

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, АВТОМАТИЗАЦІЯ ТА ОБЧИСЛЮВАЛЬНА ТЕХНІКА В ТЕХНОЛОГІЧНИХ ПРОЦЕСАХ

УДК 004.491.2

DOI: 10.31891/2219-9365-2019-64-10

БОБРОВНИКОВА К. Ю., ДЕНИСЮК Д. О., БОЙКО А. О.
Хмельницький національний університет

МОДЕЛЬ БОТ-МЕРЕЖ З ВРАХУВАННЯМ СИСТЕМИ ДОМЕННИХ ІМЕН

В роботі представлено модель бот-мереж з врахуванням системи доменних імен (Domain Name System, DNS), а також використання бот-мережами технологій ухилення від виявлення на основі DNS, таких як періодична зміна IP-відображення (cycling of IP mapping), «потік доменів» (domain flux), «швидкозмінні мережі» (fast flux) та DNS-тунелювання (DNS-tunneling), і різних способів комунікації ботів з командно-контролюючими центрами бот-мереж (C&C-серверами). Використання розробленої моделі надає можливість здійснювати виявлення ботів бот-мереж централізованої, розподіленої та гібридної архітектури на різних етапах життєвого циклу бот-мереж. Залучення моделі дозволяє підвищити точність виявлення бот-мереж на основі аналізу DNS-трафіка вже на початкових етапах інфікування мереж.

Ключові слова: бот-мережа, виявлення бот-мереж, DNS, DNS-трафік, технології ухилення від виявлення бот-мереж, періодична зміна IP-відображення, «потік доменів», «швидкозмінні мережі», DNS-тунелювання, життєвий цикл бот-мереж, шкідливе програмне забезпечення.

BOBROVNIKOVA K., DENYSIUK D., BOYKO A.
Khmelnitskyi National University

MODEL OF BOTNETS TAKING INTO ACCOUNT THE DOMAIN NAME SYSTEM

The paper presents a model of botnets taking into account the Domain Name System (DNS), as well as the using of botnets of DNS-based evasion techniques, such as cycling of IP mapping, "domain flux", "fast flux" and DNS-tunneling. Using of the developed model will allow detecting bots of botnets of centralized, distributed and hybrid architecture, taking into account their using of the Domain Name System at different stages of the botnet life cycle. The involvement of the model makes it possible to detect botnets that use the DNS-based evasion techniques and different DNS-based communication methods between bots and botnet's command and control centers (C&C-servers).

Using of the developed model will allow detecting botnets based on the DNS traffic analysis already at the initial stages of network infection with high efficiency. The experimental results demonstrate the ability to detect botnets that use DNS-based evasion techniques up to 99,1%, while the level of false positives was about 0,11%.

Keywords: bot, botnet, botnet detection, DNS, DNS traffic, botnet's evasion techniques, cycling of IP mapping, "domain flux", "fast flux", DNS-tunneling, botnet's life cycle, malware.

Вступ. Відомо багато різноманітних підходів до побудови моделей бот-мереж [6]. В [7-9] моделі життєвого циклу бот-мереж розглянуто з погляду на їх шкідливу діяльність. В [10] таксономія та життєвий цикл (ЖЦ) бот-мережі представлені з використанням моделі розробки продукту. Згідно цієї моделі, ЖЦ бот-мережі складається з етапів: Концепція, Вербування, Взаємодія, Маркетинг та Виконання атаки (Conception, Recruitment, Interaction, Marketing, and Execution of attack, CRIME).

Однією з найбільш ранніх моделей шкідливих програм є спрощена математична епідеміологічна модель [11]. Вдосконалення цієї моделі є модель «Чутливий-Інфікований-Контрольований» (Susceptible-Infected-Control, SIC), яка описує бот-мережі з використанням стохастичної моделі еволюції популяції. В [8] припущення моделі SIC полягає в тому, що всі можливі хости чутливі до інфікування шкідливим програмним забезпеченням бот-мережі, та існує невелика ймовірність того, що пристрій буде інфікований і стане контрольованим бот-мережею. З огляду на це, модель передбачає дві мети – відстежувати динаміку зростання популяції бот-мережі та порівнювати результати застосування різних стратегій пом'якшення атак бот-мереж. В [12] запропоновано модель «Чутливий-Інфікований-Чутливий» (Susceptible-Infected-Susceptible, SIS). Розроблена модель застосовує поняття коефіцієнту щеплення. Відомі потенційні вразливості комп'ютерних систем або вразливості, виявлені за допомогою зворотного інжинірингу зразків шкідливих програм, можуть бути векторами інфікування. Після визначення вектору інфікування в якості контрзаходу для нього може бути створений спеціальний експлойт, який усуває вразливість – «щеплення».

В [13] для моделювання потоків пакетів у бот-мережі P2P застосовуються методи глибокого навчання. Глибоке навчання забезпечує більш точне розпізнавання структури трафіку, порівняно з не ієрархічним машинним навчанням.

В [14] для моделювання DDoS-атаки на хмарний центр даних (Cloud Data Center, CDC) було використано стохастичне моделювання. Ця модель використовує модулі для фільтрації трафіку в межах порогових значень пропускної здатності, балансування навантаження та оптимізації черг до центрів обробки

даних, спрямовані на зменшення потоків запитів.

В [15] використовуються модель еволюційних ігор та епідеміологічна модель для аналізу стійкості взаємодіючих бот-мереж в ситуаціях, коли власники бот-мереж взаємодіють у співпраці або конкуренції. В [16] запропонована динамічна ігрова модель, в якій зловмисник – це бот-майстер (власник бот-мережі), який ініціює DDoS-атаку, а захисник – брандмауер. Модель припускає, що гравці можуть діяти як раціонально, так і нерационально. Мета цієї моделі – покращити продуктивність брандмауерів. В [17] використовується теорія ігор середнього поля в поєднанні зі стохастичними процесами та епідеміологічними станами інфікування для моделювання поширення бот-мереж в залежності від рішень власника пристрою стосовно системи захисту.

В [18] для відокремлення трафіку бот-мереж від нешкідливого трафіку використана непараметрична байєсівська модель. Складність ідентифікації шкідливого трафіку полягає у тому, що схеми комунікацій бот-мереж постійно розвиваються, і зловмисний трафік передається одночасно з нешкідливим трафіком. Запропонована модель асоціює байти, пов'язані з трафіком і комунікаціями, з конкретними бот-мережами або з доброякісними програмами. Для представлення діяльності бот-мережі використовується прихована марківська модель, яка намагається виявити шкідливий трафік з метою пом'якшення атаки бот-мережі.

В [19] взаємодія користувачів соціальних мереж моделюється у вигляді графу, який використовується для візуалізації трафіку, моделювання поведінки програм та виявлення аномалій чи атак бот-мереж.

Економічні моделі [20-22] намагаються пом'якшити загрозу бот-мереж, порушуючи мотивацію до одержання прибутку власників бот-мереж.

Зазвичай виявлення бот-мереж стає можливим лише під час атаки бот-мережі або після неї. Відомі моделі надають можливість перевірити існуючі припущення та пом'якшити наслідки атак бот-мереж. Беручи до уваги складність ідентифікації шкідливого трафіка бот-мереж, а також те, що більшість бот-майстрів для керування бот-мережами та ухилення від виявлення бот-мереж використовують сервіси DNS, актуальною задачею є розроблення моделі бот-мереж з врахуванням DNS. Це дозволить виявляти бот-мережі на етапі передачі трафіку командування та контролю бот-мережі із застосуванням DNS, до початку здійснення атаки.

Життєвий цикл бот-мережі з врахуванням DNS. Для побудови моделі бот-мережі розглянемо її життєвий цикл з врахуванням системи доменних імен. Життєвий цикл бот-мережі можна розділити на п'ять фаз: (1) інфікування; (2) первинна реєстрація або з'єднання; (3) здійснення шкідливої активності; (4) супроводження; (5) припинення функціонування. Схему ЖЦ бот-мережі подано на рис. 1.

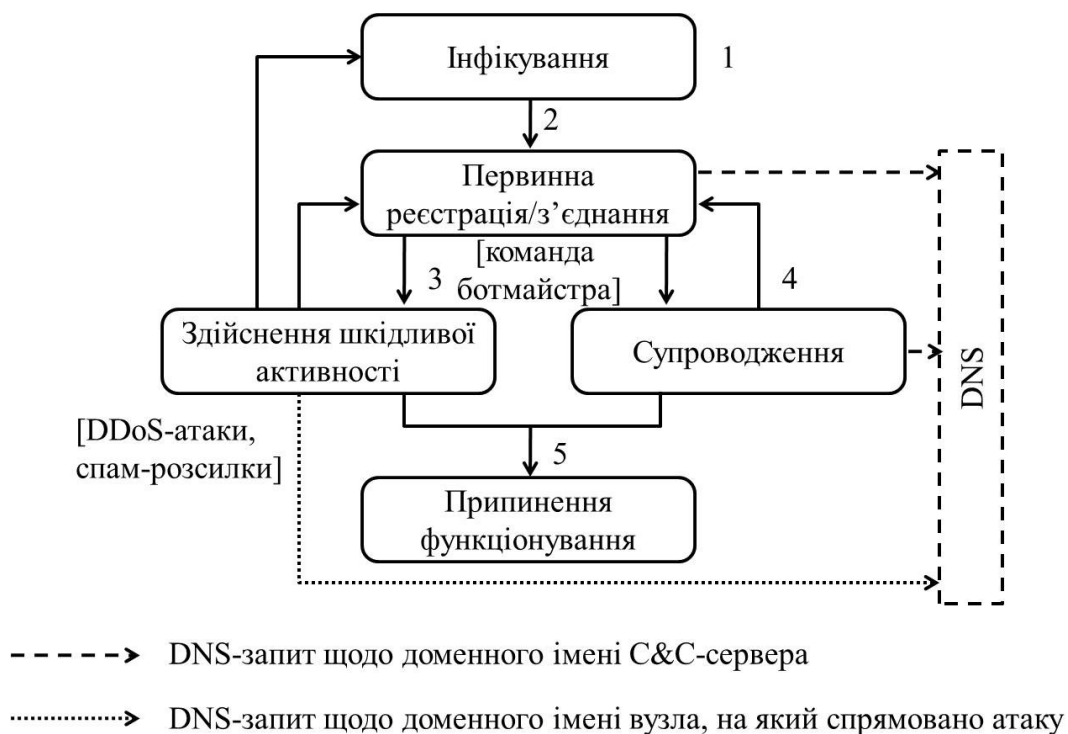


Рис. 1. Схема життєвого циклу бот-мережі з врахуванням системи доменних імен

Кожен бот, який входить до складу бот-мережі, може здійснювати пошук вразливих КС з метою їх інфікування та розширення бот-мережі. На фазі інфікування відбувається подолання існуючих в КС механізмів захисту, виконання шкідливого скрипта (shell-коду) та встановлення програми-бота на інфікованій комп'ютерній системі.

Інфікування КС ботом може бути як наслідком проведення успішної атаки на етапі здійснення шкідливої активності, так і окремою фазою ЖЦ. Наприклад, одним з найпоширеніших способів інфікування є drive-by завантаження, коли інфікування КС здійснюється під час відвідування скомпрометованого веб-сайту, на якому розміщено шкідливий код. Шкідливий код може бути завантажений на жорсткий диск КС або впроваджений у вигляді динамічних бібліотек в пам'ять легітимного процесу без збереження на диску. В цьому випадку шкідливий процес існує лише в ОП інфікованої КС до перезавантаження системи (так званий «безтілесний» бот). Здійснення інфікування з використанням популярного ресурсу, який відвідується користувачем регулярно, надає можливість багаторазового інфікування КС та утримання її в бот-мережі.

В разі успішного інфікування КС бот ініціює DNS-запит з метою встановлення відповідності між доменним ім'ям C&C-сервера та його IP-адресою. Використовуючи DNS-клієнт (stub resolver), інфікована КС надсилає рекурсивному DNS-серверу рекурсивний DNS-запит щодо доменного імені C&C-сервера. В разі успішного DNS-запиту бот здійснює первинну реєстрацію в бот-мережі та встановлює канал зв'язку з C&C-сервером – C&C-канал. Після спливання TTL-періоду DNS або перезавантаження КС бот знову ініціює DNS-запит і оновлює записи локального кешу DNS та кешу рекурсивного DNS-сервера.

На відміну від неінфікованих КС в локальній мережі, які зазвичай використовують локальні DNS-сервери для DNS-запитів, інфіковані ботами КС можуть використовувати також безкоштовні сервіси DNS (OpenDNS, FreeDNS) або власні DNS-сервери.

На рис. 2 подано узагальнену схему встановлення зв'язку між ботами бот-мережі та C&C-сервером з використанням DNS (на схемі послідовність запитів та відповідей представлено позначеннями 1...10).

Дії на фазі здійснення шкідливої активності визначаються функційним призначенням ботів бот-мережі та є наслідком виконання ботами команд власника бот-мережі (бот-майстра). Фаза супроводження полягає у внесенні змін до коду ботів з метою додавання ботам бот-мережі нової функціональності, забезпечення ухилення від виявлення, виправлення дефектів в ШПЗ, надання вказівки щодо зміни локації командно-контролюючого сервера бот-мережі тощо. Настання фази припинення функціонування може бути зумовлене як виконанням відповідної команди власника бот-мережі, так і ліквідацією бот-мережі експертами в галузі IT-безпеки, правоохоронними органами тощо.

Фази первинної реєстрації / з'єднання та супроводження вимагають надсилання ботами DNS-запитів з метою отримання інформації щодо локації C&C-сервера або сервісів оновлень ШПЗ бот-мережі. Частина атак на фазі здійснення шкідливих дій також вимагають використання сервісу DNS.

DNS-запити ініціюються ботом в наступних ситуаціях:

- (1) для первинної реєстрації бота та згуртування з бот-мережею після успішного інфікування КС;
- (2) після збоїв з'єднання з C&C-сервером (після збою 3-етапного «рукостискання» боти розпочинають надсилати запити до DNS-сервера);
- (3) після міграції C&C-сервера;
- (4) після зміни IP-адреси C&C-сервера;
- (5) при здійсненні шкідливих дій (DDoS-атак, спам-розсилок);
- (6) після перезавантаження інфікованої КС;
- (7) з метою підвищення завадостійкості (для отримання додаткових доменних імен, пов'язаних з функціонуванням бот-мережі, бот здійснює зворотні DNS-запити).

DNS-запити інфікованих КС вирізняються характерною ознакою – скоординованістю (груповою активністю), яка полягає в тому, що боти бот-мережі здійснюють одночасні або зосереджені в невеликому проміжку часу DNS-запити під час спроб доступу до C&C-серверів, їх міграціях, виконанні команд або скачуванні оновлень ШПЗ.

В разі, якщо ресурсні записи DNS для доменного імені були кешовані DNS-клієнтом КС, повторний DNS-запит не виходить за межі локального кеша DNS комп'ютерної системи до спливання TTL.

Для багатьох видів бот-мереж характерним є ігнорування ботами TTL-періоду, тривалість якого містилась у відповіді від авторитетного DNS-сервера на DNS-запит. Це означає, що бот виконує очищення локального кеша DNS та здійснює повторний DNS-запит щодо доменного імені до завершення TTL-періоду, що надає можливість підвищити гнучкість та надійність керування бот-мережею (рис. 3, а, б, на схемі послідовність дій представлено позначеннями 1...6).

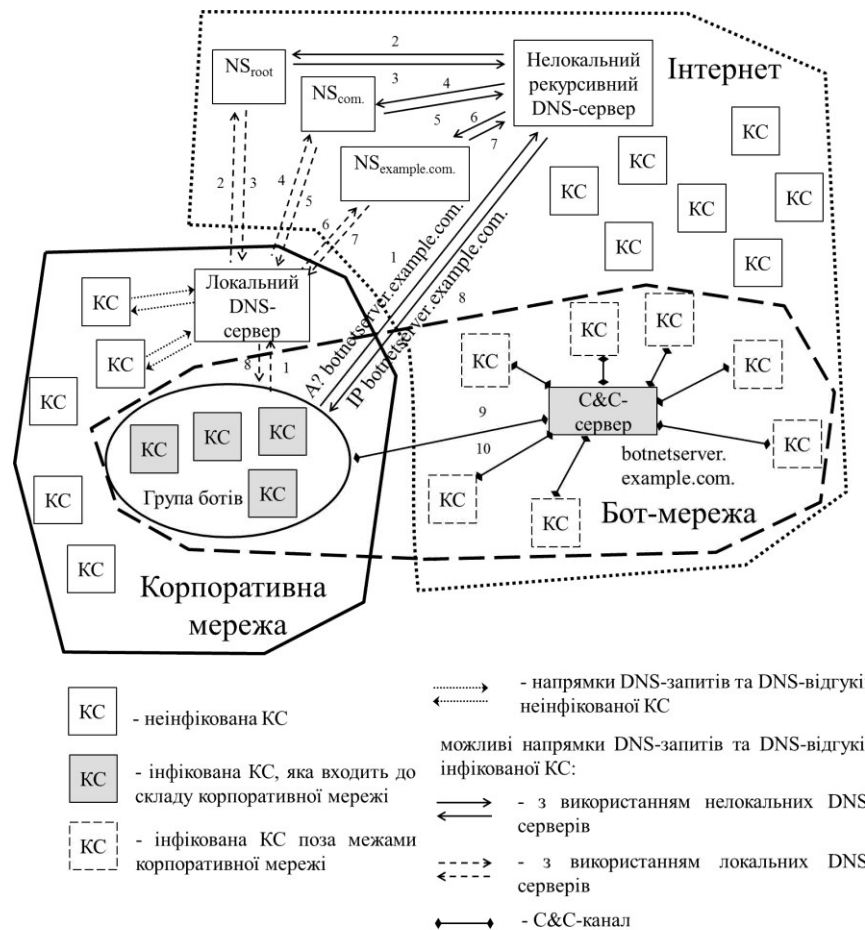


Рис. 2. Узагальнена схема встановлення зв'язку між ботами бот-мережі та C&C-сервером з використанням DNS

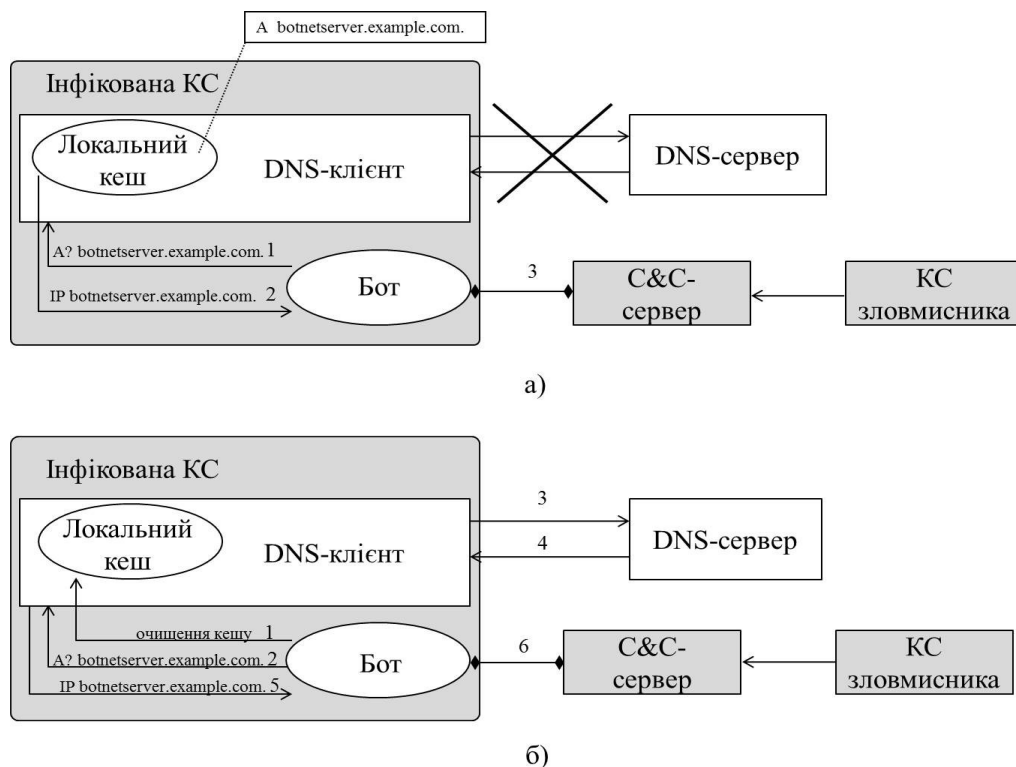


Рис. 3. Здійснення повторного DNS-запиту щодо доменного імені до завершення TTL-періоду: а) за наявності в локальному кеші DNS кешованої DNS-відповіді щодо доменного імені; б) після очищення локального кешу DNS

Модель бот-мережі з врахуванням DNS. Розглянемо модель бот-мережі як систему керування інфікованими ботами КС з врахуванням використання DNS на різних фазах життєвого циклу бот-мережі.

Представимо модель бот-мережі у вигляді кортежу:

$$M_{BN} = \langle C, A, B, \Psi, Z, L, F \rangle, \quad (1)$$

де $C = \{c_j\}_{j=1}^{N_C}$ – множина контролюючих елементів бот-мережі, N_C – кількість контролюючих елементів бот-мережі; $A = \{a_j\}_{j=1}^3$ – тип архітектури бот-мережі; $B = \{b_j^p\}_{j=1}^{N_B}$ – множина мережних протоколів, що використовуються для керування бот-мережею, N_B – кількість мережних протоколів, $p \in P$, $P = \{1..65535\}$ – множина портів, що використовуються для керування бот-мережею; $\Psi = \{\psi_j\}_{j=1}^4$ – множина технологій ухилення від виявлення бот-мереж на основі DNS; $Z = \{z_j\}_{j=1}^{N_Z}$ – множина ботів, що входять до складу бот-мережі, N_Z – кількість ботів бот-мережі; $L = \{l_j\}_{j=1}^5$ – множина етапів життєвого циклу бот-мережі; $F = \{f_j\}_{j=1}^{N_F}$ – множина функцій ботів, що визначається відповідною фазою життєвого циклу бот-мережі, N_F – кількість функцій ботів бот-мережі; функція інфікування вузла $l_1 \Rightarrow Y \xrightarrow{f_1} \{h_{inf} | h_{inf} \in H\}$, де Y – множина шкідливих дій, закладених в функціонал бот-мережі, H – множина КС в глобальній мережі, h_{inf} – інфікована ботом КС; функція приєднання інфікованої КС до бот-мережі $l_2 \Rightarrow Z \cup \{h_{inf} | h_{inf} \in H\} \xrightarrow{f_2} Z'$; функція оновлення версії ШПЗ бота бот-мережі $l_3 \Rightarrow z \times z' \xrightarrow{f_3} z'$; функція виконання команди на здійснення шкідливої активності $l_4 \Rightarrow Z \times \{p | p \in P\} \xrightarrow{f_4} Y$, де P – множина команд, які можуть бути виконані ботами бот-мережі; функція припинення функціонування бота бот-мережі $l_5 \Rightarrow Z \setminus \{z | z \in Z\} \xrightarrow{f_5} Z'$.

Розглянемо більш детально принципи функціонування складових моделі бот-мережі. Представимо множину командно-контролюючих елементів бот-мережі наступним чином:

$$C = \{c_j\}_{j=1}^{N_C} = \left\langle \langle D, I \rangle, \langle N, E \rangle \right\rangle_j \Big|_{j=1}^{N_C}, \quad (2)$$

де $D = \{d_j\}_{j=1}^{N_D}$, $I = \{i_j\}_{j=1}^{N_I}$, $N = \{n_j\}_{j=1}^{N_N}$, $E = \{e_j\}_{j=1}^{N_E}$ – множини доменних імен та IP-адрес контролюючих елементів бот-мережі, множини доменних імен та IP-адрес авторитетних серверів імен для d відповідно, N_D , N_I , N_N , N_E – кількість доменних імен, які відповідають контролюючим елементам бот-мережі, IP-адрес, які співставляються з цими доменними іменами, доменних імен авторитетних серверів імен та їх IP-адрес відповідно.

За командно-контролюючі елементи прийемо командно-контролюючі сервери та сервіси технічного обслуговування бот-мереж.

Представимо тип архітектури бот-мережі наступним чином: $A = \{a_j\}_{j=1}^3$, де a_1 – централізована, a_2 – розподілена, a_3 – гібридна. В бот-мережі централізованої архітектури кожен бот з множини Z встановлює зв'язок з одним (або декількома) командно-контролюючим сервером з множини C (рис. 4, а).

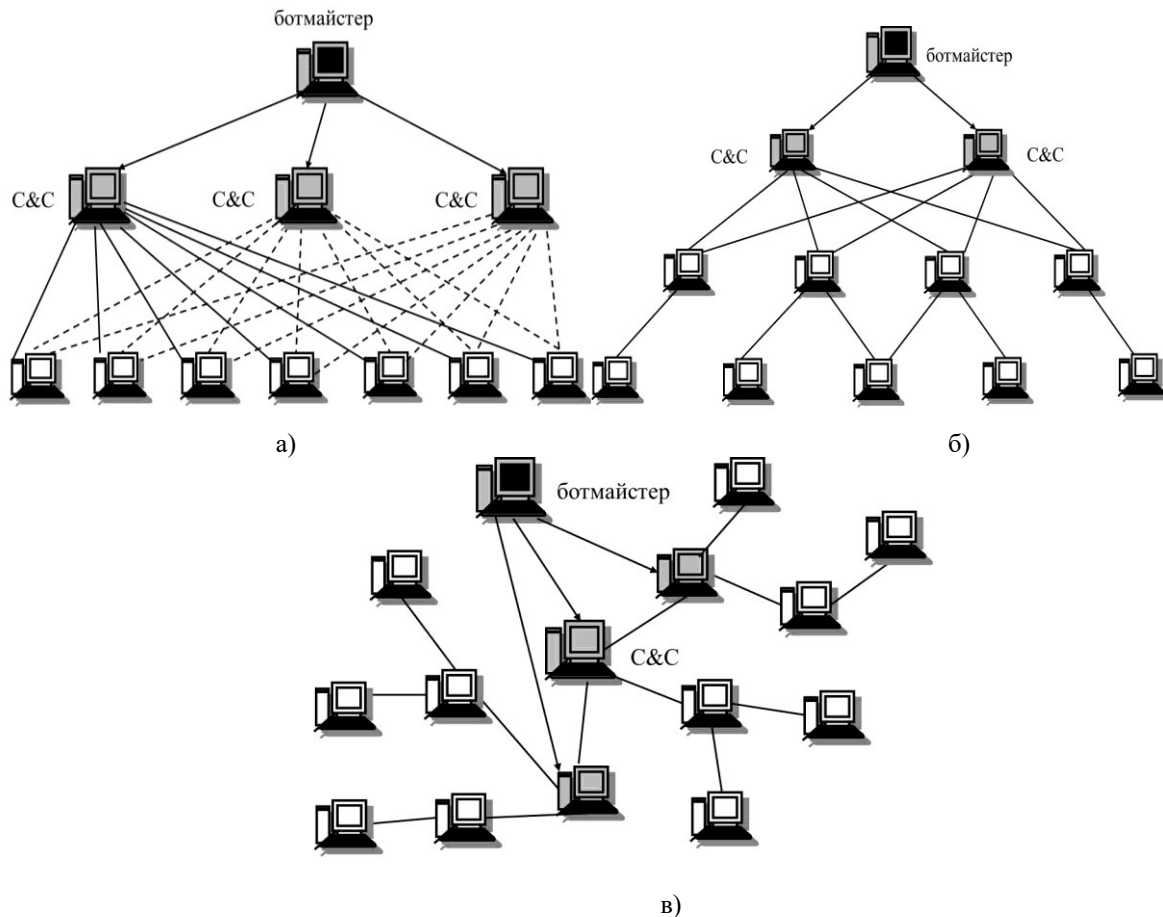


Рис. 4. Схематичне зображення бот-мережі: а) централізованої архітектури;
 б) розподіленої архітектури; в) гібридної архітектури

Розподіленість архітектури бот-мережі може бути забезпечена шляхом застосування власних розподілених DNS-сервісів або сервісів динамічної DNS, що дозволяє оновлювати інформацію на DNS-сервері в режимі реального часу та в автоматичному режимі і уможлиблює використання великої кількості інфікованих КС з метою перенаправлення трафіка від ботів бот-мережі до групи контролюючих вузлів з множини C та навпаки (рис. 4, б). В бот-мережі гібридної архітектури частина ботів виконують функції клієнтів та серверів одночасно, тобто належать до множини $C \cap Z := \{x \mid x \in Z \wedge x \in C\}$, а решта ботів є тільки клієнтами, що дозволяє підвищити надійність та швидкість реакції такої бот-мережі. Бот-мережа може бути розподілена на кластери, які використовують протокол P2P для внутрішнього спілкування. В кожному кластері виокремлюється головний вузол, який може з'єднуватись з головними вузлами інших кластерів вищого рівня ієрархії, підтримуючи зв'язок між кластерами (рис. 4, в).

Представимо множину технологій ухилення від виявлення бот-мереж на основі використання DNS наступним чином: $\Psi = \{\psi_j\}_{j=1}^4$, де ψ_1 – періодична зміна IP-відображення (cycling of IP mapping), ψ_2 – «потік доменів» (domain flux), ψ_3 – «швидкозмінні мережі» (fast flux), ψ_4 – DNS-тунелювання (DNS-tunneling).

При періодичній зміні IP-відображення С&С-сервер бот-мережі з множини $c \in C$ періодично змінює локацію, при цьому доменне ім'я d , пов'язане з С&С-сервером, співставляється з однією IP-адресою з множини $i \in I$, $d \rightarrow \{i_1, \dots, i_n\}$ (рис. 5, на схемі періодичну зміну локації С&С-сервера бот-мережі представлено позначеннями 1...n). Зазвичай функції С&С-серверів виконують географічно розподілені в глобальній мережі скомпрометовані вузли, підконтрольні зловмиснику. Тип архітектури бот-мережі – централізована, $\psi_1 \Rightarrow a_1$.

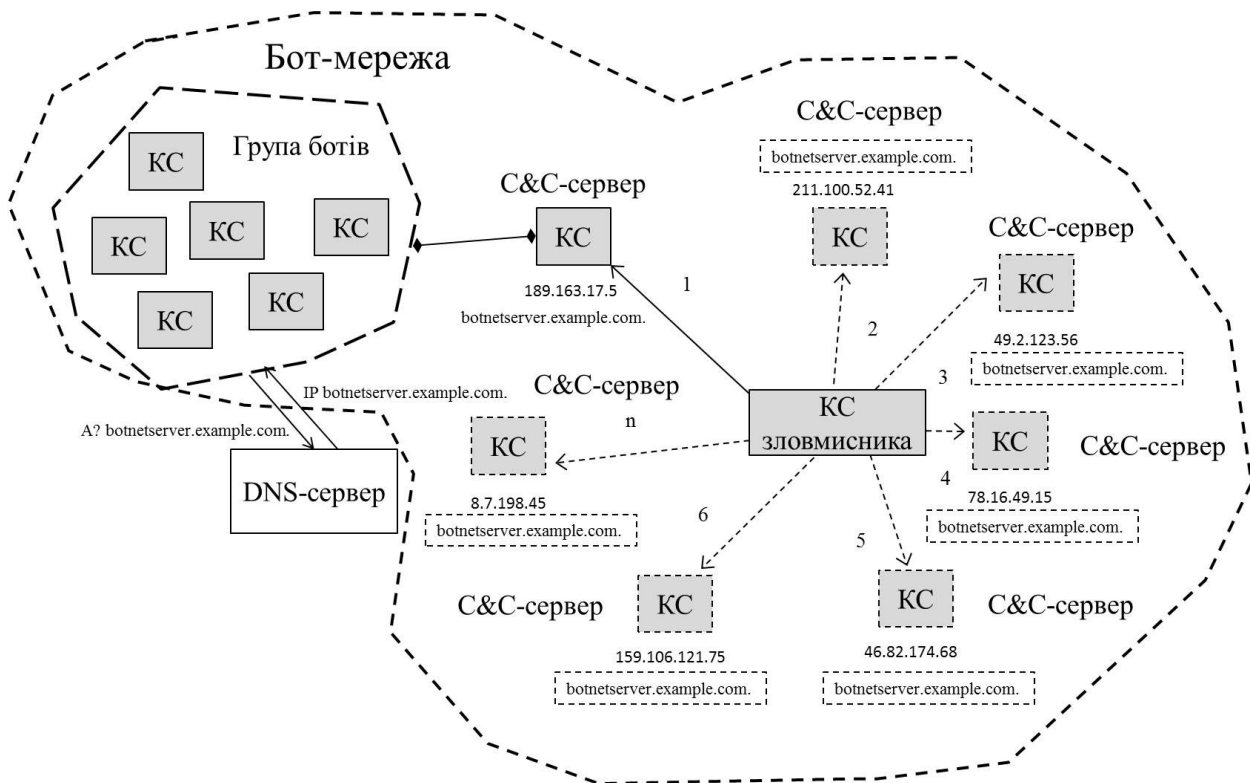


Рис. 5. Періодична зміна IP-відображення для доменного імені C&C-сервера

При використанні технології ухилення «потік доменів» C&C-сервер бот-мережі $c \in C$ періодично мігрує на нові доменні імена зі списку, який формується за допомогою використання алгоритму генерації доменних імен, DGA (рис. 6). Ботом та командно-контролюючим сервером за визначеним алгоритмом генерується дуже велика кількість доменних імен, певна незначна частина з яких використовуються в якості доменних імен C&C-сервера бот-мережі.

З метою встановлення зв'язку з C&C-сервером бот послідовно перебирає список згенерованих доменних імен та надсилає стосовно них DNS-запити, доки не отримає успішну DNS-відповідь з IP-адресою C&C-сервера. Таким чином, IP-адресі C&C-сервера $i \in I$ в межах кожного інтервалу часу, визначеного TTL-періодом для записів DNS щодо доменного імені C&C-сервера, може відповідати нове доменне ім'я $d \in D$, тобто $\{i\} \rightarrow \{d_1, \dots, d_n\}$, або в разі, якщо C&C-сервер також змінює і локацію, то $\{i_1, \dots, i_n\} \rightarrow \{d_1, \dots, d_m\}$.

Тип архітектури бот-мережі – централізована, $\psi_2 \Rightarrow a_1$.

В зв'язку з частою зміною доменного імені C&C-сервера або проблемами з його доступністю для бот-мережі є характерною підвищена кількість DNS-відповідей з кодом помилки RCODE=3 (NXDOMAIN).

Контролюючі елементи «швидкозмінної» мережі мають більш розширений функціонал в порівнянні з типовими C&C-серверами бот-мережі та приховані за мережею з множини зовнішніх проксі-серверів – «флюк-агентів», які слугують для перенаправлення запитів ботів бот-мережі та даних до та від внутрішніх контролюючих серверів (рис. 7, 8, на схемах послідовності запитів та відповідей представлено позначеннями 1...11).

Таким чином, в межах інтервалу часу, визначеного TTL-періодом DNS, для «однопоточної» мережі (рис. 7) доменне ім'я d , яке використовується для зв'язку ботів з контролюючими елементами $\{c_1, \dots, c_n\}$, співставляється з новою підмножиною IP-адрес, що циклічно змінюється $d \rightarrow \{i_1, \dots, i_n\}$, і які насправді є IP-адресами географічно розподілених інфікованих вузлів бот-мережі, що перенаправляють трафік до контролюючих елементів, тобто фактично $\{c_1, \dots, c_n\} := \{x \mid x \in Z \wedge x \in C\}$.

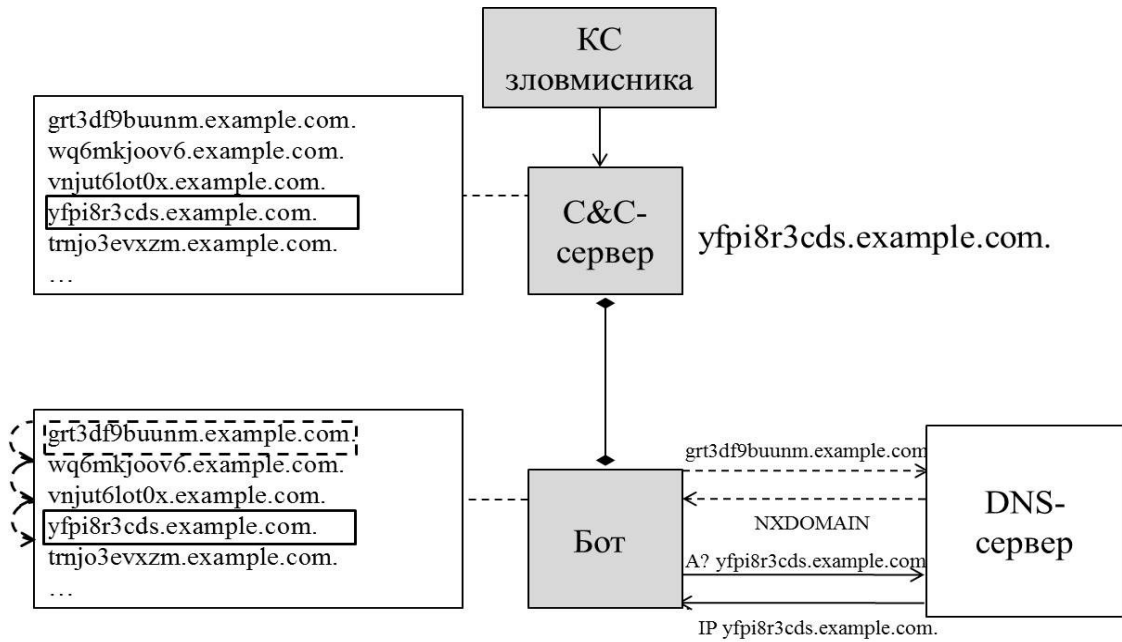


Рис. 6. Схема застосування алгоритму генерації доменних імен (DGA)

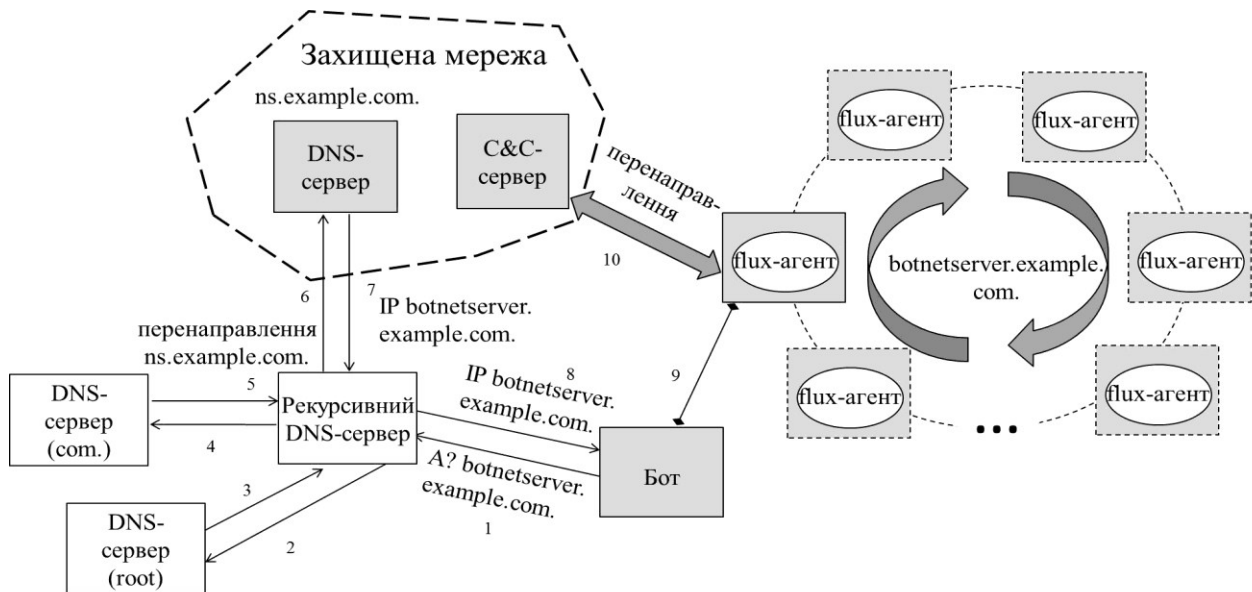


Рис. 7. Схема функціонування однопоточної «швидкозмінної» мережі

Для «двопоточної» мережі (рис. 8) додатково доменне ім'я кожного авторитетного сервера імен n співставляється з підмножиною IP-адрес, що циклічно змінюється, тобто $d \rightarrow \{i_1, \dots, i_n\}$, $n \rightarrow \{e_1, \dots, e_m\}$, і які насправді є IP-адресами географічно розподілених інфікованих вузлів бот-мережі, тобто $\{n_1, \dots, n_m\} := \{x \mid x \in Z \wedge x \in N\}$. Враховуючи, що кількість серверів імен для таких бот-мереж зазвичай більше одного, то $\{n_1, \dots, n_m\} \rightarrow \{e_1, \dots, e_n\}$. Використання в якості проксі-серверів великої кількості інфікованих вузлів, які знаходяться в різних частинах світу, та численні перенаправлення ускладнюють відстеження та відключення командно-контролюючих центрів такої бот-мережі. Тип архітектури бот-мережі – розподілена, $\psi_3 \Rightarrow a_2$.

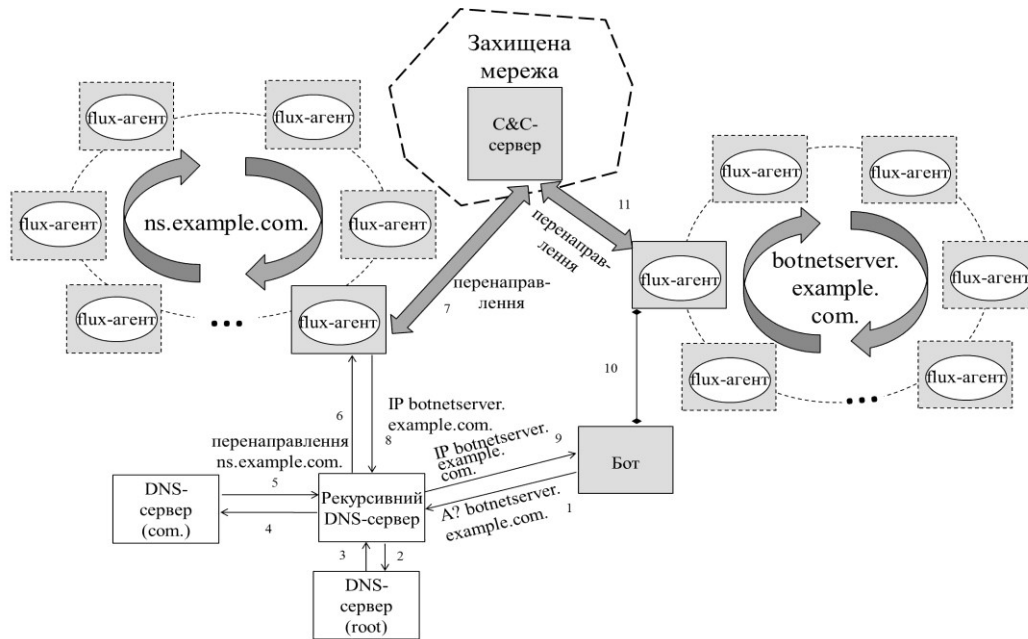


Рис. 8. Схема функціонування двопоточної «швидкозмінної» мережі

При використанні технології DNS-тунелювання для передачі трафіку командування та контролю зловмисником використовується фальшивий DNS-сервер, який є авторитетним DNS-сервером імен для зони DNS, щодо доменних імен якої здійснюються DNS-запити ботами бот-мережі. Це уможливорює надсилання ботами на сервер зловмисника закодованих повідомлень під виглядом DNS-запитів стосовно доменних імен цієї зони, та отримання ботами команд від зловмисника, також закодованих в DNS-відповідях (рис. 9, на схемі послідовність запитів та відповідей представлено позначеннями 1...4).

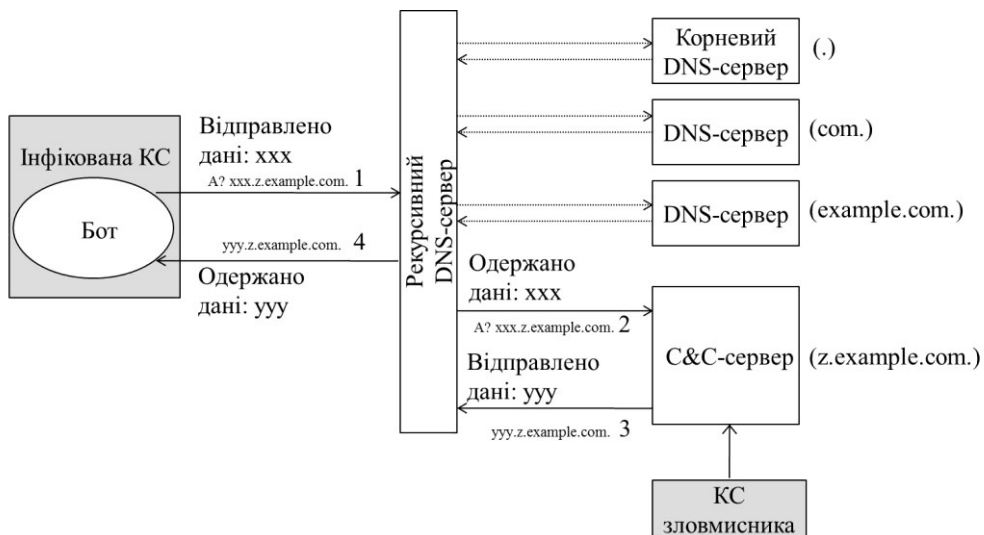


Рис. 9. Схема передачі довільного трафіку всередині полів DNS-повідомлення

В цьому випадку множина доменних імен D зони DNS фактично виступає в ролі аналога доменних імен C&C-сервера бот-мережі, IP-адреса e фальшивого DNS-сервера зазвичай залишається сталою, тобто $\{d_1, \dots, d_n\} \rightarrow e$. Тип архітектури бот-мережі – централізована або гібридна, $\Psi_4 \Rightarrow a_1 \vee a_3$.

Представимо DNS-трафік мережі у вигляді кортежу:

$$M_T = \langle \chi, H, S, D \rangle, \quad (3)$$

де χ – множина DNS-повідомлень, надісланих від та до множини H комп’ютерних систем мережі, $\chi = \chi^O \cup \chi^I$, де χ^O – множина вихідних DNS-повідомлень мережі, χ^I – множина вхідних DNS-повідомлень мережі; S – множина DNS-серверів, до яких було надіслано DNS-запити та від яких було одержано DNS-відповіді комп’ютерними системами мережі, $S = S^L \cup S^N$, де S^L – множина локальних DNS-серверів мережі, S^N – множина нелокальних DNS-серверів; D – множина запитаних комп’ютерними системами мережі доменних імен, $D = \{d_i\}_{i=1}^{N_D}$, де N_D – кількість різних доменних імен.

Представимо множину КС мережі, які здійснювали DNS-запити протягом часу спостереження, наступним чином: $H = \bigcup_{j=d_1}^{d_{N_D}} \bigcup_{k=1}^{N_{TTL}} H_{j,k}$, де H_j – підмножини MAC-адрес КС, які надсилали DNS-запити щодо певного доменного імені протягом часу спостереження; $H_{j,k}$ – підмножини MAC-адрес КС, які надсилали DNS-запити щодо певного доменного імені в межах певного TTL-періоду; N_{TTL} – загальна кількість таких підмножин; $H_{j,k} = \{h_{j,k,i}\}_{i=1}^{N_{H,j,k}}$, де $h_{j,k,i}$ – MAC-адреса певної КС мережі; $N_{H,j,k}$ – кількість КС мережі, які надсилали DNS-запити в межах певного TTL-періоду.

Аналогічно множину захоплених вхідних DNS-повідомлень представимо як $\chi^T = \bigcup_{j=d_1}^{d_{N_D}} \bigcup_{k=1}^{N_{TTL}} \chi_{j,k}$,

де χ_j – підмножини вхідних DNS-повідомлень щодо певного доменного імені, захоплені протягом часу спостереження; $\chi_{j,k}$ – підмножини вхідних DNS-повідомлень щодо певного доменного імені, захоплені в межах певного TTL-періоду; $\chi_{j,k} = \{\chi_{j,k,i}\}_{i=1}^{N_{\chi,j,k}}$, де $\chi_{j,k,i}$ – DNS-повідомлення, захоплене в межах певного TTL-періоду, $N_{\chi,j,k}$ – кількість DNS-повідомлень, захоплених в межах певного TTL-періоду.

З врахуванням полів вхідного DNS-повідомлення, дані з яких можуть бути використані для виявлення DNS-запитів бот-мереж, згідно стандарту RFC 1035 опишемо захоплений DNS-відгук щодо певного доменного імені кортежем:

$$\chi_{j,k,i} = \left\langle \chi_{j,k,i,H}, \chi_{j,k,i,TS}, \chi_{j,k,i,IP}, \left\langle \chi_{j,k,i,HD}, \chi_{j,k,i,ANS}, \chi_{j,k,i,ATH}, \chi_{j,k,i,ADD} \right\rangle \right\rangle, \quad (4)$$

$$j = d_1, \dots, d_{N_D}, k = 1, \dots, N_{TTL}, i = 1, \dots, N_{\chi,j,k},$$

де $\chi_{j,k,i,H}$ – MAC-адреса КС, що здійснювала DNS-запит; $\chi_{j,k,i,TS}$ – часовий штамп (час надходження DNS-пакета); $\chi_{j,k,i,IP}$ – IP-адреса джерела DNS-пакета; $\chi_{j,k,i,HD}, \chi_{j,k,i,ANS}, \chi_{j,k,i,ATH}, \chi_{j,k,i,ADD}$ – секції DNS-повідомлення: заголовок (Header), секція відгуків (Answer), секція серверів імен (Authority) та секція додаткової інформації (Additional) відповідно.

Заголовок DNS-повідомлення може бути описаний наступним чином:

$$\chi_{j,k,i,HD} = \left\langle \chi_{j,k,i,HD,ID}, \chi_{j,k,i,HD,OPC}, \chi_{j,k,i,HD,RC}, \chi_{j,k,i,HD,QDC}, \chi_{j,k,i,HD,ANC}, \chi_{j,k,i,HD,NSC}, \chi_{j,k,i,HD,ARC} \right\rangle,$$

$$j = d_1, \dots, d_{N_D}, k = \overline{1, N_{TTL}}, i = \overline{1, N_{\chi_{j,k}}}, \quad (5)$$

де $\chi_{j,k,i,HD,ID}$ – ідентифікатор, що дозволяє пов'язати DNS-запит та DNS-відгук (поле ID); $\chi_{j,k,i,HD,OPC}$ – тип запиту (поле OPCODE), $\chi_{j,k,i,HD,OPC} \in \{0, \dots, 2\}$, 0 – стандартний, 1 – інверсний, 2 – запит стану сервера; $\chi_{j,k,i,HD,RC}$ – код відгуку (поле RCODE), $\chi_{j,k,i,HD,RC} \in \{0, \dots, 5\}$, 0 – немає помилки, 1 – помилка в форматі запиту, 2 – збій сервера, 3 – доменне ім'я не існує тощо; $\chi_{j,k,i,HD,QDC}$ – кількість записів в секції запитів (поле QDCOUNT); $\chi_{j,k,i,HD,ANC}, \chi_{j,k,i,HD,NSC}, \chi_{j,k,i,HD,ARC}$ – кількість ресурсних записів в секціях відгуків, серверів імен та додаткової інформації (поля ANCOUNT, NSCOUNT, ARCOUNT) відповідно.

Секції відгуків, серверів імен та додаткової інформації мають однаковий формат та можуть бути описані як множини ресурсних записів наступним чином:

$$\chi_{j,k,i,S} = \left\{ \chi_{j,k,i,S,NM}, \chi_{j,k,i,S,TP}, \chi_{j,k,i,S,TTL}, \chi_{j,k,i,S,RDL}, \chi_{j,k,i,S,RDT} \right\}_{n=1}^{N_{RR,S}},$$

$$j = d_1, \dots, d_{N_D}, k = \overline{1, N_{TTL}}, i = \overline{1, N_{\chi_{j,k}}}, \quad (6)$$

де $S \in \{ "ANS", "ATH", "ADD" \}$, $\chi_{j,k,i,S,NM}$ – ім'я домена, до якого відноситься ресурсний запис (поле NAME); $\chi_{j,k,i,S,TP}$ – тип коду ресурсного запису (поле TYPE), визначає значення та формат даних в полі RDATA; $\chi_{j,k,i,S,TTL}$ – час життя записів DNS (поле TTL); $\chi_{j,k,i,S,RDL}$ – довжина поля RDATA (поле RDLENGTH); $\chi_{j,k,i,S,RDT}$ – рядок, що описує ресурс (поле RDATA); $N_{RR,S}$ – кількість ресурсних записів в секції (дорівнює значенню $\chi_{j,k,i,HD,ANC}, \chi_{j,k,i,HD,NSC}, \chi_{j,k,i,HD,ARC}$ для відповідної секції).

Множину ознак, які вказують на застосування періодичної зміни IP-відображення для шкідливого домена, визначимо наступним чином:

$$T_{\Psi_1} = \{ t_{mod}, t_{med}, t_{aver}, n_{IP}, S_{IP} \}, \quad (7)$$

де t_{mod} – TTL-період, мода; t_{med} – TTL-період, медіана; t_{aver} – TTL-період, середнє арифметичне значення; n_{IP} – кількість IP-адрес, пов'язаних з доменним ім'ям; S_{IP} – середня дистанція між IP-адресами, пов'язаними з доменним ім'ям.

Множину ознак, які вказують на застосування технології ухилення від виявлення бот-мереж «потік доменів», визначимо наступним чином:

$$T_{\Psi_2} = \{ t_{mod}, t_{med}, t_{aver}, f_S, n_D \}, \quad (8)$$

де f_S – бінарна ознака успішності DNS-запиту; n_D – кількість доменних імен, які спільно використовують IP-адресу.

Множину ознак, які вказують на застосування технології ухилення від виявлення бот-мереж «швидкозмінні» мережі, визначимо наступним чином:

$$T_{\Psi_3} = \{ t_{mod}, t_{med}, t_{aver}, n_A, S_A, n_{UA}, S_{UA} \}, \quad (9)$$

де n_A – кількість А-записів, що відповідають доменному імені, у вхідному DNS-повідомленні; s_A – середня дистанція між IP-адресами в множині А-записів для доменного імені у вхідному DNS-повідомленні; n_{UA} – кількість унікальних IP-адрес в множинах А-записів, що відповідають доменному імені, у вхідних DNS-повідомленнях; s_{UA} – середня дистанція між унікальними IP-адресами в множинах А-записів, що відповідають доменному імені, у вхідних DNS-повідомленнях.

Множину ознак, які вказують на застосування технології DNS-тунелювання, визначимо наступним чином:

$$T_{\Psi_4} = \{l_N, n_U, e_N, e_R, f_{UR}, l_P, f_{E_B}\}, \quad (10)$$

де l_N – довжина доменного імені; n_U – кількість унікальних символів в доменному імені; e_N – ентропія доменного імені; e_R – максимальне значення ентропії ресурсних записів DNS, які містяться в DNS-повідомленнях; f_{UR} – бінарна ознака використання рідковживаних типів записів DNS, або таких, які зазвичай не використовуються клієнтами; l_P – середній розмір DNS-повідомлень щодо доменного імені; f_{E_B} – функція залежності ентропії поля DNS-повідомлення від його довжини.

T_A – множина додаткових ознак, які вказують на застосування технологій ухилення від виявлення бот-мереж на основі DNS та можуть бути отримані засобами активного DNS-зондування:

$$T_A = \{n_{NS}, s_{NS}, v_{retry}, n_{ASN}, n_{ASA}\}, \quad (11)$$

де n_{NS} – кількість NS-записів у DNS-відповіді; s_{NS} – середня дистанція між IP-адресами для множини NS-записів щодо доменного імені; v_{retry} – значення поля retry, отримане у DNS-відповіді на SOA-запит; n_{ASN} – кількість різних номерів автономних систем (ASN), до яких належать IP-адреси, пов'язані з серверами імен; n_{ASA} – кількість різних номерів автономних систем, до яких належать IP-адреси, пов'язані з доменним іменем.

Наявність застосування технологій ухилення від виявлення бот-мереж на основі DNS може бути представлена із застосуванням знань, які можуть бути подані у вигляді наступних правил:

$$\begin{aligned} & \text{if } t_{\text{mod}} \in [0,900] \text{ and } t_{\text{med}} \in [0,900] \text{ and } t_{\text{aver}} \in [0,900] \text{ and} \\ & \text{and } n_{IP} \in (5, \infty) \text{ and } s_{IP} \in (65535, \infty) \text{ and } n_{ASA} > 2 \Rightarrow \text{cycling of IP mappings} \\ & \text{if } (t_{\text{mod}} \in [0,900] \text{ and } t_{\text{med}} \in [0,900] \text{ and } t_{\text{aver}} \in [0,900]) \text{ and} \\ & \text{and } ((n_A \in (5, \infty) \text{ and } s_A \in (65535, \infty)) \text{ or } (n_{UA} \in (8, \infty) \text{ and } s_{UA} \in (65535, \infty)) \text{ or } n_{AS} > 2) \text{ and} \\ & \text{and } (s_{NS} \in (65535, \infty) \text{ or } n_{ASN} > 2 \text{ and } n_{NS} > 3 \text{ and } v_{\text{retry}} \in [0,900]) \Rightarrow \text{fast_flux} \\ & \text{if } n_D \in [8, \infty] \Rightarrow \text{domain_flux} \end{aligned} \quad (12)$$

Представимо концептуально процес виявлення бот-мереж на основі аналізу DNS-трафіка із залученням розробленої моделі наступним чином:

$$M_D = \langle \chi^T, f_1^T, C_{GA}^T, C_{ET}^T, f_2^T, T \rangle, \quad (13)$$

де χ^T – множина захоплених вхідних DNS-повідомлень (DNS-відгуків) до множини H комп’ютерних систем мережі; f_1^T – функція співставлення доменних імен з «білим» та «чорним» списками; C_{GA}^T – множина алгоритмів ідентифікації бот-мереж на основі їх групової активності в DNS-трафіку [23]; C_{ET}^T – множина алгоритмів виявлення бот-мереж, які застосовують технології ухилення від виявлення на основі DNS [24]; f_2^T – функція локалізації КС, інфікованих ботами, та блокування дій ботів; $T = \{t_m\}_{m=0}^{N_T}$ – інтервал часу спостереження, де N_T – кількість ітерацій спостереження.

Формалізована схема процесу виявлення бот-мереж на основі аналізу DNS-трафіка подана на рис. 10, а.

Процес виявлення бот-мереж на основі аналізу DNS-трафіка також може бути поданий у вигляді часової діаграми (рис. 10, б), де t_0 – час початку спостереження (збору вхідного DNS-трафіка), $\{t_1, \dots, t_n\}$ – тривалості ітерацій спостереження.

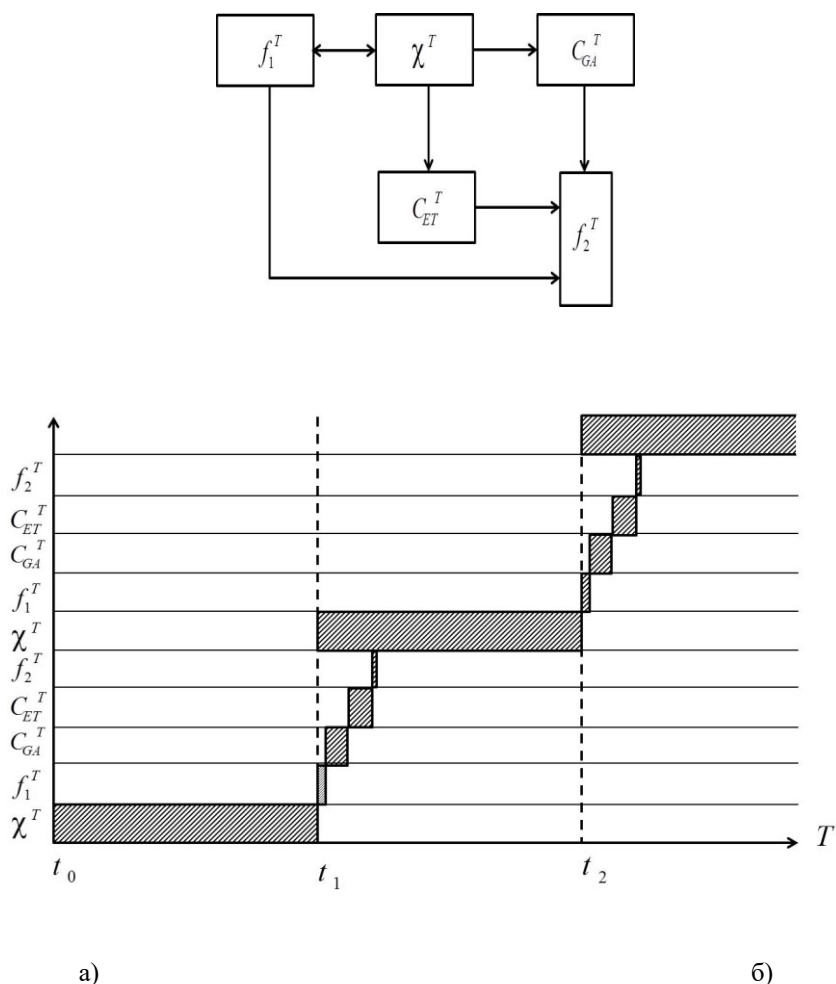


Рис. 10 – Формалізована схема (а) та часова діаграма (б) процесу виявлення бот-мереж на основі аналізу DNS-трафіка

Експериментальна частина. З метою перевірки моделі на адекватність було проведено ряд експериментів по виявленню бот-мереж із залученням розробленої моделі. Для проведення експериментів було згенеровано множину спеціального програмного забезпечення (ПЗ) з функціональними властивостями ботів. Створені боти були розподілені на чотири групи, кожна з яких відповідала одній з технологій

ухилення бот-мереж – «потік доменів», «швидкозмінні» мережі, DNS-тунелювання та періодична зміна IP-відображення.

З метою імітації C&C-серверів бот-мереж було зареєстровано множину доменних імен, що надало можливість імітувати застосування зазначених вище технологій ухилення бот-мереж на основі DNS. C&C-сервери здійснювали такі дії, як періодична зміна IP-відображення для доменного імені, зміна доменного імені, циклічна зміна A-записів та NS-записів DNS для доменного імені, а також здійснювали прийом та передачу прихованого трафіка за допомогою технологій DNS-тунелювання.

Створеними ботами було інфіковано мережу з 100 хостів, кожна бот-мережа відтворювала різні сценарії здійснення DNS-запитів щодо доменних імен C&C-серверів. Також було імітовано активність користувачів, для чого хости мережі здійснювали DNS-запити щодо легітимних ресурсів.

Експеримент тривав протягом 24 годин, за цей час було проаналізовано 6169 DNS-відповідей. В табл. 1 представлено кількість виявлених DNS-відповідей щодо шкідливих доменів. Таким чином, експерименти по виявленню бот-мереж із залученням розробленої моделі продемонстрували здатність виявлення бот-мереж на етапі передачі трафіку командування та контролю бот-мережею, до початку етапу здійснення атаки, на рівні до 99,1%, в той час як рівень хибних спрацювань становить близько 0,1 %.

Таблиця 1

Результати експериментів: кількість DNS-запитів, здійснених ботами, виявлені DNS-відповіді ботів та хибні спрацювання

Назва технології ухилення	Кількість DNS- запитів, здійснених ботами / з них групові	Виявлені DNS-відповіді / з них на групові DNS-запити	Хибні спрацювання, %
Періодична зміна IP-відображення	962 / 680	951 / 665	0,03
«Потік доменів»	3595 / 3047	3571 / 3034	0,05
«Швидкозмінні» мережі	1230 / 843	1189 / 836	0,03
DNS-тунелювання	382 / 91	372 / 84	0
Всього	6169 / 4661	6083 / 4619 (98,6% / 99,1%)	0,11

Висновки. В роботі представлено модель бот-мереж з врахуванням використання бот-мережами DNS на різних етапах життєвого циклу бот-мережі, а також застосування бот-мережами технологій ухилення від виявлення на основі DNS і різних способів комунікації ботів з командно-контролюючими центрами бот-мереж. Використання розробленої моделі надає можливість здійснювати виявлення ботів бот-мереж централізованої, розподіленої та гібридної архітектури на різних етапах життєвого циклу бот-мереж. Експерименти показали здатність виявлення бот-мереж із залученням розробленої моделі на етапі передачі трафіку командування та контролю бот-мережею, до початку етапу здійснення атаки, на рівні до 99,1%, в той час як рівень хибних спрацювань становить близько 0,1 %.

References

1. OpenDNS. Security Whitepaper. The Role of DNS in Botnet Command & Control [Електронний ресурс] – Режим доступу: http://info.opendns.com/rs/opendns/images/OpenDNS_SecurityWhitepaper-DNSRoleInBotnets.pdf. – 9.12.2019 р.
2. DAMBALLA. Botnet detection for communications service providers [Електронний ресурс] – Режим доступу: https://www.damballa.com/downloads/r_pubs/WP_Botnet_Detection_for_CSPs.pdf. – 9.12.2019 р.
3. DAMBALLA. Botnet Communication Topologies. Understanding the intricacies of botnet command-and-control [Електронний ресурс] – Режим доступу: https://www.damballa.com/downloads/r_pubs/WP_Botnet_Communications_Primer.pdf. – 9.12.2019 р.
4. VIRUS BULLETIN. Grooten, M. VB2017 videos on attacks against Ukraine, 2017 [Електронний ресурс] – Режим доступу: <https://www.virusbulletin.com/blog/2017/12/vb2017-videos-attacks-against-ukraine/>. – 9.12.2019 р.
5. NEXUSGUARD. DDoS Threat Report 2019 Q3 [Електронний ресурс] – Режим доступу: <https://www.nexusguard.com/threat-report-q3-2019>. – 9.12.2019 р.
6. Wainwright, P. An Analysis of Botnet Models / P. Wainwright, H. Kettani // Proceedings of the 2019 3rd International Conference on Compute and Data Analysis. – 2019. – pp. 116-121.
7. Feily, M. A survey of botnet and botnet detection / M. Feily, A. Shahrestani, S. Ramadass // Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies (SECURWARE '09). Washington, DC: IEEE Computer Society. – 2009. – pp. 268-273.
8. Khosroshahy, M. The SIC botnet lifecycle model: A step beyond traditional epidemiological models / M. Khosroshahy, M.K.M. Ali, D. Qiu // Computer Networks. – 2013. – Vol. 57, No. 2. – pp. 404-421.
9. Liu, J. Botnet: Classification, attacks, detection, tracing, and preventive measures / J. Liu, Y. Xiao, K. Ghaboosi, H. Deng, J.

Zhang // EURASIP Journal on Wireless Communications and Networking . – 2009. – Vol. 1. – pp. 1-11.

10. Rodríguez-Gómez, R. Survey and taxonomy of botnet research through life-cycle / R. Rodríguez-Gómez, G. Maciá-Fernández, P. García-Teodoro // ACM Computing Surveys (CSUR) . – Vol. 201345, No. 4. – pp. 1-33.

11. Kephart, J. Measuring and modeling computer virus prevalence / J. Kephart, S. White // Proceedings of the 1993 IEEE Computer Security Symposium on Research in Security and Privacy. – New York: Institute of Electrical and Electronic Engineers (IEEE). – 1993. – pp. 2-15.

12. Jerkins, J. Mitigating IoT insecurity with inoculation epidemics / J. Jerkins, J. Stupiansky // Proceedings of the ACMSE 2018 Conference (ACMSE '18). – New York: Association for Computing Machinery (ACM). – 2018. – Vol. 4. – pp. 1 - 6.

13. van Roosmalen, J. Applying deep learning on packet flows for botnet detection / J. van Roosmalen, H. Vranken, M. van Eekelen // Proceedings of the 33rd Annual ACM Symposium on Applied Computing (SAC '18). – New York: Association for Computing Machinery (ACM) . – 2018. – pp. 1629-1636.

14. El Mir, I. Towards a stochastic model for integrated detection and filtering of DoS attacks in cloud environments / I. El Mir, D. Kim, A. Haqiq // Proceedings of the 2nd International Conference on Big Data, Cloud and Applications. – New York: Association for Computing Machinery. – 2017. – Vol. 10. – pp. 1-6.

15. Song, L. Modeling and analyzing of botnet interactions / L. Song, Z. Jin, G. Sun // Physica A: Statistical Mechanics and its Applications. – 2011. – Vol. 390, No. 2. – pp. 347-358.

16. Wang, Y. Dynamic game model of botnet DDoS attack and defense / Y. Wang, J. Ma, L. Zhang, W. Ji, D. Lu, X. Hei // Security and Communication Networks . – 2016. – Vol. 9, No. 16. – pp. 3127-3140.

17. Kolokoltsov, V. N. Mean-field-game model for botnet defense in cyber-security / V. N. Kolokoltsov, A. Bensoussan // Applied Mathematics & Optimization. – 2016. – Vol. 74, No. 3. – pp. 669-692.

18. Divita, J. An approach to botnet malware detection using nonparametric Bayesian methods / J. Divita, R. Hallman // Proceedings of the 12th International Conference on Availability, Reliability and Security (ARES'17). – New York: Association for Computing Machinery (ACM). – 2017. – Vol. 75. – pp. 1-9.

19. Faloutsos, M. Detecting malware with graph-based methods: traffic classification, botnets, and Facebook scams / M. Faloutsos // Proceedings of the 22nd International Conference on World Wide Web (WWW '13 Companion). – New York: Association for Computing Machinery (ACM). – 2013. – pp. 495-496.

20. Li, Z. Botnet economics: Uncertainty matters / Z. Li, Q. Liao, A. Striegel // Managing information risk and the economics of security. Johnson, M. E. (Ed.). – Springer, Boston, MA . – 2009. – pp. 245-267.

21. Bottazzi, G. The botnet revenue model / G. Bottazzi, G. Me // Proceedings of the 7th International Conference on Security of Information and Networks (SIN '14). – New York: Association for Computing Machinery (ACM). – 2014. – pp. 459-465.

22. Li, Z. Toward a monopoly botnet market / Z. Li, Q. Liao // Information Security Journal: A Global Perspective. – 2014. – Vol. 23, No. 4-6. – pp. 159-171.

23. Pomorova, O. A technique for the botnet detection based on DNS-traffic analysis / O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, K. Bobrovnikova // Computer Networks: Communications in Computer and Information Science. Springer International Publishing. – 2015. – Vol. 522. – pp. 127-138.

24. Pomorova, O. Anti-evasion technique for the botnets detection based on the passive DNS monitoring and active DNS probing / O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, K. Bobrovnikova // Computer Networks: Communications in Computer and Information Science. Springer International Publishing. – 2016. – Vol. 608. – pp. 83-95.

Рецензія/Peer review : 28.11.2019

Надрукована/Printed : 03.01.2020