DOI 10.31891/2307-5732-2020-283-2-94-100
УДК 004.056.5

С.М. ЛИСЕНКО, Є.С. БУРДАШ
Хмельницький національний університет

МЕТОД ТА ЗАСОБИ ІДЕНТИФІКАЦІЇ БОТ-МЕРЕЖ, ЩО ВИКОРИСТОВУЮТЬ ТЕХНОЛОГІЮ «ДИНАМІЧНА ПЕРЕАДРЕСАЦІЯ IP-АДРЕС»

В роботі представлено метод, що використовує технологію динамічної "переадресації IP-адресів". Даний метод зосереджений на виявленні бот-мережі, за допомогою сканування DNS трафіку та отримання його ознак. За допомогою алгоритму машинного навчання SVDD виконується виявлення аномалій у заданих ознаках та співставлення їх з відповідними умовами, які свідчать про наявність інфікованого ботнету у даному DNS трафіку. Цей метод дає змогу виявляти шкідливі бот-мережі із високою ефективністю та швидкістю. Цей метод може стати основою для програмного забезпечення виявлення бот-мереж, які використовують технологію "динамічна переадресація IP-адрес".

Ключові слова: *бот-мережа, DNS, SVDD, машинне навчання.*

S. LYSENKO, Y. BURDASH
Khmelnytskyi National University

METHOD AND SOFTWARE OF FAST-FLUX BOTNET DETECTION

Fast flux is a method that a criminal can use to prevent the identification of the IP address of his or her own computer. The main idea of this paper is to create a method for fast-flux botnet detection based on the SVDD(support vector data description) machine learning and anomalies detection algorithm that achieves better performance and efficiency. Using this method gives an opportunity to easily detect malware in botnets and notify the user about that. It makes possible to save and protect user's private data. We focus on detection fast-flux botnets based on the scanning Domain name system (DNS). The method has a unique structure and can be extended with new parameters in the future. The method collects all received data and extracts only useful parameters from each DNS message and transforms this data into valid and understandable for the algorithm. In this article represented a method which uses anomalies detection approach. SVDD algorithms it is a powerful tool that allows us to identify malware botnet in the system in the earlier stages before they occur and infect the system. Using the SVDD algorithm can improve the detection of the botnets based on the fast-flux approach. To provide the most efficient machine learning algorithm it should be trained by the special data. In this case, the system provides the highest level of accuracy and low level of the fault. This algorithm can detect the anomalies that were unknown in the training step, it can increase number of the botnets if the future. The proposed methods and algorithm was tested on the implemented locally system and showed a good result of detection fast-flux botnet. The level of accuracy showed 97.8%.

Keywords: *fast-flux, malware, SVDD, DNS. machine learning, anomalies detection.*

Вступ

Сучасні технології розвиваються швидкими темпами, разом із тим швидко розвиваються засоби атак на приватні дані користувачів. Зловмисники знаходять слабкі місця у систем та використовують ці місця для того, щоб отримати приватні дані звичайних користувачів без їх згоди або зашкодити власникам даних мати до них доступ, що є прямим порушенням закону про захист даних.

Зі стрімким розвитком інтернету також зросла і кількість вірусів та їх методів поширення. Одним із таких методів є поширення вірусів за допомогою ботнету [1]. Ботнет використовують для крадіжки даних користувача, таких як номери та паролі кредитних карт, паролі та іншої протиправної діяльності – розсилання спаму, запуску іншого шкідливого програмного забезпечення. Бот встановлюється на комп'ютері користувача без його відому і являє собою зачасти приховану програму. Для поширення ботнетів найбільш дієвим методом є інтернет або мережа комп'ютерів. Вони використовують DNS-систему доменних імен [2]. Зловмисники ж за допомогою DNS поширюють інфіковані ботнети.

Оскільки DNS трафік часто є нефільтрованим або таким, який сприймається браузером як безпечний, за допомогою нього можна встановлювати стійкий та надійний канал зв'язку між комп'ютерами мережі. Тим самим зловмисники обходять існуючі системи захисту.

Отже, дана тема є актуальною для дослідження. В основі даної роботи полягає розробка нового та ефективного методу для виявлення бот-мереж, які використовують технологію "динамічної переадресації IP-адрес". Виявлення бот-мереж, які використовують дану технологію, повинно бути із застосуванням найбільш нових та ефективних технологій. Однією із таких технологій є застосування алгоритмів машинного навчання та штучного інтелекту. Оскільки DNS трафік містить певні набори ознак, яких за допомогою аналізу алгоритмами штучного інтелекту можна зробити певні висновки про стан даного трафіку, тобто чи він є інфікованим чи ні, і за допомогою якої технології ботнету. Оскільки алгоритми штучного інтелекту працюють достатньо швидко, то можна забезпечити виявлення шкідливого вмісту мережі ще на ранніх стадіях, що забезпечує високу ефективність системи та захисту приватних даних користувачів.

Пов'язані роботи

Ботнети становлять загрозу із сотнями мільйонів заражених комп'ютерів. Дослідження показує, що 40% усіх комп'ютерів, які підключені до інтернету, заражені ботами та контролюються зловмисниками [3].

Одним із таких методів є метод виявлення технології «динамічна переадресація IP-адрес» в основу якого закладено аналіз на математична модель через отримання даних DNS трафіку та обрахунку різниць в отриманому часі відповіді [4].

В основі даного методу лежить мережа, що використовує динамічну переадресацію IP-адрес і використовує динамічний DNS для відображення динамічного домену на різні IP-адреси та застосовує потокові боти для перенаправлення трафіку мережі. Завдяки можливості приховувати хости ботнетом зловмисники широко використовують дану технологію як прикриття різних афер.

Згідно з даним методом сервіси, що використовують динамічний DNS, щоб збалансувати їх навантаження на хости та зловмисники користуються цим, щоб приховати ботнетів. У результаті час відповіді наступних записів на домен стає більш нестабільним. Виходячи із різниці у часі відповіді, даний метод пропонує підхід виявлення ботнету. Дана система встановлюється на комп'ютері, який може бути як кінцевим хостом, так і проміжним комп'ютером звичайного користувача.

Користувач із набором невідомих URL-адрес, які можуть бути отримані наприклад зі спаму чи соціальних мереж дає можливість визначити чи домени є доброякісними або шкідливими, що використовують технологію динамічної переадресації IP-адрес.

В основі методу лежить отримання набору доменів на їх аналіз за допомогою математичних моделей на обрахунків. Додаток надсилає запит на DNS для запиту IP-адрес, які відображаються на домені. Після того як TTL (time to live) відповіді на запит закінчувався, система повторно надсилає запит на сервер. Для кожного із таких запитів, зазначені етапи повторюються протягом 10-и хвилин. У результаті для кожного IP вимірюється час відповіді, і якщо середній результат відрізняється від відповіді, система помічає його як можливого шкідливого.

Виходячи із результатів запропонованих у даному методі видно, що даний метод може точно виявляти шкідливі домени лише із 0,3 % похибкою. Для того, щоб визначити, чи домен є шкідливим даному методу, необхідно менше 20 хв. Окрім того ця система не потребує спеціальної підтримки з боку провайдера або будь-якого іншого мережевого сервісу.

У основу [5] даного підходу покладено обробку даних [6, 7] як і у попередньому прикладі, але додано нові функції для підвищення ефективності та швидкодії даного методу. Даний метод має можливість виявити шкідливі домени в онлайн режимі та використовує нейронну мережу [8] (ADeSNN).

До уваги методу приймаються наступні параметри:

1) кількість IP-адрес у секції відповіді;
2) кількість IP-адрес у додатковій секції відповіді;
3) кількість ASN для IP-адрес у секції відповіді;
4) кількість ASN для IP-адрес у додатковій секції відповіді;
5) розмір повідомлення.

На основі аналізу цих даних система дає висновок чи містить даний ботнет шкідливі дані. Даний алгоритм показує досить хорошу швидкодію та ефективність визначення шкідливого ботнету. Також адаптивний DeSNN показав підвищення ефективності його класифікації. Алгоритм виконаний на загальнодоступних даних. Крім того, адаптивний алгоритм сприяв проблемі налаштування параметрів як згадувалося раніше.

У “Identifying Fast-Flux Botnet With AGD Names at the Upper DNS Hierarchy” підході [9] пропонується нова схема виявлення технології динамічної переадресації IP-адрес. Запропонований підхід може розпізнати групи доменів, породжених зловмисниками, алгоритми генерації доменів або їх варіанти, які є репрезентативними для різних ботнетів. На додачу до цього, він також може визначити, чи алгоритмічно генеровані доменні імена в кластері використовують технологію динамічної переадресації IP-адрес чи ні, застосовуючи двоступеневий механізм виявлення. Трафік DNS збирається з магістральних маршрутизаторів.

За допомогою верхнього рівня ієрархії DNS, генеровані специфічним DGA домени, які є репрезентативними для відповідних ботнетів окрім того, визнано, чи використовуються імена AGD у кластері технологія динамічної переадресації IP-адрес чи ні також не визначені.

Описані вище алгоритми не дають змоги швидко та ефективно виявляти бот-мережі, що використовують технологію «динамічна переадресація IP-адрес, оскільки на виявлення потрібно занадто багато часу та дані алгоритми опрацьовують лише раніше відомі бот-мереж, що дає змогу у подальшому обходити дані алгоритми за допомогою новіших версій бот-мереж.

Метод виявлення бот-мережі, що використовує технологію «динамічна переадресація IP-адрес»

Технологія динамічної переадресації IP-адрес – це техніка, яку кіберзлочинці застосовують для запобігання ідентифікації IP-адреси свого ключового хост-сервера. Зловживаючи способом роботи DNS злочинець може створити ботнет з вузлами, які приєднуються та виходять із мережі швидше, ніж їх можна простежити [2].

Дана технологія використовує перевагу способу збалансованого навантаження на систему доменних імен. DNS дозволяє адміністратору зареєструвати декілька IP-адрес з одним ім’ям хоста. Альтернативні адреси законно використовуються для розподілу інтернет трафіку між декількома серверами. Зазвичай IP-адреси, пов’язані з доменом хоста, якщо вони є, не змінюються дуже часто.

Однак кіберзлочинці виявили, що вони можуть приховувати ключові сервери, використовуючи шістдесяти секундний час існування (TTL – time to live) для своїх записів ресурсів DNS та змінювати пов’язані IP-адреси записів із надзвичайною частотою. Оскільки зловживання системою вимагає співпраці з

реєстром доменних імен, вважається, що найбільш швидкі потокові DNS ботнети походять із країн, що розвиваються, або з інших країн не кіберзлочинність погано регулюється із сторони влади та її законів.

Для того, щоб виявити, що застосована технологія – це технологія "Динамічна переадресація IP-адрес", необхідно отримати корисні функції та параметри з повідомлень відповідей DNS, які відрізняють домени динамічної переадресації IP-адрес від законних доменів.

Повідомлення DNS складається із заголовка DNS та чотирьох розділів: запиту, відповіді, авторизації та із додаткового поля (question, answer, authority and additional). Розділ запиту (question) містить запит, що надсилається на сервер DNS. Дані наступних трьох розділів зберігаються у наборі записів ресурсів (RRs – resource records). Для визначення нам необхідні два типи RRs: записи, що містять IPv4 адреси та NS записи що вказують на імена серверів. Розділ відповідей (answers) містить відповіді RR відповідні до запису. Розділ authority містить NS записи домену. IP-адреси NS зазвичай знаходяться у додатковому розділі [2, 3].

Для виявлення необхідні наступні ознаки функцій для аналізу пакетів:

1) N_a – кількість записів у секції відповіді (answers);

2) N_{asn} – кількість різних ASN для всіх записів;

3) N_{ns} – кількість записів для NSs (у додатковій секції повідомлень DNS);

4) $N_{\mathbf{nsasn}}$ – число кількість різних ASN для всіх NS;

5) $TTL_{\mathbf{a}}$ – кількість TTL записів у секції відповіді (answers);

6) TTL_{ns} – кількість TTL записів для NS домену.

Основними ознаками динамічної переадресація IP-адрес є N_a, $TTL_{\mathbf{a}}$ та N_{asn}. Зазвичай ці ботнети встановлюють (у загальному більше п'яти) записів для одного домену, щоб забезпечити роботу принаймні одному доступному із них також встановлюють N_a більший за норму, дуже низький TTL ($TTL_{\mathbf{a}}$) для швидкої зміни списку адрес та високий N_{asn} для розповсюдження IP-адрес.

Застосування алгоритму SVDD для виявлення бот-мереж, що використовують технологію «динамічна переадресація IP-адрес»

Даний етап є основою для ефективного виявлення бот-мереж, що використовують технологію «динамічна переадресація IP-адрес».

Виявлення бот-мережі доволі складний процес. Зазвичай сучасні системи можуть виявити бот-мережу вже занадто пізно, тоді коли система вже є інфікованою, що потім тільки ускладнює процес як виявлення вже інфікованих даних, так і забезпечення захисту системи в цілому. Тому доцільно виявляти бот-мережу, ще на початковій стадії DDoS атаки, що дозволяє якісно та швидко реагувати на дії зловмисників.

Задача полягає у тому, що потрібно розробити нову систему виявлення бот-мережі за допомогою відслідковування інтернет трафіку та виявлення у ньому ознак, за допомогою яких можна буде робити висновки, чи відбуваються якісь дії зловмисників.

Етапи виявлення бот-мережі поділяються на такі основні кроки: збір та отримання трафіку DNS мережі, виявлення ознак мережі, перевірка отриманих ознак за допомогою алгоритму машинного навчання та повідомлення про стан трафіку на наявність інфікованої бот-мережі у отриманому трафіку.

Схема роботи алгоритму представлена на рис. 1.

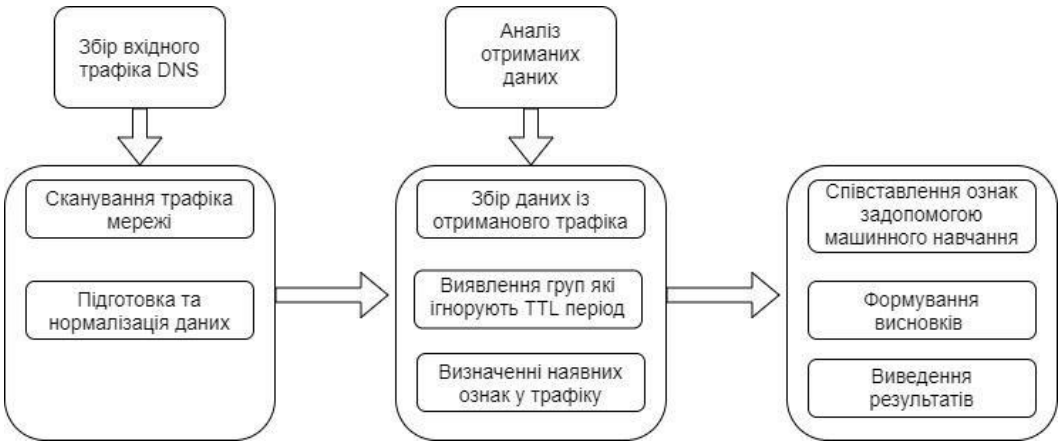

Рис. 1. Схема функціонування алгоритму виявлення бот-мережі, що використовує технологію «динамічна переадресація IP-адрес»

Підготовка поділяється на такі кроки:

1. Сканування мережевого трафіку.
2. Отримання мережевого трафіку.
3. Визначення усіх наявних ознак в отриманому трафіку.

Наступним кроком є аналіз отриманих ознак на основі алгоритму машинного навчання, який описується наступним чином:

1. Формування та створення моделі алгоритму.
2. Завантаження отриманих ознак.
3. Обробка та формування лише необхідних ознак та їх представлення у правильній та зручній формі для алгоритму.
4. Навчання на основі отриманих ознак та їх значень.
5. Тренування моделі на кількох групах різних даних, для підвищення чіткості роботи алгоритму.

Етап навчання є одним із найважливіших етапів, оскільки якісна робота цього етапу, забезпечить ефективну роботу алгоритму.

На даному етапі відбувається формування даних, які завідома є правильними та не правильними. Потім отримують вектор ознак, який проходить додаткову перевірку даних згідно раніше ідентифікованих вимог. Якщо дані не є такими, вони повертаються на попередній етап – етап нормалізації та формування вектора ознак. Якщо дані не можуть бути перетворені для обробки алгоритмом – вони помічаються, як дані що є не правильними, та система сповіщає про це користувача.

Якщо ж дані перевірені та провалідовані вони передаються на етап навчання. Чим більша кількість ітерацій, тим це краще для алгоритму, та він буде мати менший відсоток похибки, що у свою чергу підвищить ефективність та якість роботи алгоритму машинного навчання. Також дані, які будуть передаватися на вхід етапу навчання, мають містити у собі дані, які відхиляються від норми.

Алгоритм SVDD є більш ефективним представленням алгоритму SVM [10].

Оскільки алгоритм SVDD, є алгоритмом виявлення аномалій, то саме дані які містять у собі відхилення від норми, будуть основними до опрацювання [11, 12].

Основною перевагою даних алгоритмів є те, що в результаті після етапу навчання алгоритму виявлення аномалій, у тому числі алгоритм SVDD, може виявляти раніше не відомі аномалії та відхилення, які будуть опрацьовуватися надалі. Це дає змогу для широких можливостей застосування алгоритмів даного типу. Вектори, які містять у собі набори параметрів, будуть опрацьовуватися алгоритмом.

Алгоритм приймає дані у вигляді *N* об’єктів даних $\{x_i, i = 1, N\}$. На основі цих даних необхідно вивести сферу із мінімальним обсягом, що містить всі, або більшу частину об’єктів, даних.

На випадок, коли один або кілька об’єктів знаходяться у тренувальному набору, та є найбільш рівновіддаленими від центра, виходить дуже велика сфера, яка не відображатиме дані добре [13]. Щоб мінімізувати похибку, вводяться так звані “слабкі змінні” ξ_i. Зі сфер, описаних центром *a* та радіусом *R*, мінімізується радіус.

$$F(R, a, \xi_i) = R^2 + C\sum \xi_i \quad (1)$$

де змінна *C* являє собою кількість векторів, які можуть вийти за межі сфери.

Отриману формулу необхідно звести до мінімуму за відповідними обмеженнями:

$$[(x]_i - a)^T(x_i - a) \le R^2 + \xi_i \quad \forall_i, \xi_i \ge 0. \quad (2)$$

Відповідно до обмежень (посилання на формулу F()), побудуємо рівняння методу Лагранжа:

$$L(R, a, \alpha_i, \xi_i) = R^2 + C\sum_i \xi_i - \sum_i \alpha_i \{R^2 + \xi_i - (x_i^2 - 2ax_i + a^2)\} - \sum_i \gamma_i \xi_i \quad (3)$$

з множниками Лагранжа $\alpha_i \ge 0$ та $\gamma_i \ge 0$. Нові обмеження визначаються рівністю нулю часткових похідних:

$$\sum_i \alpha_i = 1, a = \frac{\sum_i \alpha_i x_i}{\sum_i \alpha_i} = \sum_i \alpha_i x_i \ , C - \alpha_i - \gamma_i = 0 \ \forall_i \quad (4)$$

$\alpha_i \ge 0$ та $\gamma_i \ge 0$, можна виділити змінні γ_i з третього рівняння у (формула вище) і використовувати обмеження $0 \le \alpha_i \le C, \forall_i$. З посилання на номери попередніх двох рівнянь максимізуються щодо α_i:

$$L = \sum_i \alpha_i (x_i \cdot x_i) - \sum_{ij} \alpha_i \alpha_j (x_i \cdot x_j) \quad (5)$$

з обмеженнями $0 \le \alpha_i \le C, \sum_i \alpha_i = 1.$

З другого рівняння у (2) слідує, що центр сфери є лінійною комбінацією об'єктів даних з ваговими коефіцієнтами, які отримують шляхом оптимізації рівняння (3). Тільки для невеликого набору об'єктів рівність у рівнянні (1) відповідає об'єктам, які знаходяться на межі самої сфери. Для цих об'єктів коефіцієнти будуть ненульовими і називаються опорними об'єктами. У описі сфери потрібні лише ці об'єкти. Радіус *R* сфери можна отримати, розраховуючи відстань від центру сфери до опорного вектора з вагою менше, ніж *C*. Об'єкти, для яких, потрапляють у верхню межу в (2) і виходять за межі сфери. Ці вектори підтримки вважаються перевершеними.

Гіперсфера – це чітка границя навколо даних та зазвичай не дає правильного представлення про їх структуру. В основі методу SVDD лежить нелінійне відображення даних із тренувального набору у простір із великим розміром та побудова роздільної гіперплощини в цьому просторі. Таким чином можна отримати

нелінійні границі у вхідному просторі. З використанням функції ядра можна вирахувати роздільну гіперплощину без конкретного відображення даних на простір великим розмірів [12, 13].

Розгорнута діаграма функціонування етапу навчання представлена на рис. 2.

Рис. 2. Розгорнута діаграма функціонування етапу навчання

Алгоритм машинного навчання SVDD (Support Vector Domain Description) [10–13] приймає на вхід

Після сканування та підготовки даних необхідно провести етап навчання та встановлення моделі машинного навчання. Для цього необхідно передати підготовлені ознаки, вже у правильному вигляді, на вхід, як параметри у систему машинного навчання.

На цьому етапі необхідно налаштувати модель на виявлення аномалій у отриманому DNS трафіку, на основі тестових даних, які будуть вказувати на ці аномалії.

Набір навчальних даних складається із N кортежів (x_i, y_i) та функції класифікатора, яка представлена у наступному вигляді:

$$f(x) \rightarrow y \tag{6}$$

де x – представлений у вигляді вектора ознак;

y – являє собою вектор класу.

У системі представимо $x = \{x_1, x_2, x_3, \dots, x_n\}$, як множину DNS пакетів. Кожен об'єкт пакета представляє у собі вектор ознак, і характеризується значеннями у вигляді:

$$x_1 i = \{x_1 ij + |1 \leq j \leq m\} \tag{7}$$

де m – це кількість ознак;

x_{ij} – значення j-ї особливості у i-му DNS пакеті.

Відповідно до даних які необхідні для виявлення динамічної переадресації IP-адрес, x_i можна представити у вигляді кортежу із параметрами які необхідні для алгоритму машинного:

$$x_i = \{n_a, n_{asn}, n_{NS}, n_{nsasn}, TTL_a, TTL_{ns}\} \tag{8}$$

Нехай також представимо y як набір класів доменних імен:

$$y = \{y_1, y_2, y_3, \dots, y_n\} \tag{9}$$

Для даного випадку представимо, що y = (1,0), y якому y = 1 та вказує на технологію динамічної переадресації IP-адрес і тоді необхідно застосувати алгоритм машинного навчання для виявлення бот-мережі. У випадку y = 0, це інша технологія чи бот-мережі та застосування даного алгоритму не буде

доречним [12, 13].

Експерименти

Для того, щоб можна було оцінити ефективність даного підходу, що використовує технологію «динамічна переадресація IP-адрес», було проведено ряд експериментів. Для експериментів було використано доменні імена та їх IP-адреси, які є інфікованими та не інфіковані доменні імена.

Для цього було обрано та проаналізовано різні доменні імена та IP-адреси які відносяться до різних країн на мають різні дані та значення параметрів.

Інфіковані доменні імена були отримані із набору даних Malware Domains [14]. У таблиці 1 показаний приклад вхідних даних, які будуть подані на опрацювання алгоритму, де:

#IP – кількість IP-адрес, що асоціюються із даним доменним ім'ям;

#ASN – кількість асоційованих номерів автономної системи;

#PREF – кількість пов'язаних префіксів: префікси IP-адреси також дають інформацію про те, чи є домен законним або він є частиною інфікованої бот-мережі;

#C – кількість країн, із якими асоціюється даний домен;

ND – затримка мережі – відноситься до часу, необхідного для передачі пакетів туди та назад через інтернет між клієнтом та сервером;

PD – затримка обробки запиту;

DFD – затримка отримання документа: посилається на необхідний час з сервера, щоб отримати веб-сторінку.

Таблиця 1

Приклад даних

Доменне ім'я	#IP	#ASN	#PREF	#C	ND	PD	DFD	Тип мережі
google.com.ua	11	1	1	1	0.0481	0.0280	0.2593	legitimate
mastereduc.com	3	3	3	3	0.1362	1.67	1.43	malicious
youtube.com	11	1	1	1	0.0489	0.0414	0.3420	legitimate
dapcopharma.co.ke	2	2	2	2	0.4468	0.1236	0.1818	malicious

Так як даний підхід в працює із IP-адресами, для кожного доменного ім'я, система буде запитувати його IP-адресу та їх кількість для того, щоб перевірити наявність інших IP-адрес що асоціюються з даним доменним ім'ям.

Одним із важливих етапів для даного підходу був етап навчання. Згідно з отриманими даними, частина яких була завідома інфікованою [14], був проведений етап навчання. На цьому етапі було опрацьовано 364 різних інфікованих доменних адрес. Також було опрацьовано 100 адрес, які вважаються легітимними та не містять ознак бот-мережі.

Наступним кроком є етап виявлення. На даному етапі алгоритм розпізнає інфіковані імена, повідомляє про це користувача. Для простоти та зручності усі інфіковані імена будуть виводитися у окремий файл, у якому можна перевірити адреси, які саме були інфіковані.

```
segurocaixaatualizacao.com
token-caixa.sg.tf
nrpexhausts.co.uk
dapcopharma.co.ke
internetbanking-recadastro-caixa.net
mlas.org.sg
zedocaixao2016.xpg.uol.com.br
utrerainmobiliaria.com
branainmobiliaria.com
arquined.com
casasyseguros.com
caixaeconomicafederalhabitacao.com
petiscos.com
mail.nosweatwtloss.com
ultragene.pt
laanpenger.net
margot-salon.ro
perfectparties1.com
punkrockbabyclothing.com
edarural.com
hospitalsadda.com
mushroomworldbpl.com
superbongo.com
fencing-vulkan.ru
lifeokna.ru

Загальна кількість - 356
```

Рис. 3. Результати роботи алгоритму

Експерименти показали, що обрані нами ознаки дають достатню точність роботи алгоритму. Із завідома відомих 364 інфікованих імен, система успішно ідентифікувало 356. Тобто, виходячи із результатів роботи, можна сказати, що точність роботи системи становить 97,8%. Результати роботи даного алгоритму є хорошим показником того, що даний підхід має місце для подальшого розвитку. При подальшому використанні та повторному проведенні етапу навчання похибка може знизитися.

Висновки

Описаний метод ідентифікації бот-мереж, що використовують технологію "динамічна переадресація IP-адрес", показав хороші результати на експериментах з доволі високою чіткістю визначення інфікованої бот-мережі. Даний метод дає змогу ефективно виявляти бот-мережі даного типу та технології та повідомляти про це користувача на більш ранньому етапі ще до того, як бот-мережа інфікує систему користувача.

Отже, даний метод має місце для подальшого розвитку та розробки.

References

1. Botnet Wikipedia. URL: https://en.wikipedia.org/wiki/Botnet. (date 21.03.20)
2. Domain name system Wikipedia. URL: https://en.wikipedia.org/wiki/Domain_Name_System (date 21.03.20)
3. Botnet scams are exploding Google Scholar. URL: http://www.contentagenda.com/articleXml/LN760999245.html?industryid=45177 (date 23.03.20)
4. Detect Fast-Flux Domains Through Response Time Differences IEEE Xplore. URL: https://ieeexplore.ieee.org/abstract/document/6905768 (date 21.03.20)
5. Chahal P. S., and Khurana S. S. TempR: Application of Stricture Dependent Intelligent Classifier for Fast Flux Domain Detection, International Journal of Computer Network & Information Security, vol. 8, 10.11.2016
6. Celik Z. B., and Oktug S. Detection of fast-flux networks using various dns feature sets. p. 868–873.
7. Nafarieh Z., Mahdipur E., Haj Seyed Javadi H. (2019). Detecting Active Bot Networks Based on DNS Traffic Analysis. Journal of Advances in Computer Engineering and Technology, 5(3), 129–142.
8. Learning to link human objects in video and advertisements with clothes retrieval. IEEE Xplore. URL: https://ieeexplore.ieee.org/abstract/document/7727859/ (date 23.03.20)
9. Alieyan K., ALmomani A., Manasrah A., and Kadhum M. M. A survey of botnet detection based on DNS', Neural Comput. Appl., vol. 28, no. 7, p. 1541–1558, 2017.
10. Vapnik V.N. Statictical learning theory. Wiley, 1998, 740 p.
11. Manolakis D., Marden D., Shaw G., Hyperspectral image processing for automatic target detection applications, Lincoln Lab. J., vol. 14, no. 1, pp. 79–114, 2003.
12. Tax D.M., Duin R.P. Support Vector Data Description. Machine Learning 54, 45–66 (2004). https://doi.org/10.1023/B:MACH.0000008084.60811.49
13. Ruirui J., Ding L., Min W., Liu J. The application of SVDD in gene expression data clustering, Proc. Int. Conf. Bioinformat. Biomed. Eng., pp. 371–374, 2008.
14. Malware Domain Blocklist DNS-BH – Malware Domain Blocklist by RiskAnalytics. URL: https://www.malwaredomains.com/ (date 31.03.20)

Рецензія/Peer review : 25.4.2020 р. Надрукована/Printed : 16.6.2020 р.
Стаття рецензована редакційною колегією