

УДК 621.395.74

МОНІТОРИНГ ДОСТУПНОСТІ ВЕБ-СЕРВІСУ В РОЗПОДІЛЕНИХ ІНФОКОМУНІКАЦІЙНИХ СИСТЕМАХ

DOI 10.36994/2707-4110-2020-1-28-12

Климаш М.М., д.т.н., проф., Національний університет «Львівська політехніка», Львів, Україна. mklimash@polynet.lviv.ua

Шпур О.М., к.т.н., Національний університет «Львівська політехніка», Львів, Україна. olha.m.shpur@lpnu.ua

Пелех Н.В., Національний університет «Львівська політехніка», Львів, Україна. van_plus_k@ukr.net

Анотація. У цій роботі була вдосконалено техніку захисту веб-сервісів від DDOS-атак на основі аналізу службової інформації, що базується на моніторингу тривалості сеансу, прописаного у файлах журналів. Це підвищує ефективність моніторингу безпеки веб-додатків у розподілених інформаційних системах після атаки першого етапу її захисту. Запропоновано алгоритм контролю стану системи на основі повідомлень журналу, які записуються у файл на етапі брандмауера. Кожне повідомлення отримує таку інформацію, як час запиту веб-служби та IP-адресу. Використовуючи час доступу до веб-сервісу, можна відстежувати відвідуваність, а у випадку різких відмінностей у кількості відвідувань у попередні інтервали, можна виявити можливу атаку. За допомогою IP, відстежуючи кількість запитів від кожного джерела запитів, можна з'ясувавши, хто викликає атаку. Дослідження ефективності розробленої системи моніторингу безпеки веб-додатків у розподілених інформаційних комунікаційних системах показало, що в той час, коли кількість відвідувань різко зростає, графік швидко зростає. Це вказує на можливу атаку.

Ключові слова: веб-сервіс; безпека; Ddos атаки; розподілені інформаційні системи.

WEB SERVICE AVAILABILITY MONITORING IN DISTRIBUTED INFOCOMMUNICATION SYSTEMS

Klymash M. M., Dr. habil., Prof., Lviv Polytechnic National University, Lviv, Ukraine. mklimash@polynet.lviv.ua

Shpur O. M., Ph. D., Lviv Polytechnic National University, Lviv, Ukraine. olha.m.shpur@lpnu.ua

Peleh N. V., Lviv Polytechnic National University, Lviv, Ukraine. van_plus_k@ukr.net

Abstract. In this paper has been improved technique of security of web services against DDOS attacks based on the analysis of service information, which is based on

monitoring of session duration spelled out in log files. This will increase the effectiveness of monitoring the security of web applications in distributed information systems after the attack of the first stage of its protection. The algorithm for monitoring the status of system based on log messages that are written to a file during the firewall stage has been proposed. Each message receives information such as web service request time and ip address. Using the time of access to the web service, you can track the attendance, and in the case of sharp differences with the number of visits in the previous intervals, to say about a possible attack. IP addresses us by tracking the number of requests from each request source, tracking who is causing the attack. Investigation of the effectiveness developed web application security monitoring system in distributed information communications systems has shown that at a time when the number of visits is increasing dramatically, the schedule is growing rapidly. This indicates a possible attack.

If the current number of visits increases 10 times more than the average per day, the system will notify you. To do this, the data on visits for the year were uploaded to the database and the system was checked when the load increased. From the test results, we were able to verify that as the number of visits for the current hour increased, the system reported an attack. To visually check the operation of the algorithm, a graphical interface was proposed, which shows the state of the system, and a schedule of visits at 3 intervals: per day, per month, per year. Visual inspection allows you to quickly make sure that the conclusion about the attack is not wrong

Keywords: *web service; security; Ddos attacks; distributed information systems.*

Вступ

Сьогодні, весь світ все більше переносить різні аспекти нашого життя у мережу Інтернет. До таких аспектів можуть належати як соціальні мережі, інтернет-магазини, різноманітні сервіси, так і більш критичні для суспільства аспекти - наприклад, банкінг. Такий процес приносить дуже багато користі та є зручним у використанні, однак, з іншого боку піддає загрозам приватне життя звичайних людей. Для компаній такі загрози можуть коштувати великих грошей. Тому дуже важливо знати, які загрози небезпечні для веб-сервісів, та як їх попередити або вчасно зреагувати на них [1]. Для цього необхідні певні системи моніторингу стану кожної системи, яка буде аналізувати можливі проблеми, що можуть виникати під час роботи, і запускати системи захисту в самому сервісі або повідомляти людей про можливу загрозу.

Такі системи повинні в дуже короткий час відслідковувати всі зміни, які відбуваються із системою - як різноманітні ін'єкції, так і атаки, при яких сервіс перестає нормально функціонувати. Одним із можливих варіантів розвитку аналізатора може бути самодіагностика сервісу, коли аналізатор сам перевіряє підключені системи на діри в системі захисту, що дозволяє повідомляти власникам про проблему. Це, в свою чергу, дасть змогу вирішити її, поки вона не завдала не тільки фінансової шкоди компанії, а і призвела до можливих втрат конфіденційних даних користувачів.

Тому метою цієї роботи - дослідження методів моніторингу доступності веб застосувань у розподілених інфокомунікаційних системах на основі аналізу загроз, які можуть впливати на систему, та знаходження способів вирішення проблем захисту від Ddos атак.

Аналіз особливостей безпеки веб сервісів

Сучасні веб системи повинні протидіяти різним загрозам зі сторони зловмисників. Але для цього необхідно знати які типи загроз можуть спричинити технічні проблеми на стороні серверу, де компанія буде втрачати кошти кожну секунду, чи більш поганий варіант – доступ до приватних даних користувачів. Або зміни в базах даних, що може призвести до великих фінансових втрат компаній. Тож, є кілька типів загроз веб сервісу: ін'єкції, слабка авторизація, Dos/Ddos атаки, Man-in-the-Middle атака (MitM)

Ін'єкційні атаки та їх реалізація. Ін'єкційні атаки відносяться до широкого класу векторів атак, які дозволяють зловмиснику подавати ненадійний вхід до програми, яка обробляється інтерпретатором як частина команди або запиту, що змінює хід виконання цієї програми. Ін'єкційні атаки є одними з найдавніших та найнебезпечніших атак веб-додатків. Вони можуть спричинити крадіжку даних, втрату даних, втрату цілісності даних, відмову в обслуговуванні, а також повний системний компроміс.

Ін'єкція є головною проблемою веб-безпеки. Він занесений як ризик безпеки веб-додатків номер один у Top-10 OWASP. Ін'єкційні атаки, зокрема інжекція SQL (SQLi) та міжсайтовий сценарій (XSS), є не лише дуже небезпечними, але й дуже широко поширені, особливо у застарілих програмах. Те, що робить ін'єкційні атаки особливо небезпечними, це те, що вони можуть буди виконані в дуже великій кількості сценаріїв (особливо для SQLi та XSS).

DDOS-атаки. DDOS-атаки - розподілені атаки, спрямовані на відмову в обслуговуванні, продовжують залишатися однією з найважливіших загроз в мережі. Атаки такого типу можуть швидко виснажити мережеві ресурси або потужності сервера, що призведе до неможливості отримати доступ до ресурсу і викличе серію негативних наслідків: втрачений прибуток, неможливість скористатися послугами і зробити різні транзакції і т.д. На даний момент не існує якогось універсального засобу для протидії DDOS-атакам. Для протидії розподіленим атакам, спрямованим на відмову в обслуговуванні, слід дотримуватися двох основних завдань:

1. Діагностувати DDOS-атаку на самих ранніх стадіях. Чим раніше буде виявлена DDOS-атака, тим раніше зможе включитися в гру мережевий адміністратор і тим раніше можна буде почати проводити анти DDOS-захід. Крім того, при виявленні DDOS-атаки можна буде, не чекаючи реагування адміністратора, автоматично запустити заходи з протидії: задіяти резервні канали зв'язку, включити фільтри і т.д.

2. Друге завдання пов'язана з поділом загального потоку трафіку на шкідливий і звичайний. Зрозумівши, які з клієнтських запитів є результатом DDOS-атаки, можна буде створити відповідні правила для брандмауера або ACL правила для маршрутизатора або ж, у разі масштабної атаки, передати ці дані на вищі маршрутизатори.

Методи виявлення Ddos атак. HADEC – це структура для виявлення високошвидкісної DDoS-атаки, що відбувається на мережесхемних і прикладних рівнях, таких як TCP-SYN, HTTP GET, UDP та ICMP. Рамка складається з двох основних компонентів: сервера виявлення та сервера захоплення. Виявлення DDoS в реальному часі починається з захоплення сервера, який відповідає за захоплення живого мережевого трафіку та передачу журналу на сервер виявлення для обробки. Виявлення обчислює вхідний пакет для UDP, ICMP та HTTP для виявлення атаки, якщо кількість з'єднань джерела перевищує заданий поріг. На основі такого удосконалення такого методу проведено дослідження науковцями [2, 3, 13]. Однак запропоновані виявлення забезпечують високу затратність як мережесхемних так і фінансових ресурсів.

D-FACE - для виявлення чотирьох типів трафіку: законного користувача, низькошвидкісного, високошвидкісного та швидкого трафіку. Для виявлення використовується різниця ентропії, яка містить нормальний потік трафіку, тоді як значення ентропії IP-джерела є матрицею виявлення для врахування атаки. Виявлення починається з вилучення пов'язаного заголовка, який класифікує мережу в унікальний мережесхемний потік. Розмежування трафіку подій низької швидкості, швидкості та миттєвій події базується на порівнянні поточної швидкості вхідного трафіку у кожному часовому вікні та на основі значення інформаційного трафіку.

Метод, який виявляє атаку HTTP DDoS за допомогою машинного навчання, щоб відрізнити ботнет від законних користувачів при виявленні трафіку атаки, справжнього трафіку та флеш трафіку. Запропонована система розміщується як проксі-сервер і здійснює перевірку поведінки користувачів замість моніторингу всього трафіку. Пропонована робота виявляє джерело ботнету та вивчає поведінку користувачів, щоб виявити шкідливий запит проти веб-сервера. На основі такого удосконалення такого методу проведено дослідження науковцями [4-6].

Хмарне виявлення HTTP DDoS за допомогою статистичного підходу з коваріаційною матрицею. Виявлення запровадило два алгоритми, відомі як тренінг та тестування для розпізнавання різного типу атаки потоку HTTP на основі поведінки атаки. Алгоритм навчання був використаний для побудови нормальних моделей мережевого трафіку, а алгоритм тестування використовувався для визначення типів отриманого трафіку. Результати, отримані в [7, 8, 11-12, 14], були оцінені за допомогою матриці плутанини для вимірювання ефективності виявлення та надання результатів внутрішнього та зовнішнього. Однак проведене дослідження не дозволяє враховувати різноманітність трафіку і працює лише на класичних його моделях.

MLP-GA алгоритм. Його дослідження наведені у [9]. Пропоноване виявлення використовувало чотири параметри для генерування виявлення на прикладних рівнях. Метод виявлення підраховує кількість HTTP GET-запитів, отриманих веб-сервером, і обчислює кількість IP-адрес, орієнтованих на сервер протягом 20 секунд. Пропоноване виявлення також перевіряє номер порту, який використовується HTTP DDoS, оскільки порти, які

використовуються зломисниками HTTP DDoS, змінюються і залишаються відкритими.

Методика виявлення на основі завантаженості вихідного вузла [10]. Ця методика використовувала кілька рівнів для захисту веб-сервера. Перший шар дозволяє або відхиляє отримане з'єднання, перевіривши вихідну IP-адресу з білим списком. Зареєстрованим IP-адресам було дозволено встановити з'єднання з веб-серверами для отримання сервісу, тоді як підключення незареєстрованих IP-адрес переривалося. Дозволені IP-адреси були перевірені, і якщо вони поводитимуть зломисні дії – з'єднання буде перервано, а IP-адреси додаються у чорний список.

Реалізація захисту від DDOS атак на основі аналізу службової інформації

На основі аналізу типів загроз, які найчастіше впливають на роботу веб сервісів пропонуємо свій варіант пошуку рішення щодо захисту системи на основі аналізу службової інформації.

Оскільки більшість систем захисту функціонують на рівні ядра мережі і частина вхідних запитів перенаправляється у Black Hole, єдиним можливим варіантом підвищення ефективності захисту системи буде логування всіх запитів на етапі проходження етапу фаєрволу вхідними пакетами. Тоді розпарсивши дані логу, можна дізнатися всю потрібну інформацію. Майбутній аналізатор загроз буде побудовано після фаєрволів.

Оскільки постійне зчитування з файлу буде доволі затратним в плані ресурсів, було прийнято рішення переносити логи до бази даних. Це дозволить пришвидшити пошук та фільтрування «потрібних» логів. Найбільш ідеальним варіантом було би використання так званої in-методу бази даних. Такі бази даних тримають дані в операційній пам'яті, що дозволяє швидко виконувати операції запису і пошуку.

На рис. 1 наведені спрощені кроки реалізації запису/зчитування логу до бази даних.

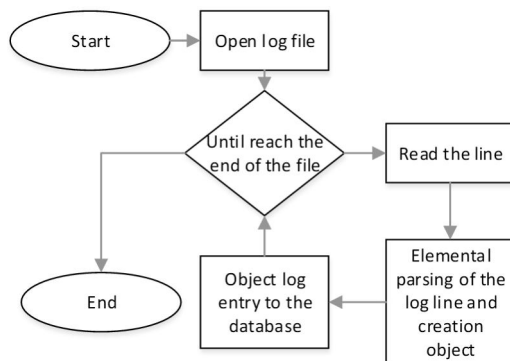


Рис. 1. Спрощені кроки переносу логів до бази даних

Для спрощення, за приклад був взятий лог по типу:

`'ip_address'--['yyy'/'MM'/'dd':'HH':'mm':'ss']`

(1)

Такий тип логу дозволяє сконцентруватися виключно на потрібній інформації. Але за потреби лог можна розширити. В якості парсера був вибраний Antlr. Він дає змогу швидко змінювати лексер дуже швидко. Типовий парсер є комбінацією лексера і парсера. Для початку необхідно створити граматичну. Для логу типу (1), необхідно виділити ір адресу, а також дату і час. Все інше відкидаємо. Для цього був створений граматичний словник для парсера (рис. 2).

```
grammar RQL;
    leg: query;
    query: ip '-' - '['daceAndTime'] ' WORD*;
    ip: NUMBER* '.' NUMBER* '.' NUMBER* '.' NUMBER*
        daceAndT ime:dace ':'time;
    dace: NUMBER '/' NUMBER '/' NUMBER;
    cime: hour ':' minuce ':' second;
    hour: HNUMBER;
    minuce: NUMBER;
    second: NUMBER;
    NUMBER : [0-9]+;
    WORD : [a-zA-Z_]+;
    WHITESPACE : (' '|'\t'|'.')+ -> skip;
```

Рис.2. Граматичний словник для парсеру

В такому вигляді граматичного словника було окремо виділено ір адресу. Більш складним є розбирання дати і часу. Оскільки може бути необхідно брати лише час або лише дату, було вирішено виділити їх в окремі лексеми, а потім об'єднати їх разом. Після цього створюється клас, який розбирає вхідний текст на лексеми. Створюється синтаксичний аналізатор. Далі вхідний текст розбирається, аналізується і виділяються необхідні елементи на етапі візиту. Це клас, який ходить по дереві лексери і дозволяє обробляти лексеми на вході чи на виході.

Для логу '52.14.235.178 - - [2019/11/06:19:40:00]' розбір на лексичне дерево буде виглядати наступним чином:

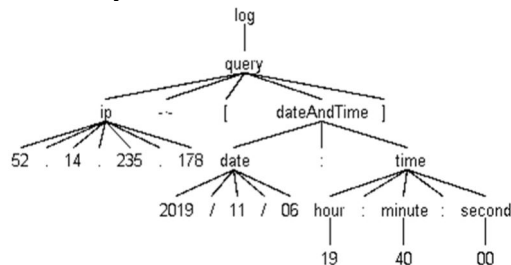


Рис. 3. Розклад логу на лексичне дерево

Наступним кроком після парсингу даних є створення об'єкту, в який поміщаються розпарсені дані. Після цього використовуючи Hibernate ми зберігаємо їх в базу даних для подальшої роботи з ними при аналізі.

Для розпізнавання Dos-атаки було розроблено алгоритм, який дозволяє виділити користувача серед інших запитів, який атакує сервіс. Для цього до бази даних відправляється запит, який знаходить всі звернення до веб-серверу в проміжку часу від 24 годин до запиту і до самого моменту цього ж запиту. Після цього для пришвидшення пошуку сортується список всіх запитів за ір-адресою. Далі в окремому модулі шукається максимальна кількість запитів для кожної окремої ір-адреси і отримане число повертається до головного методу, після чого серед усіх запитів, здійснених протягом 24 годин, знаходиться ір-адреса, з якої і було відправлено найбільшу кількість запитів. Для того, щоб перевірити, чи максимальна кількість запитів не відрізняється від кількості запитів інших користувачів, треба знайти медіану кількості запитів серед всіх інших користувачів, за виключенням числа максимальної кількості запитів. Надалі проводиться порівняння, чи максимальна кількість запитів буде більшою ніж медіана, помножена на певний коефіцієнт. Таким коефіцієнтом було обрано число 10, оскільки воно дозволяє відсікти сплески трафіку в пікові години навантаження від реальних Dos-атак, де з однієї ір-адреси надсилається велика кількість запитів. Оскільки це буде виділятися серед загального веб-трафіку, можна буде визначити ір-адресу, з якої ймовірно здійснюється Dos-атака.

Необхідно врахувати той факт, що атака відбувається з використанням великої кількості джерел трафіку. Для цього по світу заражається велика кількість 'стандартних' пристроїв, які після зараження продовжують і далі функціонувати належним чином, допоки зловмисник не задіє внесений програмний код. В результаті це приносить фінансові втрати та непрацюючі сервіси, що в свою чергу створює незручності для користувачів і репутаційні втрати. Тож оскільки запити надсилаються з великої кількості ір-адрес, під час пошуку ми можемо виключити це поле з рахунку і користуватися лише часом звертання до веб-сервісу. Загальну схему роботи даного алгоритму наведено на рисунку 5.

Результати моніторингу безпеки веб застосувань за допомогою логів

Для дослідження ефективності запропонованого алгоритму виявлення Ddos атак на веб-сервісі було проведено імітаційне моделювання з використанням мови програмування Java. Оскільки виявлення DDos атак вимагає дані логів з файлу, для цього згенеровано дані за нормальним законом розподілу. Перед початком роботи веб-серверу генеруються випадкові дані, які і записуються у лог файл. З цієї причини дані у файлі не будуть послідовними в часі, але це вирішується сортуванням всіх логів при записі в базу даних.

В файл eventLogs.txt, вноситься 150000 записів, для яких ір-адреса і час генеруються за законом розподілу Гауса. У випадку помилки вона прологується.

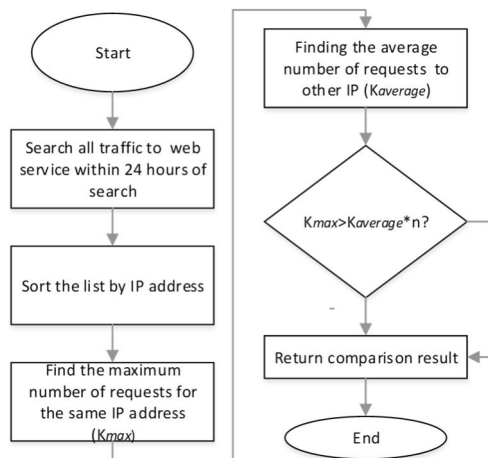


Рис.4. Загальна блок-схема алгоритму пошуку Dos-атаки на веб-сервіс
Після цієї процедури, вміст файлу буде виглядати наступним чином:

```

234.93.206.83 - - [2019/01/12:22:59:53]
180.201.249.172 - - [2019/10/11:14:40:53]
237.111.62.215 - - [2019/06/07:10:46:53]
241.160.173.222 - - [2019/11/09:14:53:53]
2.167.107.30 - - [2019/02/01:16:03:53]
102.167.1.158 - - [2019/04/23:05:44:53]
196.202.29.150 - - [2019/02/23:19:56:53]
234.69.110.139 - - [2019/09/02:12:36:53]
120.39.191.219 - - [2019/09/06:00:10:53]
151.105.100.161 - - [2019/03/24:12:37:53]
67.200.48.93 - - [2019/07/11:16:32:53]
139.92.117.146 - - [2019/06/05:17:24:53]
170.170.14.119 - - [2019/08/31:00:23:53]
74.184.195.154 - - [2019/07/14:05:14:53]
    
```

Рис. 6. Приклад записів згенерованих логів у файлі

Для деякого спрощення задачі, при виявленні джерела атаки, MAC-адреса не бралася до уваги. Для логування повідомлень в якості аналізатора було обрано Logback, що надає єдиний інтерфейс для роботи з іншими логерами. Єдине, що потрібно, – налаштувати все під конкретну реалізацію. Дана обгортка дозволяє зручно налаштувати різноманітні варіанти запису логів у різні напрями. Можна виводити певні рівні логів в консоль, паралельно записуючи їх до файлу. При цьому запис до файлу можна налаштувати таким чином, щоб розбивати логи по різних файлах, а коли логи застаріють, вони будуть автоматично перезаписані.

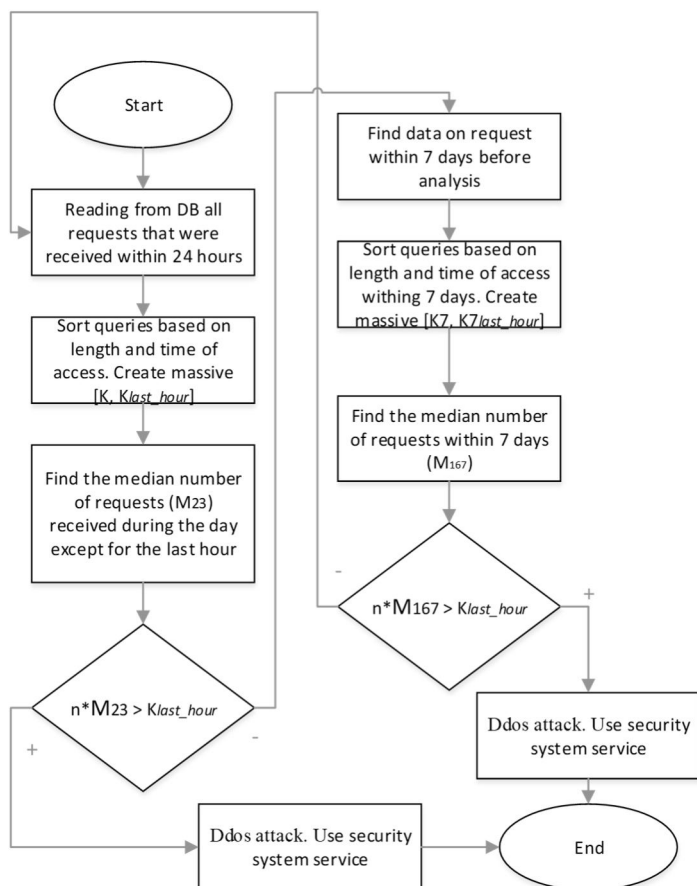


Рис. 5. Блок-схема роботи алгоритму, який розпізнає Ddos атаку, на веб-сервісі

Для Ddos атаки, логування може ділитись на 2 етапи. 1 етап відповідає за перевірку навантаження порівняно с середнім за день. 2 етап - відповідає за перевірку кількості запитів у поточну годину і в середньому за 1 тиждень. Це дає змогу виділяти атаки, які відбуваються більше 1 години. Такий метод має місце, оскільки тримати під атакою веб-сервер дуже довгий час - затратний процес. На рис.7 наведено результат того, як відбувається перевірка системи на атаку за допомогою пропонованого методу, і як її результат - відповідь, що на даний момент атака не відбувається.

```

: if number of visits for current hour more than average number of visits for 7 day * 10: false
: number of visits for current number 0
: average number of visits for 7 day 222
: Ddos false
    
```

Рис. 7. Логування для Ddos атаки

Для перевірки правильності роботи, було змінено дані в базі даних, що дозволить зрозуміти, чи правильно працює аналізатор, та що зміниться при умовах, коли перевірка повинна дати позитивний результат, що буде свідчити

про атаку. Для цього буде проведено 2 симуляції. Перша симуляція буде відповідати за перевірку наявності атаки в поточну годину, при перевірці з запитами за один день. Друга перевірка буде відповідати ситуації, коли перша перевірка дала негативний результат, а тоді необхідно перевірити, чи не проходить атака великий проміжок часу. Тому буде перевірятися на даних за 1 тиждень часу.

Результат першої перевірки показав: при кількості запитів за поточну годину рівною 260 і середньою кількістю запитів рівною 25, проводиться атака на наш веб-сервер. На рис.8 видно, що логіка аналізатора виділила запити в базі даних і порахувавши їх дала відповідь, що відбувається атака, як і очікувалось.

```
number of visits for current hour 260
average number of visits for day 25
Number of visits for last hour more than average number of visits for a day
Ddos true
```

Рис. 8. Перший етап перевірки на атаку

При другій перевірці враховувалися дані за поточну годину і середнє значення запитів за тиждень. На рис.9 видно, що середня кількість запитів за поточний день не дуже відрізняється від максимальної, що дозволяє пройти перевірку з негативним результатом на поточний день і перейти до перевірки за тиждень.

```
number of visits for current hour 3000
average number of visits for day 2500
if number of visits for current hour more than average number of visits for 7 day * 10: true
number of visits for current number 3000
average number of visits for 7 day 221
Ddos true
```

Рис. 9. Другий етап перевірки на атаку

Не зважаючи на логування, перевірка статусу роботи системи та підрахунок навантаження в різні моменти часу можуть викликати незручності. Для цього було вирішено розробити і запускати разом із сервером окремий модуль, який буде відповідати за відображення стану ресурсу та навантаження на нього. В якості UI платформи було взято Vaadin, який дозволить швидко та без особливих незручностей створити користувацький інтерфейс, який зможе надати необхідну інформацію. Моніторинг додатка, збирання показників, поведінку трафіку або стану відповідної бази даних проводитиметься на основі Spring Actuator

Як можемо бачити з рис. 10 наявний один елемент, який відповідає нашому веб-сервісу. Для цього елемента відображається його назва, статус та можливі помилки. З правого боку розміщений графік навантаження на веб-сервіс. По замовчуванню встановлено зображення навантаження за 30 календарних днів. Але під таблицею є можливість перемикає періоди навантаження.



Рис.10. Візуалізація роботи системи моніторингу стану веб-сервісу

Для перевірки ефективності запропонованих рішень проведемо атаку на відповідний веб сервіс. З рис. 11 можна зробити висновок, що аналізатор трафіку, який працює на основі запропонованого нами алгоритму відслідкував момент, коли відбулася Ddos атака. На рис.11. ми можемо бачити, наскільки різко змінилося навантаження на систему, порівняно з попереднім днем. Це вже вказує на те, що така ситуація не є звичною і відбувалася атака. Проте ми можемо перевірити все більш детально.

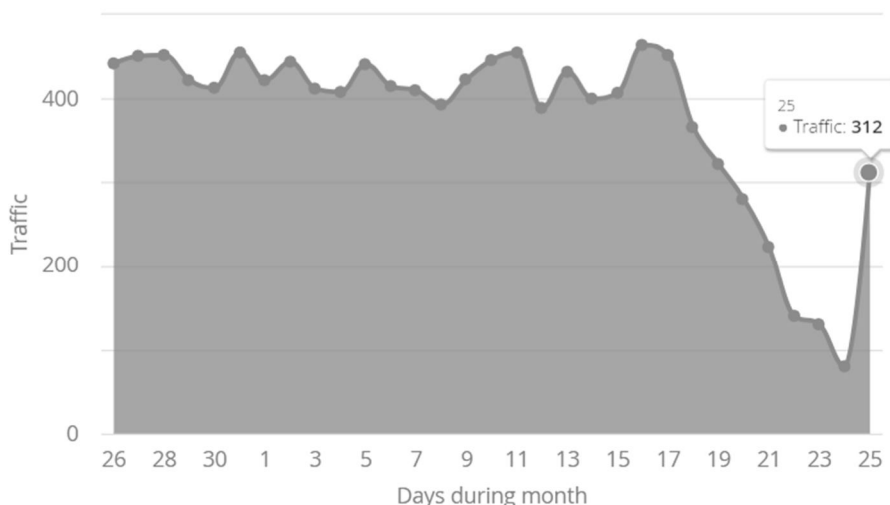


Рис.11. Результат роботи аналізатора трафіку

Для того, щоб зробити це, перейдемо на відображення навантаження погодинно. Результат наведений на рис.12. Бачимо, що в останню годину різко зросло навантаження на сервіс, хоча перед цим воно було настільки низьким, що його складно відрізнити від нуля.

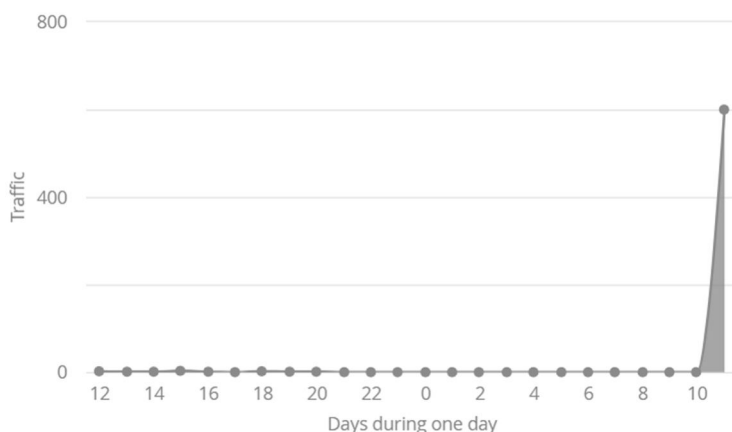


Рис.12. Погодинне відображення навантаження на веб-сервіс
У той же час, змінився статус в полі, яке відповідає за можливі помилки.

Application Name	Status	Issue
service anylizer	UP	DDos

Рис. 13.Поведінка таблиці при наявності атаки

Висновки

У роботі удосконалено методику захисту веб сервісів від DDOS атак на основі аналізу службової інформації, яка базується на аналізі тривалості сесії прописаної лог файлами, що дасть змогу підвищити ефективність системи моніторингу безпеки веб застосувань у розподілених інфокомунікаційних системах після проходження атаки першого етапу її захисту. Запропоновано алгоритм моніторингу стану системи на основі лог-повідомлень, які записуються в файл на етапі фаєрволу. З кожного повідомлення дістаються такі данні як час запиту на веб-сервіс, та ір-адреса. За допомогою часу звертання до веб-сервісу, можна відслідковувати відвідуваність, та у разі різких відмінностей з кількістю відвідувань в попередні проміжки часу, сказати про можливу атаку. IP-адреса дозволяє нам відслідковуючи кількість запитів з кожного джерела запитів, відслідковувати хто спричиняє атаку.

Дослідження ефективності роботи розробленої системи моніторингу безпеки веб застосувань у розподілених інфокомунікаційних системах довело, що в момент часу, коли кількість відвідувань різко збільшується, графік стрімко зростає. Це свідчить про можливу атаку. Якщо на поточний момент кількість відвідувань зростає в 10 разів більше ніж в середньому за день, система повідомить про це. Для цього данні про відвідування за рік, було завантажено в базу даних і відбувалась перевірка системи при підвищенні навантаження. З результатів тестування ми могли перевірити, що коли кількість відвідувань на поточну годину зростали, система повідомляла про атаку.

Для візуальної перевірки роботи алгоритму, було запропонований графічний інтерфейс, який показує стан системи, та графік відвідувань на 3-ох

проміжках: за день, за місяць, за рік. Візуальна перевірка дає змогу швидко пересвідчитись, що висновок про атаку не є хибним

Література

1. Klymash Mykhailo, Shpur Olga, Lavriv Orest, Peleh Nazar. Information security in virtualized data center network // Advanced information and communication technologies, AICT-2019 : proceedings of the 3rd International conference (Lviv, Ukraine, July 2–6 2019). – 2019. – С. 419–422
2. Hameed, S., Ali, U. HADEC: Hadoop-based live DDoS detection framework. EURASIP J. on Info. Security 2018, 11 (2018) doi:10.1186/s13635-018-0081-z
3. Hameed, S.; Ali, U. Efficacy of Live DDoS Detection with Hadoop. In Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS), Istanbul, Turkey, 25–29 April 2016.
4. Ashraf J, Latif S. Handling intrusion and DDoS attacks in software defined networks using machine learning techniques. Proceedings of National Software Engineering Conference, 2014; 55– 60.
5. Gabriel IM, Valeriu PV. Achieving DDoS resiliency in a software defined network by Intelligent Risk Assessment based on neural networks and danger theory. 5th IEEE International Symposium on Computational Intelligence and Informatics, 2014; 319– 324.
6. J. D. Ndibwile and A. Govardhan, "Web Server Protection against Application Layer DDoS Attacks using Machine Learning and Traffic Authentication," pp. 261–267, 2015.
7. A. Aborujilah and S. Musa, "Cloud-based DDoS HTTP attack detection using covariance matrix approach," Journal of Computer Networks and Communications, vol. 2017, Article ID 7674594, 8 pages, 2017.
8. Girma, A., Garuba, M., Li, J., Liu, C.: Analysis of DDoS attacks and an introduction of a hybrid statistical model to detect DDoS attacks on cloud computing environment. In: IEEE 12th International Conference on Information Technology-New Generations (ITNG), pp. 212–217 (2015)
9. Khundrakpam Johnson Singh, TanmayDe. MLP-GA based algorithm to detect application layer DDoS attack. Journal of Information Security and Applications, Volume 36, October 2017, Pages 145-153
10. Xiaohui Yang , Yue Yu DDoS Attacks Detection and Traceback Method Based on Flow Entropy Algorithm and MPLS Principle. International Conference on Cloud Computing and Security ICCCS 2018: Cloud Computing and Security, pp 670-683
11. A. Iswardani and I. Riadi, "Denial Of Service Log Analysis Using Density K-Means Method," vol. 83, no. 2, pp. 299–302, 2016.
12. T. A. Cahyanto and Y. Prayudi, "Web Server Logs Forensic Investigation to Find Attack's Digital Evidence Using Hidden Markov Models Method ," Snati, pp. 15–19, 2014.
13. B. Meng, W. Andi, X. Jian, and Z. Fucai, "DDOS attack detection system based on analysis of users' behaviors for application layer," in Proceedings of IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, July 2017.
14. C. E. Kayataş et al., "Statistical measures: Promising features for time series based DDoS attack detection," 2018 26th Signal Processing and Communications Applications Conference, Izmir, Turkey, 2018, pp. 1-4.

References

1. Klymash Mykhailo, Shpur Olga, Lavriv Orest, Peleh Nazar. Information security in virtualized data center network // Advanced information and communication technologies, AICT-2019 : proceedings of the 3rd International conference (Lviv, Ukraine, July 2–6 2019). – 2019. – С. 419–422
2. Hameed, S., Ali, U. HADEC: Hadoop-based live DDoS detection framework. EURASIP J. on Info. Security 2018, 11 (2018) doi:10.1186/s13635-018-0081-z

3. Hameed, S.; Ali, U. Efficacy of Live DDoS Detection with Hadoop. In Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS), Istanbul, Turkey, 25–29 April 2016.
4. Ashraf J, Latif S. Handling intrusion and DDoS attacks in software defined networks using machine learning techniques. Proceedings of National Software Engineering Conference, 2014; 55– 60.
5. Gabriel IM, Valeriu PV. Achieving DdoS resiliency in a software defined network by Intelligent Risk Assessment based on neural networks and danger theory. 5th IEEE International Symposium on Computational Intelligence and Informatics, 2014; 319– 324.
6. J. D. Ndibwile and A. Govardhan, "Web Server Protection against Application Layer DDoS Attacks using Machine Learning and Traffic Authentication," pp. 261–267, 2015.
7. A. Aborujilah and S. Musa, "Cloud-based DDoS HTTP attack detection using covariance matrix approach," Journal of Computer Networks and Communications, vol. 2017, Article ID 7674594, 8 pages, 2017.
8. Girma, A., Garuba, M., Li, J., Liu, C.: Analysis of DDoS attacks and an introduction of a hybrid statistical model to detect DDoS attacks on cloud computing environment. In: IEEE 12th International Conference on Information Technology-New Generations (ITNG), pp. 212–217 (2015)
9. Khundrakpam Johnson Singh, TanmayDe. MLP-GA based algorithm to detect application layer DDoS attack. Journal of Information Security and Applications, Volume 36, October 2017, Pages 145-153
10. Xiaohui Yang , Yue Yu DDoS Attacks Detection and Traceback Method Based on Flow Entropy Algorithm and MPLS Principle. International Conference on Cloud Computing and Security ICCCS 2018: Cloud Computing and Security, pp 670-683
11. A. Iswardani and I. Riadi, "Denial Of Service Log Analysis Using Density K-Means Method," vol. 83, no. 2, pp. 299–302, 2016.
12. T. A. Cahyanto and Y. Prayudi, "Web Server Logs Forensic Investigation to Find Attack's Digital Evidence Using Hidden Markov Models Method ," Snati, pp. 15–19, 2014.
13. B. Meng, W. Andi, X. Jian, and Z. Fucai, "DDOS attack detection system based on analysis of users' behaviors for application layer," in Proceedings of IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, July 2017.
14. C. E. Kayataş et al., "Statistical measures: Promising features for time series based DDoS attack detection," 2018 26th Signal Processing and Communications Applications Conference, Izmir, Turkey, 2018, pp. 1-4.