж, тоді здійснити аналіз із залученням решти ПМ розподіленої системи на основі обробки варіантів подій;

6.3) якщо близькість для включення до певного підкласу є нечіткою, але додатково із залученням решти ПМ визначено, що вектор містить зловмисні дії, тоді треба здійснити створення нового класу для бот-мереж, занести дані, оновити налаштування класифікатора, передати результат решті КС;

6.4) якщо перевірка встановила, що досліджуваний вектор не містить зловмисного навантаження, тоді варто здійснити зупинку дослідження процесу, на основі якого він був сформований;

6.5) якщо встановлено, що досліджуваний вектор містить зловмисне навантаження, тоді треба здійснити зупинку відповідного процесу;

6.6) здійснення пошуку й дослідження на основі отриманих відомостей аналогічних процесів в інших КС мережі, де встановлена РБС її програмними модулями;

7) обчислення значення ймовірностей в станах ПМ і формування вимоги для інших ПМ здійснити обчислення ймовірності бути ураженою для всієї РБС [20] − цей крок здійснюється через дослідження наявного зловмисного прояву.

Розроблений метод виявлення бот-мереж дозволяє виявляти бот-мережі, використовує баєсівський класифікатор, здійснює обмін отриманими результатами між програмними модулями РБС.

Експерименти. Метою експериментів була перевірка застосування методу виявлення, роботи класифікатора в структурі розподіленої системи та визначення залежності відсотку виявлених вузлів бот-мережі від їх представлення векторами та різними класифікаторами.

Для підготовки до проведення експериментів було здійснено конструювання 28 штучних бот-мереж та отримано коди відомих виявлених бот-мереж. Всі згенеровані бот-мережі згруповано за класами, в яких виділено 25 структурних елементів на трьох стадіях функціонування та 81 функцію. Варто зазначити, що не всі отримані таким чином бот-мережі містили повністю всі структурні елементи та функції.

Кожну функцію задано векторами зловмисних дій та атак з урахуванням варіацій та на основі яких побудовано зразки задля їх включення в підкласи і класи. Експеримент проводився для класифікатора без додавання екземплярів створених бот-мереж та з ними, тобто здійснювалась перевірка без навчання класифікатора на створених зразках і з попереднім зарахуванням зразків по класах. Другий варіант є необхідним для перевірки точності зарахування до класів тестових зразків, з яких ці класи були сформовані, оскільки у процесі здійснення моніторингу API функцій можуть бути похибки, а також для встановлення величини різниці у двох випадках без попереднього навчання і з ним. Це необхідно, щоб перевірити залежність виявлення за векторами по кожному класу і загальну кількість виявлених вузлів бот-мереж та точність класифікації. Експеримент здійснювався в межах 19 комп'ютерних систем локальної мережі. Кожна з КС містила ПМ РБС. Інших антивірусних засобів у КС не встановлено. Спочатку було встановлено ПМ із класифікатором, в якому не було зразків створених бот-мереж. В одній із КС було розміщено командно-контролюючий центр, у трьох КС було розміщено контролюючі центри, до кожного з яких було під'єднано по п'ять вузлів у кожній із п'ятнадцяти КС. Встановлення штучно згенерованих бот-мереж здійснювалось почергово. Після завершення експерименту з окремо згенерованою бот-мережею здійснювалось повне оновлення всіх КС, при цьому класифікатор залишався без змін для кожного випадку. Тривалість моніторингу КС становила 96 годин для кожного екземпляра

бот-мережі кожного з двох класифікаторів. Атака з вузлів бот-мережі не здійснювалась. Вузли бот-мережі працювали тільки в режимі контролю КС та підтримки структури бот-мережі через відправлені повідомлення. Таким чином, для ПМ РБС об'єктами дослідження були запущені в КС процеси і, відповідно, побудова векторів за ними. З метою проведення експерименту було обрані бот-мережі, які використовують стратегію отримання повного контролю в КС. Для здійснення експерименту засобами API моніторингу в КС було отримано вектори, які почергово оброблено класифікатором ПМ. Результати обробки представлено в табл. 1.

Експерименти передбачали визначення таких показників ефективності виявлення вузлів бот-мереж для класів і підкласів баєсівського класифікатора:

1) $P_{1,1}$ − відсоток векторів зловмисних дій та атак для вузлів бот-мереж, що належать цьому класу щодо всіх тестових зразків, які система зарахувала до цього класу з використанням попереднього навчання;

2) $P_{1,2}$ – аналогічно до 1) тільки без використання попереднього навчання;

3) $P_{2,1}$ – відсоток векторів зловмисних дій та атак для вузлів бот-мереж, що належать цьому підкласу класу щодо всіх тестових векторів, які система зарахувала до цього підкласу класу в тестовій вибірці (ті, які були вірно зараховані до підкласів) із використанням попереднього навчання;

4) $P_{2,2}$ – аналогічно до 3) тільки без використання попереднього навчання;

5) $P_{3,1}$ – відсоток вірно виявлених вузлів бот-мереж із використанням попереднього навчання;

6) $P_{3,2}$ – аналогічно до 5) тільки без використання попереднього навчання;

7) $P_{4,1}$ – відсоток хибно класифікованих вузлів бот-мереж як корисних додатків (помилка 1-го роду)із використанням попереднього навчання;

8) $P_{4,2}$ – аналогічно до 7) тільки без використання попереднього навчання;

9) $P_{5,1}$ – відсоток не вірно класифікованих вузлів бот-мереж, як таких що є вузлами бот-мереж, але віднесені не до того класу (помилка 3 - го роду) з використанням попереднього навчання;

10) $P_{5,2}$ – аналогічно до 9) тільки без використання попереднього навчання.

Результати оцінки ефективності виявлення програмного забезпечення вузлів бот-мереж на основі роботи двох класифікаторів для введених класів та підкласів у класифікаторі наведено у таблиці 1.

У результаті проведення експерименту отримано зарахування до потрібного підкласу та класу отриманих на основі моніторингу векторів із точністю до 66% для класифікатора без введених векторів 28 штучно згенерованих бот-мереж та 88% для класифікатора, в який попередньо було додано вектори шляхом здійснення його навчання, зберігаючи в ньому шаблони попередніх наповнень. Перевірка здійснювалась окремо для класів, їх підкласів та загалом для вузлів. Результати були усереднені та їх дисперсія відносно середнього значення становить 1%. Різниця відхилення для двох класифікаторів по кожному класу, підкласу і вузлів бот-мереж і загалом становить 21,5%. Відхилення між різницями відхилень для двох класифікаторів становить по кожному окремому класу, підкласу та вузлу бот-мережі менше 5%, що вказує на точність визначення в різних класах і підкласах. Це означає, що результат виявлення програмного забезпечення вузлів бот-мереж збігається в розрізі класів та підкласів для векторів зловмисних дій та атак.

Помилки 1-го роду склали для першого і другого класифікаторів 11,7% та 31,97%, що пояснюється їх різним наповненням. Помилки 3-го роду – 0,01% та 2,14% відповідно, що пояснюється ширшим полем класифікації другого класифікатора через менший обсяг навчальної вибірки. Загалом результати роботи класифікаторів показують можливість їх застосування для задач виявлення бот-мереж.

Висновки. У статті представлено метод виявлення бот-мереж на основі застосування розподілених систем та класифікатора. Метод виявлення складається з двох частин: хостового і мережного рівнів. На рівні хостової частини процедура виявлення базується на реалізації класифікації Баєса. Мережний рівень розширює результати, отримані на рівні хоста, до решти локальної мережі. Розроблений метод забезпечує обмін результатами, отриманими за класифікацією Баєса, для подальшого використання іншими компонентами розподіленої системи. Результати розробленого класифікатора показують, що представлення зразків бот-мереж для різних класів і підкласів достатньо для ефективного виявлення бот-мереж. Результати експерименту показали, що точність виявлення бот-мереж сягає 88%.

Напрямами подальших досліджень є розробка нових методів виявлення ЗПЗ, які б були орієнтовані на архітектуру розподіленої системи, де вони будуть реалізовані, та використовували цю перевагу над іншими хостовими методами.

Таблиця 1

Результати експерименту

Показники експерименту	Отриманні значення для різних класів							Середні значення
	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	
$P_{1,1}$, %	90,74	84,29	73,66	86,30	94,04	94,18	96,60	89,44
$P_{1,2}$, %	75,93	63,57	60,22	70,32	68,77	67,60	69,36	67,71
$\lvert P_{1,1}-P_{1,2}\rvert$, %	14,81	20,72	13,44	15,98	25,27	26,58	27,24	21,73
$P_{2,1}$, %	85,80	83,57	72,58	85,39	98,88	93,92	96,60	88,42
$P_{2,2}$, %	74,69	63,57	59,14	70,32	67,37	66,58	67,66	66,80
$\lvert P_{2,1}-P_{2,2}\rvert$, %	11,11	20	13,44	15,07	31,57	27,34	28,94	21,62
$P_{3,1}$, %	92,11	84,21	71,93	89,47	90,53	88,42	93,68	87,72
$P_{3,2}$, %	76,32	57,89	63,16	64,91	71,58	54,74	75,79	65,89
$\lvert P_{3,1}-P_{3,2}\rvert$, %	15,79	26,32	8,77	24,56	18,95	33,68	17,89	21,83
$P_{4,1}$, %	7,89	14,47	28,07	10,53	7,37	11,58	6,32	11,70
$P_{4,2}$, %	21,05	40,79	36,84	31,58	24,21	44,21	22,11	31,97
$\lvert P_{4,1}-P_{4,2}\rvert$, %	13,16	26,32	8,77	21,05	16,84	32,63	15,79	20,27
$P_{5,1}$, %	0	1,32	0	0	2,11	0	0	0,01
$P_{5,2}$, %	2,63	1,32	0	3,51	4,21	1,05	2,11	2,14
$\lvert P_{5,1}-P_{5,2}\rvert$, %	2,63	0	0	3,51	2,1	1,05	2,11	2,13

Список літератури:

1. Машинне навчання та знання людини в динамічній рівновазі. URL: https://eset.ua/ua/products/for_business/security/endpoint_security (дата звернення: 26.11.2018).

2. Доктор Веб. URL: https://curenet.drweb.ru/ (дата звернення: 26.11.2018).

3. Обзор Symantec Endpoint Protection 12. URL: https://www.anti-malware.ru/reviews/Symantec_Endpoint_Protection_12_2 (дата звернення: 26.11.2018).

4. Malwarebytes Endpoint Security. URL: Ошибка! Недопустимый объект гиперссылки. (дата звернення: 26.11.2018).

5. Решение по безопасности беспроводных сетей на базе Cisco Network Admission Control. URL: https://www.cisco.com/web/RU/products/hw/wireless/secure/cnac.html (дата звернення: 26.11.2018).

6. Антивирусная защита сети Kaspersky Administration Kit. URL: https://support.kaspersky.ru/learning/courses/kl_102.80/intro/section1 (дата звернення: 26.11.2018).

7. Rostami M.R., Eslahi M., Shanmugam B. and Ismail Z. Botnet evolution: Network traffic indicators. Biometrics and Security Technologies (ISBAST), 2014 International Symposium on, 2014. Pp. 274−279.

8. Kotenko I. (2013). Experiments With Simulation Of Botnets And Defense Agent Teams, ECMS 2013 Proceedings edited by: W. Rekdalsbakken, R. T. Bye, H. Zhang, European Council for Modeling and Simulation. doi:10.7148/2013-0061.

9. Sheng L., Zhiming L., Jin H., Gaoming D. and Wen H. A Distributed Botnet Detecting Approach Based on Traffic Flow Analysis. Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012 Second International Conference on, 2012. Pp. 124−128.

10. Li S.-H., Kao Y.-C., Zhang Z.-C., Chuang Y.-P. and Yen D.C. A Network Behavior-Based Botnet Detection Mechanism Using PSO and K-means. ACM Transactions on Management Information Systems (TMIS). 2015. Vol. 6. P. 3.

11. Stevanovic M. and Pedersen J.M. An analysis of network traffic classification for botnet detection. Cyber Situational Awareness, Data Analytics and Assessment (CyberSA). 2015 International Conference on, 2015. Pp. 1−8.

12. Gu G., Zhang J. and Lee W. BotSniffer: Detecting botnet command and control channels in network traffic, 2008.

13. Pomorova O., Savenko O., Lysenko S., Kryshchuk A. and Bobrovnikova K. A Technique for the Botnet Detection Based on DNS-Traffic Analysis. Computer Networks, ed: Springer, 2015. Pp. 127−138.

14. Zhang J., Perdisci R., Lee W., Sarfraz U. and Luo X. Detecting stealthy P2P botnets using statistical traffic fingerprints. Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference on, 2011. Pp. 121−132.

15. Gu G., Porras P., Yegneswaran V., Fong M. and Lee W. BotHunter: detecting malware infection through IDS-driven dialog correlation. Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, Boston, MA, 2007.

16. Goebel J. and Holz T. Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation. HotBots. 2007. Vol. 7. P. 8.

17. Savenko O., Lysenko S. and Kryshchuk A. Multi-agent based approach of botnet detection in computer systems. Computer Networks, ed: Springer, 2012. Pp. 171−180.

18. Sochor T. & Zuzcak M. Attractiveness Study of Honeypots and Honeynets in Internet Threat Detection. 22nd Int. Conf. Computer Networks: Communications in Computer and Information Science 2015-06-16 Brunow. Cham: Springer International, 2015. Pp. 69−81. ISBN 978-3-319-19418-9.

19. Савенко О.С. Архітектура розподіленої багаторівневої системи виявлення шкідливого програмного забезпечення в локальних комп'ютерних мережах. Вчені записки Таврійського національного університету. 2018. Том 29(68). № 2. С. 172−181.

20. Markowsky G., Savenko O. and Sachenko A. (2019) Distributed Malware Detection System Based on Decentralized Architecture in Local Area Networks. Shakhovska N., Medykovskyy M. (eds) Advances in Intelligent Systems and Computing III. CSIT 2018. Advances in Intelligent Systems and Computing, vol. 871. Springer, Cham. P. 582−598.

ВИЯВЛЕНИЕ БОТ-СЕТЕЙ РАСПРЕДЕЛЕННЫМИ СИСТЕМАМИ НА ОСНОВЕ КЛАССИФИКАЦИИ

В статье представлены типы бот-сетей на основе их характерных особенностей архитектуры, функционального назначения элементов и с учетом архитектуры средств обнаружения. Данное представление формализовано к компонентам векторов, и на его основе осуществлено создание образцов для подклассов и классов, используемых в классификаторе компонентов бот-сетей Байеса. Разработанный метод обнаружения бот-сетей на основе классификатора содержит два уровня. Первый (хостовой) уровень при обнаружении использует классификатор Байеса, исследует вектор вызовов API функций. На основе полученного результата на втором (сетевом) уровне осуществляется оценка вероятности наличия бот-сетей всей распределенной системой.

Ключевые слова: *распределенная система, вредоносное программное обеспечение, бот-сети, сетевые антивирусы, наивный Байесовский классификатор, классификация.*

DISTRIBUTED SYSTEM OF DETECTION OF BOTNET BASED ON CLASSIFICATION

The article presents the types of botnets based on their characteristic features of architecture, functional purpose of elements and taking into account the architecture of detection tools. This representation is formalized to the component of the vectors and on its basis the creation of samples for subclasses and classes used in the Bayesian classification of botnet components was made. The developed method for detecting botnets based on the classifier has two levels. The first (host) level when detected uses a Bayesian classifier that examines the API function calls vector. On the basis of the obtained result at the second (network) level, an estimation of the probability of the availability of botnets by the entire distributed system is carried out.

Key words: *distributed system, malware, botnet, network antivirus, naive Bayes classifier, classification.*