УДК 681.04 DOI: 10.26565/2519-2310-2022-2-03

СУЧАСНІ ЗАГРОЗИ ТА СПОСОБИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ВЕБ-ЗАСТОСУНКІВ

Кирило Яремчук, Денис Воскобойников, Ольга Мелкозьорова

Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна
kir.yaremchuk@gmail.com, denisvoskoboinykov@gmail.com, olha.melkozerova@karazin.ua

Надійшла: вересень 2022. Прийнята: жовтень 2022.

Анотація: *Складність розроблюваних веб-застосунків зростає з кожним роком, що, в свою чергу, робить важкоздійсненним забезпечення їхньої безпеки. Саме тому, доцільно приділяти особливу увагу критичним проблемам захисту програмного забезпечення. Вміння оцінювати ризики та запобігати вразливостям ще на етапі проектування продукту є вкрай важливою задачею, котра знижує потенційні складності при експлуатації застосунку. За останні роки кількість випадків витоку даних у всіх галузях ринку зменшилася, але, їх руйнівність стала значнішою. Серед усіх атак, атаки на веб-застосунки становлять більш ніж 50 відсотків. Згідно зі списком вразливостей OWASP Top Ten, в роботі розглянуто актуальні категорії вразливостей та напрямки атак на існуючі веб-застосунки. Було розглянуто ефективні способи їх запобіганню. Наведені рекомендації щодо реалізації та підтримки захищеності додатків, розроблених з використанням бібліотеки ReactJS. Було виділено найпоширеніші загрози безпеки продуктів на базі React на протязі життєвого циклу додатку. Розглянуті основні способи оптимізації ReactJS.*

Ключові слова: *вразливість; веб-застосунки; загрози веб-застосунків; методи безпеки ReactJS*

1. Вступ

Завдяки тривалому часу свого розвитку, веб-застосунки почали являти собою дещо куди значніше, аніж просто сайти з контентом. На просторах мережі Інтернет с кожним днем з'являються все більш складні веб-застосунки за своїми цілями й можливостями, котрі пропонують нові рішення задля задоволення вимог споживачів в усіх галузях ринку. Веб-застосунки являють собою найбільш зручний та ефективний засіб для представлення інформації й надання послуг у мережі. Компанії з різнобічних галузей ринку продовжують створювати веб-застосунки для просування своїх товарів та послуг у Інтернеті, займаючи свої ніші у цифровому світі. Мобільні інструменти та веб-технології захопили вершину списку на довгі роки. Такі цифрові сервіси часто бувають критично значущими й потребують захисту задля забезпечення безпеки різного роду конфіденційної інформації. Галузь веб-технологій розвивається невпинними кроками, однак несе з собою і певні ризики безпеки: - забезпечення захисту значної кількості різнотипної інформації досягти досить складно. Недотримання вимог безпеки може загрожувати втратою ресурсів, а інколи й проблемами з законом [1]. На рис. 1 представлені тенденції злому систем безпеки у 2014-2021 роках [2].

Відповідно звіту *Risk Based Security* про тенденції порушення даних за 2021 рік, у 2020 та 2021 роках було втрачено 27,81 і 18,88 млрд записів, тоді як у 2019 році усього 4,681 млрд. записів [2]. Попри те, що кількість таких випадків зменшилася у 2020 та 2021 роках, втрати даних стали більш масштабними. В цілому, можна спостерігати відсутність кореляції між кількістю випадків втрати даних та кількістю втрачених даних, що говорить про те, що навіть одне малозначне порушення безпеки може спричинити серйозні наслідки. Щоб на практиці знизити можливі ризики безпеки, потрібно ретельно контролювати усі потенційні загрози та уважно дотримуватися існуючих стандартів інформаційної безпеки (ІБ). На рис. 2 представлено огляд основних тенденцій атак (зломів) по галузям економіки у 2021 році [2].

Так, найбільший вплив загроз ІБ у 2021 році зазнали сфери охорони здоров'я, фінансів, страхування, ІТ- індустрія й наукова галузь, промисловість та галузь державного управління, котрі являють собою значну долю від усіх атак.

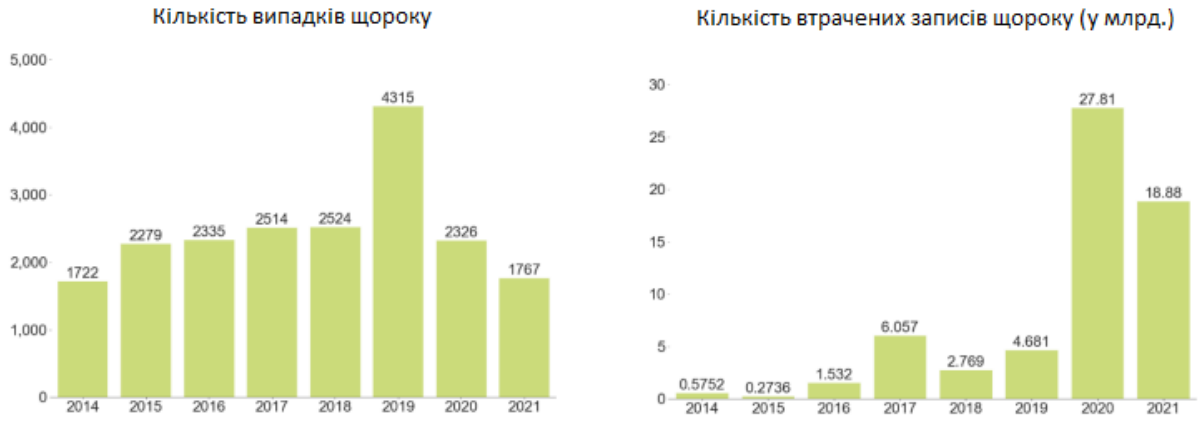

Рис. 1 – Тенденції атак систем ІБ у 2014-2021 роках

Попри те, що у даному дослідженні розкривається лише проблематика загроз та вразливостей веб-застосунків, джерела даних, стосовно випадків витоку даних, можуть бути абсолютно різними.

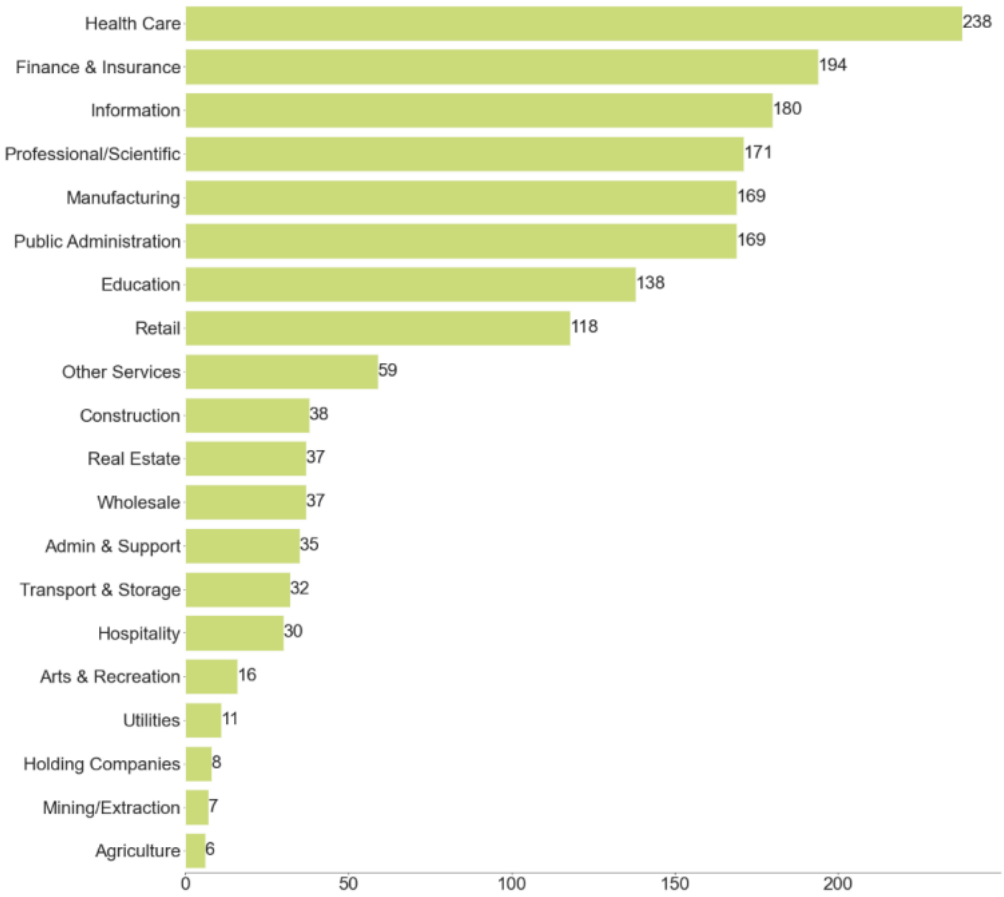

Рис. 2 – Тенденції типових атак за галузями у 2021 році

2. Загальний стан в сфері безпеки веб-застосунків

Атаки на веб-застосунки являють собою найнебезпечнішу загрозу ІБ багатьох сучасних організацій. Атаки, що спрямовані на веб-застосунки становили 40% від усіх зламів у 2015 році й вже понад 50% у 2021 році [3]. Згідно з аналізу *WhiteHat Security* на момент 2021 року, так чи інакше, принаймні 50 відсотків веб-застосунків у таких сферах, як: освіта, промисло-

вість, роздрібна торгівля, комунальні служби та охорона здоров'я, мають принаймні одну вразливість, котра можливо використати для зламу відповідних додатків. В ході дослідження було виявлено, що через те, що з кожним роком все більша кількість галузей фокусуються на розвиток у мережі, ризики витоку їх корпоративних даних залишаються досить високими, що позначається на збільшенні вразливостей [4]. Також було виявлено, що, наприклад, лише деякі з найпоширеніших ризиків від *WhiteHat* можна побачити в списку OWASP (*Open Web Application Security Project*), що регулярно оновлює найбільш розповсюджені вразливості.

При цьому, знаходження проблеми ІБ не вирішує проблему безпеки саме по собі. Для усунення виявлених вразливостей потрібен деякий час. Так, час на виправлення критичних вразливостей застосунків, в середньому, сягає 194 діб. Середній час викриття критичних вразливостей становить 300 діб, а вразливостей с високим рівнем ризику понад 500 діб [4]. Чим більший термін виявлення, тим більші шанси злодіїв на «успіх» від вразливості, що експлуатується. Також, великий відсоток веб-застосунків залишається вразливим завжди, що каже про те, що розробники не в змозі впоратися з усіма вразливостями своїх продуктів.

Усунення проблем безпеки та виявлення вразливостей ще на етапі проектування й розробки веб-застосунку потребує значно менших ресурсів, аніж їх виправлення у вже існуючому продукті. У існуючому застосунку процес виправлення вразливостей здатний призвести до втрати працездатності веб-застосунку, що в першу чергу несе втрату репутації, коштів й ресурсів на вирішення проблеми. Вдосконалення вже існуючого веб-застосунку у більш захищене рішення може коштувати значно більше сил і коштів, аніж його початкова реалізація у вигляді захищеного застосунку.

На сьогоднішній день, розробка веб-застосунків, беручи до уваги стандарти безпеки, є неодмінним обов'язком. Список загроз та вразливостей постійно розширюється, не стоячи на місці, тому розробники веб-застосунків мають мати відповідні компетенції у сфері ІБ, щоб залишати свій продукт безпечним. На жаль, впоратися з цією задачею буває не часто, тому у якості допомоги розробники можуть звертатися до стандартів безпеки або користуватися іншими інструментами, котрі здатні допомогти у забезпеченні безпеки їх програмних рішень.

3. Стислий огляд поширених загроз безпеки веб-застосунків та способів захисту на базі бібліотеки ReactJS

OWASP являє собою головний ресурс з інформацією про стандарти, відомі методи захисту та інструменти безпеки. Через велику базу користувачів та відкриту спільноту, OWASP роками залишається своєрідною бібліотекою, в котрій містяться різноманітні документи, що допомагають вирішити проблеми безпеки при розробці продуктів.

Головною метою проекту OWASP Top Ten є формулювання основних загроз ІБ, визначення найважливіших та найбільш критичних недоліків захищеності веб-застосунків. Цей список створено за сприяння об'єднаної спільноти проекту. На початку існування ціллю проекту було покращити знання розробників, щодо безпеки, але згодом проект став справжнім стандартом безпеки веб-застосунків. Наразі актуальною є версія 2021 року, котра вийшла відносно недавно й включає в себе найбільш актуальний список загроз та вразливостей.

Останню версію OWASP складено на базі опитувань у відповідній галузі та більш ніж 40 відгуків від компаній, що займаються забезпечення безпеки. Також, була зібрана інформація з більш ніж ста тисяч застосунків та API (*Application Programming Interface*), що дозволило створити переконливу інформаційну базу задля подальшого аналізу даних.

Останнім часом, веб-інструменти та технології зазнали суттєвих змін. Так, мову програмування JavaScript та її фреймворки *AngularJS* та *ReactJS* [5], можливо впевнено назвати своєрідним фундаментом або основою мережі. Завдяки цим фреймворкам деяка «відповіда-

льність» серверів була перенесена на сторону браузера. В той же час величезну популярність набрала серверна технологія розробки Node.js, котра працює на базу подій [6]. На рис. 3 можна бачити зміни у списку OWASP Top Ten [7].

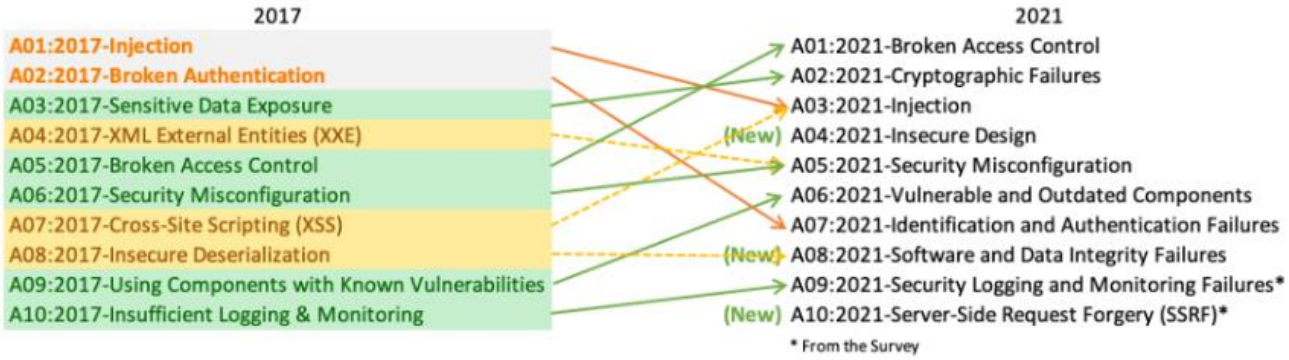

Рис. 3 – Рейтінг OWASP Top Ten за 2017 та 2021 рр..

Як можна бачити, за цей термін, перелік OWASP зазнав великих змін [7]. Так, *Broken Access Control* - піднялася в списку й опинилася на 1-й позиції. В реалізованих веб-застосунках має бути реалізовано доступ до інформації згідно з привілегіями користувачів. У випадку ігнорування безпеки доступу до інформації, це може дозволити користувачам отримати доступ до чутливої інформації без наданих на це повноважень, що може призвести до незворотних наслідків та/або втрати критично важливих даних.

Cryptographic Failures (раніше *Sensitive Data Exposure*) - піднялася на 2-ге місце. Ця вразливість пов'язана з втратою конфіденційних даних чи зломом системи, якщо використовувані криптографічні методи не здатні забезпечити достатній рівень ІБ: наприклад, неактуальні криптографічні шифри та протоколи.

*Injection (*до цієї категорії було додано і XSS *(Cross Site Scripting))* - міжсайтове виконання сценаріїв наразі стало невід'ємною частиною цієї категорії (опустилася на 3 місце). Дана вразливість надає злочинцям змогу впроваджувати у веб-застосунок потрібні їм недекларовані дані та/або інструкції, що забезпечує отримання несанкціонованого доступу до цільових даних. Також, ін'єкції здатні власноруч змінювати веб-застосунок.

Insecure Design - поєднує у собі ризики безпеки, котрі пов'язані з недоліками дизайну (тобто загальної структури і окремих алгоритмічних конструкції додатків). Тому, дизайн має охоплювати шаблони проектування й безпечну еталонну архітектуру програмного забезпечення (ПЗ). Являє нову категорію у релізі загроз 2021 року (4-те місце).

Security Misconfiguration – ця категорія піднялася вгору (на 5-ту позицію), через розширення кількості використань ПЗ з широкими можливостями їх налаштування, що обумовлює потенційну можливість нелегітимного доступу до інформації та маніпулювання даними, у томи числі, модифікацію і зміну даних тощо.

Vulnerable and Outdated Components (раніше, *Using Components with Known Vulnerabilities)* – є єдиною категорією, яка не має жодних CVE (*Common Vulnerabilities and Exposures*), зіставлених із включеними CWE (*Common Weakness Enumeration*), піднялася на 6 місце [7]. Охоплює всі застарілі складові ПЗ, експлуатація складає критичні ризики ІБ системи. Потребує регулярне сканування системи, що дозволяє завчасно виявляти наявні проблеми ІБ (*намічі безпеки, використання експлойт-паків*).

Identification and Authentication Failures (раніше, *Broken Authentication*) – ця категорія, відтепер, включає CWE, котрі більшою мірою відносяться до помилок ідентифікації, опустилася на 7 місце. Категорія охоплює вразливості, що включають в себе фальсифікацію обліко-

вих даних та "brute force" атаки. Вразливості цієї категорії є похідними недоліків застосованих механізмів автентифікації та перевірки сеансу користувача.

Software and Data Integrity Failures - нова категорія (8 місце рейтингу), яка фокусується на ідеях оновлень ПЗ, критичних даних і конвеєрів CI/CD (*Continuous Integration/Continuous Delivery*) без перевірки на цілісність, що забезпечує злочинців можливостями використовувати дані, котрі надходять до серверу, для забезпечення можливостей проведення атак [7].

Security Logging and Monitoring Failures (раніше, *Insufficient Logging & Monitoring*) – яка займає 9 місце, була розширена задля включення більшій кількості типів вразливостей, що пов'язані з незадовільним поточним контролем і фіксуванням подій ІБ. Наявність цих вразливостей обумовлює труднощі зі своєчасністю реагування на збої у роботі застосунків.

Server-Side Request Forgery - нова категорія, яка була додана з опитування галузевих фахівців. Її наявність надає зловмиснику можливість реалізувати недекларованe надсилання (сервером додатку) запитів до обраного зловмисником домену. Ця вразливість може виникнути у разі, якщо веб-застосунок отримує дані без перевірки адреси, яку надає користувач.

Бібліотека *ReactJS* (*або React*) завдяки своїй гнучкості є найбільш поширеним рішенням для створення клієнтської частини веб-застосунків (завдяки JSX (*JavaScript XML*) синтаксису)) [8]. У зв'язку зі зростанням складності розроблюваних застосунків і збільшенням обсягів даних, проблеми безпеки таких продуктів невпинно зростають. Так, серед найпоширеніших загроз для веб-застосунків, що створені із застосуванням *ReactJS* слід виділити [8]:

- Вразливість *Zip Slip*, є критично важливою і дозволяє маніпулювати застосунком за рахунок вилучення архіву та інтегрування будь-яких файлів у систему;
- *XSS* (або *Cross-Site Scripting*), що передбачає можливість виконання шкідливого програмного коду, котрий приймається за справжній і виконується за стосунком;
- *XXE* (*XML External Entity*) атаки на XML парсери у випадках, коли ті некоректно опрацьовують посилання на деякі зовнішні сутності;
- *Arbitrary Code Execution*, яка працює за принципом впровадження зловмисником експлойту (та/або експлуатації вразливості 0-го дню), що забезпечує віддалене ініціювання виконання нештатного програмний коду та/або маніпулювання діями скомпрометованого додатку та/або апаратного засобу;
- Вразливість *Broken Authentication* здійснюється завдяки існування передумов компрометації діючих процедур авторизації (тобто недосконаліть автентифікації);
- Атаки типу *SQL Injection*, де зловмисник може відправляти шкідливі *SQL*-запити до бази даних та маніпулювати даними бази даних задля отримання своїх цілей.

Серед способів покращення *ReactJS*, що обумовлюють показники безпеки веб-застосунків, можна виділити наступні [8]:

- Захист від XSS, за допомогою використання технології прив'язки даних;
- Поточний моніторинг URL посилань та контроль впровадження стороннього шкідливого ПЗ, котре може здійснюватися через URL;
- Своєчасне оновлення використовуваних бібліотек. Актуальні версії React позбавлені від багатьох вразливостей минулих релізів;
- При розміщенні коду до вузлів об'єктної моделі документа (DOM - *Document Object Model*), слід не допускати прямого відкритого доступу;.

- Забезпечити протидію уразливостям, що діють за принципом впровадження даних *JSON* (*JavaScript Object Notation*). Зазвичай ці дані відображаються на боці сервера і передаються разом зі сторінками *React* у форматі обміну даними *JSON*;
- При розгортанні HTML коду, його слід розміщувати відразу у *DOM* вузли, що потребує ретельної перевірки всіх процедур;
- Забезпечити регулярність перевірок поточного стану вразливостей (інструмент *OWASP Dependency-Check*), що надає змогу виявляти відомі різновиди вразливостей, які розташовуються у залежностях проекту, перш ніж додавати їх до проекту;
- Виключити застосування сумнівного коду із бібліотек та використовувати ПЗ, яке оптимізує створюваний код;
- Впроваджувати корисні функції *React* для серверного відображення даних на клієнті, що надасть змогу екранувати дані при їх візуалізації на клієнтській частині застосунку у автоматичному режимі.

4. Висновки

1. Використання методів та практик щодо написання безпечного коду, так само як і проектування веб-застосунків з використанням захищеного дизайну являють собою досить складну задачу.

2. Постійний аналіз стану проблематики, щодо поточного стану відомих вразливостей веб-застосунків та появи сучасних методів їх парирування, є принциповою умовою роботи для фахівців, які професійно пов’язані з розробкою та тестуванням веб-застосунків.

3. Питанню постійного моніторингу вразливостей має приділятися певна увага протягом усього життєвого циклу роботи створеного застосунку. Для контролю за вразливостями слід використовувати діючи рейтинги ІБ, та відповідні засоби відстеження актуальних вразливостей та загроз ІБ.

4. Веб-застосунки, що реалізовані на базі використання бібліотеки *ReactJS*, дозволяють значно зменшити зусилля, які необхідно докласти розробникам програмних додатків для досягнення певного рівня ІБ. Ця бібліотека сприяє дотримання сучасних стандартів безпеки веб-застосунків, та дозволяє уникати типових помилок при розробці відповідних додатків.

Список літератури

[1] Inamdar, D. M., & Gupta, S. (2020). A Survey on Web Application Security. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, (6), 223-228.

[2] RiskBased Security, 2021 Mid Year Data Breach QuickView Report. (2021). Вилучено з https://pages.riskbasedsecurity.com/download-the-2021-mid-year-data-breach-quickview-report-today

[3] Help Net Security. Web app attacks are skyrocketing, it’s time to protect APIs. (2021). Вилучено з https://www.helpnetsecurity.com/2021/12/27/web-app-attacks-increased/

[4] Dark Reading, WhiteHat Security: 50% of Apps Are Vulnerable. (2021). Вилучено з https://www.darkreading.com/application-security/whitehat-security-50-of-apps-are-vulnerable

[5] Imaginary Cloud, Angular vs React: a comparison of both frameworks. (2020). Вилучено з https://www.imaginarycloud.com/blog/angular-vs-react/

[6] AltexSoft, The Good and the Bad of Node.js Web App Programming. (2022). Вилучено з https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-node-js-web-app-development/

[7] OWASP Top Ten – 2021. (2021). Вилучено з https://owasp.org/www-project-top-ten/

[8] React.js security best practices. (2020). Вилучено з https://upplabs.medium.com/react-js-security-best-practices-62b9a281cc42

Received: September 2022. Accepted: October 2022.

Authors:
Kyrylo Yaremchuk, student (magistracy), Faculty of Computer Science, V. N. Karazin Kharkiv National University, Kharkov, Ukraine.

E-mail: kir.yaremchuk@gmail.com

Denys Voskoboinykov, student (magistracy), Faculty of Computer Science, V. N. Karazin Kharkiv National University, Kharkov, Ukraine.

E-mail: kir.yaremchuk@gmail.com

Olha Melkozerova, Ph.D., Associate Professor Department of Security of Information Systems and Technologies, V.N. Karazin Kharkiv National University, Ukraine.

ORCID ID https://orcid.org/0000-0002-1134-2925

E-mail: olha.melkozerova@karazin.ua

Modern threats and ways to secure web applications.

Abstract. The complexity of the developed web applications is growing every year, which, in turn, makes it difficult to ensure their security. That is why it is advisable to pay special attention to the critical problems of software protection. The ability to assess risks and prevent vulnerabilities at the product design stage is an extremely important task, which reduces the potential difficulties in the operation of the application. In recent years, the number of data breaches in all market sectors has decreased, but their consequences have become more dangerous. Among all attacks, attacks on web applications account for more than 50 percent. According to the OWASP Top Ten list of the vulnerabilities, the relevant categories of vulnerabilities and directions of attacks on existing web applications were worked out in the work. Effective ways of their prevention are considered. Recommendations for implementing and maintaining the security of applications developed using the ReactJS library are provided. The most common security threats to React-based products throughout the application life cycle have been identified. Modern way of ReactJS optimization are considered.

Keywords: vulnerability; web applications; web application threats; ReactJS security methods.