https://www.proconference.org/index.php/usc/article/view/usc19-01-012
DOI: 10.30888/2709-2267.2023-19-01-012

УДК 004-049.5

REVIEW OF SECURE ACCESS TO A WEB RESOURCE USING MACHINE LEARNING METHODS

ОГЛЯД БЕЗПЕЧНОГО ДОСТУПУ ДО ВЕБ-РЕСУРСУ ЗА ДОПОМОГОЮ МЕТОДІВ 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 досліджено зростаючу загрозу безпеці в глобальному кіберпросторі. Подано заходи для визначення ризиків доступу до інформації, захисту web-серверів від CSRF та розвитку методів машинного навчання (МН) для виявлення небезпечних запитів. Розглянуто значення web-програм у захисті даних. Проаналізовано атаки та їх наслідки. Розглянуті методи захисту, включаючи токени, перевірку HTTP-заголовків та автоматизоване запобігання з використанням cookie SameSite. Використання МН для виявлення CSRF є ефективною стратегією. Описано важливість збагачення інструментів виявлення CSRF семантичною інформацією для зниження помилкових результатів. Застосування контрольованого навчання та класифікаторів сприяє виявленню вразливостей web-об'єктів, зокрема HTTP-запитів. Пропонується архітектура для виявлення вразливостей з використанням евристик та МН.*

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

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.*

Вступ.

Останнім часом глобальна кіберсфера стає пріоритетом безпеки [1-3]. Мілітаризація кіберпростору зростає, а заходи попередження недостатньо ефективні. Пропонуються заходи для визначення ризиків доступу до інформації [5-6]. Машинне навчання допомагає у безпеці web-додатків [7-10]. CSRF атаки загрожують web-серверам [11-12]. Досліджується підвищення рівня захисту web-серверів за допомогою машинного навчання.

1 Виявлення вразливого місця web-сервера.

Web-програми - основний інтерфейс для захисту конфіденційних даних і функцій [13]. Вони часто використовуються для подання податкових декларацій, доступу до медичних результатів і фінансових операцій [13]. Проте web-додатки привабливі для зловмисників, що може завдати економічних збитків і отримати доступ до даних [13]. Виявлення вразливостей через методи чорної скриньки стало популярним [14], дозволяючи працювати на рівні HTTP-трафіку [14]. Проте, аналіз таких вразливостей є складним [15], особливо у розумінні семантики web-додатків [15].

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.

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.

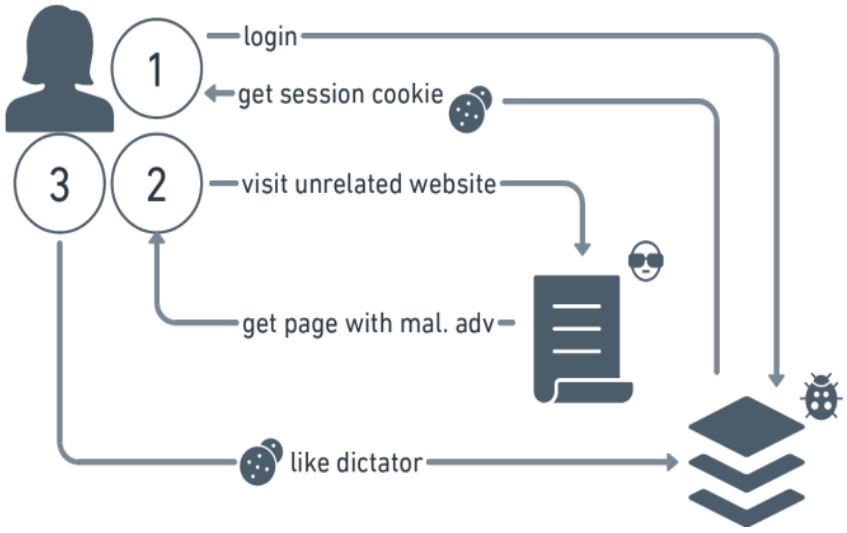

Рисунок 1 – Приклад Cross-site request forgery

Джерело [22]

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.

Автоматизоване запобігання CSRF здатне усунути загрозу, особливо з атрибутом cookie SameSite [1]. Але web-додатки використовують інші методи, такі як перевірка заголовків HTTP-запиту та анти-CSRF токенів [2-3]. Ці методи мають обмеження і вимагають точного розміщення перевірок безпеки [2-3]. Пошук оптимального розташування захисту виходить за межі фреймворків і вимагає ефективних інструментів виявлення CSRF [4]. Використання машинного навчання може стати рішенням для автоматизованої підтримки виявлення CSRF [4]2

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 → 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 ∈ 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. Запропоновану 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.

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.

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

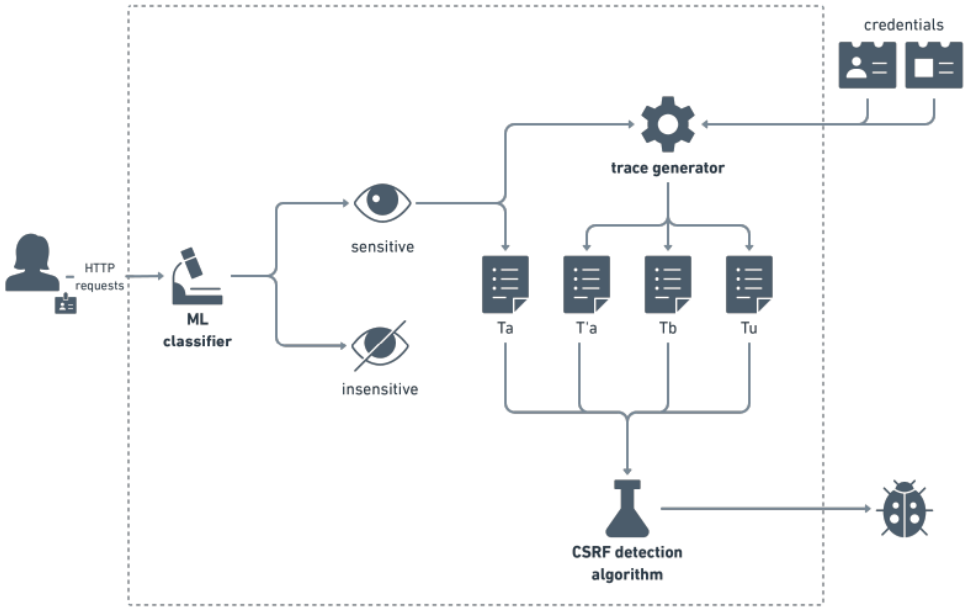

Рисунок 2 – Архітектура рішення

Авторська розробка

2.3 Класифікатор машинного навчання. Для класифікації захисту web-додатків використовується класифікатор машинного навчання. Його навчають на наборі даних з 6000 HTTP-запитів із позначенням двома експертами. Простір ознак класифікатора має 49 вимірів, розділених на структурні, текстові та функціональні категорії. Структурні: описують структурні властивості HTTP-запиту, включають загальну кількість параметрів, параметри, пов'язані з логічними значеннями, ідентифікаторами та блобами. Текстові: фіксують текстові характеристики запитів, використовуючи словник ключових слів, таких як слова в шляху або параметрах. Функціональні: вказують методи HTTP-запиту, наприклад, GET або POST. Словник містить 21 ключове слово, що можуть вказувати на конфіденційні запити. Ці ознаки допомагають виявляти вразливості CSRF, що робить класифікатор ефективним інструментом для захисту веб-додатків.

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

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

Стаття надіслана: 30.07.2023 р.