МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет «Одеська юридична академія»

ЄВРОПЕЙСЬКІ ОРІЄНТИРИ РОЗВИТКУ УКРАЇНИ В УМОВАХ ВІЙНИ ТА ГЛОБАЛЬНИХ ВИКЛИКІВ XXI СТОЛІТТЯ: СИНЕРГІЯ НАУКОВИХ, ОСВІТНІХ ТА ТЕХНОЛОГІЧНИХ РІШЕНЬ

МАТЕРІАЛИ
Міжнародної науково-практичної конференції

19 травня 2023 року

У двох томах

Том 1

Видавництво
«Юридика»
2023

УДК 005.332.2(4):316.42(477)"364""20"(062.552)
Є24

Рекомендовано до друку Вченою радою
Національного університету «Одеська юридична академія»
(протокол № 3 від 16.06.2023 р.)

За загальною редакцією **С. В. Ківалова.**

Відповідальний за випуск **М. Р. Аракелян.**
Матеріали видано в авторській редакції.

Є24 **Європейські** орієнтири розвитку України в умовах війни та глобальних викликів XXI століття: синергія наукових, освітніх та технологічних рішень : у 2 т. : матеріали Міжнар. наук.-практ. конф. (м. Одеса, 19 травня 2023 р.) / за загальною редакцією С. В. Ківалова. – Одеса : Видавництво «Юридика», 2023. – Т. 1. – 790 с.

ISBN 978-617-8263-39-3

Матеріали Міжнародної науково-практичної конференції «Європейські орієнтири розвитку України в умовах війни та глобальних викликів XXI століття: синергія наукових, освітніх та технологічних рішень». У першому томі збірника містяться наукові напрацювання вчених, практиків, військовослужбовців у теоретичній та емпіричній площині в умовах повномасштабного військового вторгнення у сферах філософських основ, загальної теорії та історичних досліджень держави і права, актуальних проблем світових соціально-політичних процесів, соціології та психології. Висвітлено питання загроз національній безпеці в їхньому конституційному вимірі, у рамках міжнародного та європейського права, трудового права та права соціального забезпечення, земельного, аграрного та екологічного права, пов'язані з функціонуванням економіки та підприємництва в умовах європейського вибору України. Розглянуто проблеми інформатизації та цифровізації суспільства, захисту інформації та кібербезпеки в умовах військового вторгнення, питання методики викладання іноземних мов, теорії та практики перекладу, проблеми лінгвістики та журналістики.

Збірник розраховано на наукових та науково-педагогічних працівників, здобувачів вищої освіти, практичних працівників у сферах юридичної, економічної, соціологічної, політологічної, психологічної, філологічної наук, журналістики та кібербезпеки тощо.

УДК 005.332.2(4):316.42(477)"364""20"(062.552)

ISBN 978-617-8263-37-9 (у 2 т.)
ISBN 978-617-8263-39-3 (т. 1)

© Національний університет
«Одеська юридична академія», 2023

СЕКЦІЯ 12
ІНФОРМАТИЗАЦІЯ ТА ЦИФРОВІЗАЦІЯ СУСПІЛЬСТВА В ЕПОХУ СТРІМКОГО РОЗВИТКУ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ

на принципах конфіденційності, доступності та цілісності при зберіганні даних облікових записів, доступі і підключеннях користувачів. Управління ризиками безпеки й відповідність нормативним вимогам належать до політик і процесів, які організації мають забезпечити для дотримання законів, правил та нормативних актів.

Проведений аналіз проблем тестування вебзастосунків виявив наявність різних підходів для захисту програмних продуктів. Доволі не просто зорієнтуватися і зрозуміти, які саме методи використовувати або коли і в якій послідовності застосовувати ті чи інші засоби тестування. Доцільним є баланс у використанні декількох різних методів і підходів, які будуть доповнювати і посилювати один одного. Збалансований підхід до тестування безпеки залежить від багатьох чинників, у тому числі і зрілості процесу тестування та корпоративної культури, але важливо, щоб компанії приділяли увагу інтеграції тестування безпеки на якомога ранніх етапах життєвого циклу розробки ПЗ. Тестування безпеки має поєднувати різні ефективні техніки і методології для оцінки безпеки програми та її оточення.

Список використаних джерел:

1. Web Security Testing Guide. URL: https://owasp.org/www-project-web-security-testing-guide/stable/2-Introduction/
2. Keshav M. Automated VS Manual Security Testing – Which One to Choose? URL: https://www.getastra.com/blog/security-audit/manual-security-testing/
3. Трофименко О. Г., Пастернак Ю. Ю., Манаков С. Ю., Лобода Ю. Г. Автоматизація тестування вебсайтів електронної комерції. *Сучасна спеціальна техніка*. 2021. № 2(65). С. 46–59. DOI: 10.36486/mst2411–3816.2021.2(65).5
4. Trofymenko O. H., Dyka A. I. Problems of testing e-commerce websites. *International scientific conference «Information technologies and management in higher education and sciences»*: conference proceedings (November 28, 2022). Fergana, the Republic of Uzbekistan. Riga, Latvia: “Baltija Publishing”, 2022. Part 3. P. 195–199. DOI: 10.30525/978-9934-26-277-7-230

Ключові слова: тестування безпеки, вебзастосунок, безпека вебзастосунку, уразливості безпеки, засоби тестування.

Key words: security testing, web application, web application security, security vulnerabilities, testing tools.

ТОЛОКНОВ АНАТОЛІЙ АРНОЛЬДОВИЧ

Національний університет «Одеська юридична академія», асистент кафедри інформаційних технологій

ОГЛЯД МЕТОДІВ ЗАХИСТУ ВЕБФОРМ

Завдяки формам користувач «спілкується» з сайтом, вебсервісом: замовляє продукти, користується інтернет-банкінгом і просто автентифікується. Не треба говорити, що зберігання конфіденційної інформації, захист фінансової інформації – це одна з головних задач вебмайстра

при створенні вебформ. Нижче наведений короткий огляд методів, які необхідно застосовувати при створені безпечних вебформ.

Використання GET і POST. Щоразу, коли ви вводите URL-адресу у свій браузер і відвідуєте вебсайт, ви робите запит GET. Запит GET повідомляє вебсерверу, що ви хочете отримати (GET) деякі дані. Хорошим прикладом форми, яка збирає дані для отримання інформації, є пошукова форма. Коли ви шукаєте інформацію, ви зазвичай просите «отримати» інформацію, тому ваша пошукова форма має видати запит GET. За замовчуванням усі форми видають запити GET після надсилання. Слід мати на увазі, що браузери зазвичай кешують запити GET.

Пошуковий запит передається у незахищеному вигляді в рядку адреси браузера, наприклад: https://example.com/?some-sweet-**data=search+data**.

Щоразу, коли форма надсилається за допомогою запиту POST, усі дані у формі кодуються в URL-адресі, а потім поміщаються в тіло HTTP-запиту, який потім надсилається на вебсервер для обробки [1]. Коли розробляється логіка на стороні сервера, можна отримати дані форми, переглянувши тіло вхідного запиту та проаналізувати те, що потрібно.

Метод приманки (honeypot). Один елементарний спосіб запобігання спаму називається «приманка» (honeypot) [1], і концепція досить проста. Можна включити приховане поле у форму. Буде відомо, що справжня людина ніколи не повинна змінювати це поле. Отже, якщо форма надсилається з даними з цього поля, можна припустити, що це був бот, і відхилити надсилання.

На практиці це може виглядати так:

```
<input name=«honeypot» class=«visually-hidden» tabindex=
«-1» autocomplete=«off»>
```

- Атрибут name важливий, щоб знати, що перевіряти на сервері. Значення атрибуту може бути будь-яким.
- До поля введення застосовується клас для призначення стильових правил, які ховають його для користувачів. Наприклад: visibility: hidden.
- Атрибут tabindex=«-1» вилучає поле з навігації з клавіатури. Це важливо для користувачів допоміжних технологій.
- І нарешті, для запобігання автоматичному заповненню введених даних форми браузером використовуємо autocomplete=«off».

Реалізація такого методу може коштувати дуже мало часу та зусиль. Але ж багато ботів достатньо розумні, щоб визначити, коли поле є приманкою, і вони пропустять його. Але навіть якщо це зупинить 10 % спаму, зусилля того варті.

Шифрування даних (SSL). Вебсайти, які не є захищеними (які не використовують SSL), передають усе між браузером і вебсервером у вигляді звичайного тексту. Це означає, що будь-який користувач, який, наприклад, намагатиметься увійти на вебсайт, надсилатиме свою адресу електронної пошти та пароль через загальнодоступний Інтернет (через свого провайдера та багатьох інших провайдерів) на ваш вебсайт.

Одним із рішень проблеми надсилання паролів та інших цінних даних, таких як номери кредитних карток, у вигляді звичайного тексту через мережу є шифрування всієї транзакції HTTP. Основним стандартом у цій галузі є Secure Sockets Layer (SSL) [2]. SSL працює шляхом встановлення зашифрованого сеансу на окремому порту HTTP-сервера. Потім клієнт виконує те, що називається рукостисканням (handshake), під час якого відбувається обмін криптографічною інформацією між клієнтом і сервером, щоб вони могли автентифікувати один одного й обмінюватися криптографічними ключами.

- Без сертифіката SSL: username: someName; password: iLoveNuts
- Із сертифікатом SSL (зашифрований):

SIUFJaYQNtsn+y73mfBYv3fVfjJ2GdHl4A7XnTJXxgUyd4/TrU3nN+g3aJ4BVXSJ

Перевірка даних. Перевірка даних введених у форму може відбуватися на стороні клієнта, у браузері, і на стороні сервера. Рекомендується виконувати перевірку і в браузері і на сервері. На стороні клієнта перевірка даних відбувається для запобігання «простих» помилок, які робить користувач під час введення даних у поля форми. Перевірка у формі зазвичай виконується за допомогою атрибутів HTML або JavaScript. Це робиться для підвищення зручності користування формою шляхом надання негайного зворотного зв'язку [1, 3]. Будь-який технічний користувач може змінити атрибути перевірки HTML або взагалі обійти перевірку на стороні клієнта, зробивши запит HTTP поза вашою формою. Тому рекомендується завжди робити перевірку даних на боці сервера.

Очистка даних (екранування). Очищення даних відбувається, коли ви перетворюєте або видаляєте небезпечні дані. Це відрізняється від перевірки, оскільки ви не відхиляєте введені дані. Ви змінюєте їх, щоб використовувати безпечно. Наприклад, скажімо, у вас є форма, яка запитує ім'я та прізвище. Користувач може ввести наступне:

Ім'я: l33t; DROP TABLE user, Прізвище: <script>alert('h4x0r')</script>

Такі дані можуть піддавати сайт двом типам атак: SQL injection і міжсайтовий скриптинг (XSS – cross-site scripting).

Якщо спробувати додати ім'я користувача до бази даних як є, можна скинути всю таблицю user (SQL Injection). Якщо ви збережете прізвище як є, ваша база даних буде нормальною, але якщо ви додасте це прізвище до свого HTML, це може вставити довільний JavaScript на сторінку (XSS) [1, 5].

Використання маркерів CSRF. Маркер CSRF (Cross-Site Request Forgery, підробка міжсайтового запиту) – це секретний, непередбачуваний, випадковий набір символів, створений на стороні сервера та надісланий клієнту, щоб клієнт пізніше міг перевірити свою особу та/або сеанс. CSRF зазвичай створюється за допомогою секретного ключа та мітки часу, хоча ми можемо включити в цей алгоритм деякі особливості користувача [1, 4, 5].

Коли користувач входить у вашу програму, ви призначаєте користувачеві унікальний токен CSRF і зберігаєте цей токен десь на боці сервера, як-от сеанс, файл, база даних тощо. Щоразу, коли користувач

робить запит до серверної частини (особливо якщо цей запит потребує даних або надсилає дані), цей CSRF буде надіслано в запиті, щоб сервер міг перевірити користувача.

У інтерфейсі можна зберегти цей маркер CSRF у прихованому полі введення або у файлі cookie.

```
<input type=«hidden» name=«csrf» value=«d4f3e48f-7ae3-4398-0dca81383e6c»>
```

Отже, коли користувач надсилає форму, можна перевірити, чи CSRF, надісланий користувачем, і той, який ви зберегли на стороні сервера, збігаються.

Використання CAPTCHA. Простим методом підвищення рівня безпеки форми від спаму є включення до перевірки математичного питання, наприклад: «Скільки буде 10 + 6?», а ще краще цифри писати словами. Приймаються лише дані з правильною відповіддю. Також можна скористатися стороннім сервісом, наприклад reCAPTCHA (https://developers.google.com/recaptcha/), який належить Google. Заміна цього сервісу буде hCaptcha (https://www.hcaptcha.com/) [2, 5].

Наведений вище огляд має деякі нюанси у використанні і може бути розширений. Але використовуючи вже ці методи вебмайстер значно підвищить безпеку використання вебформ.

Список використаних джерел:

1. Gil A. How to Build HTML Forms Right: Security [Електронний ресурс] / Austin Gil. – 2020. – Режим доступу до ресурсу: https://austingil.com/how-to-build-html-forms-right-security/
2. Sikira A. Handling HTML form security [Електронний ресурс] / Amer Sikira. – 2021. – Режим доступу до ресурсу: https://webinuse.com/handling-html-form-security/
3. Degges R. Everything You Ever Wanted to Know About Secure HTML Forms [Електронний ресурс] / Randall Degges. – 2017. – Режим доступу до ресурсу: https://www.twilio.com/blog/2017/09/everything-you-ever-wanted-to-know-about-secure.html-forms.html
4. Knight J. Securing HTML FORMs [Електронний ресурс] / Jon Knight // Ariadne – Режим доступу до ресурсу: http://www.ariadne.ac.uk/issue/5/securing-forms/
5. Coyier C. Serious Form Security [Електронний ресурс] / Chris Coyier // CSS-Tricks. – 2020. – Режим доступу до ресурсу: https://css-tricks.com/serious-form-security/

Ключові слова: автентифікація, вебформа, персональні дані, безпека.
Key words: authentication, web form, personal data, security.