https://www.sworldjournal.com/index.php/swj/article/view/swj20-01-041
DOI: 10.30888/2663-5712.2023-20-01-041

УДК 004-049.5

ORGANIZATION OF PROTECTED ACCESS TO WEB SERVERS BY MEANS OF MACHINE LEARNING

ОРГАНІЗАЦІЯ ЗАХИЩЕНОГО ДОСТУПУ ДО WEB-СЕРВЕРІВ ЗАСОБАМИ МАШИННОГО НАВЧАННЯ

Korobeinikova T.I. / Коробейнікова Т.І.
c.t.s., as.prof. / к.т.н., доц.
ORCID: 0000-0003-2487-8742
Kravchuk N.V. / Кравчук Н.В.
аспірант / postgraduate
Lviv Polytechnic National University, S. Bandera St. 12, Lviv, 79013
Національний університет «Львівська політехніка», Львів, Бандери, 12, 79013

Анотація. *У статті досліджується зростаюча загроза безпеці в глобальному кіберпросторі. Запропоновано заходи з визначення ризиків доступу до інформації з урахуванням рівня відповідальності осіб. Особлива увага приділена захисту web-серверів від атак типу CSRF, розвитку методів підвищення рівня захисту web-серверів за допомогою машинного навчання для виявлення небезпечних запитів; розглянуто значення web-програм як інтерфейсу для захисту конфіденційних даних. Досліджено методи виявлення вразливостей через чорну скриньку, робота акцентує увагу на аналізі типової атаки CSRF. Проаналізовано сценарій атаки та її наслідки для користувачів та веб-серверів. Розглянуто такі методи захисту: токени та перевірку заголовків HTTP-запитів; також розглянуто автоматизоване запобігання за допомогою атрибуту cookie SameSite. Застосування машинного навчання для виявлення CSRF розглядається як можлива ефективна стратегія. Описана важливість збагачення інструментів виявлення CSRF семантичною інформацією для зниження помилкових результатів. Застосування контрольованого навчання та класифікаторів допомагає виявляти безпеку web-об'єктів, зокрема HTTP-запитів. Пропонується архітектура для виявлення вразливостей з використанням евристик та машинного навчання.*

Ключові слова: *Web-програми, захист конфіденційних даних, web-додатки, машинне навчання (ML), класифікатор, HTTP-запит, вразливості веб-додатків.*

Вступ.

Останнім часом глобальна кіберсфера усе більше розглядається світовим суспільством як один із найважливіших пріоритетів безпеки, оскільки його функціонування стає вагомим чинником у розвитку військового, соціального, економічного та інших секторів [1-3]. Стає очевидніша і зростаюча мілітаризація кіберпростору, а зусилля світових держав попередити цей процес, залишаються, на жаль, малоефективними. У зв'язку з ризиками, що зумовлені взаємодією людського персоналу з інформацією, пропонуються заходи, які можуть визначати ризики під час надавання доступу до інформації конкретним особам із урахуванням рівня відповідальності, що покладається на них разом із доступом до цієї інформації [5-6].

Web-програми є особливо складними для аналізу через їхню різноманітність і широке застосування нестандартних методів програмування. Таким чином, машинне навчання (machine learning, ML) є дуже корисним для безпеки web-додатків: він може використовувати дані, позначені вручну, щоб перенести людське розуміння семантики web-додатку в інструменти автоматизованого аналізу [7-10].

Одною з основних загроз для web-сервераа є несанкціонований доступ до інформаційних ресурсів та інформаційно-телекомунікаційних систем. Напад CSRF вводить користувача в оману взаємодії зі сторінкою або сценарієм на віддаленому ресурсі. Це спровокує шкідливий запит на сайт користувача. Але сервер припускає, що це запит від авторизованого web-сайту. Коли користувач введений в оману, зловмисник може взяти на себе контроль над використанням даних, надісланих у запиті 11-12].

Робота присвячена подальшому розвитку і дослідженню підвищення рівню захисту web-сервераів за рахунок використання машинного навчання для визначення небезпечних запитів.

1 Виявлення вразливого місця web-сервера.

Web-програми є найпоширенішим інтерфейсом для захисту конфіденційних даних і доступних функцій. Вони регулярно використовуються для подання податкових декларацій, доступу до результатів медичних оглядів, здійснення фінансових операцій і обміну думками тощо.

З іншого боку, це означає, що web-додатки є привабливими мішенями для зловмисників, які мають намір завдати економічних збитків, неправомірно отримати доступ до конфіденційних даних. Відомо, що захистити web-додатки важко [13]. Причини такі: неоднорідність та складність web-платформ, впровадження сумнівних мов сценаріїв, які не гарантують безпеку тощо.

У таких умовах особливо популярними є методи виявлення вразливостей чорної скриньки [14]. На відміну від методів білої скриньки, які вимагають доступу до вихідного коду web-програми, методи чорної скриньки працюють на рівні HTTP-трафіку, тобто HTTP-запитів і відповідей і пропонує підхід до виявлення вразливості без урахування мови, пропонує єдиний інтерфейс для найширшого можливого діапазону web-програм.

Це звучить привабливо, але попередні роботи показали, що такий аналіз далеко не тривіальний [15]. Однією з головних проблем є те, як надати автоматизованим інструментам компонент ефективного виявлення вразливостей, тобто розуміння семантики web-додатків.

1.1 Типова атака. Прикладом може стати міжсайтова підробка запитів (Cross-Site Request Forgery, CSRF). CSRF – web-атака, яка змушує користувача надсилати небажані HTTP-запити, контрольовані зловмисником, до вразливої web-програми, у якій він наразі пройшов автентифікацію. Ключова концепція CSRF полягає в тому, що зловмисні запити направляються до web-програми через браузер користувача, отже, їх можна не відрізнити від призначених легітимних запитів, які фактично авторизував користувач. Типова атака CSRF працює так (рис. 1):

1) Жертва входить у чесну, але вразливу web-програму, наприклад, у улюблену соціальну мережу. Автентифікація сеансу реалізується через файл cookie сеансу, який автоматично додається браузером до будь-якого наступного запиту до web-програми;

2) Жертва відкриває іншу вкладку та відвідує непов’язаний web-сайт, який перенаправляє на web-сторінку, що містить шкідливу рекламу;

3) Зловмисна реклама надсилає міжсайтовий запит до соціальної мережі за

допомогою HTML або JavaScript, наприклад, із проханням поставити «подобається» певній політичній партії. Оскільки запит містить файли cookie жертви, він обробляється в контексті її автентифікації в соціальній мережі. Таким чином, шкідлива реклама може змусити жертву поставити «подобається» бажаній політичній партії, що може спотворити результати онлайн-опитувань.

Рисунок 1 – Приклад Cross-site request forgery

Джерело [22]

Зауважте, що CSRF не вимагає від зловмисника перехоплювати або змінювати запити та відповіді користувача: достатньо, щоб жертва відвідала web-сайт зловмисника, з якого була розпочата атака. Таким чином, будь-який шкідливий web-сайт в Інтернеті може використовувати вразливості CSRF.

1.2 Запобігання CSRF. Щоб запобігти CSRF, web-розробники мають реалізувати явні механізми захисту [16]. Якщо додавання додаткової взаємодії з користувачем не надто впливає на зручність використання, можна примусово повторити автентифікацію або використати одноразові паролі чи captcha, щоб запобігти непоміченим міжсайтовим запитам.

Однак у багатьох випадках перевага надається автоматизованому запобіганню: нещодавно введений атрибут cookie SameSite можна використовувати для запобігання вкладенню файлів cookie під час міжсайтових запитів, що усуває основну причину CSRF і рекомендується для web-додатків. На жаль, часто web-додатки відфільтровують міжсайтовий запит за допомогою таких технік:

1) Перевірка значення стандартних заголовків HTTP-запиту, як Referrer і Origin, із зазначенням сторінки, з якої надходить запит;

2) Перевірка наявності користувацьких заголовків HTTP-запитів, як X-Requested-With, які не можна встановити з міжсайтової позиції;

3) Перевірка наявності непередбачуваних анти-CSRF токенів, встановлених сервером у конфіденційні форми.

Однак усі вони мають однакові обмеження: вимагають точного розміщення перевірок безпеки. Наприклад, маркери слід прикріплювати до чутливих HTTP-запитів, щоб забезпечити повний захист без шкоди для взаємодії з користувачем.

Використання токена для захисту кнопки «Подобається» корисне для запобігання атаці, про яку йдеться вище, однак мати токен на домашній сторінці соціальної мережі небажано, оскільки це може призвести до відхилення легітимних міжсайтових запитів, наприклад, клацання результатів пошукової системи, нихщо індексує соціальну мережу.

Зрештою, пошук «оптимального» розташування засобів захисту від CSRF зазвичай є складним завданням для web-розробників. Сучасні фреймворки забезпечують автоматизовану підтримку, але вразливості CSRF досі регулярно виявляються навіть на web-сайтах з найвищим рейтингом. Це формує потребу в ефективних інструментах виявлення CSRF. Вирішенням може бути застосування машинного навчання для автоматичної підтримки інструментів виявлення CSRF.

2 Застосування машинного навчання.

Приклад CSRF показує, що корисно збагачувати інструменти виявлення вразливостей семантичною інформацією, щоб мінімізувати кількість помилкових спрацьовувань і помилкових негативів. Принаймні, можна було б автоматично класифікувати запити HTTP як чутливі до безпеки. Однак це HTTP-запити мають відносно слабку синтаксичну структуру наприклад, є кілька способів реалізації кнопки «подобається», ідентифікованого унікальним рядком *3aa5bf*:

1) запит GET до сторінки like.php з одним параметром id = 3aa5bf;

2) запит GET до сторінки manage.php з параметром id = 3aa5bf і параметром action = like;

3) запит POST до сторінки manage.php, включаючи об’єкт JSON {id: 3aa5bf, action: upvote}.

Усі ці запити виглядають семантично схожими для досвідчених тестувальників безпеки, але вони синтаксично відрізняються, і може бути важко ідентифікувати всі найпоширеніші способи кодування тієї самої інформації в дикій природі.

2.1 Контрольоване навчання. Машинне навчання (ML) надає ефективні інструменти для автоматизації завдань класифікації. Класифікатор можна розглядати як функцію $f : X \rightarrow Y$, що відображає будь-який об’єкт із простору ознак X у відповідний клас із Y. Підполе навчання під керівництвом вивчає ефективні методи автоматичного генерування класифікаторів, починаючи з набору позначених даних. Таким чином, щоб використовувати контрольоване навчання, потрібно:

1) зібрати набір об’єктів O, це наприклад, HTTP-запити, надіслані репрезентативним web-додаткам;

2) визначити набір класів Y. Наприклад, можна встановити $Y = \{+1, -1\}$, щоб відрізнити чутливі до безпеки запити (+1) від усіх інших (-1);

3) визначте простір ознак X, вручну визначивши основні аспекти, які виглядають корисними для призначення об’єктів в O їх правильному класу в Y. Наприклад, можна використовувати довжину запиту, метод запиту або наявність вибраних ключових слів у тіло запиту;

4) побудувати навчальний набір D пар (~x, y), де кожен ~x є кодуванням в X об’єкта $o \in O$, а y — його клас.

Після цього контрольоване навчання може автоматично витягнути найефективніший класифікатор із набору можливих гіпотез H шляхом оцінки його ефективності на навчальному наборі D. Поки в D є достатньо підібраних вручну даних, ефективність керованого навчання може конкурувати з людьми-експертами або навіть перевершувати їх [17].

2.2 Виявлення web-вразливостей. Зрештою, запропоновану нами методологію можна описати архітектурою (на рис. 2):

1) Використовуйте контрольоване навчання для автоматичного навчання класифікатора, який розділяє вибрані цікаві web-об'єкти, наприклад, HTTP-запити, HTTP-відповіді або файли cookie, на основі web-семантика додатка. Наприклад, у випадку виявлення CSRF, класифікатор буде використовуватися для ідентифікації чутливих до безпеки запитів HTTP;

2) Для кожного можливого класу, повернутого класифікатором, визначте евристику для виявлення вразливості. Навіть тривіальна евристика, що позначає кожен об'єкт у даному класі як невразливий, є ймовірною. Наприклад, нечутливі запити не можуть бути використані для CSRF, тому їх можна відразу позначити як невразливі;

3) Використовуйте класифікатор, щоб вибрати відповідну евристику виявлення вразливості для кожного цікавого web-об'єкта, наприклад, як частину розширення браузера.

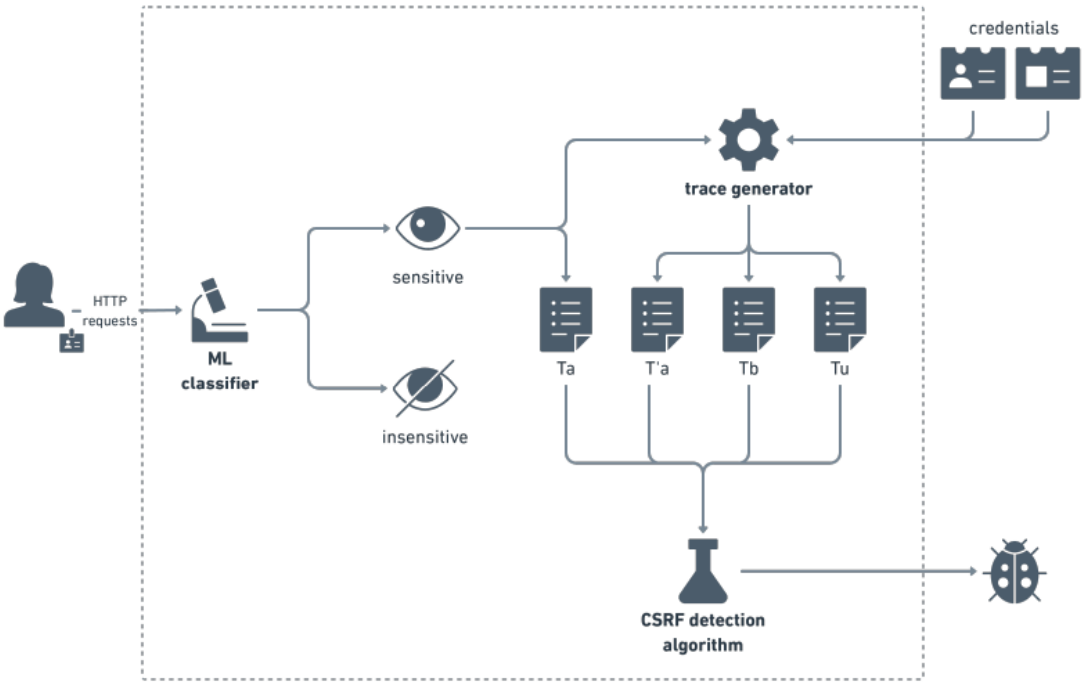

Рисунок 2 – Архітектура рішення

Авторська розробка

2.3 Класифікатор машинного навчання. Класифікатор ML, який можна використовувати, потрібно навчати на наборі даних близько 6000 HTTP-запитів із існуючих web-сайтів, зібраних і позначених двома експертами. Простір ознак X класифікатора має 49 вимірів, кожен з яких фіксує певну властивість HTTP-запитів. Їх можна розділити на 3 категорії: структурні, текстові та функціональні.

1) Структурні: ця категорія функцій описує структурні властивості запиту HTTP. Точніше, ми визначаємо наступний набір числових ознак: numOfParams: загальна кількість параметрів; numOfBools: кількість параметрів запиту, прив'язаних до логічного значення; numOfIds: кількість параметрів запиту, пов'язаних з ідентифікатором, тобто шістнадцятковим рядком, використання якого було емпірично спостережено як звичайне в нашому наборі даних; numOfBlobs: кількість параметрів запиту, прив'язаних до блобу, тобто будь-який рядок, який не є ідентифікатором; reqLen: загальна кількість символів у запиті, включаючи назви та значення параметрів. Хоча можна розробити більш складні методи «набору» параметрів запиту, запити HTTP мають дуже слабку структуру, і важко придумати загальні, але точні методи введення для них.

2) Текстові: Ця категорія функцій фіксує текстові характеристики HTTP-запитів і базується на невеликому вручну підготовленому словнику ключових слів V, які можуть зустрічатися в запиті, в результаті ручної перевірки конфіденційних запитів із вибірки реальних web-сайтів, розглянутих у наш набір даних. Більш конкретно, ми розглядаємо двійкові ознаки лише таких форм: wordInPath, де слово ∈ V означає наявність слова рядка в шляху запиту; wordInParams, де слово ∈ V означає наявність рядкового слова в будь-якому імені параметра запиту.

Словник V включає наступні 21 ключове слово, які були вибрані як можливі сигнали конфіденційних запитів відповідно до здорового глузду та попередньої перевірки частини нашого набору даних, яка зарезервована для навчання: створити, додати, встановити, видалити, оновити, видалити, друг, налаштування, пароль, маркер, змінити, дія, оплатити, увійти, вийти, опублікувати, прокоментувати, підписатися, підписати та переглянути.

3) Функціональні: ця категорія функцій вказує на метод HTTP, пов'язаний із запитом. Ми розглядаємо лише наступні дві двійкові функції: isGET: метод HTTP-запиту GET; isPOST: методом запиту HTTP є POST.

Немає додаткових альтернатив, оскільки наш набір даних включає лише запити GET і POST. Усі інші запити можна легко позначити як конфіденційні чи ні лише на основі їх методу, наприклад, запити OPTIONS завжди є нечутливими.

Висновки.

Web-програми є особливо складними для аналізу через їхню різноманітність і широке застосування нестандартних методів програмування. Таким чином, ML є дуже корисним у web-налаштуваннях, оскільки він може використовувати дані, позначені вручну, щоб надати людині розуміння семантики web-додатку автоматизованим інструментам аналізу. Комплексна система оцінки запитів до web-сервераа та оцінка ключових його параметрів дозволить підвищити ефективність знаходження шкідливих несанкціонованих запитів за рахунок ML класифікатора побудованого на базі знань про уразливості web-систем.

Література.

1. Aggarwal C.C., Charu C. Data Classification Algorithms and Applications. 2015: Chapman & Hall /CRC.

2. Chandola V., Banerjee A., Kumar V. Anomaly Detection for Discrete

Sequences: A Survey // IEEE Transactions on Knowledge and Data Engineering, No. 24(5), 2012. pp. 823–839.

3. Трояновська Т. І. Методи та засоби популяризації комерційних веб-ресурсів / Т. І. Трояновська, Л. А. Савицька, В. Ю. Тарануха // Інформаційні технології та комп'ютерна інженерія. – Вінниця, 2017. – №2, С. 23-30.

4. Захарченко С. М. Застосування односторінкових веб-орієнтованих інтерфейсів в соціально значущих проектах. / С. М. Захарченко, Т. І. Трояновська, О. В. Бойко В. С. Рибаченко // Вісник ХНУ, №3, 2016р., с. 33-39.

5. EM-алгоритм [Електронний ресурс] URL: https://uk.wikipedia.org/wiki/EM-алгоритм

6. Гороховський О. І. Модель формування автоматичних розкладів за алгоритмом Парето / О. І. Гороховський, Т. І. Трояновська, О. В. Бойко // Інформаційні технології та комп'ютерна інженерія – 2016. – №1, с. 4-12.

7. Гороховський О. І. Розробка формалізованого опису автоматизованої системи дистанційного навчання / О. І. Гороховський, Т. І. Трояновська, А. В. Снігур // Інформаційні технології та комп'ютерна інженерія – 2007. – № 2. – С. 192–198. – ISSN 1999–9941..

8. Manevitz L. M. Y.M. Document Classification on Neural Networks Using Only Positive Examples // SIGIR. 2000.

9. Markou M., Singh S. Novelty detection: A Review, Part 2: Neural Network-based Approaches // Signal Processing, No. 83(12), 2003. pp. 2481–2497.

10. Markou M..S.S. Novelty detection: A Review, Part 1: Statistical Approaches // Signal Processing, No. 83(12), 2003. pp. 2481–2497.

11. Peacock A., Ke X., Wilkerson M. Typing patterns: A key to user identification // IEEE Security and Privacy, Vol. 2, no.5, pp.40–47, Sep. 2004.

12. Коробейнікова Т.І. Відмовостійкість та автомасштабування веб-ресурсу. / Коробейнікова Т.І., Захарченко С. М. // International scientific journal «Grail of Science» – 2022. – № 14-15 (May, 2022). – С. 312–319. ISSN: 2710–3056. ISBN 979-8-88526-799-1.

13. Shelestov A., Skakun S., Kussul O. Complex neural network model of user behavior in distributed systems // International Conference «Knowledge- Dialogue-Solutions». 2007.

14. Sun P., Chawla S. On Local Spatial Outliers // IEEE ICDM Conference. 2004.

15. Загальна лінійна модель [Електронний ресурс] URL: https://uk.wikipedia.org/wiki/Загальна лінійна модель

16. Колодчак О.М. Сучасні методи виявлення аномалій в системах виявлення вторгнень // Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua, 2012.

17. Рубан І.В., Мартовицький В.О., Партика С.О. Класифікація методів виявлення аномалій в інформаційних системах // Системи озброєння і військова техніка, No. 3(47), 2016.

Abstract: *The article explores the increasing security threat in the global cyberspace. Measures for assessing information access risks considering individuals' responsibility are proposed. Special attention is given to safeguarding web servers from CSRF attacks and developing methods to enhance*

their protection using machine learning for detecting malicious requests. The significance of web programs as an interface for safeguarding confidential data is highlighted. Vulnerability detection methods through black-box testing, focusing on CSRF attacks, are examined. The attack scenario and consequences for users and web servers are analyzed. Protection methods, such as tokens and HTTP header checks, and automated prevention using the SameSite cookie attribute, are discussed. The potential of machine learning for detecting CSRF is considered as an effective strategy. The importance of enriching CSRF detection tools with semantic information to reduce false positives is emphasized. Controlled learning and classifiers aid in identifying web object security, particularly HTTP requests. An architecture for vulnerability detection utilizing heuristics and machine learning is proposed.

Keywords: *Web programs, confidential data protection, web applications, machine learning (ML), classifier, HTTP request, web application vulnerabilities.*

Науковий керівник: к.т.н., доц. Коробейнікова Т.І.

Стаття надіслана: 30.07.2023 р.