УДК 004.056.5
DOI 10.31891/2307-5732-2020-285-3-6

С. М. ЛИСЕНКО, В. І. КОМАРОВ
Хмельницький національний університет

МЕТОД ТА ЗАСОБИ ІДЕНТИФІКАЦІЇ БОТ-МЕРЕЖ, ЩО ВИКОРИСТОВУЮТЬ ТЕХНОЛОГІЮ «ПОТІК ДОМЕНІВ»

У роботі представлено метод ідентифікації бот-мереж, що використовують технологію «потік доменів». Метод дозволяє виявляти як відомі, так і нові невідомі раніше загрози на основі комплексного аналізу DNS-трафіку. Даний метод поєднує в собі опрацювання збоїв у DNS-запитах, використання частотного лексичного аналізу доменних імен та аналіз множини ознак отриманих з DNS-повідомлень за допомогою алгоритму машинного навчання Random Forest, що дозволяє підвищити ефективність та достовірність виявлення даного типу бот-мереж, а також дає змогу виявляти атаки на ранніх стадіях або навіть до їх виникнення. Запропонований метод може бути основою для побудови програмного забезпечення систем виявлення бот-мереж, що використовують технологію «потік доменів».

Ключові слова: бот-мережа, потік доменів, шкідливе програмне забезпечення, Random Forest, DNS.

S. LYSENKO, V. KOMAROV
Khmelnytskyi National University

METHOD AND SOFTWARE FOR DOMAIN-FLUX BOTNET IDENTIFICATION

The purpose of this paper is to develop a method for detecting domain-flux botnet. In this paper, we focus on detecting domain-flux botnets based on Domain Name System (DNS) traffic features. We have explored the peculiarities of the domain-flux botnets and developed a botnet model based on DNS, DNS traffic model and model of the detection process determine all features. This method passively captures all DNS traffic from network and then extract all useful data from each DNS message. This method combines handling DNS query failures, the use of frequency domain lexical analysis and the analysis of multiple features derived from DNS messages using the Random Forest machine learning algorithm. We have analyzed a large number of legitimate domains and pseudorandom domain names generated by different domain-flux botnets to get expected values for domain names generated by humans and bots. In addition, this method use white list database to filter known domain names queries. The method allows to identify both known and new previously unknown threats based on a comprehensive analysis of DNS traffic. Comprehensive analysis of DNS traffic improves the efficiency and reliability of the detection of this type of botnets, and allows the detection of attacks in the early stages or even before they occur. In order to evaluate the effectiveness of the proposed approach, Random Forest machine learning algorithm has applied to train predictive model for our detection system. This proposed scheme has implemented and tested in a real local area network. The experimental results show that our proposed method achieves the highest detective efficiency with an average overall true positive rate of up to 96.08% and a false positive rate of 0.8 %. In addition, the proposed method can be the basis for the construction of other software systems for detection of domain-flux botnets.

Keywords: botnet, domain-flux, malware, Random Forest, DNS.

Вступ. Бот-мережі відіграють важливу роль у розповсюдженні зловмисних програм, і вони широко використовуються для поширення шкідливої діяльності в інтернеті. Бот-мережі часто зловживають доменними іменами, оскільки DNS трафік, як правило, не фільтрований або дозволений через брандмауер, тим самим забезпечується стійкий і безперешкодний канал зв'язку [1].

«Потік доменів» – це техніка для збереження шкідливої бот-мережі в роботі шляхом постійної зміни доменного імені Command and Control (C&C) сервера. Доменні імена змінюються з часом на основі певного алгоритму, який відомий лише власнику бот-мережі, що ускладнює виявлення шкідливого трафіку, серверів команд та управління [1]. Зареєструвати, заблокувати або закрити ці доменні імена важко. Навіть відстеження стану цих доменних імен вимагає великої кількості різних ресурсів. Використовуючи цю техніку, бот-мережа може гнучко переносити свої C&C сервери на кілька доменних імен [2].

Одним з підходів до ідентифікації бот-мереж є залучення механізмів машинного навчання, що дозволяє виявляти атаки на ранніх стадіях або навіть до їх виникнення. Такі підходи покладаються на евристичні механізми і нечіткі відповідності, що надає перевагу перед звичайними методами, зокрема перевірці сигнатур.

Тому актуальною задачею є розроблення методу та засобів виявлення бот-мереж, що використовують технологію «потік доменів». Виявлення нових невідомих раніше загроз має здійснюватися на основі поєднання всіх знань про бот-мережі, які використовують технології ухилення на основі DNS. Використання даної технології ухилення може бути виявлене шляхом аналізу ознак, вилучених з DNS-повідомлень, за допомогою машинного навчання. Метод повинен забезпечувати виявлення атак бот-мереж, що використовують технологію «потік доменів», на ранніх стадіях або навіть до їх виникнення.

Пов'язані роботи. Сучасні бот-мережі, зазвичай використовують технологію «потік доменів» та алгоритм генерації домену (DGA), щоб генерувати велику кількість псевдовипадкових доменних імен [1]. Зазвичай боти бот-мережі генерують велику кількість запитів DNS, зареєстрованих на одну і ту ж IP-адресу, і вони часто генерують багато збоїв у DNS-трафіку [3]. Невдалі DNS-запити можуть вказувати на наявність ботів на клієнтах, тоді як успішні запити, які відбуваються в часі поруч з невдалими, ймовірно пов'язані з неінфікованими клієнтами.

У роботі [3] описано метод виявлення алгоритмічно сформованих доменних імен та представлений підхід для виявлення DGA за допомогою частотного аналізу розподілу символів та зважених балів доменних імен. Доцільність підходу демонструється з використанням ряду законних доменів та ряду зловмисних

алгоритмічно створених доменних імен. Як зазначено у даній роботі, розроблений метод показує хороші результати виявлення, але використовує лише букви для частотного аналізу доменних імен. Проте велика частина доменів також містить в собі цифри 0–9, та також символ дефіс «–».

У роботі [4] також зосереджено увагу на виявленні бот-мереж, що використовують технологію «потік доменів», на основі функцій трафіку системи доменних імен (DNS). У роботі було проаналізовано велику кількість законних доменів, а також псевдовипадкових доменних імен і було виявлено, що в правилах побудови доменних імен є чітке зміщення. Було зроблено частотний розподіл буквено-цифрових символів для знаходження очікуваного значення доменного імені та аналіз цього розподілу алгоритмами машинного навчання. Результати експериментів показують, що запропонований метод досягає найвищої ефективності виявлення для алгоритмів дерева рішень (J48) із середньою загальною точністю до 92,3 % та Random Forest із середньою загальною точністю 91,6 %.

У роботі [5] представлено прототип системи виявлення бот-мереж, яка використовує пасивний аналіз трафіку DNS для виявлення присутності бот-мережі в локальній мережі. В роботі використовували алгоритм машинного навчання Naive Bayes. Він проходив навчання на ознаках, витягнутих як з доброякісних, так і зловмисних слідів трафіку DNS. Оскільки запропонований метод оснований на аналізі DNS-трафіку, він дозволяє раннє виявлення ботів у мережі. Крім того, метод не залежить від кількості ботів, що працюють в локальній мережі, і ефективний, коли присутня лише невелика кількість заражених машин. Проте запропонований підхід, спираючись на мінімальний набір особливостей, страждає від високої помилкової негативної оцінки (false negative rate).

Таким чином, методи [3–5] демонструють непогані результати виявлення, але, разом з тим, мають високий рівень хибних спрацювань (false negative rate). Крім того, дані методи використовують для виявлення лише частотний аналіз доменних імен або покладаються на невелику кількість ознак, отриманих з DNS-трафіку, а це означає, що при зміні алгоритму генерації доменів або при зміні поведінки ботів бот-мережі точність виявлення буде суттєво падати.

Метод виявлення бот-мереж, що використовують технологію «потік доменів» на основі алгоритму Random Forest

В роботі пропонується метод виявлення бот-мереж, що використовують технологію «потік доменів» на основі використання алгоритму Random Forest («випадковий ліс»). Метод дозволяє забезпечити виявлення бот-мереж, що використовують технологію «потік доменів» на основі ознак характерних даній технології шкідливого програмного забезпечення.

Робота методу полягає у відслідковуванні мережевих пакетів, зокрема DNS-трафіку, формуванні множини ознак, які вказують на діяльність бот-мережі, що використовує технологію «потік доменів», формування та використання білих списків доменних імен для фільтрування трафіку та відсіювання заздалегідь відомих DNS адрес. Крім того застосовуватиметься частотний лексичний аналіз доменних імен для виявлення тих доменів, які найімовірніше сформовані алгоритмічно, а також здійснення висновку на основі алгоритму дерева рішень – Random Forest.

Метод складається з 3-х етапів: етап підготовки, етап навчання та етап виявлення. Укрупнена схема функціонування методу виявлення бот-мереж, що використовують технологію «потік доменів» подана на рис. 1.

Рис. 1. Укрупнена схема функціонування методу виявлення бот-мереж, що використовують технологію «потік доменів»

Етап підготовки включає наступні кроки:

1) проведення аналізу, побудова моделей та визначення ключових ознак, які будуть застосовуватись для ідентифікації бот-мереж, що використовують технологію «потік доменів»;

2) збір тестових даних (мережевого трафіку) для навчання.

Етап навчання включає наступні кроки:

– завантаження тестових даних (мережевого трафіку);

– перетворення даних – у більшості випадків наявні дані не підходять для використання безпосередньо для навчання моделі машинного навчання, необхідні дані потрібно попередньо обробити;

– частотний лексичний аналіз доменних імен;

– формування бази даних білих списків доменних імен;

– тренування моделі – побудова «випадкового лісу» за допомогою алгоритму Random Forest, на основі ознак, визначених на етапі підготовки для ідентифікації бот-мереж, що використовують технологію «потік доменів»;

– оцінка моделі.

Етап виявлення діяльності бот-мереж, що використовують технологію «потік доменів»:

– моніторинг мережевого трафіку;

– фільтрування DNS-трафіку, яке використовує відсіювання відомих DNS-запитів, які містять легітимні доменні імена;

– збір усіх наявних параметрів та ознак у відфільтрованому зібраному трафіку;

– виявлення груп, в яких DNS-запит є невдалим;

– виявлення запитів, в яких доменні імена за методом статистичного аналізу найімовірніше сформовані алгоритмічно;

– співставлення кількох груп ознак та їх аналіз за допомогою штучного інтелекту та методу машинного навчання Random Forest;

– формування висновків.

Загальна схема функціонування методу виявлення бот-мереж, що використовують технологію «потік доменів», надана на рис. 2.

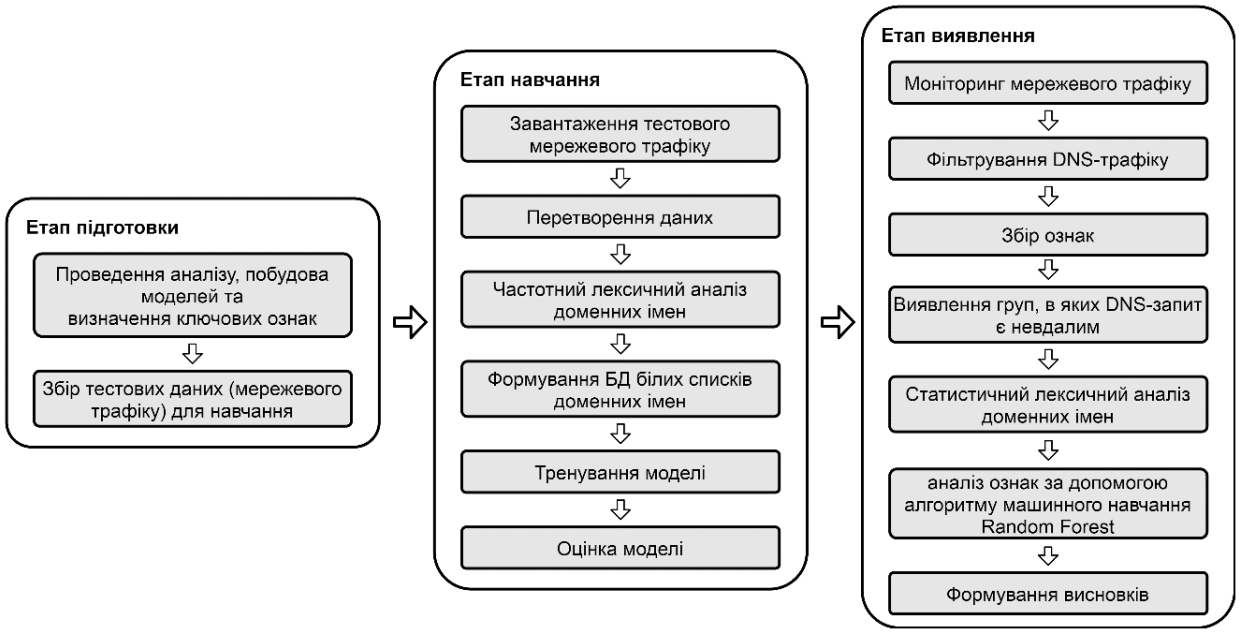

Рис. 2. Загальна схема функціонування методу виявлення бот-мереж, що використовують технологію «потік доменів»

Етап підготовки методу виявлення бот-мереж, що використовують технологію «потік доменів»

На етапі підготовки ***першим кроком*** є проведення аналізу, побудова моделей та визначення ключових ознак, які будуть застосовуватись для ідентифікації бот-мереж, що використовують технологію «потік доменів». Оскільки виявлення бот-мереж, що використовують технологію «потік доменів», ґрунтується на моделі бот-мереж з урахуванням системи доменних імен та моделі DNS-трафіку, то важливою задачею є розроблення моделі DNS-трафіку та DNS-пакета.

Представимо модель DNS-трафіку у вигляді кортежу:

$$DNS_{traffic} = \langle M, C, S, D \rangle, \quad (1)$$

де M – множина DNS-повідомлень надісланих та отриманих від множини C комп'ютерних систем мережі, $M = M_O \cup M_I$, де M_O та M_I – множина вихідних та вхідних DNS-повідомлень, відповідно;

C – множина комп'ютерних систем мережі;

S – множина DNS-серверів, до яких було надіслано та отримано DNS-запити та DNS-відповіді відповідно, $S = S_L \cup S_N$, де S_L та S_N – множина локальних та нелокальних DNS-серверів відповідно;

D – множина запитаних доменних імен множиною C комп'ютерних систем мережі, $D = \{d_j\}_{j=1}^{N_D}$, де N_D – кількість різних доменних імен.

Враховуючи поля вхідного DNS-повідомлення, дані з яких можуть бути використані для виявлення бот-мереж, що використовують технологію «потік доменів», згідно стандартів RFC 1034 [6] та RFC 1035 [7], представимо модель DNS-повідомлення $R \in M = \{M_j\}_{j=1}^{N_M}$, де N_M – кількість DNS-повідомлень, у вигляді кортежу:

$$R = \langle R_{Mac}, R_{IP}, R_{Port}, R_T, \langle R_H, R_{Req}, R_{Ans}, R_{Ath}, R_{Add} \rangle \rangle, \quad (2)$$

де R_{Mac} – MAC-адреса КС, що здійснювала DNS-запит;

R_{IP} – IP-адреса джерела DNS-пакета;

R_{Port} – порт джерела DNS-пакета;

R_T – час надходження DNS-пакета;

$R_H, R_{Req}, R_{Ans}, R_{Ath}, R_{Add}$ – секції DNS-повідомлення: заголовок (Header), секція запиту (Question), секція відгуків (Answer), секція серверів імен (Authority), секція додаткової інформації (Additional) відповідно.

Відповідно до [6,7] заголовок DNS-повідомлення опишемо наступним чином:

$$R_H = \langle R_{H,ID}, R_{H,QR}, R_{H,OP}, R_{H,FLG}, R_{H,RC}, R_{H,QD}, R_{H,AN}, R_{H,NS}, R_{H,AR} \rangle, \quad (3)$$

де $R_{H,ID}$ – унікальний ідентифікатор транзакції, що дозволяє пов'язати DNS-запит та DNS-відгук;

$R_{H,QR}$ – ідентифікатор, який вказує на те, чи є пакет запитом, коли $R_{H,QR} = \mathbf{0}$, чи відповіддю – $R_{H,QR} = \mathbf{1}$;

$R_{H,OP}$ – тип запиту, $R_{H,OP} \in \{0, \dots, \mathbf{3}\}$, де 0 – стандартний, 1 – інверсний, 2 – стан сервера, 3 – резерв;

$R_{H,FLG}$ – прапорці;

$R_{H,RC}$ – код відгуку, $R_{H,RC} \in \{0, \dots, \mathbf{5}\}$, де 0 – немає помилки, 1 – помилка в форматі запиту, 2 – некоректна робота сервера, 3 – доменне ім'я не існує, 4 – сервер не може виконати запит даного типу, 5 – сервер не може виконати запит клієнт, оскільки має адміністративне обмеження безпеки;

$R_{H,QD}$ – кількість записів в секції запитів;

$R_{H,AN}$ – кількість записів в секції відповіді;

$R_{H,NS}$ – кількість записів в секції серверів імен;

$R_{H,AR}$ – кількість записів в секції додаткової інформації.

Структуру секції запиту DNS-повідомлення [6, 7] опишемо наступним чином:

$$R_{Req} = \langle R_{Req,QN}, R_{Req,QT}, R_{Req,QC} \rangle, \quad (4)$$

де $R_{Req,QN}$ – поле QNAME, доменне ім'я, до якого прив'язаний даний запис;

$R_{Req,QT}$ – тип запису DNS, який шукається;

$R_{Req,QC}$ – визначальний клас запиту.

Структуру секції відповіді DNS-повідомлення [6,7] може бути описана наступним чином:

$$R_{Ans} = R_{Ath} = R_{Add} = \langle R_{A,N}, R_{A,T}, R_{A,C}, R_{A,TTL}, R_{A,RDL}, R_{A,RD} \rangle, \quad (5)$$

де $R_{A,N}$ – поле NAME, ідентичне полю $R_{Req,QN}$;

$R_{A,T}$ – поле TYPE, визначає формат і призначення даного ресурсного запису;

$R_{A,C}$ – поле CLASS, клас ресурсного запису;

$R_{A,TTL}$ – допустимий час зберігання даного ресурсного запису в кеші невідповідального DNS-сервера;

$R_{A,RDL}$ – довжина поля даних;

$R_{A,RD}$ – поле даних, формат і зміст якого залежить від типу запису.

Алгоритм вилучення ознак з вхідних DNS-повідомлень, щодо певного доменного імені представимо у вигляді кортежу:

$$AF = \langle D, M, R, V \rangle, \quad (6)$$

де V – множина ознак, які вказують на застосування технології «потік доменів».

Множина ознак, що вказують на діяльність бот-мережі, яка використовує технологію ухилення «потік доменів», складається з наступних елементів:

$$V = \{N_{dom}, S_{bit}, T_{ttl}, L_{dom}, N_{num}, W_{dom}\}, \quad (7)$$

де N_{dom} – кількість доменних імен, які спільно використовують IP-адресу;

S_{bit} – бінарна ознака успішності DNS-запиту (якщо S_{bit} = false – невдалий, а якщо S_{bit} = true – вдалий);

T_{ttl} – TTL-період;

L_{dom} – довжина доменного імені;

N_{num} – кількість цифр в доменному імені;

W_{dom} – зважена оцінка частотного лексичного аналізу доменних імен, визначається за формулою:

$$W_{dom} = \frac{\sum_{i=\mathbf{0}}^{n} X_i}{n} \quad (8)$$

де n – кількість літер в доменному імені;

X_i – частота використання *i*-ї літери.

Другим кроком збирається DNS-трафік за допомогою моніторингу мережі через SPAN-порт комутатора мережі, який дублює пакети від одного або декількох портів на окремо взятий порт.

Етап навчання методу ідентифікації бот-мереж, що використовують технологію «потік доменів»

На етапі навчання ***першим кроком*** є завантаження тестових даних, зібраних на другому кроці етапу підготовки. На ***другому кроці*** з тестового мережевого трафіку необхідно вилучити ознаки та дані, які будуть безпосередньо використовуватись для виявлення бот-мереж, що використовують технологію «потік доменів» і представити їх в нормалізованому вигляді.

З тестового мережевого трафіку вибираються наступні дані:

$$PV_j = \begin{Bmatrix} R_{Mac,j}, R_{IP,j}, R_{Port,j}, R_{T,j}, R_{H,ID,j}, R_{H,QR,j}, \\ R_{H,RC,j}, R_{A,N,j}, R_{Req,QN,j}, R_{A,TTL,j} \end{Bmatrix} \quad (9)$$

$j = d_1, \dots, d_{N_D}$.

Розділимо наші дані на групи представлені наступними кортежами:

1. $G_{1,j} = \langle R_{Mac,j}, R_{IP,j}, R_{Port,j} \rangle$ – для відслідковування запитів від конкретного адресу;
2. $G_{2,j} = \langle R_{H,ID,j}, R_{H,QR,j} \rangle$ – унікальний ідентифікатор транзакції та ідентифікатор, який вказує на те, чи є пакет запитом чи відповіддю, що дозволяє пов'язати DNS-запит та DNS-відгук;
3. $G_{3,j} = \langle R_{H,RC,j} \rangle$ – код відгуку та бінарна ознака успішності DNS-запиту дозволяють дізнатись чи DNS-запит пройшов без помилок та був успішним;
4. $G_{4,j} = \langle R_{T,j}, R_{A,TTL,j} \rangle$ – час надходження пакету та TTL період дозволяють відслідковувати чи від конкретного адресу не надходили запити на інші доменні імена;
5. $G_{5,j} = \langle R_{A,N,j}, R_{Req,QN,j} \rangle$ – остання група буде використовуватись для частотного аналізу доменних імен.

Поєднавши дані з наступних груп $N_{dom} = G_1 \cup G_2 \cup G_4$, отримують кількість доменних імен, які спільно використовують IP-адресу. З групи G_3 вибираються дані для бінарної ознаки успішності DNS-запитів. Дані з групи G_4 використовуються для визначення T_{ttl} – TTL-період. Група G_5 дає можливість підрахунку наступних ознак: L_{dom}, N_{num} та n (кількість літер в доменному імені без врахування доменних зон) для формули зваженої оцінки лексичного аналізу доменних імен W_{dom}.

На ***третьому кроці*** використовується частотний лексичний аналіз, обраховуючи частоту появи букв, цифр та символів в доменних іменах сформованих людиною та алгоритмічно, для використання цих даних в обрахунку зважених оцінок частотного лексичного аналізу доменних імен W_{dom}, а саме для параметра X_i (частота використання i -ї літери). Модель вхідних даних представимо кортежем:

$$DData = \langle LDD, MDD \rangle, \quad (10)$$

де LDD – множина доменних імен сформованих людиною, $LDD = TDD \cup WLDD$, де TDD – множина топ використовуваних доменних імен; $WLDD$ – множина білих списків доменних імен;

MDD – множина доменних імен сформованих алгоритмом генерації доменних імен (DGA), $MDD = BLDD \cup DGAD$, де $BLDD$ – множина чорних списків доменних імен; $DGAD$ – множина доменних імен сформованих алгоритмічно.

На ***четвертому кроці*** формується базу даних білих списків доменних імен, необхідних нам для початкової фільтрації DNS-трафіку. За допомогою співставлення доменних імен отриманих з трафіку та доменних імен з бази даних білих списків дозволить відсіяти не шкідливі запити, які ідуть від легальних серверів. База даних білих списків доменних імен формується з множини LDD, вибираючи унікальні записи з кожної з множин. Множина LDD включає в себе множину TDD – множина доменних імен з різних джерел, таких як Alexa, Quantcast, Cisco Umbrella, DomCop, The Majestic та інші популярні джерела.

На ***п'ятому кроці*** здійснюється тренування моделі та побудова «випадкового лісу» за допомогою алгоритму Random Forest, на основі ознак, визначених на етапі підготовки та перетворені на попередньому кроці під використання в машинному навчанні для ідентифікації бот-мереж, що використовують технологію «потік доменів». Початок алгоритму випадкових лісів починається з випадкового вибору k ознак із загальної кількості m ознак, визначених на етапі підготовки для ідентифікації бот-мереж, що використовують технологію «потік доменів» [8].

Далі використовуються довільно вибрані k ознаки, щоб знайти кореневий вузол, використовуючи найкращий розділений підхід для дерева. Наступним етапом обчислюються дочірні вузли, використовуючи той самий найкращий розділений підхід. Перші 3 етапи повторюються, поки не сформується дерево з кореневим вузлом та буде досягнуто листкового вузла. На кінець, повторюються 1–4 етапи, щоб створити n випадково створених дерев. Ці випадково створені дерева утворюють випадковий ліс [8].

Для найкращого розподіленого підходу пропонується використовувати такий критерій як приріст інформації та коефіцієнт (індекс) Джині. Значення сортуються так, що атрибут з високим значенням розміщується в корені. Використовуючи інформаційний приріст як критерій, оцінюється інформація, що

міститься в кожному атрибуті. Обрахувавши міру ентропії для кожного атрибута обчислюється їх інформаційний приріст [9]. Міра ентропії розраховується за формулою:

$$E(X) = -\sum_{x \epsilon X} p(x) \log p(x), \tag{11}$$

де $p(x)$ – частота ознаки x у даному вузлі; X – кількість унікальних ознак.

Інформаційний приріст обчислює очікуване зменшення ентропії за рахунок сортування за атрибутом та розраховується за формулою:

$$IG(T, X) = E(T) - E(T, X), \tag{12}$$

де T – цільова ознака; X – ознака, яку потрібно розділити.

Індекс Джині – це показник нерівності розподілу деякої величини, який вказує наскільки часто випадково вибраний елемент буде неправильно ідентифікований. Індекс Джині приймає значення від 0 і до 1, де 0 означає абсолютну рівність, а 1 позначає повну нерівність. Це означає, що слід віддати перевагу атрибуту з нижчим показником Джині [9]. Коефіцієнт Джині розраховується за наступною формулою:

$$I_G(f) = \sum_{i=1}^{m} f_i(1 - f_i) = \sum_{i=1}^{m} (f_i - f_i^2) = \sum_{i=1}^{m} f_i - \sum_{i=1}^{m} f_i^2 = 1 - \sum_{i=1}^{m} f_i^2, \tag{13}$$

де f_i – частота ознаки i у даному вузлі; m – кількість унікальних ознак.

Прогнозування навченого за допомогою алгоритму випадковий ліс складається з наступних кроків: 1. беруться тестові ознаки та використовуються правила кожного випадково створеного дерева рішень для прогнозування результату; 2. підраховуються голоси за кожен прогнозований результат; 3. вибирається прогнозований результат з найвищою оцінкою як остаточний прогноз алгоритму випадковий ліс[8].

Шостим кроком оцінюється продуктивність системи, яка зазвичай визначається такими показниками: True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), False Negative Rate (FNR), Accuracy (ACC) [4], коефіцієнт помилок (ERR) та коефіцієнт кореляції Matthews (MCC). Для визначення оцінок продуктивності системи використовуються наступні поняттями та показники: condition positive (P) – кількість реальних позитивних випадків виявлення запитів бот-мережі; condition negative (N) – кількість реальних негативних випадків виявлення запитів бот-мережі; True Positive (TP) – правильно ідентифіковані запити бот-мереж; True Negative (TN) – правильно ідентифіковано звичайний запит; False Positive (FP) – неправильно визначені запити бот-мереж; False Negative (FN) – помилково визначені запити бот-мережі як звичайний запит[4].

True Positive Rate (TPR) вимірює відсоток правильно класифікованих запитів бот-мереж, що використовують технологію «потік доменів». TPR розраховується за формулою: $TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$. False Positive Rate (FPR) вимірює відсоток звичайних запитів, хибно класифікованих як запити бот-мереж. FPR розраховується за формулою: $FPR = \frac{FP}{N} = \frac{FP}{FP + TN}$. True Negative Rate (TNR) вимірює відсоток правильно класифікованих звичайних запитів. TNR розраховується за формулою: $TNR = \frac{TN}{N} = \frac{TN}{TN + FP}$. False Negative Rate (FNR) вимірює відсоток запитів бот-мереж, хибно класифікованих як звичайні запити. FNR розраховується за формулою: $FNR = \frac{FN}{P} = \frac{FN}{FN + TP}$. Позитивні прогнозовані значення (Accuracy (ACC)) вимірює ступінь близькості між вимірами класифікованих запитів та сумою фактичних запитів бот-мереж та звичайних запитів. ACC розраховується за формулою: $ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$. Коефіцієнт помилок (або помилка неправильної класифікації) вимірює відношення неправильно класифікованих запитів до загальної кількості класифікованих запитів бот-мереж та звичайних запитів. Коефіцієнт помилок розраховується за формулою: $ERR = \frac{FP + FN}{TP + TN + FP + FN} = 1 - ACC$ [4]. Коефіцієнт кореляції Matthews (Matthews Correlation Coefficient (MCC)) використовується в машинному навчанні як міра якості двійкових класифікацій. MCC розраховується за формулою: $MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$.

Етап виявлення методу ідентифікації бот-мереж, що використовують технологію «потік доменів»

На ***першому кроці*** отримують DNS-трафік за допомогою моніторингу мережі через SPAN-порт комутатора мережі (Switched Port Analyzer), який дублює пакети від одного або декількох портів на окремо взятий порт. Отримання DNS-трафіку через SPAN-порт опишемо наступною функцією:

$$f_{get} \Rightarrow \{DNS | DNS \in Traff\} \xrightarrow{f_{get}} DNS_{real}, \tag{14}$$

де $Traff$ – мережевий трафік, який проходить через SPAN-порт комутатора мережі; DNS – множина DNS-пакетів трафіку.

На ***другому кроці*** виконується пошук доменних імен з множини N_D доменних імен отриманих з DNS-трафіку в множині LDD доменних імен бази даних білих списків. Отриманий трафік представимо кортежем:

$$DNS_{real} = \langle M, C, S, D \rangle \tag{15}$$

З множини M DNS-повідомлень вибирається множина M_I вхідних DNS-повідомлень, де $M_I \in M$. Для кожного M_I переглядається секція R_{Req} запиту DNS-повідомлення. Далі дістається значення поля $R_{Req,QN}$. Функцію пошуку доменних імен отриманих з DNS-трафіку в базі даних легітимних доменних імен представимо наступним чином:

$$f_{find} \Rightarrow R_{Req,QN} \cap LDD \xrightarrow{f_{find}} RF_{Req,QN}, \tag{16}$$

де $RF_{Req,QN}$ – множина знайдених доменних імен в множині LDD, $RF_{Req,QN} \in R_{Req,QN}$.

Дану множину $RF_{Req,QN}$ потрібно відняти від множини $R_{Req,QN}$ для отримання відфільтрованої множини доменних імен для подальшого аналізу. Опишемо дану операцію такою функцією:

$$f_1 filter \Rightarrow R_1(Req, QN) \; [\![RF]\!]_{\downarrow}(Req, QN) \; (\rightarrow \neg(\; f_{\downarrow} filter \;)) \; [\![R']\!]_{\downarrow}(Req, QN), \tag{17}$$

де $R'_{Req,QN}$ – множина доменних імен не знайдених в базі даних білих списків доменних імен.

На ***третьому кроці*** подається множина даних, отриманих з вхідних DNS-повідомлень щодо певного доменного імені наступним чином:

$$PV = \begin{Bmatrix} R'_{Mac}, R'_{IP}, R'_{Port}, R'_{T}, R'_{H,ID}, R'_{H,QR}, \\ R'_{H,RC}, R'_{A,N}, R'_{Req,QN}, R'_{A,TTL} \end{Bmatrix} \tag{18}$$

Аналогічно другому кроку етапу навчання перетворюються отримані дані в ознаки, розбиваючи дані на групи: $G_1 = \langle R'_{Mac}, R'_{IP}, R'_{Port} \rangle$, $G_2 = \langle R'_{H,ID}, R'_{H,QR} \rangle$, $G_3 = \langle R'_{H,RC} \rangle$, $G_4 = \langle R'_{T}, R'_{A,TTL} \rangle$, $G_5 = \langle R'_{A,N}, R'_{Req,QN} \rangle$. Після перетворення даних отримують множину нормалізованих ознак, необхідних для аналізу та виявлення бот-мереж, що використовують технологію «потік доменів»:

$$V = \{N_{dom}, S_{bit}, T_{ttl}, L_{dom}, N_{num}, W_{dom}\} \tag{19}$$

На ***четвертому кроці*** виявляються групи запитів, в яких DNS-запит є невдалим. Використовуючи групу G_3 визначається бінарна ознака успішності запиту за допомогою наступної функції:

$$f_{RC} \Rightarrow R'_{H,RC} = 0 \rightarrow true \vee R'_{H,RC} \in \{1, \dots, 5\} \rightarrow false \xrightarrow{f_{RC}} S_{bit} \tag{20}$$

На ***п'ятому кроці*** виявляються запити, в яких доменні імена за методом статистичного аналізу найімовірніше сформовані алгоритмічно. Береться множина $R'_{Req,QN}$ та для кожного доменного імені, визначається значення зваженої оцінки частотного лексичного аналізу за формулою:

$$W_{dom,j} = \frac{\sum_{i=0}^{n_j} X_{i,j}}{n_j}, \tag{21}$$

де j – кількість доменних імен отриманих з DNS'_{real}, $j = d'_1, \dots, d'_{N'_R}$, де $d' \in R'_{Req,QN}$; n_j – кількість літер в j -му доменному імені; $X_{i,j}$ – частота використання i -ї літери j -го доменного імені.

Зважені оцінки W_{dom} частотного лексичного аналізу доменних імен DNS-повідомлень використовуються для подальшого виявлення бот-мереж, що використовують технологію «потік доменів», на основі алгоритму Random Forest, як одна із ознак, що вказують на застосування даної технології ухилення.

На ***шостому кроці*** всі вибрані та проаналізовані дані з відфільтрованого DNS-трафіку об'єднуються в множину ознак, що дозволить виявляти бот-мережі, які використовують технологію «потік доменів», на основі алгоритму Random Forest. Для всіх отриманих згрупованих даних застосовується перетворення та нормалізація, в результаті отримують множину ознак щодо кожного доменного імені, яка матиме наступний вигляд:

$$V_j = \{N_{dom,j}, S_{bit,j}, T_{ttl,j}, L_{dom,j}, N_{num,j}, W_{dom,j}\}, \tag{22}$$

де $j \in R'$ – кількість зібраних DNS-повідомлень після фільтрування.

Сьомим кроком формуються висновки на основі аналізу множини ознак V_j щодо кожного доменного імені алгоритмом машинного навчання Random Forest. Оскільки, використовується бінарна класифікація алгоритмом машинного навчання, то на виході отримують відповідно до кожного запитаного доменного імені результат, який може набувати двох значень: інфікований запит від бота бот-мережі або неінфікований запит.

Експерименти

Для оцінки ефективності даного методу ідентифікації бот-мереж, що використовують технологію «потік доменів», було проведено ряд експериментів. Спочатку було зібрано базу зразків доменних імен сформованих алгоритмічно різними алгоритмами генерації доменів відомих бот-мереж, а також базу легітимних доменів [10].

Для забезпечення неупереджених результатів на етапі тренування набір даних було розділено на дві частини. Перший – 75 % для тренування, а решта 25 % використовується для перевірки правильності. Кількість проаналізованих доменних імен подано в таблиці 1.

Таблиця 1

Кількість доменних імен використаних для навчання та тестування

Набір даних	Зразки	Навчальний набір даних	Тестовий набір даних
DGA	Conficker	75000	25000
	Cryptolocker	75000	25000
	Zeus	75000	25000
	GameoverZeus	1250	417
	NewGameoverZeus	1250	416
	Kraken (v1)	1500	500
	Kraken (v2)	1500	500
	Matsnu	75000	25000
	PushDO	76013	25337
	Ramdo	75000	25000
	Rovnix	75000	25000
	Tinba	76051	25350
Легітимні	Majestic	750000	250000
	Quantcast	348816	116272
	Cisco Umbrella	750000	250000
	Alexa	633245	211083
	DomCom	750000	250000
Всього		**3839625**	**1279875**

Всього було проаналізовано 5119500 доменних імен, серед яких 3839625 (75 %) було відібрано для навчання, а 1279875 (25 %) – використовувались для перевірки правильності. Загальна кількість проаналізованих доменних імен сформованих алгоритмічно – 810084 (15,8 % від загальної кількості проаналізованих доменних імен), серед яких 607564 використано для навчання, а 202520 – для перевірки правильності. Загальна кількість проаналізованих легітимних доменних імен – 4309416 (84,2 % від загальної кількості проаналізованих доменних імен), серед яких 3232061 використано для навчання, а 1077355 – для перевірки правильності.

Було проведено експеримент, використовуючи файл-знімок мережі, який було накопичено шляхом моніторингу мережі через SPAN-порт комутатора. Загальна кількість запитів становила 3547. Також були присутні запити від ботів бот-мережі, що використовують технологію «потік доменів». Загальна кількість їх становила 306. Правильно ідентифіковано 294 запити, що складає 96,08 % від загальної кількості. Загальну кількість правильно та неправильно ідентифікованих запитів подано в таблиці 2.

Таблиця 2

Кількість та правильність ідентифікованих запитів

Назва	Кількість / з загальної кількості
Правильно ідентифіковані інфіковані запити (TP)	294 / 306
Неправильно ідентифіковані інфіковані запити (FN)	12 / 306
Правильно ідентифіковані неінфіковані запити (TN)	3215 / 3241
Неправильно ідентифіковані неінфіковані запити (FP)	26 / 3241

Оцінку продуктивності системи в результаті експериментальних досліджень подано в таблиці 3.

Таблиця 3

Оцінка продуктивності системи

TPR	TNR	FPR	FNR	ACC	ERR	MCC
96,08 %	99,2 %	0,8 %	3,92 %	98,93 %	1,07 %	0,9337

Таким чином, запропонований метод продемонстрував можливість виявлення бот-мереж, що використовують технологію «потік доменів», з високою достовірністю (96,08 %).

Висновки. Запропоновано метод виявлення бот-мереж, що використовують технологію «потік доменів», на основі комплексного аналізу DNS-трафіку. Метод дозволяє виявляти як відомі так і нові невідомі раніше загрози на основі комплексного аналізу DNS-трафіку.

Метод поєднує в собі опрацювання збоїв у DNS-запитах, використання частотного лексичного аналізу доменних імен та аналіз множини ознак отриманих з DNS-повідомлень за допомогою алгоритму машинного навчання Random Forest, що дозволяє підвищити ефективність та достовірність виявлення даного типу бот-мереж, а також дає змогу виявляти атаки на ранніх стадіях або навіть до їх виникнення.

Експериментальні дослідження продемонстрували здатність запропонованого методу до виявлення бот-мереж, що використовують технологію «потік доменів» з високою достовірністю (до 96,08 %).

Література

1. Dodopoulos R. DNS-based Detection of Malicious Activity: master's thesis, Eindhoven University of Technology. 2015.

2. Лисенко С.М. Методи виявлення бот-мереж в комп'ютерних системах / С.М. Лисенко, К.Ю. Бобровнікова, В.С. Харченко // Сучасні інформаційні системи. – 2019. – Т. 3. – С. 87–95.

3. Agyepong E., Buchanan W., Jones K. Detection of Algorithmically Generated Malicious Domain Using Frequency Analysis. International Journal of Computer Science and Information Technology. 2018. DOI: 10.5121/ijcsit.2018.10306.

4. Truong D., Cheng G. Detecting domain-flux botnet based on DNS traffic features in managed network. Security Comm. Networks. 2016. 9: 2338– 2347. DOI: 10.1002/sec.1495.

5. Wielogorska M., O'Brien D. DNS Traffic analysis for botnet detection: Proceedings of the 25th Irish Conference on Artificial Intelligence and Cognitive Science (Dublin, December 7 – 8, 2017). P. 261–271.

6. Mockapetris P. RFC-1034. Domain names – concepts and facilities ISI, 1987. URL: http://www.ietf.org/rfc/rfc1034.txt?number=1034.

7. Mockapetris P. RFC-1035. Domain names – concepts and facilities. ISI, 1987. URL: http://www.ietf.org/rfc/rfc1035.txt?number=1035.

8. Polamuri S. How the Random Forest algorithm works in machine learning. URL: https://data/aspirant.com/2017/05/22/random-forest-algorithm-machine-learing (application date: 13.03.2020).

9. Ronaghan S. The Mathematics of Decision Trees, Random Forest and Feature Importance in Scikit-learn and Spark. URL: https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3 (application date: 13.03.2020).

10. Trost J. Getting Started with DGA Domain Detection Research. URL: http://www.covert.io/getting-started-with-dga-research (application date: 13.03.2020).

References

1. Dodopoulos R. DNS-based Detection of Malicious Activity: master's thesis, Eindhoven University of Technology. 2015.

2. Lysenko S.M. Metody vyiavlennia bot-merezh v kompiuternykh systemakh / S.M. Lysenko, K.Iu. Bobrovnikova, V.S. Kharchenko // Suchasni informatsiini systemy. – 2019. – T. 3. № . – S. 87–95.

3. Agyepong E., Buchanan W., Jones K. Detection of Algorithmically Generated Malicious Domain Using Frequency Analysis. International Journal of Computer Science and Information Technology. 2018. DOI: 10.5121/ijcsit.2018.10306.

4. Truong D., Cheng G. Detecting domain-flux botnet based on DNS traffic features in managed network. Security Comm. Networks. 2016. 9: 2338– 2347. DOI: 10.1002/sec.1495.

5. Wielogorska M., O'Brien D. DNS Traffic analysis for botnet detection: Proceedings of the 25th Irish Conference on Artificial Intelligence and Cognitive Science (Dublin, December 7 – 8, 2017). P. 261–271.

6. Mockapetris P. RFC-1034. Domain names – concepts and facilities ISI, 1987. URL: http://www.ietf.org/rfc/rfc1034.txt?number=1034.

7. Mockapetris P. RFC-1035. Domain names – concepts and facilities. ISI, 1987. URL: http://www.ietf.org/rfc/rfc1035.txt?number=1035.

8. Polamuri S. How the Random Forest algorithm works in machine learning. URL: https://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing (application date: 13.03.2020).

9. Ronaghan S. The Mathematics of Decision Trees, Random Forest and Feature Importance in Scikit-learn and Spark. URL: https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3 (application date: 13.03.2020).

10. Trost J. Getting Started with DGA Domain Detection Research. URL: http://www.covert.io/getting-started-with-dga-research (application date: 13.03.2020).

Надійшла / Paper received: 04.05.2020 Надрукована / Paper Printed : 01.06.2020