DOI 10.31891/2307-5732-2019-273-3-173-179
УДК 621.391 160164

С.М. ЛИСЕНКО, В.О. ЛІСОВИЙ
Хмельницький національний університет

МЕТОД ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ВИЯВЛЕННЯ ШКІДЛИВИХ ЗАПИТІВ В КОМП'ЮТЕРНИХ МЕРЕЖАХ НА ОСНОВІ ПРОТОКОЛУ DNS

В роботі представлено метод, спрямований на виявлення і блокування доменів, які запитуються в потоковому трафіку DNS і використовуються для зловмисного видалення даних DNS. Згідно з методом потоковий трафік DNS збирається і перетворюється на вектори відповідності доменів. Після цього попередньо навчений класифікатор класу використовується для виявлення доменів, які обмінюються даними через DNS. Одразу після цього запити на домени, які класифікуються як такі, що використовуються для обміну даними, блокуються на невизначений час. Метод дозволяє його реалізацію в DNS-серверах, якщо вони підтримують протоколювання DNS-трафіку і чорного списку доменів. Описаний метод дає можливість виявляти шкідливі потоки пакетів даних серед звичайних, а постійний моніторинг визначених шкідливих потоків дає можливість виявити зловмисника та дозволяє ізолювати звичайний мережевий потік даних від шкідливого.

Ключові слова: DNS протокол, DoS-атака, кіберзагроза, кібератака, виявлення кібератак, мережний трафік.

S. LYSENKO, V. LISOVYI
Khmelnytskyi National University

METHOD AND SOFTWARE FOR MALICIOUS QUERIES DETECTION IN THE NETWORKS BASED ON THE DNS PROTOCOL

Today's network attacks are one of the most dangerous cyber threats, as well as one of the main sources of illegal earnings on the Internet. Most often, such attacks are carried out by botnets and used for DDoS attacks (distributed attacks such as "denial of service"), collecting confidential information, sending spam, using adware, phishing, clicking clicks (click traffic), creating spam searches, the use of infected computers for storing illegal material (pirated software, etc.) and as proxies for anonymizing access to the Internet. Antivirus software using signature-based technologies cannot normally detect harmful zero-day software, since such new signatures are not available for newly created malware. An analysis of known methods to combat cyberattacks shows their lack of efficiency, so building a new method for detecting cyber-threats is an extremely urgent task. The article presents a new method for detecting DNS-attack type. Article proposes a method for detecting and blocking domains that are requested in DNS streaming traffic and used to maliciously delete DNS data. According to the method, DNS traffic is collected and converted into domain matching vectors. After that, a pre-trained classifier class is used to identify domains that exchange data through DNS. Immediately thereafter, queries for domains that are classified as used for data exchange are blocked indefinitely. The method allows its implementation on DNS servers if they support DNS traffic logging and the blacklist of domains. The described method makes it possible to detect harmful streams of data packets among ordinary, and continuous monitoring of certain malicious flows makes it possible to detect an attacker and allows to isolate the usual network data stream from the harmful one.

Keywords: DNS protocol, DNS tunnelling, DoS-attack, cyberattack, cyberattacks detection, network traffic.

Вступ

На сьогоднішній день мережні атаки є однією з найбільш небезпечних кіберзагроз [1], а також одним з основних джерел нелегального заробітку в мережі Інтернет. Найчастіше такі атаки здійснюють бот-мережі і використовуються для атак DDoS (розподілені атаки типу «відмова в обслуговуванні»), збору конфіденційної інформації, розсилання спаму, застосування засобів нав'язування реклами, фішингу, накрутки клік-лічильників (клікфрод), створення пошукового спаму, використання інфікованих комп'ютерів для зберігання нелегального матеріалу (піратське ПЗ тощо) та в якості проксі-серверів для анонімізації доступу в мережі Інтернет. Щороку по всьому світу бот-мережами інфікується близько 500 млн персональних комп'ютерів, кожну секунду – близько 18 ПК. За останній рік бот-мережі нанесли $110 млрд збитків світовій економіці [1].

Ряд технологій ухилення від виявлення бот-мереж базуються на використанні системи доменних імен (DNS) – DNS-тунелювання (DNS-tunnelling), «швидкозмінні» мережі (fast-flux service network), технологія «потік доменів» (domain flux) та періодична зміна IP-відображення для шкідливого домена (cycling of IP mapping) [2–5].

На сьогоднішній день існує багато підходів виявлення шкідливого мережного трафіку [6, 7]. В [5] запропоновано систему виявлення шкідливих доменів на основі пасивного DNS-аналізу, що здійснює класифікацію доменних імен за 4 групами ознак, які можуть бути вилучені з DNS-трафіка: (1) часові ознаки; (2) ознаки, що базуються на DNS-відповідях; (3) ознаки, що базуються на значеннях TTL; (4) ознаки, що базуються на доменному імені. В [6] запропоновано підхід, який дозволяє виявляти доменні імена бот-мереж, які використовують метод «швидкозмінних» мереж. Висновок щодо шкідливості доменного імені здійснюється за рядом ознак, отриманих на основі аналізу даних, вилучених з A-, NS-, SOA- та BGP-запитів щодо доменного імені: значення часу життя A-записів (TTL-періоду), IP-адрес в A- та NS-записах, а також значення таймера "retry" (визначає, як довго вторинний сервер імен повинен чекати перед тим, як зробити повторну спробу запиту первинного сервера щодо зміни серійного номера зони, якщо попередня спроба була невдалою). В [7] проведено аналіз можливостей використання DNS-запитів для встановлення прихованих комунікацій. Надано оцінку статистичних методів виявлення аномалій у вмісті DNS-пакетів

шляхом порівняння ймовірнісних розподілів нормального та тунельованого DNS-трафіка. З метою висвітлення потенційної загрози розглянуто можливість здійснення контрзаходів з боку зловмисника. В [8, 9] проведено аналіз великої кількості реальних DNS-запитів та обчислено порогові значення довжини запитаного імені хоста та кількості унікальних символів в ньому, що дозволяють відрізнити легітимний DNS-трафік від тунельованого.

Описані методи мають наступні недоліки: необхідність залучення інформації, отриманої від інших сервісів (WHOIS тощо); необхідність активного DNS-зондування, тому неможливість реалізації на основі пасивного аналізу DNS-трафіка; зосередження на виявленні вузького кола шкідливого ПЗ.

Метод та програмне забезпечення виявлення шкідливих запитів в комп'ютерних мережах на основі протоколу DNS

Запропонований метод являє собою постійно діючий процес, спрямований на виявлення і блокування доменів, які запитуються в потоковому трафіку DNS і використовуються для зловмисного видалення даних DNS. По-перше, потоковий трафік DNS збирається і перетворюється на вектори відповідності доменів. Після цього попередньо навчений класифікатор класу використовується для виявлення доменів, які обмінюються даними через DNS. Відразу після цього запити на домени, які класифікуються як такі, що використовуються для обміну даними, блокуються на невизначений час. Метод дозволяє його реалізацію в DNS-серверах, якщо вони підтримують протоколювання DNS-трафіку і чорного списку доменів (рис. 1).

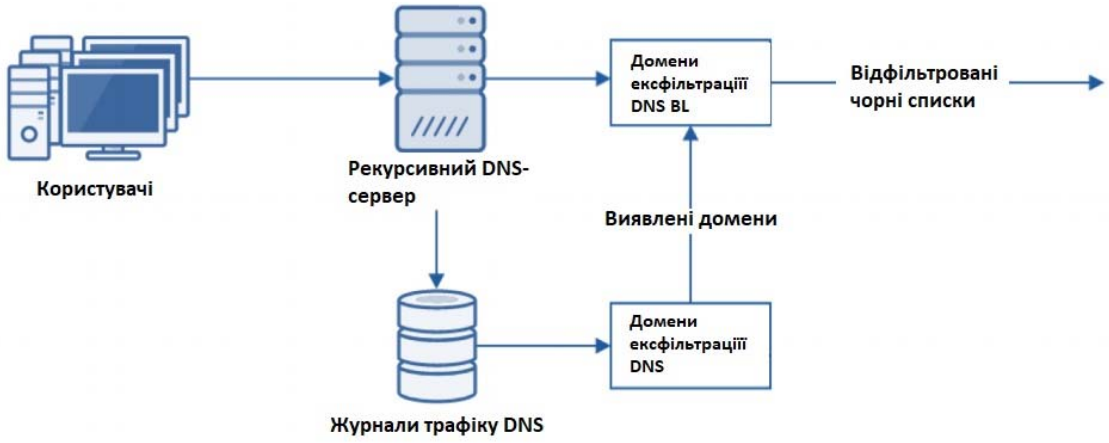

Рис. 1. Архітектура системи виявлення шкідливих запитів

Метод базується на наступних припущеннях:

1) Аномалія обміну даними DNS: на основі загальної схеми для видалення даних (підрозділ II-B) встановлюється аномалія трафіку DNS при використанні для обміну даними. Ми припускаємо, що домени, які використовуються для обміну даними через DNS, ймовірно, будуть характеризуватися більше, ніж середні запити та відповіді, кодовані корисні навантаження (підзапити), а також безліч унікальних запитів.

2) Використання єдиного домену: на основі нещодавно виявленого шкідливого програмного забезпечення ми припускаємо, що для ексфільтрації використовується єдиний домен, що дозволяє виявити його аномалію запитів і заперечення його запитів DNS для припинення витоку. Більш складний випадок з кількох доменів обговорюється в майбутньому розділі роботи.

Запропонований метод має три регульовані параметри (підсумовані в таблиці 1). Налаштування параметрів можуть впливати на швидкість виявлення та їх затримку.

Налаштування ν. Вище прийнятна швидкість помилкових спрацьовувань (позначена як ν) потенційно збільшить швидкість виявлення за рахунок додаткових помилкових позитивних доменів. Рекомендується встановити значення, яке не перевищує 0,1% (через нестачу загрози), і зменшувати його до того моменту, коли буде досягнуто прийнятну норму помилкових позитивних доменів, яку буде перевірено експертом безпеки.

Таблиця 1

Параметри методу

Ім'я	Опис	Приклад значення
ν	Прийнятна швидкість помилкових позитивних доменів	$2 \cdot 10^{-5}$
λ	Частота збору та класифікації даних [хвилини]	15
n_S	Розмір вікна для перевірки (кількість наборів збору даних)	24

У роботі $\nu = 2 \cdot 10^{-5}$, що дає лише 18 помилкових позитивних доменів у великомасштабному DNS-трафіку з піковою швидкістю 47 мільйонів годинних запитів протягом шести днів. Більше того, оскільки кількість помилкових позитивних доменів зменшувалася в геометричній прогресії з моменту початку

виконання, ми очікуємо, що з огляду на попередні помилкові позитивні результати, цей показник буде значно нижчим. Налаштування λ. Більш низька частота збору даних і класифікації (позначена як λ) дозволяє краще виявляти затримку, наприклад, до λ хвилин з моменту початку ексфільтрації. Однак, це вимагатиме, щоб витяг ознак застосовувався на $n_S \cdot \lambda$ хвилин журналів кожних λ хвилин. Тому рекомендується встановити його як низький рівень зберігання, потужності обробки та виділення пам'яті.

Налаштування n_S. Довший розмір вікна інспекції в одиницях λ хвилин (позначений як n_S) дозволяє виявляти «низькі та повільні» атаки, навіть якщо вони відбуваються протягом декількох годин (наприклад, десять номерів кредитних карт вилучаються протягом періоду шести годин). Так само, як і λ, його слід якомога більше збільшувати в межах обмежень зберігання і пам'яті. Більше того, компроміс між негайним блокуванням і виявленням тихої атаки повинен бути відображений у встановленні як n_S, так і λ. Наприклад, захищена бездротова мережа готелю, яка хоче зосередитися на блокуванні безкоштовного доступу Wi-Fi на основі тунелювання DNS, може вибрати $\lambda = 1$, $n_S = 10$, тоді як мережа торговельної точки може використовувати $\lambda = 60$, $n_S = 10$.

Збір даних. Головною метою фази збору даних є генерування компактного представлення доменного трафіку, що сприятиме подальшому ефективному вилученню функцій. Цей етап починається з обробки потокового трафіку DNS кожні λ хвилин і представляє кожну лінію журналу DNS як наступний запис:

$$\langle Q, R, T\rangle_j ,$$

таким чином, що *Q* містить повне ім'я запиту (наприклад, google.com.), *R* містить повний список значень записів відповіді (наприклад, 8.8.8.8), *T* містить тип запиту (наприклад, "A"), і *j* індекс рядка журналу. Зокрема, для випадку неіснуючих відповідей (статус NXDOMAIN), *R* призначається порожнім рядком. На основі цього подання групуємо DNS-журнали за первинними доменами.

Основне доменне ім'я, таким чином, визначається як конкатенація домену [2] другого рівня і домену верхнього рівня [3]. Наприклад, для повноцінного доменного імені login.example.org.de. домен другого рівня є example. і домен верхнього рівня є org.de.; отже, основним доменом є example.org.de. Основне використання основного доменного імені призначене для реєстрації домену. Тому, використовуючи його як точку вибірки для завдання класифікації, він призначає клас для кожної реєстрації і є ефективним проти доменів, які були зареєстровані лише для зловмисних дій. Первинний домен може бути вилучений з кожної лінії журналу (як пояснено вище) з використанням функції *prim*, як зазначено нижче:

$$prim(\langle Q, R, T\rangle_j) = P_j,$$

де P_j є основним доменом для *j*-го рядка журналу. Використовуючи функцію *prim*, лог-лінії можуть бути згруповані за їх первинним доменом, а також за їх дискретними часовими рамками збору *t*:

$$L_t^{Pi} = \{\langle Q, R, T\rangle_j \mid prim(\langle Q, R, T\rangle_j) = P_j\}.$$

Таке групування журналів DNS дозволяє ефективно сканувати послідовні журнали, оскільки кожен журнал може бути зібраний один раз, а використаний n_S разів за допомогою розсувного вікна. Наприклад, рядок журналу, який буде зібрано в момент часу t_i, буде споживатися в часи $[t_i, ..., t_i + n_s)$. Це розсувне вікно протягом останніх n_S λ хвилин виявляється корисним у подальшій фазі вилучення ознак.

Вилучення ознак. Фаза вилучення ознак працює раз на кожні λ хвилин на кожному доменному розсувному вікні, тобто комбінація останніх журналів *n*:

$$W^{Pi}_{tnow} = \{ L_t^{Pi} \mid t_{now} - n_S <= t < t_{now} \}.$$

Визначимо функцію вилучення ознак, *fe*, щоб перетворити вікноспостережень W^{Pi}, на вектор ознак, який представляє конкретний первинний домен у певному вікні спостережень:

$fe(W^{Pi}_{tnow}) = \langle P_i, Ent, NI, Uniq, Vol, Len, LMW\rangle$,

де кожна з наступних ознак є обчислюваною функцією на домені P_i в тимчасовому кадрі t_{now} 1) *E* відповідає ентропії символу, 2) *NI* відповідає коефіцієнту типу non-IP, *Uniq* відповідає унікальному співвідношенню запитів,

3) *Vol* відповідає обсягу запиту, *Len* відповідає середній довжині запиту, і *LMW* відповідає співвідношенню між довжиною найдовшого значущого слова і довжиною субдомену.

Ентропія символів. Поняття ентропії, або щільність інформації, можна інтерпретувати як вимірювання середньої невизначеності букви A, заданої $\{A1, .., An - 1\}$ [10]. Ентропія обчислюється над дискретною випадковою величиною *X*, використовуючи формулу:

$$H(X) = -\sum_{i=0}^{n} \Pr(xi) * \log \Pr(xi),$$

де $\Pr(x_i)$ – ймовірність *i*-го символу інформації (наприклад, символу) в серії *X*, що складається з *n* символів.

Серед інших застосувань ентропія широко використовується як евристика для виявлення шифрування в потоці бітів [11–13]. Обчислення ентропії здійснюється за запитами домену для виявлення зашифрованої або закодованої поведінки запитів. Формально, якщо випадкова величина *X* є серією символів, а дозволеними символами є літери, цифри і дефіси (LDH), як у запиті на коментарі (RFC) протоколу DNS [27], то:

$$H(X) = - \sum_{Xi \in LHD}^{n} \Pr(xi) * \log \Pr(xi),$$

і функція обчислюється наступним чином:

$$E(W^{Pi}{}_{tnow}) = H(Q_1||..||Q_m),$$

де $Q_1||..||Q_m$ – конкатенація повністю кваліфікованих доменних імен в лог-рядках $W^{Pi}{}_{tnow}$

Розповсюдження типу RR в IP-вузлі обміну. Згідно з [15] 99,4% запитаних записів ресурсів (RRs) мають наступні типи: A (IPv4), AAAA (IPv6) і PTR (покажчики зворотного пошуку).

Однак ці RR обмежені короткою довжиною відповіді (тобто до довжини IP-адреси) порівняно з іншими типами RR (наприклад, TXT, SRV), і, отже, розподіл типів RR може бути різним у використовуваному домені для обміну даними. Створена нами функція обчислює швидкість записів A та AAAA для кожного домену за час:

$$E(W^{Pi}{}_{tnow}) = \frac{\sum_{W^{Pi}_{tnow}} | (T = "A" \wedge T = "AAAA")^1}{\sum_{W^{Pi}_{tnow}} 1}.$$

Унікальний коефіцієнт запиту. Домен, піддомени якого використовуються як повідомлення, навряд чи повторить їх. Таким чином, при порівнянні доменів, що використовуються для ексфільтрації до звичайних доменів, очікується набагато вищий коефіцієнт унікального запиту. Функція обчислюється так:

$$Uniq(W^{Pi}{}_{tnow}) = \frac{| \{Q | Q \in W^{Pi}_{tnow} |}{\sum_{W^{Pi}_{tnow}} 1}.$$

Унікальний об'єм запитів: у звичайному режимі трафік DNS є досить розрідженим, оскільки відповіді в основному кешуються в межах заглушки. Однак, у разі обміну даними через DNS, трафік, специфічний для домену, повинен уникати кеш-пам'яті за допомогою неповторюваних повідомлень або короткого часу для того, щоб дані перетворилися на сервері зловмисника. Як уникнути кешу, так і тривалого обміну даними, може призвести до більшого обсягу запитів у порівнянні зі звичайними налаштуваннями. Функція обчислюється наступним чином:

$$Vol\left(W^{Pi}_{tnow}\right) = | \{Q | Q \in W^{Pi}_{tnow} |.$$

Середня тривалість запиту. Як доповнення до функції обсягу та обмеження розміру запиту, існує компроміс між обсягом запитів та їх довжиною, що робить його ефективною функцією виявлення.

Найдовше змістовне слово над середньою довжиною домену: для кожного основного домену кожен субдомен розкладається на його ієрархічні мітки, упорядковані за їх довжиною. Починаючи від найдовшого до найкоротшого підрядка, пошук слова англійського словника виконується над поточним підрядком. Якщо пошук є успішним (тобто підрядок є дійсним англійським словом), то довжина підрядка вважається довжиною найдовшого значущого слова (LMW). Цю довжину ділять на довжину субдомену та усереднюють за всіма субдоменами. Хоча англійська мова не єдина мова, яка використовується в Інтернеті, це може допомогти розрізнити домени з великою кількістю доменів, які або читаються, або не читаються.

Виявлення обміну даними DNS. Виявлення обміну даними DNS для вхідного вектора ознак обчислюється з використанням моделі виявлення аномалій, а саме ізоляційного лісу [14]. Модель ізоляційного лісу є класичним класифікатором (тобто навчається тільки на існуючих законних даних) і може виявляти аномальну поведінку на майбутніх даних. Тому тут слід обговорити два аспекти: (1) навчання моделі та (2) застосування моделі на нових даних.

Фаза навчання приймає набір раніше зібраних векторів ознак і виводить модель аномалії, яка, по суті, є функцією, яка діє на вибірку і виводить показник аномалії. У нашому випадку вхід до моделі є множиною W^{Pi}_{t} для кожного домену та для t в певний період часу (наприклад, попередній день), а результат – аномальна оцінка від 0 до 1. Аномалія оцінювання – це функція рівня забруднення, позначена як ν, що відноситься до частки шуму в даних. Виходом фази навчання є модель аномалії разом з T_s, поріг аномалії, який буде застосований на нових даних. Коли приходить нова вибірка, до неї застосовується така модель, що кожному зразку присвоюється оцінка, *s*, використовуючи функцію *iforest*, наступним чином:

$$Iforest(fe(W^{Pi}_{tnow})) = s,$$

де *fe* – функція вилучення ознак з рівняння 1, яка переводить зразок до вектора ознак. Якщо оцінка перевищує показник аномалії (тобто $s > T_s$), вибірка вважається аномальною, а домен, на який він посилається, буде позначений як домен, який використовується для обміну даними через DNS.

Блокування шкідливих доменів. Домени, які є аномальними з точки зору трафіку (тобто використовуються для обміну даними), можна розділити на дві категорії: шкідливі та легітимні. Як правило, коли система розгортається в мережі, вона легко знайде легітимні (і, можливо, нелегітимні) служби, які використовують DNS для обміну даними. Як тільки ці домени відображаються та видаляються експертом з безпеки, будь-який новий домен, що з'являється у фазі виявлення аномалій, вважається зловмисним видаленням даних і негайно блокується.

Оцінка ефективності запропонованого методу. Оцінка зосереджується на двох основних цілях:

виявлення малопропускної ексфільтрації шкідливих програм та виявлення високопродуктивних тунелів DNS. З цією метою ми представляємо наш набір даних DNS-трафіку, що складається з доброякісного трафіку, а також тестових об'єктів ексфільтрації DNS. Тестові об'єкти включають (1) Iodine і (2) Dns2tcp (як високопродуктивні інструменти тунелювання), і (3) FrameworkPOS і (4) Backdoor.Win32.Denis (як DNS ексфільтрація зловмисного програмного забезпечення).

Виявлення і коефіцієнти помилкових позитивних результатів для запропонованого методу розглядаються для кожного з тестованих суб'єктів при встановленні сильного обмеження на прийнятний параметр швидкості помилкових позитивних (тобто менше $2 * 10^{-5}$). Виходячи з ручної класифікації випадків використання помилкового позитивного виявлення, робиться висновок, що більшість помилкових позитивних доменів потрапляють під випадки легального обміну даними через DNS (наприклад, пошук антивірусної сигнатури), тому слід виявити, що експерт з питань безпеки повинен перейти до білого списку, щоб уникнути майбутніх помилкових спрацювань. Таким чином, враховуючи, що перші перераховані в білий список, аналізуються помилкові позитивні показники протягом часу (замість кожного домену), вимірюючи кількість нових помилково виявлених доменів на день.

Тестовий набір даних. Наш основний набір даних (позначений як DS) – це один тиждень трафіку DNS, зібраного з підмножини рекурсивних резонансів DNS, керованих компанією «Akamai Technologies». DS розглядається як великомасштабний зразок трафіку DNS з принаймні сто тисячами кінцевих користувачів.

Для проведення додаткового чесного тесту було використано DSpartial. DSpartial містить трафік підмножини трафіку користувачів DS. Тому вона трохи менше десятої частини DS, що робить її більш придатною для порівняння з попередніми дослідженнями.

Таблиця 2

Опис тестових наборів даних

	значення	std	min	середнє	max
DS	$3.5 \cdot 10^7$	$7.8 \cdot 10^6$	$2.2 \cdot 10^7$	$3.7 \cdot 10^7$	$4.7 \cdot 10^7$
$DS_{partial}$	$2.9 \cdot 10^6$	$4.9 \cdot 10^5$	$1.8 \cdot 10^6$	$2.7 \cdot 10^6$	$4.0 \cdot 10^6$

Протягом одного тижня трафіку DNS, до складу якого входять DS і $DS_{partial}$, вівся трафік реальних шкідливих комунікацій та інструментів DNS-тунелювання. Через нестачу обміну даними через DNS, можна припустити, що окрім нашого введеного трафіку немає жодних додаткових випадків обміну даними DNS на будь-якому з наборів даних. Кожен з доменів у наборі даних DS представляється у вигляді вектора ознак.

Домени, які використовуються для видалення даних, мають відповідні вектори ознак, які з'являються як відхилення з відносно характеристичних векторів нормальних доменів. Наприклад: Iodine DNS тунелювання має середню довжину запиту, що перевищує 100 символів, тоді як менше 0.01 нормальних доменів поводяться таким чином. Крім того, з Iodine кодуванням за замовчування, встановленим на Base128, ентропія символів зазвичай перевищує 3.0, тоді як у випадку чистого трафіку менше 0.05 трафіку слідує за такою поведінкою.

Об'єкти тестування. Існує чотири об'єкти тестування, які були протестовані:

1) *FrameworkPOS*: шкідливе програмне забезпечення FrameworkPOS [15].

2) *Backdoor.Win32.Denis*: Троянське зловмисне програмне забезпечення Backdoor.Win32.Denis [16] (в цьому документі називається Denis).

3) *Iodine*: Iodine є інструментом тунелювання DNS з відкритим вихідним кодом.

4) *Dns2Tcp*: Подібно до Iodine, Dns2tcp також є інструментом туннелювання.

Результати виявлення. Фаза навчання методу застосовується до першого дня набору даних DS, який містить лише доброякісний трафік. Заради ефективності замість тренування понад $24 \cdot 3{,}5 \cdot 10^7 \approx 8{,}4 \cdot 10^8$ записів ми відкидаємо первинні домени з менш ніж десятьма субдоменами за останні $\lambda \cdot n_s$ хвилини як з фаз навчання, так і з етапів виконання. Хоча цей фільтр ігнорує ексфільтрацію зі швидкістю, меншою, ніж 2.5kb на $n_s \cdot \lambda$ (щонайбільше 10 запитів по 255 байтів кожна), нам вдається скоротити наш навчальний набір до $2{,}1 \cdot 10^5$, тобто 0,25% від його початкового розміру. Параметри моделі тимчасового вікна встановлюються в $n_s = 6$, $\lambda = 60$, таким чином, метод залишається здатним виявляти атаки, які так само повільні, як 0.11b/s, а при гіршому випадку аналіз блокує і повідомляє про свою активність до шести годин після їх перших зібраних даних. Однак, фактично, для двох реальних шкідливих програм, які оцінюються в розділі V, вони сприймаються негайно протягом перших λ хвилин після їх виконання.

Припустимий коефіцієнт помилкового позитивного ставлення дорівнює $\nu = 2 \cdot 10^{-5}$, тобто він виявить лише домени, які принаймні такі аномальні, як чотири найбільш аномальних доменів у доброякісному трафіку. Фаза тренування з наведеними вище параметрами дає поріг оцінки аномалії $T_s = 0.653$. Виходячи з вихідної моделі і T_s, ми застосовуємо модель до нового трафіку в DS, який не використовувався для навчання.

Фаза виконання моделі працює так само. Кожні $\lambda = 60$ хвилин домени, які мали щонайменше десять субдоменів протягом останніх $\lambda \cdot n_s = 360$ хвилин, стають кандидатами для класифікації. Журнали доменів перетворюються на вектори ознак і модель призначає їм аномалію. Виходячи з фази навчання, домени, аномалії яких перевищують $T_0.653$, будуть розглядатися як домени, які використовуються для ексфільтрації

і будуть негайно заблоковані. Перша частина нашого методу оцінювання фокусується на швидкості успішно виявлених досліджуваних. Оцінку аномалії для кожного з досліджуваних досліджують кожні λ хвилин після початку ін'єкції трафіку. Очікуваний результат полягає в тому, щоб кожний показник аномалії кожного випробуваного перевищував заданий поріг T_s = 0.653 принаймні в одному зразку для того, щоб вважатися успішним виявленням. Результати вказують на те, що всі тестовані суб'єкти були успішно виявлені методом під час їх виконання.

Оскільки наш метод класифікує домени в дискретному періоді часу (тобто, λ n_s хв.), можна оцінити його на основі загальної кількості доменів, які були неправильно класифіковані як зловмисне вилучення.

Для цього використовується набір даних DS. Так само, як і в оцінці швидкості виявлення, ми тренуємо дані про трафік з першого дня, включеного в DS, і застосовуємо нашу модель до решти трафіку, включеного в DS. Кількість нових помилкових позитивних доменів у часі істотно падає і збігається до максимум одного домену на добу лише через два дні, коли модель була використана для класифікації (див. рис. 2).

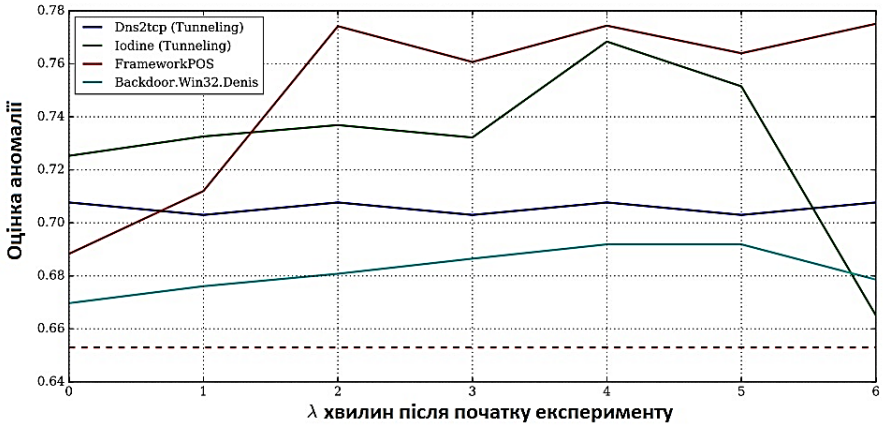

Рис. 2. Аномалія балів доменів, відповідних Dns2tcp, Iodine, FrameworkPOS і Backdoor.Win32.Denis, T_s = 0.653, ν= 2 10^{-5}, ns = 6, λ = 60

Список неправильно класифікованих доменів в основному складається з законних служб безпеки, які неправильно використовують протокол DNS для обміну даними для узагальненої класифікації помилково виявлених доменів). У нашому дозволі ці домени розглядаються як помилкові спрацьовування, навіть якщо вони явно використовуються для обміну даними, оскільки передбачається, що вони навмисно встановлені користувачем і тому не повинні бути блоковані. У реалізації реального світу можна зібрати білий список цих доменів перед застосуванням автоматичного блокування, щоб уникнути помилкового вимкнення служб легалізації. Класифікація хибних спрацювань надана в таблиці 3.

Таблиця 3

Класифікація хибних спрацювань

Категорія	Частота	Приклади
Пошук служб безпеки	12 з 18	sophosxl.net, l2.nessus.org, avts.mcafee.com, a.e.e5.sk
Загальні служби пошуку	4 з 18	kr0.io, dsipsl.net, drtst.com, dsipsl.net
Інші (*)	2 з 18	jacksonriverdev.com, groupinfra.com

Таким чином, запропонований метод продемонстрував можливість виявлення шкідливих запитів на основі DNS.

Висновки

В даній статті запропонований метод, спрямований на виявлення і блокування доменів, які запитуються в потоковому трафіку DNS і використовуються для зловмисного видалення даних DNS. Згідно методу потоковий трафік DNS збирається і перетворюється на вектори відповідності доменів. Після цього попередньо навчений класифікатор класу використовується для виявлення доменів, які обмінюються даними через DNS. Відразу після цього запити на домени, які класифікуються як такі, що використовуються для обміну даними, блокуються на невизначений час. Метод дозволяє його реалізацію в DNS-серверах, якщо вони підтримують протоколювання DNS-трафіку і чорного списку доменів. Описаний метод дає можливість виявляти шкідливі потоки пакетів даних серед звичайних, а постійний моніторинг визначених шкідливих потоків дає можливість виявити зловмисника та дозволяє ізолювати звичайний мережевий потік даних від шкідливого.

References

1. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. Bilge, L., Kirda, E., Kruegel,

C., Balduzzi. M.: NDSS, 2017. P. 1–17.

2. Quantitatively analyzing stealthy communication channels. Butler, P. Xu, K., Yao, D.: Proc. Ninth Int'l Conf. Applied Cryptography and Network Security, 2016. P. 238–254.

3. DAMBALLA. Botnet Communication Topologies. Understanding the intricacies of botnet command-and-control. Retrieved from https://www.damballa.com/downloads/r_pubs/ WP_Botnet_Communications_ Primer.pdf

4. The Federal Bureau of Investigation. Demarest, J. (2017). Statement Before the Senate Judiciary Committee, Subcommittee on Crime and Terrorism, Washington, D.C. Retrieved from http://www.fbi.gov/news/testimony/taking-down-botnets.

5. On Botnets that use DNS for Command and Control. Dietrich, C.J., Rossow, C., Freiling, F. C., Bos, H., van Steen, M., Pohlmann, N.: In: Proceedings of European Conference on Computer Network Defense, 2016. P. 9–16.

6. Detecting DNS Tunneling. Farnham, G., Atlasis, A.: SANS Institute InfoSec Reading Room, 2013. P. 1–32.

7. A comparison of distance-based semi-supervised fuzzy c-means clustering algorithms. Lai, D.T.C., Garibaldi, J.M.: Fuzzy Systems (FUZZ), In IEEE International Conference, 2015. P. 1580–1586.

8. As the Net Churns: Fast-Flux Botnet Observations. Nazario, J., Holz, T.: In: Conference on Malicious and Unwanted Software (Malware'08), 2008. P. 24–31.

9. Schiller, C. Botnets: The Killer Web Application. Craig Schiller, James R. Binkley. Syngress Publishing, 2012. 464 p.

10. Winning with DNS failures: Strategies for faster botnet detection. Yadav, S., Reddy, A.L.N.: In: Proc. of the 7th International ICST Conference on Security and Privacy in Communication Networks, 2011.

11. R. Matherw, V. Katkar. Survey of Low Rate DoS Attack Detection Mechanisms. International Conference and Workshop on Emerging Trends in Technology (ICWET 2011) – TCET, Mumbai, India. P. 955–958.

12. Zhang Sheng, Zhang Qifei, Pan Xuezeng, Zhu Xuhui. Detection of Low-rate DDoS-Attack Based on Self-Similarity. 2010 Second International Workshop on Education Technology and Computer Science. P. 333–336.

13. Junhan Park, Keisuke Iwai, Hidema Tanaka and Takakazu Kurokawa. Analysis of Slow Read DoS attack. ISITA2014, Melbourne, Australlia, October 26–29, 2014. P. 60–64.

14. L. Buczak, P. A. Hanke, G. J. Cancro, M. K. Toma, L. A. Watkins,and J. S. Chavis, "Detection of tunnels in pcap data by random forests," in Proceedings of the 11th Annual Cyber and Information SecurityResearch Conference. ACM, 2016, p. 16.

15. Alexey Shulmin S. Y. (2017) Use of dns tunneling for cnc communications. Retrieved from https://securelist.com/use-of-dns-tunneling-for-cccommunications/78203/.

16. P. Security. (2016) What is multigrain? learnwhat makes this pos malware different. Retrieved from https://www.pandasecurity.com/mediacenter/malware/multigrainmalwarepos/.

Рецензія/Peer review : 6.5.2019 р. Надрукована/Printed : 2.6.2019 р.
Рецензент: д.т.н., проф. Говорущенко Т.О.