DOI 10.31891/2307-5732-2020-283-2-141-149
УДК 621.391 160164

А.О. НІЧЕПОРУК, А.А. НІЧЕПОРУК, Ю.О. НІЧЕПОРУК, А.Д. КАЗАНЦЕВ
Хмельницький національний університет

МЕТОД ВИОКРЕМЛЕННЯ ФРАГМЕНТІВ БОТ-МЕРЕЖ НА ОСНОВІ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ

В роботі запропоновано метод виявлення фрагментів бот-мереж на основі аналізу мережевого трафіку. Метод заснований на представленні шкідливої активності, що здійснюють боти в локальній мережі у вигляді зваженого орієнтованого графу, де вершинами виступають хости мережі, а ребрами – зв'язки між хостами. З метою виявлення шкідливої активності на хості використаємо IDS Snort – мережеву систему виявлення вторгнень, що працює за принципом мережевих сніферів. Всі шкідливі активності розподілено сім категорій: контроль, сканування, спам, отримання інформації, завантаження, атака та категорія інші Зв'язність графу забезпечується наявністю ребер, які мають ваги. Вагою ребра, що з'єднує два вузли, є імовірність того, що два вузли є частиною однієї бот-мережі. Для визначення імовірності того, що два вузли є частиною однієї бот-мережі використовується правило Байєса. Оновлення вагів ребер відбувається після кожного інтервалу часу в межах загального часу моніторингу шкідливої активності. Наприкінці часу моніторингу здійснюється розбиття отриманого графу на підграфи, що відповідають окремим бот-мережам. Для розбиття графу на підграфи розроблений алгоритм, що дозволяє виділити фрагменти різних бот-мереж, які присутні в локальній мережі. Представлений алгоритм використовує "жадібний" підхід та ґрунтується на обчисленні максимального виграшу, який може принести перенесення будь-якої вершини в той чи інший підграф розбиття.

Ключові слова: *бот-мережа, мережевий трафік, орієнтований граф, хост.*

A.O. NICHEPORUK, A.A. NICHEPORUK, Y.O. NICHEPORUK, A.D. KAZANTSEV
Khmelnytskyi National University

METHOD OF DETECTING FRAGMENTS OF BOTNETS BASED ON THE ANALYSIS OF NETWORK TRAFFIC

The paper proposes a method of detecting fragments of botnets based on the analysis of network traffic. The proposed method depends primarily on the temporary relationships of malicious actions between computers on the network and does not depend on the architectures of botnets and the tools used to manage them. The method is based on the representation of malicious activity performed by bots in the local network in the form of a weighted oriented graph, where the vertices are the hosts of the network, and the edges are the connections between the hosts. In order to detect malicious activity on the host, we use IDS Snort - a network intrusion detection system that works on the principle of network sniffers. All malicious activities are divided into seven categories: control, scanning, spam, information retrieval, downloads, attacks and other categories. The connectivity of the graph is ensured by the presence of edges that have weights. The weight of the edge connecting the two nodes is the probability that the two nodes are part of the same bot network. The Bayesian rule is used to determine the probability that two nodes are part of the same bot network. The edge weights are updated after each time interval within the total time of harmful activity monitoring. At the end of the monitoring time, the obtained graph is divided into subgraphs corresponding to individual bot networks. An algorithm has been developed to divide a graph into subgraphs, which allows to select fragments of different bot networks that are present in the local network. The presented algorithm uses a "greedy" approach and is based on the calculation of the maximum gain that can bring the transfer of any vertex in a subgraph of the partition. To verifying the effectiveness of the proposed method, a number of experiments were performed, which included determining the fact of the presence of a botnet on a local computer network.

Keywords: *botnet, network traffic, oriented graph, host.*

Вступ

У сучасному світі бот-мережі стали однією з найбільших загроз безпеці для корпоративних мереж підприємств. Зловмисники або ботмайстри використовують бот-мережі для запуску DDoS, щоб паралізувати великі веб-сайти. Вони також використовують «фішингові» атаки, щоб викрасти конфіденційну інформацію (наприклад, облікові записи користувачів та паролі), надіслати об'ємну рекламу електронною поштою та/або здійснити шахрайство з кліками. Навіть незважаючи на те, що технології їх виявлення були значно удосконалені, загроза бот-мереж все ще залигається актуальною і сьогодні.

Сучасні методи в першу чергу спрямовані на виявлення бот-мереж та попередження їх поширення [1]. Проте при виявленні таких загроз розробники досить часто нехтують умовою наявності декількох бот-мереж, що інфікували одну локальну мережу [2]. Це значно знижує ефективність залучених методів по виявленню бот-мереж в локальній мережі. Тому актуальною є задача розроблення методу виокремлення фрагментів бот-мереж на основі аналізу мережевого трафіку.

Попередні дослідження

Розглянемо детальніше відомі методи виявлення та локалізації бот-мереж у локальних мережах.

Методи виявлення бот-мереж, що представлені в роботах [3, 4], базуються на аналізі трафіку та залучають методи, які засновані на порівнянні отриманих результатів аналізу трафіку з шаблонами бази аномалій. Головним недоліком цієї групи підходів є необхідність постійного розбору трафіку та виділення важливих характеристичних ознак, які можуть змінюватись зловмисниками для підвищення ступеня прихованості бот-мереж. Окрім того, представлені підходи не враховують архітектури існуючих бот-мереж, в наслідок чого блокування підозрілих пакетів, в подальшому не гарантує їх повторення від інших вузлів бот-мережі.

У роботі [5] авторами запропоновано систему виявлення P2P бот-мереж на основі машинного навчання. Запропонований підхід здійснює отримання згорткових ознак, на основі яких здійснюється

формування моделі класифікатора із залученням штучної нейронної мережі з прямим зв'язком. Результати експерименту показують, що ефективність виявлення з використанням згорткових ознак вища, у порівнянні із використанням традиційних ознак. За результатами проведених експериментів представленого методу ефективність виявлення P2P бот-мереж склала 94,7% із рівнем помилкових спрацювань 2,2%.

Іншим підхід до виявлення P2P бот-мереж представлений у роботі [6]. Запропонована система виявлення бот-мереж ґрунтується на проведені обліку мережевого трафіку із використанням протоколу NetFlow. Потік пакетів аналізується з використанням трьох основних модулів: експорту, колектора і аналізатора. Модуль експорту захоплює пакет і виконує моніторинг вмісту пакета. Колектор фіксує потік потоку, а компонент аналізатора ініціює автоматизований аналіз трафіку із захопленою інформацією про пакет. Інформація про потік пакетів збирається за допомогою віртуального інтерфейсу та фізичного зонда. Віртуальний інтерфейс використовується для збору інформації про зловмисний трафік між віртуальними машинами, а фізичний зонд збирає зловмисну інформацію про трафік між мережевими мостами, що з'єднують віртуальні машини.

В останні час цільовим напрямком атаки бот-мереж окрім, локальних комп'ютерних мереж, стали IoT мережі. Серед основних напрямків по виявленню бот-мереж в IoT мережах можна відзначити використання ентропійного аналізу, визначення належності досліджуваного трафіку до одного із законів розподілу випадкових величин, залучення систем на основі правил, використання методів машинного навчання, обчислення та порівняння статистичних величин, наприклад, кількості пакетів, часу затримки між пакетами, кількість одиничних пакетів, протокол передачі, тощо.

У роботі [7] було запропоновано алгоритм захисту IoT мереж перед DDoS-атакам, шляхом надання IoT пристроям інтелектуальних можливостей, подібних до ботів. Щоб зрозуміти різницю між доброякісним і шкідливим запитом, вузол здійснює аналіз вмісту пакету. Хоча результати показали, що такий підхід допомагає запобігти атакам, проте продуктивність роботи цього методу сильно залежить від обмежених ресурсів кожного бота.

Основи методу виокремлення фрагментів бот-мереж на основі аналізу мережевого трафіку

Згідно із визначенням "бот" є саморозповсюджуваною частиною коду, який інфікує кінцеві хости через різні вразливості системи, зокрема вразливості, засновані на переповненні буфера та інші подібні вразливості засновані на атаках соціальної інженерії із запуском троянських програм. У зв'язку з цим вони схожі на віруси і хробаки. Боти, однак, відрізняються від ізольованих вірусів і черв'яків у тому сенсі, що вони демонструють певні специфічні комунікаційні схеми і контролюються зовнішнім об'єктом, який зазвичай називають C&C сервером. Схеми спілкування, пов'язані з ботами, відображають їхню здатність брати участь у скоординованій шкідливій діяльності.

Виявлення бот-мереж є головною метою адміністраторів мережної безпеки. Якщо бот-мережа може бути ідентифікована, і успішно виділена в мережі, таку ж стратегію дезінфекції можна швидко застосувати до всіх хостів відразу. Це значно знижує ймовірність повторного зараження від все ще інфікованих хостів, коли мережа одночасно очищається від одного вузла. Адміністратори мережевої безпеки також загалом вважають, що якщо одночасно можна було виявити всі інфіковані хости, то такий підхід був би більш ефективніший, ніж тестування кожного хоста в ізоляції на наявність або відсутність бот-мережі. Тому головною мета даного методу є виявлення всіх ботів у кожній з бот-мереж, які можуть інфікувати комп'ютерну мережу.

Запропонований метод залежить, в першу чергу, від тимчасових взаємозв'язків шкідливих дій між комп'ютерами в мережі і не залежить від архітектур бот-мереж і засобів, які використовуються для їхнього керування. В процесі життєвого циклу, наприклад, час життя бота, а також тривалість і вибір шкідливих дій, замовлених керуючим ботом, можна очікувати, що ці параметри змінюватимуться з часом. Запропонований метод включає механізми, які дозволяють графу, що представляє інфікованими комп'ютерами, розвиватися з часом. Що стосується того, як такий граф змінюється з часом, то особливо важливими є граничні ваги, які виводяться з тимчасових взаємозв'язків шкідливих дій на кінцевих точках ребер.

Визначення точної мітки кожному інфікованому хосту в мережі досягається шляхом представлення інфікованих хостів зваженим повнозв'язним графом:

$$G = (V, E) \tag{1}$$

де V – набір вузлів, причому кожен інфікований хост – вузол, а E – набір ребер.

Вузол додається до набору V, коли новий хост проявляє шкідливу активність. З іншого боку, вузол видаляється з V, коли відповідний хост припиняє виявляти шкідливу активність і надійно відомо, що був дезінфікований. Ці вузли можуть бути додані повторно до V, якщо відповідні хости знову виявляють шкідливу активність.

Зв'язність графу G забезпечується наявністю ребер, які мають ваги. Вагою ребра, що з'єднує два вузли, є імовірність того, що два вузли є частиною однієї бот-мережі. Ваги ребра можуть приймати значення в діапазоні від 0 до 1. Вага ребра, яка близька до 1, вказує на те, що існує велика ймовірність того, що два вузли є частиною однієї бот-мережі, тоді як вага ребра, близька до 0, вказує на те, що ці два вузли однозначно належать до різних бот-мереж. Вагові коефіцієнти оновлюються в кінці кожного часового інтервалу, щоб відобразити спостереження за певний час. Наприклад, якщо два вузли виконували одну і ту ж зловмисну активність протягом цього періоду, відповідна вага ребра збільшується, оскільки імовірність того, що два вузли є частиною однієї бот-мережі, збільшиться. Оскільки комп'ютерні системи, що належать

до однієї бот-мережі, прагнуть показувати подібні за часом дії протягом тривалого часу, ребра між такими хостами в графі G набувають великих ваг. З іншого боку, ребра між хостами, які належать до двох різних бот-мереж, демонструють низькі граничні ваги.

Припускаючи, що мережа комп'ютерів була атакована множиною бот-мереж, для виявлення та ідентифікації бот-мереж, існує необхідність в розбитті графа G на непересічні підграфи, кожен з яких відповідає унікальній бот-мережі.

Узагальнену схему методу виокремлення фрагментів бот-мереж на основі аналізу мережевого трафіку наведено на рис. 1.

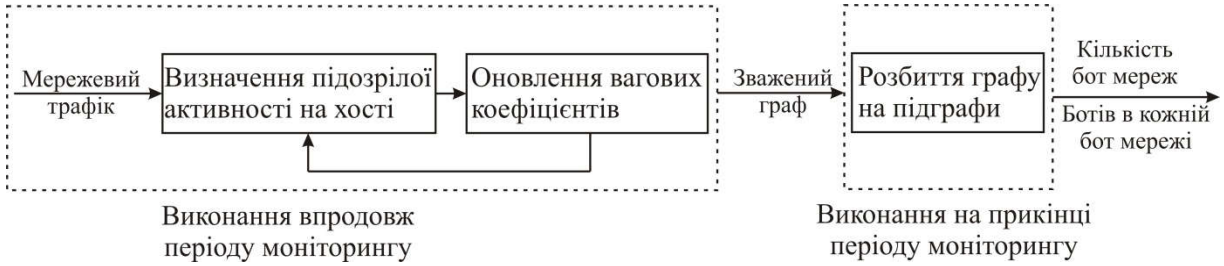

Рис. 1. Узагальнена схема методу виокремлення фрагментів бот-мереж на основі аналізу мережевого трафіку

Таким чином запропонований метод складається з наступних кроків:

1. Визначення підозрілої активності на хостах комп'ютерної мережі;
2. Оновлення вагових коефіцієнтів на протязі всього періоду моніторингу шкідливої активності;
3. Формування зваженого повнозв'язного графу;
4. Розбиття отриманого графу на підграфи, що відповідають окремим бот-мережам.

Розглянемо детальніше кроки запропонованого методу.

Виявлення шкідливої активності на хості

Першим кроком запропонованого методу виокремлення фрагментів бот-мереж є виявлення шкідливої активності на хості.

Щоб виявити і виміряти спільні події, необхідно мати інструменти, необхідні для виявлення шкідливих дій. На протязі попередніх років були розроблені різні методи. Наприклад, маршрутизатор, відповідальний за локальну мережу, може бути обладнаний монітором, який може реєструвати мережеві траси, пов'язані з кожним клієнтом, у списках протоколу конфігурації динамічного хоста (DHCP) і статичних IP-адрес. Таке ведення мережевих трас підтримується багатьма маршрутизаторами, такими як Cisco і Juniper. Згодом ці журнали можуть бути проаналізовані, щоб виявити, чи відбулися шкідливі дії на хостах.

Наприклад, спам від хоста може бути виявлений шляхом моніторингу наступних ознак: запити системи доменних імен для записів обмінників поштових повідомлень, що надходять від хоста; частоту, з якою хост ініціює підключення протоколу передачі простих повідомлень до зовнішніх поштових серверів. Виявлення зловмисних завантажень може бути здійснено за допомогою механізму виявлення підписів BotHunter та Snort. Інші шкідливі дії можуть бути аналогічно виявлені за допомогою вільно доступних систем виявлення та запобігання вторгнення в мережу.

З метою виявлення шкідливої активності на хості використано IDS Snort [9] – мережеву систему виявлення вторгнень, що працює за принципом мережевих сніферів. IDS Snort за певними сигнатурам шукає шкідливий трафік, який проходить через мережу, що захищається, і попереджають про його наявність адміністратора. Snort може бути встановлений на численних платформах операційних систем, таких як Windows, Linux і т.д. Snort працює в режимі реального часу і має можливість оповіщення про дані мережевого трафіку та їх аналіз. Повідомлення буде надіслано в системний журнал або окремі файли "попередження" або у спливаючі вікна. Snort логічно поділяється на кілька компонентів. Ці компоненти забезпечують покроковий процес виявлення конкретних атак і генерують вивід у необхідному форматі з системи виявлення. Компонентами Snort є пакетний декодер, препроцесор, движок виявлення, система реєстрації і оповіщення, а також модулі виведення.

Оновлення вагових коефіцієнтів

Припускаючи, що активності R були визначені під час дискретних інтервалів часу, розглянемо формування графа G та оновлення його вагових коефіцієнтів. Оновлення вагових коефіцієнтів графу G є необхідним у зв'язку із динамічною зміною в часі активностей хостів, що перебувають під контролем бот-мережі. Для визначення та оновлення вагових коефіцієнтів будемо вважати, що кожен дискретний часовий інтервал має одиничну тривалість.

Нехай i, j дві вершини графа G, такі, що представляють комп'ютерну систему в локальній мережі, і які проявляють підозрілу активність. Нехай $C_{i,j}$ латентна змінна, що визначає чи є комп'ютерні системи i та j складовими однієї бот-мережі, так, що $C_{i,j}=1$ визначає, що дві комп'ютерні системи i та j є складовими

однієї бот-мережі, а $C_{i,j} = 0$ – ні. Тоді виникає завдання визначення $p(C_{i,j} = 1)$, тобто оцінити ймовірність того, що два хости *i* та *j* є частиною однієї і тієї ж бот-мережі на підставі спостережень до поточного часу *t*. У будь-який заданий час *t* вага ребра між двома вузлами *i, j* визначимо як $p(C_{i,j}^t = 1)$. Оскільки вагою ребра є імовірність, то вона приймає значення від 0 до 1. Оскільки час є дискретною величиною в одиничні проміжки часу, то *t* = {0, 1, 2,…}. Оновлення ваги ребра в кожний момент часу *t* буде базуватися на спостереженнях шкідливої активності між *t - 1* і *t*.

Нехай $A_i^{(t)}$ та $A_j^{(t)}$ змінні, що визначають активність, що спостерігається *i* та *j* комп'ютерною системою відповідно, між часом *t* - 1 та *t*. Нехай a_i та a_j поточні значення, що приймають змінні $A_i^{(t)}$ та $A_j^{(t)}$. a_i та a_j відображають значення з таблиці 3.1. a_i може приймати значення від 0,1,…,*K,* де значення $a_i = 0$ визначає, що хост не показав шкідливої активності протягом розглянутого часового інтервалу, в той час як $a_i = m\,(1 \le m \le K)$ вказує, що хост виконував m-у шкідливу активність в інтервалі часу *t* - 1 та *t* .

В кожний дискретний момент часу оновлюватимемо попереднє значення імовірності $p(C_{i,j}^{t-1})$ між кожною парою вузлів до останнього значення $p(C_{i,j}^t)$ використовуючи правило Баєса. Оновлена вага ребра між двома вузлами залежить від попередньої ваги ребра $p(C_{i,j}^{t-1})$, тобто ймовірності побачити цю групу активностей $A_i^{(t)}$ та $A_j^{(t)}$ з урахуванням значення $C_{i,j}$, і загальної ймовірності побачити цю пару дій $p(A_i^{(t)}, A_j^{(t)})$:

$$p(C_{i,j}^{(t)} \mid A_i^{(t)} = a_i, A_j^{(t)} = a_j) = \frac{p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid C_{i,j}^{(t)})\, p(C_{i,j}^{(t-1)})}{p(A_i^{(t)} = a_i, A_j^{(t)} = a_j)} \tag{2}$$

Ймовірність $p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid C_{i,j} = 1)$ та $p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid C_{i,j} = 1)$, які є важливими для обчислення лівої частини рівняння (3), оцінюються для кожної пари можливих активностей ботів, перелічених у таблиці 3.1. Слід зазначити, що ніколи не потрібно безпосередньо обчислювати знаменник у правій частині рівняння (3), тобто тому, що обмеження нормалізації на ймовірності, яка пов'язана з двома результатами для $C_{i,j}$, опосередковано дає знаменник у рівнянні (3). Тобто, якщо ми оцінюємо ці дві ймовірності без знання знаменника, то той факт, що ці ймовірності повинні скласти одиницю, дасть нам невідоме значення для знаменника.

Ваги ребер оновлюються між парою хостів *i, j* наприкінці часового інтервалу, лише якщо принаймні на одному з них виявлено шкідливу активність протягом спостережуваного часового інтервалу, тобто $A_i^{(t)} \neq 0$ або $A_j^{(t)} \neq 0$.

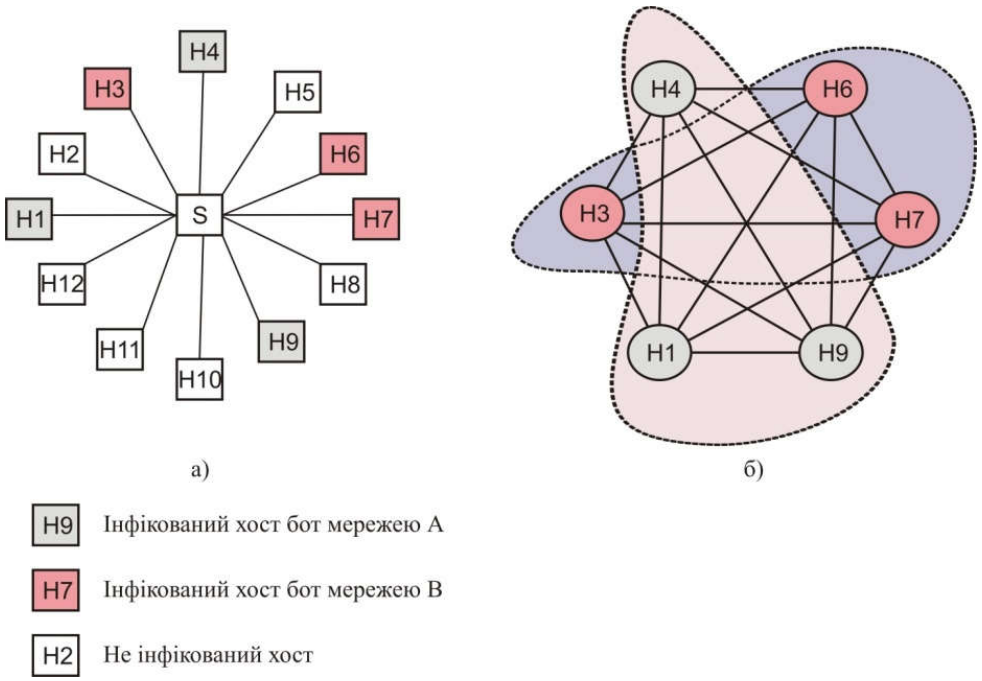

Рис. 2. Схематичне зображення розбиття хостів, що проявляють шкідливу активність на два графи: а) локальна мережа, що інфікована двома бот-мережами; б) розбиття повнозв'язного графу на два підграфи

На початку періоду моніторингу, тобто при початковій побудові графу *G*, ваги ребер між кожною парою вузлів, що виявляють шкідливу активність, ініціалізуються значенням 0,5. Вага 0,5 означає ймовірність того, що хости в двох кінцевих точках інфіковані однією бот-мережею, з одного боку, та іншою – з другого боку. Якщо мережа інфікована тільки одним або двома бот-мережами, можна стверджувати, що це початкове значення ймовірності буде вищим. Але тоді можна також заперечити, що ця початкова ймовірність буде нижчою, якщо мережа була атакована багатьма малими бот-мережами. За відсутності будь-якої попередньої інформації про кількість та/або розміри бот-мереж у локальній мережі, вирішено здійснити неупереджену ініціалізацію значенням 0,5, для всіх ребер між вузлами, що демонструють підозрілу діяльність. Граф *G* створюється, коли новий хост у мережі починає показувати ознаки шкідливої активності (на цей момент він ще не є частиною *G*), далі зазначений хост включається у *G*, а нові ребра утворюються між щойно доданим хостом та всіма хостами, які були присутні на даний момент в графі *G*, із встановлення початкової ваги кожного ребра на рівні 0,5. Схематичне зображення розбиття хостів, що проявляють шкідливу активність на два підграфи наведено на рис. 2.

Оцінка розподілу спільної діяльності

Як зазначалось раніше $A_i^{(t)}$ та $A_j^{(t)}$ є активностями, що були виявлені мережевим монітором на хостах *i* та *j* відповідно, на проміжку часу *t* - 1 та *t*. Тоді важливою задачею є оцінка розподілу спільної діяльності $p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid C_{i,j} = 1)$ та $p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid C_{i,j} = 1)$ для всіх можливих пар a_i та a_j.

Нехай $B_i^{(t)}$ та $B_j^{(t)}$ упорядковані дії бот-майстра на вузлах *i* та *j* в інтервалі часу *t*-1 та *t*, причому b_i та b_j є двома значеннями для таких дій. Подібно до $A_i^{(t)}$, $B_i^{(t)}$ змінна, що може приймати значення від 0 до *K*, де *K* – кількість активностей. Слід зазначити, що $b_i = 0$ вказує, що бот-майстер не вибрав жодної зловмисної активності для часового інтервалу (тобто його бот-мережа не активна) і $b_i = 1$ ($1 \le l \le K$) вказує, що бот-майстер вибрав *l*-у зловмисну активність. Імовірність $p(B_i^{(t)} = b)$ представляють відносні частоти різних шкідливих дій, виконаний бот-майстром. Виходячи з емпіричних спостережень, можна припустити, що в середньому бот-майстер не взаємодіє із бот-мережею на протязі 25% часу.

Конкретні значення $A_i^{(t)}$ та $A_j^{(t)}$ в момент часу *t* залежать від дій бот-майстра $B_i^{(t)}$ та $B_j^{(t)}$ для хостів *i* та *j* на протязі цього часового інтервалу. Окрім того, слід врахувати можливість того, що обидва вузли можуть контролюватися одним і тим же бот-майстром.

Тому для визначення ймовірність того, що два різних вузла показують дві задані дії, коли обидва вузли знаходяться в одній бот-мережі або знаходяться в різних бот-мережах запишемо:

$$p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid C_{i,j}^{(t)}) = \sum_{b_i=0}^{K} \sum_{b_j=0}^{K} p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid B_i^{(t)} = b_i, B_j^{(t)} = b_j, C_{i,j}^{(t)}) \times$$
$$\times p(B_i^{(t)} = b_i, B_j^{(t)} = b_j \mid C_{i,j}^{(t)}) \quad (3)$$

Покажемо більш прості версії вищезазначеної формули для двох випадків $C_{i,j} = 1$ та $C_{i,j} = 0$. Коли $C_{i,j} = 1$ обидва боти *i* та *j* контролюються одним і тим самим бот-майстром. Цей факт можна визначити відношенням $B_i = B_j = B$, тобто *B* є випадковою змінною для визначення загальної активності у двох вузлах. У цьому випадку рівняння 3 буде мати вигляд:

$$p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid C_{i,j} = 1) = \sum_{b=0}^{K} p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid B^{(t)} = b) \times p(B^{(t)} = b) \quad (4)$$

де *b* – будь-яке конкретне значення для випадкової величини *B,* загальна діяльність, напрямлена бот-майстром на бот-мережу.

Слід зауважити, що у рівнянні 4 було усунуто залежність від $C_{i,j}$, оскільки той факт, що два боти мають одного і того ж бот-майстра неявно припускає $C_{i,j}$. Для подальшого спрощення рівняння, показаного раніше, зауважимо, що $A_i^{(t)}$ не залежить від $A_j^{(t)}$, оскільки дійсно виконувана ботом *i* активність залежить тільки від активності, обраної бот-майстром бота, та не залежить від того, що відбувається у бота *j*. Отже, можна переписати вищезгадане рівняння наступним чином:

$$p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid C_{i,j} = 1) = \sum_{b=0}^{K} p(A_i^{(t)} = a_i \mid B^{(t)} = b) \times$$
$$\times p(A_j^{(t)} = a_j \mid B^{(t)} = b) \times p(B^{(t)} = b) \quad (5)$$

В іншому випадку, коли $C_{i,j} = 0$ боти *i* та *j* мають два різних бот майстри $B_i^{(t)}$ та $B_j^{(t)}$. Аналогічно залежність $C_{i,j} = 0$ від може бути вилучена у формулі для спільної ймовірності

$p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid C_{i,j} = 0$ до тих пір, поки зберігається різниця між активністю $B_i^{(t)}$ та $B_j^{(t)}$ на двох вузлах. Таким чином, у випадку $C_{i,j} = 1$ $A_i^{(t)}$ не залежить від $A_j^{(t)}$, оскільки активність, що фактично здійснюється ботом *i*, залежить тільки від активності, обраної бот-майстром бота, і не залежить від того, що відбувається на бот *j*. Перепишемо формулу 3 для випадку $C_{i,j} = 0$:

$$\begin{aligned} & p(A_i^{(t)} = a_i, A_j^{(t)} = a_j \mid C_{i,j} = 0) = \\ & = \sum_{b_i=0}^{K} p(A_i^{(t)} = a_i \mid B_i^{(t)} = b_i)\, p(B_i^{(t)} = b_i) \times \\ & \times \sum_{b_j=0}^{K} p(A_j^{(t)} = a_j \mid B_j^{(t)} = b_j)\, p(B_j^{(t)} = b_j) \end{aligned} \quad (6)$$

Обидва рівняння (5) і (6), які демонструють визначення спільних розподілів за спостережуваними зловмисними діями в парі хостів, за умови, що вони належать до однієї і тієї ж бот-мережі, і за умови, що вони належать до двох різних бот-мереж, залежать від можливості оцінювати $p(A^{(t)} = a \mid B^{(t)} = b)$ – імовірність того, що спостережена шкідлива активність бота *a* в той час, як бот-майстер вибрав *b*.

Розбиття отриманого графу на підграфи, що відповідають окремим бот-мережам

Нехай задано орієнтований зважений граф $G(X,V,w)$ порядку *n,* де $X = \{x_1,...,x_n\}$ – множина вершин (хості в локальній мережі); $V \subseteq X \times X$ – множина ребер; $w: V \to R^+$ – відображення, що визначає вагу кожного ребра, де R^+ – множина дійсних невід'ємних чисел.

Необхідно визначити розбиття множини вершин X графа $G(X,V,w)$ на k – підмножин $(X_1,...,X_k)$ таким чином, щоб для частин графа $G_1(X_1,V_1,w_1),\ldots, G_k(X_k,V_k,w_k)$ виконувались наступні вимоги:

$$\begin{gathered} X_i \cap X_j = \varnothing, \text{ для } \forall\ i \neq j, \text{ де } i,j = \overline{1,k}; \\ \bigcup_{i=1}^{k} X_i = X; \\ |X_1| = n_1, \ldots, |X_k| = n_k,\ n_1 + \ldots + n_k = n, \end{gathered} \quad (7)$$

Перерізом розбиття $C(X_1,\ldots,X_k)$ будемо називати сукупність ребер, що сполучають вершини, які належать різним підграфам.

В якості критерію оптимальності *Q,* що визначає ефективність бі-розбиття $(X_1,\ldots,X_k)$ будемо розглядати вагу розрізу – сума вагів всіх ребер перерізу:

$$Q(X_1,X_2,\ldots,X_k) = \frac{1}{2} \sum_{L=1}^{k-1} \sum_{i \in V_L} \sum_{j \notin V_L} w(x_i,x_j) \to \min \quad (8)$$

В даному випадку оптимальним *k*-розбиттям є рішення $(X_1^*,\ldots,X_k^*)$ екстремальної задачі (8), тобто розбиття $(X_1^*,\ldots,X_k^*)$ з мінімальними вагами перерізу $C(X_1^*,\ldots,X_k^*)$.

Система вимог (7), що ставиться до розбиття $(X_1,\ldots,X_k)$, визначає область пошуку *D* задачі розбиття графа. Дана задача відноситься до задач переборного типу і загальна кількість допустимих рішень $|D|$ визначити із виразу:

$$\frac{n!}{n_1! \cdot n_2! \cdot \ldots \cdot n_k! \cdot t!} \quad (9)$$

де *t* – загальна кількість під графів (ботів бот-мереж, що присутні в локальній мережі). Суть алгоритму полягає в обчисленні максимального виграшу, який може принести перенесення будь-якої вершини в той чи інший підграф розбиття і здійсненні цього перенесення.

Визначимо декомпозиційне обмеження так, щоб ваги кожного підграфа розбиття визначались наступним чином:

$$w(V_i) = \sum_{v_q \in V_i} w(v_q), \quad (10)$$

де $w(v_q)$ – ваги вершини v_q, обмежуються наступними границями:

$$L_i \le w(V_i) \le U_i, i = \overline{1,k}, \quad (11)$$

де L_i та U_i – мінімальна та максимальна вага вершин підграфів розбиття та визначають межі, в яких можуть варіюватися ваги підграфів розбиття.

Тоді з урахування декомпозиційного обмеження наведемо кроки запропонованого алгоритму:

1) Ребра графа сортуються по зростанню вагів. Нехай $E=(e_1,\ldots e_m)$ – вектор відсортованих ребер;

2) i:=0; f:=0;

3) Із *k* підграфів розбиття вибирається той, при переміщенні в який вершин ребра e_i не порушуються

обмеження задачі декомпозиції, та досягається найбільше зменшення ваги перерізу;

4) Якщо на кроці 3 відбулось переміщення, то f:=1;

5) i:=i+1;

6) Якщо $i \leq m$, то перехід на крок 3;

7) Якщо f=1, то перехід на крок 2.

Слід зазначити, що на кроці 3 алгоритму обидві вершини ребра e_i виявляються в одному підграфі розбиття, тим самим ребро перестає брати участь в перерізі, якщо до цього брало. Назвемо цей процес локалізацією ребра. Таким чином, алгоритм локалізує спочатку самі «важкі» ребра, а потім, якщо це можливо, ребра меншої ваги. Евристика працює до тих пір, поки можлива локалізація хоча б одного ребра зі зменшенням ваги перетину, тобто до досягнення деякого локального екстремуму.

Таким чином, розроблений алгоритм розбиття орієнтованого зваженого графа, що представлений ботами, які присутні в локальній мережі, на підграфи дозволяє виділити фрагменти різних бот-мереж, що присутні в локальній мережі. Представлений алгоритм використовує "жадібний" підхід та ґрунтується на обчисленні максимального виграшу, який може принести перенесення будь-якої вершини в той чи інший підграф розбиття. Запропонований алгоритм є складовою частиною методу виявлення фрагментів бот-мереж на основі аналізу мережевого трафіку.

Експерименти

Для перевірки ефективності методу виявлення фрагментів бот-мереж на основі аналізу мережевого трафіку та моделювання виявлення хостів, що інфікували локальну мережу, було проведено ряд досліджень.

З метою моніторингу активності та визначення вагів ребер між інфікованими вузлами бот-мережі, перший експеримент передбачав використання локальної мережі, що складалась з 15 хостів.

Для представленої мережі було згенеровано дві IRC бот-мережі, кожна з яких складалась з трьох хостів. Хости, промарковані від 1 до 3 складали бот-мережу 1, а значення від 4 до 6 відповідали хостам бот-мережі 2. Усі інфіковані комп'ютери брали участь у регулярному інтернет-спілкуванні, що передбачає P2P-завантаження, HTTP веб-комунікацію, використання протоколу передачі файлів, Telnet тощо. При генеруванні бо мереж існувало значне перекриття часу, коли обидві бот-мережі були задіяні в одній і тій же шкідливій діяльності. Окрім того, з метою підвищення ступеню адекватності моделювання, згенеровані бот-мережі характеризувались відсутністю унікальних шкідливих дій, які можна було б використовувати для сегментації окремих бот-мереж.

Процес моніторингу мережевого трафіку тривав 24 години, протягом якого здійснювалось виявлення шкідливої активності. Сканування портів та інших атак, що виконуються шкідливими хостами в мережі, були відстеженні за допомогою системи попередження вторгнень Snort. Для визначення типу активності бота було залучено правила Intrusion Detection System Snort. Набір правил, що визначать різні активності ботів наведено в таблиці 1.

Таблиця 1

Залучені правила для IDS Snort

№ п/п	Тип активності
	Сканування
1	alert tcp $EXTERNAL_NET 10101 -> $HOME_NET any (msg:"SCAN myscan"; flow:stateless; ack:0; flags:S; ttl:>220; reference:arachnids,439; classtype:attempted-recon; sid:613; rev:6;)
2	alert tcp $EXTERNAL_NET 10101 -> $HOME_NET any (msg:"SCAN myscan"; flow:stateless; ack:0; flags:S; ttl:>220; reference:arachnids,439; classtype:attempted-recon; sid:613; rev:6;)
3	alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"SCAN cybercop os probe"; flow:stateless; dsize:0; flags:SF12; reference:arachnids,146; classtype:attempted-recon; sid:619; rev:6;)
4	alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN FIN"; flow:stateless; flags:F,12; reference:arachnids,27; classtype:attempted-recon; sid:621; rev:7;)
	Спам
1	alert tcp $EXTERNAL_NET any -> $SMTP_SERVERS 25 (msg:"POLICY-SPAM 1.usa.gov URL in email, possible spam redirect"; flow:to_server, established; file_data; content:"http|3A 2F 2F|1.usa.gov"; pcre:"/http\x3A\x2f\x2f1\.usa\.gov\x2f[a-f0-9]{6,8}/smi"; metadata:ruleset community, service smtp; reference:url,www.symantec.com/connect/blogs/spam-gov-urls; classtype:bad-unknown; sid:24598; rev:3;)
2	alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"POLICY-SPAM local user attempted to fill out paypal phishing form"; flow:to_server,established; content:"POST"; http_method; content:"/logindo.php"; fast_pattern:only; http_uri; content:"partner="; nocase; http_client_body; content:"&login="; distance:0; nocase; http_client_body; content:"&user="; distance:0; nocase; http_client_body; content:"&pass="; distance:0; nocase; http_client_body; content:"&submit="; distance:0; nocase; http_client_body; metadata:service http; classtype:suspicious-login; sid:21637; rev:4;)

Продовження табл. 1

№ п/п	Тип активності
3	alert tcp $EXTERNAL_NET any -> $SMTP_SERVERS 25 (msg:"POLICY-SPAM appledownload.com known spam email attempt"; flow:to_server, established; content:"appledownload.com"; nocase; metadata:service smtp; reference:url,www.tuaw.com/2011/05/18/new-phishing-email-pretends-to-be-from-apples-online-store/; classtype:policy-violation; sid:19122; rev:5;)
	Атака
1	alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDOS TFN Probe"; icmp_id:678; itype:8; content:"1234"; reference:arachnids,443; classtype:attempted-recon; sid:221; rev:4;)
2	alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDOS tfn2k icmp possible communication"; icmp_id:0; itype:0; content:"AAAAAAAAAA"; reference:arachnids,425; classtype:attempted-dos; sid:222; rev:2;)
3	alert udp $EXTERNAL_NET any -> $HOME_NET 31335 (msg:"DDOS Trin00 Daemon to Master PONG message detected"; content:"PONG"; reference:arachnids,187; classtype:attempted-recon; sid:223; rev:3;)
4	alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDOS TFN client command BE"; icmp_id:456; icmp_seq:0; itype:0; reference:arachnids,184; classtype:attempted-dos; sid:228; rev:3;)
	Контроль
1	alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:3;)
2	alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP icmpenum v1.1.1"; dsize:0; icmp_id:666 ; icmp_seq:0; id:666; itype:8; reference:arachnids,450; classtype:attempted-recon; sid:471; rev:3;)

Протягом усього періоду моніторингу здійснювалось оновлення вагів ребер між хостами в мережі з використанням рівняння (3). Отримані ваги ребер між вузлами (що відповідають хостам від 1 до 6 в мережі) наведені в таблиці 2.

Згідно із даними із таблиці 2, ваги ребер між вузлами, що належать до однієї бот-мережі, є високими, тоді як крайні ваги між вузлами, що належать різним бот-мережам, є низькими. Наприклад, ваги ребер між вузлами 1 і 3 (які належать до однієї бот-мережі) становили 0,94, тоді як вага ребра між вузлами 1 і 4 (які належать до різних бот-мереж) становили 7×10^{3}.

Таблиця 2

Ваги ребер між інфікованими вузлами мережі після закінчення часу моніторингу

Хост мережі	1	2	3	4	5	6
1	1.000	0.9202	0.9472	0.007	0.0007	0.0002
2	0.9202	1.000	0.968	0.0013	0.0001	0.0003
3	0.9472	0.968	1.000	0.0823	0.0006	0.0003
4	0.007	0.0013	0.0823	1.000	0.9714	0.9575
5	0.0007	0.0001	0.0006	0.9714	1.000	0.9581
6	0.0002	0.0003	0.0003	0.9575	0.9581	1.000

Інформація із представленої матриці вагів між інфікованими хостами мережі, була використана для ідентифікації бот-мереж у мережі. Таблиця 3 демонструє результати проведеного експерименту за визначенням ботів двох бот-мереж.

Таблиця 3

Результати проведеного експерименту по визначенню ботів двох бот-мереж

Кількість повторень експерименту	Середній час виконання методу (етап розбиття)	Коефіцієнт ваги перерізу	Кількість вірного розподілу ботів бот-мереж 1 та 2
5	0,24	0,4	5
5	0,27	0,5	4
5	0,24	0,6	1

Результати проведеного експерименту свідчать, що запропонований метод дозволив здійснити вірне визначення як ботів у кожній бот-мережі, так і кількість бот-мереж, що присутні у локальній мережі.

Висновок

Запропоновано метод виявлення фрагментів бот-мереж на основі аналізу мережевого трафіку. Метод заснований на представлені шкідливої активності, що здійснюють боти в локальній мережі у вигляді зваженого орієнтованого графу, де вершинами виступають хости мережі, а ребрами – зв'язки між хостами.

Зв'язність графу забезпечується наявністю ребер, які мають ваги. Вагою ребра, що з'єднує два вузли, є імовірність того, що два вузли є частиною однієї бот-мережі. Для визначення імовірності того, що

два вузли є частиною однієї бот-мережі використовується правило Байєса. Оновлення вагів ребер відбувається після кожного інтервалу часу в межах загального часу моніторингу шкідливої активності. Наприкінці часу моніторингу здійснюється розбиття отриманого графу на підграфи, що відповідають окремим бот-мережам.

Для розбиття графу на підграфи розроблений алгоритм, що дозволяє виділити фрагменти різних бот-мереж, які присутні в локальній мережі. Представлений алгоритм використовує "жадібний" підхід та ґрунтується на обчисленні максимального виграшу, який може принести перенесення будь-якої вершини в той чи інший підграф розбиття. Для перевірки ефективності запропонованого методу проведено ряд експериментів, що включали визначення факту наявності бот-мереж у локальній комп'ютерній мережі.

References

1. Kapre A., Padmavathi B. Behaviour based botnet detection with traffic analysis and flow interavals using PSO and SVM. International Conference on Intelligent Computing and Control Systems: Proceedings (Madurai, India, 15-16 June 2017). Madurai, 2017. P. 718–722.

2. Jaikumar P., Kak A.C.A graph-theoretic framework for isolating botnets in a network. Security and Communication Networks. 2012. Vol. 5. No. 6. P. 2605–2623.

3. Li S.H., Kao Y.C., Zhang Z.C., Chuang Y.P., Yen D.C. A Network Behavior-Based Botnet Detection Mechanism Using PSO and K-means. ACM Transactions on Management Information Systems. 2015. Vol. 6. Issue 1. P. 3–12.

4. Stevanovic M., Pedersen J. M. An analysis of network traffic classification for botnet detection. Cyber Situational Awareness, Data Analytics and Assessment: Proceedings (London, UK, June 8–9 2015). London, 2015. P. 1–8.

5. Chen S.C., Chen Y.R., Tzeng W.G. Effective Botnet Detection Through Neural Networks on Convolutional Features. The 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications. 12th IEEE International Conference On Big Data Science And Engineering: Proceedings (New York, NY, USA, 1–3 August 2018). New York, 2018. P. 372–378.

6. Thangapandiyan M., Anand P. M. An efficient botnet detection system for P2P botnet. International Conference on Wireless Communications, Signal Processing and Networking: Proceedings (Chennai, India, May 23-25). Chennai,2016. P. 1217–1221.

7. Zhang C., Green R. Communication security in internet of thing:preventive measure and avoid ddos attack over iot network. Proceedings of the 18th Symposium on Communications & Networking. Societyfor Computer Simulation International, 2015, P. 8–1

8. IDS Snort. URL: https://www.snort.org/

Рецензія/Peer review : 26.5.2020 р. Надрукована/Printed : 16.6.2020 р.
Стаття рецензована редакційною колегією