УДК 004.8

Мураховський Р.М

АНАЛІЗ КОДУ ФРОНТЕНДУ НА ВРАЗЛИВОСТІ

Вінницький національний технічний університет

Анотація
Метою даної роботи є аналіз коду фронтенду на вразливості.
Ключові слова: кібербезпека, вразливості, веб-ресурс, front end, захист, аналіз, фреймворк, атака, скріпт.
Abstract
The purpose of this work is to analyze the frontend code for vulnerabilities.
Keywords: cybersecurity, vulnerability, web resource, front end, protection, analysis, framework, attack, script.

Вступ

Безперервний розвиток інтернету та інформаційних технологій в цілому зробили так, що навіть маленька компанія чи державні установи, має свій веб-застосунок, а кожен веб-застосунок має свої вразливості. В більшості випадків, веб-застосунки розробляються на базі певних загальних фреймворків (каркасів веб-застосунку) [1]. Такий підхід зобов'язує програміста будувати архітектуру програми відповідно до певної логіки, середовище надає можливість для подій, сховищ і з'єднань даних [2]. Каркаси знаходяться на більш високому рівні абстракції в порівнянні з бібліотекою і забирають непотрібні зусилля з розробки приблизно на 80%. Проте це також породжує проблеми з успадкуванням всіх вразливостей, що закладені у каркасі. Фронтенд забезпечує зовнішній вигляд і функціонування сайту, тому аналіз його вразливостей і забезпечення захисту від них є першочерговою ціллю.

Основна частина

Аналіз вразливостей веб-застосунків проводиться багатьма організаціями з безпеки. Зокрема, OWASP Top 10 [3] - це стандартний документ для розробників та безпеки веб-додатків. Він надає широкий консенсус щодо найбільш критичних ризиків безпеки для веб-додатків.

Відповідно до проведеного аналізу найбільш критичними є такі атаки та вразливості, що вони використовують.

Скрипти є найбільшою вразливістю, так як повне функціонування сайту неможливе без них. Для прискорення написаня скриптів найчастіше використовується фрейм Jquery Jquery використовується на 74 відсотках сайтів [4] він прискорює написання скриптів в кілька разів, але підключення може викликати серйозну небезпеку з захистом. Сучасні сайти та супутнє програмне забезпечення часто створюють специфічну основу для реалізації програм, написаних користувачем. Більшість мов веб-програмування є серверними, а отже сценарії, написані на них, запускаються безпосередньо на сервері, і тільки результати їх роботи надсилаються на комп'ютер користувача. На жаль, сценарії веб-сайту не завжди розробляються дійсно хорошими фахівцями. Багато інтернет-проектів використовують безкоштовне або самостійне програмне забезпечення і це викликає багато небезпек з захистом.

XSS атака - це атака на вразливість на сервері, що вставляє довільний HTML /JavaScript код в результат роботи сценарію в тих випадках, коли сценарій не фільтрує дані, які надійшли від користувача. Код може містити шкідливу інформацію, яка може скомпрометувати комп'ютер жертви через експлойти веб-браузера. Також міжсайтовий скриптинг може містити шкідливий JavaScript код, який надсилає свої облікові дані сеансу на інший веб-сервер [5].

Підробка міжсайтових запитів (CSRF) - це атака, яка змушує кінцевого користувача виконувати небажані дії в веб-додатку, для якого вони зараз проходять перевірку автентичності [6]. Вразливості CSRF можуть виникати, коли додатки використовують виключно cookie-файли HTTP для ідентифікації користувача, який відправив конкретний запит [7]. Оскільки браузери автоматично додають файли cookie в запити незалежно від джерела запиту, зловмисник може створити шкідливий веб-сайт, який підробляє міждоменні запити до вразливого додатку. Атакуючий створює підроблені HTTP-запити і змушує жертву відправляти їх через теги зображень, XSS або інші методи [8]. Атакуючий може змусити жертву виконати будь-яку операцію із зміни стану: вхід в систему, оновлення облікових даних, вчинення фінансових транзакцій.

Ін'єкція. Вади ін'єкції, такі як SQL, NoSQL, OS та ін'єкція LDAP, виникають, коли неперевірені дані надсилаються інтерпретатору як частина команди або запиту [9]. Певним чином сконфігуровані дані можуть змусити інтерпретатора виконати команди або отримати доступ до даних без належного дозволу.

Слабка автентифікація. Функції додатків, пов'язані з автентифікацією та управлінням сеансами, часто реалізуються неправильно, що дозволяє зловмисникам компрометувати паролі, ключі або маркери сеансів або використовувати інші недоліки реалізації, щоб тимчасово або назавжди прийняти ідентифікаційні дані інших користувачів [10].

Розміщення чутливих даних у відкритому вигляді. Багато веб-додатків та API не захищають належним чином конфіденційні дані, такі як фінанси, охорона здоров'я та ідентифікація [11]. Зловмисники можуть викрасти або модифікувати такі слабо захищені дані для здійснення шахрайства з кредитною карткою, викрадення особистих даних або інших злочинів. Конфіденційні дані можуть бути скомпрометовані без додаткового захисту, наприклад, шифрування в спокої або під час передачі, і вимагає особливих заходів обережності при обміні ними з браузером.

Небезпеку становлять і **інструменти зовнішнього вигляду.** Вони спрощують процес програмування (наприклад, використання препроцесора Sass замість чистого CSS, оскільки він надає можливість використання циклів, функцій, локальних змінних і багато іншого), проте оскільки браузери не розуміють синтаксис Sass / SCSS, тому код перекладається на CSS [8], а отже можуть інтерпретувати і шкідливі функції.

Головною особливістю виробничих проблем є їх залучення до конкретних версій програмного забезпечення. "дірки" часто не зустрічаються в ряді веб-додатків, а лише в деяких їх виданнях. Однак слід зазначити, що чим популярнішим є програмне забезпечення, тим більше шансів знайти нові уразливості. Це не залежить від якості написаного програмного забезпечення. З метою захисту від цього є своєчасне встановлення всіх оновлень.

Висновки

У даній роботі було визначено та проаналізовано поширенні вразливості веб-серверів, та можливості атак через вразливості, тому що важливим етапом тестування на проникнення є пошук вразливостей. Результатом є повний аналіз існуючих вразливостей і методики захисту веб-сайтів від них. Виконання існуючих методів, аналіз вразливостей і вдосконалення захисту від них забезпечить гарантію безпеки фронтенду, адже саме фронтенд є тим мостом який з'єднує користувача з базою даних сайту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Веб-фреймворки и с чем их едят [Електронний ресурс] –Режим доступу до ресурсу: http://iwsm.ru/blog/show/veb-freymvorki-i-s-chem-ih-edyat.
2. Фреймворки в веб-разработке [Електронний ресурс] –Режим доступу до ресурсу: https://web-creator.ru/articles/about_frameworks.
3. OWASP Ten [Електронний ресурс] –Режим доступу до ресурсу: https://owasp.org/www-project-top-ten/
4. Стаття [Електронний ресурс] –Режим доступу до ресурсу: https://www.anti-malware.ru/news/2019-04-22-1447/29512
5. Уязвимости веб-приложений [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://www.ptsecurity.com/upload/corporate/ruru/analytics/Web-Vulnerabilities-2019-rus.pdf.
6. Евтеев Д. SQL Injection от А до Я [Електронний ресурс] / Дмитрий Евтеев. – 2008. – Режим доступу до ресурсу: https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/PT-devteev-AdvancedSQL-Injection.pdf.
7. SQL инъекции. Проверка, взлом, защита [Електронний ресурс] // BVN2. – 2011. – Режим доступу до ресурсу: https://habr.com/ru/post/130826/.
8. How to Prevent SQL Injection Attacks [Електронний ресурс] // eSecurityPlanet. – 2018. – Режим доступу до ресурсу: https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks.html.
9. Статья [Електронний ресурс] –Режим доступу до ресурсу: https://htmlacademy.ru/tutorial/php/sql-injections
10. Статья [Електронний ресурс] –Режим доступу до ресурсу: https://www.securitylab.ru/blog/company/axxtel/348165.php
11. Результаты тестирования шести ведущих фреймворков на производительность [Електронний ресурс] –Режим доступу до ресурсу: http://www.alrond.com/ru/2007/jan/25/rezultaty-testirovanija-6-frameworks/.

Мураховський Руслан Миколайович – студент групи 2БС-17б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: murahovskiyrus@gmail.com.

Murahovskiy Ruslan Mykolayovych — student of group 1BS-17b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: murahovskiyrus@gmail.com.

Науковий керівник: ***Войтович Олеся Петрівна*** — доцент кафедри захисту інформації, Вінницький національний технічний університет, м. Вінниця

Supervisor: Voytovych Olesya Petrovna — Cand. Sc., Associate Professor of Information Protection, Vinnytsia National Technical University, Vinnytsia