**DOI: 10.6084/m9.figshare.11340671**
УДК 004.49

**Цюцюра Микола Ігорович**
Кандидат технічних наук, доцент, доцент кафедри інформаційних технологій, *orcid.org/0000-0003-4713-7568*
*Київський національний університет будівництва і архітектури, Київ*
**Криворучко Олена Володимирівна**
Доктор технічних наук, професор, завідувач кафедри програмної інженерії та кібербезпеки, *orcid.org/0000-0002-7661-9227*
*Київський національний торговельно-економічний університет, Київ*
**Жирова Тетяна Олександрівна**
Кандидат педагогічних наук, старший викладач кафедри програмної інженерії та кібербезпеки, *orcid.org/0000-0001-8321-6939*
*Київський національний торговельно-економічний університет, Київ*
**Котенко Наталія Олексіївна**
Кандидат педагогічних наук, старший викладач кафедри програмної інженерії та кібербезпеки, *orcid.org/0000-0002-2675-6514*
*Київський національний торговельно-економічний університет, Київ*

# СУЧАСНІ ТЕХНОЛОГІЇ ТЕСТУВАННЯ І ЗАХИСТУ ВЕБ-СТОРІНОК

***Анотація.*** *Використання веб-сторінок практично в усіх сферах сучасного суспільства зумовлює підвищення їхньої якості як з точки зору повноти та зручності контенту, так і з точки зору безпеки. Аналіз сучасних загроз безпеці веб-сторінок дав змогу виділити такі: Injection; Broken Authentication; Sensitive Data Exposure; XML External Entities; Broken Access Control; Security Misconfiguration; Cross-Site Scripting; Insecure Deserialization; Using Components with Known Vulnerabilities; Insufficient Logging & Monitoring. Проаналізовано методи боротьби з наведеними загрозами. Рекомендовано здійснювати перевірку веб-сайтів на вразливість за допомогою таких сканерів: Scan My Server, SUCURI, Qualys SSL Labs, Qualys FreeScan, Quttera, Detectify, SiteGuarding, Web Inspector, Acunetix, Netsparker Cloud, UpGuard Web Scan, Tinfoil Security, Observatory.*

***Ключові слова: веб-сторінка; вразливість веб-сторінок; загрози безпеці веб-сторінок; інструменти для аналізу захищеності веб-сторінок; тестування веб-сторінок***

## Постановка проблеми

Для усіх хто хоч якимось чином причетний до мережі Інтернет, будь то користувач, розробник чи замовник, питання безпеки веб-ресурсів постає досить гостро. Перш ніж визначитися як захищати, потрібно з'ясувати від чого саме необхідно захищати свої веб-сторінки, свої персональні дані та ін. Тобто з'ясувати які види хакерських атак на сьогодні є найпоширенішими та з якою метою вони здійснюються. Володіючи такою інформацією, можна скласти низку заходів для досягнення максимального рівня захищеності веб-ресурсів.

Враховуючи сучасний стан розвитку інформаційних технологій, з'ясуємо за допомогою яких програмних засобів можна перевірити безпечність власних веб-сторінок та безпечність тих веб-ресурсів, якими ми часто користуємося.

## Аналіз останніх досліджень та публікацій

За даними The Web Application Security Consortium найпоширенішими вразливостями веб-додатків є Cross-Site Scripting, Information Leakage, SQL Injection, Insufficient Transport Layer Protection та Fingerprinting HTTP Response Splitting [2].

За класифікацією Google [3] можна виокремити такі типи атак проти сайтів:

– Gibberish Hack – після зламу зловмисники автоматично створюють багато сторінок з неякісними змістом і множинними входженнями ключових фраз; такі сторінки потрапляють в індекс Google; при спробах переходу на подібні веб-ресурси, користувачі перенаправляються на інші веб-сторінки, які не мають відношення до інформації, представленої в пошуку.

 

– Japanese Keywords Hack – після атаки інтернет-злодії генерують множинні сторінки з іменами каталогів і унікальними текстами; за рахунок проставляння посилань на партнерські інтернет-магазини з підробленими товарами бренду здійснюється монетизація цих сторінок; іноді облікові записи хакерів додаються в Search Console як власників сайтів.

– Cloaked Keywords Hack – на зламаному сайті використовується шкідлива технологія, за допомогою якої маскується неякісний контент або формуються типові сторінки з помилкою 404; на таких ресурсах у величезній кількості створюються додаткові сторінки, де в текстах присутні приховані ключові фрази і приховані посилання; самі тексти позбавлені будь-якого сенсу; ці сторінки іноді містять основні елементи шаблону з оригінального сайту, тому на перший погляд вони можуть виглядати як звичайні частини цільового сайту.

Низка літературних джерел [1; 10 – 12] наводить такі типи веб-атак: Injection, Cross-Site Scripting (XSS), Broken Authentication and Session Management, Insecure Direct Object References, Cross-Site Request Forgery, Security Misconfiguration, Insecure Cryptographic Storage, Failure to Restrict URL Access, Insufficient Transport Layer Protection, Unvalidated Redirects and Forwards.

## Мета статті

Очевидно, що володіння інформацією щодо загроз в інтернет-просторі та методів захисту від цих загроз сприяє покращенню якості веб-сторінок з точки зору безпечності. Основною метою статті є перелік найпоширеніших на сьогодні загроз безпеці веб-сторінок, коротке роз'яснення шляхів впливу їх на безпечну роботу веб-сторінок; добір методів та інструментів перевірки веб-сторінок на захищеність і демонстрація їх роботи.

## Виклад основного матеріалу

Всесвітньою неприбутковою благодійною організацією, орієнтованою на підвищення безпеки програмного забезпечення OWASP (Open Web Application Security Project) проаналізовано сотні тисяч випадків вразливостей веб-додатків та складено список найпоширеніших загроз [1].

1. Injection. Зловмисник може керувати веб-додатком, змінювати команди, що подаються до його підсистем, надсилати деформовані запити. Найбільш відомими серед цих атак є SQL Injection. З їх допомогою зловмисник може увійти в систему як адміністратор, навіть не володіючи паролем. Використовуючи цю інформацію зловмисник може викрасти секретну інформацію або гроші, змінити або стерти дані. Окрім SQL Injection відомі LDAP Injection, XPath Injection, Command Injection, SMTP Injection. Аномальні дані можуть спровокувати виконання ненавмисних команд інтерпретатором.

2. Broken Authentication. Більшість додатків вимагають від користувачів аутентифікації з використанням логіну та пароля, тобто перш ніж використовувати ресурс користувач має ввести свій логін та пароль. У такій системі аутентифікації є низка недоліків і зловмисники цим користуються. Зловмисник, який зміг вгадати дійсний пароль, зможе виконати будь-які дії від імені користувача.

3. Sensitive Data Exposure. Секретні дані зазвичай повинні бути захищені шифруванням та іншими криптографічними алгоритмами. Однак дуже часто шифруванням нехтують або виконують його не у повному обсязі, що дає можливість зловмисникам захопити конфіденційну інформацію, яку вони не повинні мати (паролі, кредитні картки, особисту інформацію та інші критичні для бізнесу дані). До загальних недоліків належить використання слабких ключів шифрування, неправильне використання протоколу, недотримання стандартних алгоритмів та протоколів шифрування.

4. XML External Entities (XXE). Досить часто програми мають отримувати та обробляти XML-документи від користувачів. Старі або погано налаштовані XML-парсери можуть увімкнути функцію XML, що називається зовнішнім посиланнями на об'єкти в межах документу XML, та вбудувати вміст іншого файлу. Зловмисники можуть скористатися цим способом для читання конфіденційних даних, доступу до внутрішніх систем і навіть вимкнення програми.

5. Broken Access Control. Більшість веб-додатків обмежують те, що користувачі можуть бачити або робити, незалежно від того чи отримують вони доступ до особистих даних чи до даних іншого користувача. Проте досить часто механізми контролю доступу мають суттєві недоліки. Зловмисник може обійти ці засоби контролю або зловживати ними для доступу до несанкціонованих даних та функцій, таких як доступ до облікових записів інших користувачів, перегляду конфіденційних файлів, зміни даних інших користувачів, виконання адміністративних дій та інше.

6. Security Misconfiguration. Сервери та програми містять багато складових, які потрібно правильно налаштовувати. Це стосується всіх рівнів стека програм, від операційної системи і мережевих пристроїв до веб-сервера і самої програми. Налаштування за замовчуванням неповні або спеціальні конфігурації можуть залишати файли незахищеними (наприклад, увімкнено паролі за замовчуванням, відкриті хмарні служби, а також численні інші небезпечні налаштування, які можуть дозволити зловмисникові отримати доступ до системи або даних).

7. Cross-Site Scripting (XSS). Використовуючи XSS, зловмисник може змінювати веб-сторінки, які інші користувачі бачать у вашій програмі, будь то крадіжка інформації, такої як паролі та кредитні картки, розповсюдження фіктивних даних, захоплення сесій користувача, перенаправлення на інший сайт або виконання шкідливих скриптів у браузері жертви. Ця вразливість може виникати, коли ненадійні дані включаються до веб-сторінки або до відповіді без належної перевірки. Зловмисник може подавати форми з фрагментами HTML або JavaScript, які вбудовуватимуться безпосередньо на сторінку та відображатимуться браузером.

8. Insecure Deserialization може дозволити атаки ін'єкцій і ескалацію привілеїв, і навіть призвести до віддаленого виконання коду і захоплення сервера в певних ситуаціях. Багато додатків повинні серіалізувати об'єкти і дані у формат, який можна легко передавати та зберігати. Коли додаток відновлює ці об'єкти назад у пам'ять шляхом десеріалізації форматованих даних, отриманих від користувача, може бути здійснено втручання в пам'ять об'єкта, або навіть виконання з його допомогою довільних функцій.

9. Using Components with Known Vulnerabilities. Сучасне програмне забезпечення більше не є монолітним – воно завжди складається з все більшої кількості компонентів, фреймворків і бібліотек з відкритим кодом. Будь-які відомі вразливості, виявлені в цих залежностях, можуть безпосередньо впливати на програму. Іноді це призводить до інших вразливостей, які ми розглянули вище, таких як ін'єкція, віддалене виконання коду або будь-який інший недолік, який може дозволити зловмисникам отримати доступ до конфіденційних даних або дій.

10. Insufficient Logging & Monitoring. Незважаючи на низку заходів зі створення абсолютно захищеної системи від усіх можливих нападів, потрібно розуміти, що деякі атаки пройдуть через оборону. Стійка система захисту має складатися з кількох шарів. Це дасть можливість виявити атаки та швидко їх подолати.

Для захисту веб-ресурсів OWASP рекомендує тестувати їх протягом всього життєвого циклу розробки програмного забезпечення, а також виділяє при цьому такі основні складові:

1. Understand the Threat Model. Перш ніж почати тестування, потрібно розставити пріоритети, визначити на що важливо витрачати час. Необхідно розробити модель загроз перед тестуванням – це дасть можливість правильно розставити пріоритети.

2. Understand Your SDLC. Підхід до тестування безпеки додатків має бути сумісним з людьми, процесами та інструментами, які використовуються у життєвому циклі розроблення програмного забезпечення (SDLC). Придумування зайвих чи додаткових кроків здебільшого до добра не доводить. Необхідно використовувати усі можливості для збирання інформації про безпеку та активно її використовувати.

3. Testing Strategies. Потрібно обрати найпростішу, найшвидшу та найточнішу техніку для перевірки кожної вимоги. Врахування людських ресурсів, необхідне для боротьби з помилковими позитивними наслідками використання автоматизованих інструментів.

4. Achieving Coverage and Accuracy. Не потрібно починати все тестувати одразу. Слід зосередитися на тому, що важливо, і розширювати програму перевірки з плином часу. Під розширенням програми перевірки розуміють розширення набору захисних засобів безпеки та ризиків, які автоматично перевіряються, а також розширення набору програм і API, які охоплюються. Мета полягає в тому, щоб досягти стану, в якому здійснюється безперервна перевірка основної безпеки всіх програм і API.

5. Clearly Communicate Findings. Про результати тестування потрібно повідомляти чітко та ефективно, у зручній та зрозумілій формі. Для здійснення перевірки веб-сайтів на вразливість є сенс сканувати їх за допомогою таких безкоштовних інструментів: Scan My Server, SUCURI, Qualys SSL Labs, Qualys FreeScan, Quttera, Detectify, SiteGuarding, Web Inspector, Acunetix, Netsparker Cloud, UpGuard Web Scan, Tinfoil Security, Observatory та ін [8].

Наприклад, з допомогою SUCURI протестуємо безпечність сайтів Amazon та «Київські енергетичні послуги», результати тестування продемонстровано на рис. 1 та 2 відповідно.

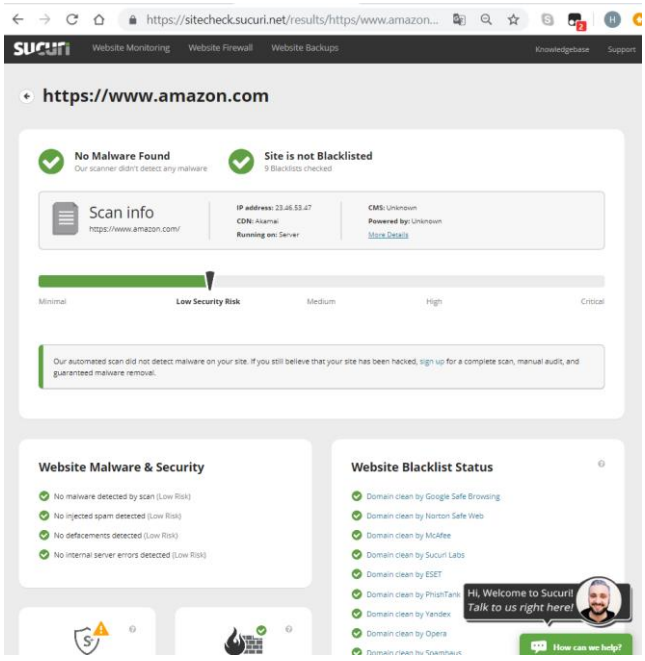


*Рисунок 1 – Результати сканування сайту Amazon з допомогою SUCURI*

Легко бачити, що сайт Amazon є цілком безпечним, а розробникам сайту «Київські енергетичні послуги» треба вжити заходів для підвищення безпечності.

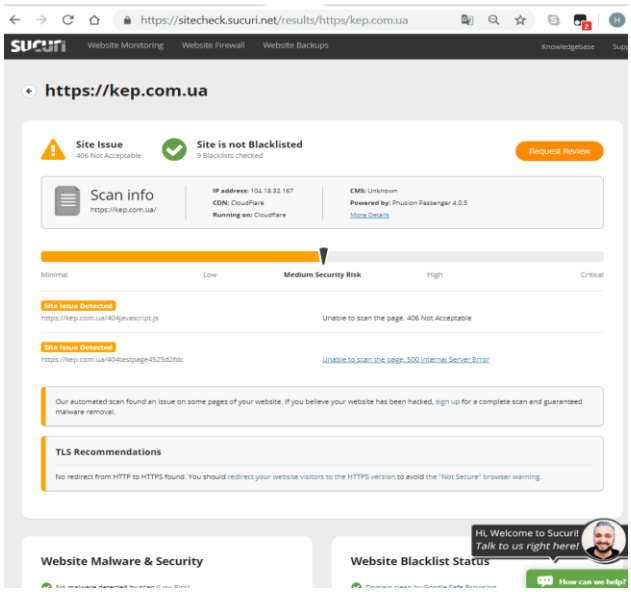


*Рисунок 2 – Результати сканування сайту «Київські енергетичні послуги» за допомогою SUCURI*

Всі наведені сервіси мають різну швидкість тестування веб-сайтів та видають результати за різними критеріями, тому треба здійснювати тестування використовуючи кілька з них.

## Висновки

При створені веб-додатків необхідно дотримуватися низки елементарних правил, таких як: підбір та правильне користування паролями; вибір безпечного хостинг-провайдера; регулярне резервне копіювання; створення різних баз даних для кожного сайту; використовувати HTTPS / TLS для шифрування даних; правильно підключати сервер під час налаштування сайту; не використовувати налаштування за замовчуванням; вимикати функції, які не використовуються; постійний моніторинг змін вмісту; своєчасне оновлення сайту відповідно до оновлення програмного забезпечення, на якому він працює; створення багатошарового захисту.

Для захисту веб-додатків від хакерських атак необхідно: використовувати інструменти для аналізу захищеності (автоматизовані засоби), які виконують тести на проникнення, тобто намагаються зламати веб-сторінку за допомогою, наприклад, SQL-інєкції; потрібно захищати призначені для користувача дані за допомогою HTTPS; періодично та вчасно оновлювати програмне забезпечення; попереджувати міжсайтовий скриптинг; перевіряти та шифрувати паролі; контролювати процес завантаження файлів; стежити за повідомленнями про помилку; перевіряти вхідні дані; розподіляти права доступу до файлів.

## Список літератури

1. *OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf*

2. *Web Application Security Statistics. Режим доступу: http://projects.webappsec.org/w/page/13246989/Web%20Application%20Security%20Statistics*

3. *State of Website Security in 2016. Режим доступу: https://webmasters.googleblog.com/2017/03/nohacked-year-in-review.html*

4. *LeBlanc J., Messerschmidt T. Identity and Data Security for Web Development: Best Practices 1st Edition O'Reilly Media; (June 6, 2016) 204 p.*

5. *Khawaja G., Practical Web Penetration Testing: Secure web applications using Burp Suite, Nmap, Metasploit, and more Kindle Edition, Packt Publishing; 1 edition (June 22, 2018) 294 p.*

6. *Marshall J., Hands-On Bug Hunting for Penetration Testers: A practical guide to help ethical hackers discover web application security flaws Packt Publishing - ebooks Account (September 12, 2018) 373 p*

7. *Brooks C., Grow C., Craig P., Short D., Cybersecurity Essentials 1st Edition, Sybex; (October 5, 2018), 784 p.*

8. *12 Online Free Tools to Scan Website Security Vulnerabilities & Malware Режим доступу: https://geekflare.com/online-scan-website-security-vulnerabilities/#SUCURI*

9. *Жирова Т.О., Котенко Н.О. Проблеми тестування інтерфейсу Web-додатків. Збірник матеріалів IX Міжнародної конференції молодих вчених «Молоді вчені 2018 – від теорії до практики». – Дніпро-Варна : «Дике Поле», 2018. – Стор. 184-188.*

10. *Котенко Н.О., Жирова Т.О. Безпека Web-додатків // Кібербезпека в Україні: правові та організаційні питання : матеріали всеукр. наук.-практ. конф., м. Одеса, 30 листопада 2018 р. – Одеса : ОДУВС, 2018. – Стор. 91-93.*

11. *Sullivan B., Liu V. Web Application Security, A Beginner's Guide 1st Edition, McGraw-Hill Education; 1 edition (November 24, 2011) 352 p.*

12. *Shema M., Hacking Web Apps: Detecting and Preventing Web Application Security Problems 1st Edition, Syngress; 1 edition (September 12, 2012), 296 p.*

*Стаття надійшла до редколегії 05.09.2019*

**Цюцюра Николай Игоревич**
Кандидат технических наук, доцент, доцент кафедры информационных технологий, *orcid.org/0000-0003-4713-7568*
*Киевский национальный университет строительства и архитектуры, Киев*
**Криворучко Елена Владимировна**
Доктор технических наук, профессор, заведующий кафедрой программной инженерии и кибербезопасности, *orcid.org/0000-0002-7661-9227*
*Киевский национальный торгово-экономический университет, Киев*
**Жировая Татьяна Александровна**
Кандидат педагогических наук, старший преподаватель кафедры программной инженерии и кибербезопасности, *orcid.org/0000-0001-8321-6939*
*Киевский национальный торгово-экономический университет, Киев*
**Котенко Наталья Алексеевна**
Кандидат педагогических наук, старший преподаватель кафедры программной инженерии и кибербезопасности, *orcid.org/0000-0002-2675-6514*
*Киевский национальный торгово-экономический университет, Киев*

## СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ТЕСТИРОВАНИЯ И ЗАЩИТЫ ВЕБ-СТРАНИЦ

***Аннотация.*** *Использование веб-страниц практически во всех сферах современного общества приводит к повышению их качества как с точки зрения полноты и удобства контента, так и с точки зрения безопасности. Анализ безопасности веб-страниц позволил выделить следующие угрозы: Injection; Broken Authentication; Sensitive Data Exposure; XML External Entities; Broken Access Control; Security Misconfiguration; Cross-Site Scripting; Insecure Deserialization; Using Components with Known Vulnerabilities; Insufficient Logging & Monitoring. Проанализированы методы борьбы с перечисленными угрозами. Рекомендуется осуществлять проверку веб-сайтов на уязвимость с помощью таких сканеров: Scan My Server, SUCURI, Qualys SSL Labs, Qualys FreeScan, Quttera, Detectify, SiteGuarding, Web Inspector, Acunetix, Netsparker Cloud, UpGuard Web Scan, Tinfoil Security, Observatory.*

***Ключевые слова: Веб-страница; уязвимости Веб-страниц; угрозы безопасности Веб-страниц; инструменты для анализа защищенности Веб-страниц; тестирование Веб-страниц***

**Tsiutsiura Mykola**
PhD (Eng.), Associate Professor, Department of Information Technology, *orcid.org/0000-0003-4713-7568*
*Kyiv National University of Construction and Architecture, Kyiv*
**Kryvoruchko Olena**
DSc (Eng.), Professor, Head of the Department of Software Engineering and Cyber Security, *orcid.org/0000-0002-7661-9227*
*Kyiv National University of Trade and Economics, Kyiv*
**Zhyrova Tetiana**
Candidate of Sciences in Pedagogy, Senior Lecturer of Program Engineering and Cybersecurity Department, *orcid.org/0000-0001-8321-6939*
*Kyiv National University of Trade and Economics, Kiev*
**Kotenko Nataliia**
Candidate of Sciences in Pedagogy. Senior Lecturer of Program Engineering and Cybersecurity Department, *orcid.org/0000-0002-2675-6514*
*Kyiv National University of Trade and Economics, Kiev*

## MODERN TECHNOLOGIES FOR WEB PAGE TESTING AND PROTECTION

***Abstract.*** *The use of web-pages in almost all spheres of modern society results in the need for improvement of their quality both in terms of content completeness and convenience and security. An analysis of current threats to the security of web-pages allowed to distinguish the following: Injection; Broken Authentication; Sensitive Data Exposure; XML External Entities; Broken Access Control; Security Misconfiguration; Cross-Site Scripting; Insecure Deserialization; Using Components with Known Vulnerabilities; Insufficient Logging & Monitoring. Methods to prevent the listed threats have been analyzed. It is recommended to scan websites for vulnerabilities with such scaners as Scan My Server, SUCURI, Qualys SSL Labs, Qualys FreeScan, Quttera, Detectify, SiteGuarding, Web Inspector, Acunetix, Netsparker Cloud, UpGuard Web Scan, Tinfoil Security, Observatory.*

***Keywords: Web-page; Web-pages vulnerabilities; security threats to Web-pages; tools for analyzing security of Web-pages, testing of Web-pages***

**References**

1. *OWASP Top 10 – 2017 The Ten Most Critical Web Application Security Risks [electronic source] https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf*
2. *Web Application Security Statistics. [electronic source] http://projects.webappsec.org/w/page/13246989/Web%20Application%20Security%20Statistics*
3. *State of Website Security in 2016. Режим доступу: https://webmasters.googleblog.com/2017/03/nohacked-year-in-review.html*
4. *LeBlanc, J., & Messerschmidt, T., (2016). Identity and Data Security for Web Development: Best Practices 1st Edition O'Reilly Media; (June 6) 204 p.*
5. *Khawaja, G. Practical Web Penetration Testing: Secure web applications using Burp Suite, Nmap, Metasploit, and more Kindle Edition, Packt Publishing; 1 edition (June 22, 2018) 294 p.*
6. *Marshall, J. Hands-On Bug Hunting for Penetration Testers: A practical guide to help ethical hackers discover web application security flaws Packt Publishing - ebooks Account (September 12, 2018) 373 p*
7. *Brooks, C., Grow, C., Craig, P., & Short, D., (2018). Cybersecurity Essentials 1st Edition, Sybex; (October 5), 784 p.*
8. *12 Online Free Tools to Scan Website Security Vulnerabilities & Malware [electronic source] https://geekflare.com/online-scan-website-security-vulnerabilities/#SUCURI*
9. *Zhyrova, T., & Kotenko, N., (2018). Problems testing the interface of Web-based applications. Collection of materials of the IX International Conference of Young Scientists "Young Scientists of 2018 - From Theory to Practice". – Dnipro-Varna: "Wild Field". – P. 184 – 188.*
10. *Zhyrova, T., & Kotenko, N., (2018). Web-Application Security // Cyber Security in Ukraine: Legal and Organizational Issues: All-In-One Materials. sci. pract. Conf., Odessa, November 30. – ODUPS, 2018. – P. 91 – 93.*
11. *Sullivan B., Liu V. Web Application Security, A Beginner's Guide 1st Edition, McGraw-Hill Education; 1 edition (November 24, 2011) 352 p.*
12. *Shema M., Hacking Web Apps: Detecting and Preventing Web Application Security Problems 1st Edition, Syngress; 1 edition (September 12, 2012), 296 p.*

---

**Посилання на публікацію**

APA *Tsiutsiura, Mykola, Kryvoruchko, Olena, Zhyrova, Tetiana, & Kotenko, Nataliia, (2019). Modern technologies for Web-page testing and protection. Management of Development of Complex Systems, 39, 100 – 105; dx.doi.org\10.6084/m9.figshare.11340671.*

ДСТУ *Цюцюра М.І. Сучасні технології тестування і захисту Веб-сторінок [Текст] / М.І. Цюцюра, О.В. Криворучко, Т.О. Жирова, Н.О. Котенко // Управління розвитком складних систем. – 2019. – № 39. – С. 100 – 105; dx.doi.org\10.6084/m9.figshare.11340671.*