УДК 004.41:004.056

О. В. Коваленко, *к.т.н., доцент*
e-mail: clashav@gmail.com
Центральноукраїнський національний технічний університет,
пр. Університетський, 8, м. Кропивницький, 25006, Україна

МАТЕМАТИЧНІ МОДЕЛІ ТЕХНОЛОГІЇ ТЕСТУВАННЯ DOM XSS ВРАЗЛИВОСТІ ТА ВРАЗЛИВОСТІ ДО SQL ІН'ЄКЦІЙ

В роботі розроблено математичні моделі технології тестування комплексу DOM XSS вразливостей і технології тестування вразливості до SQL ін'єкцій В основу математичного моделювання покладено підхід GERT-мережевого синтезу. Математична модель технології тестування комплексу DOM XSS вразливостей відрізняється від відомих урахуванням специфіки комплексного аналізу різних типів XSS вразливості («stored XSS», «reflected XSS» і DOM Based XSS), а також включенням в алгоритм процедур автоматичного аудиту DOM Based XSS окремо. Це дає можливість провести аналітичне оцінювання часових витрат тестування зазначених вразливостей в умовах реалізації стратегії розробки безпечного програмного забезпечення. Математична модель технології тестування вразливості до SQL ін'єкцій відрізняється від відомих вдосконаленим способом визначення відстані між результатами ін'єкції. Використання в запропонованому способі критерію Джаро–Вінклера для порівняння результатів ін'єкції SQL коду і введення граничного значення дозволить підвищити точність результатів тестування безпеки програмного забезпечення.

Ключові слова: *технології тестування комплексу DOM XSS, SQL ін'єкції, GERT-мережа, процес тестування безпеки Web-додатків.*

Збільшення кількості користувачів всесвітньої мережі Інтернет, постійне зростання інформаційного, фінансового і ділового контенту в кіберпросторі обумовлює підвищення попиту на Web-додатки. У той же час цей процес викликає зворотну негативну реакцію з боку зловмисників, що мають постійну можливість аналізу об'єктивно існуючих вразливостей Інтернет-додатків.

Аналіз різного роду статистичних матеріалів відомих організацій показав, що одним із найбільш небезпечних видів атак (вразливостей) є міжсайтовий скриптинг – XSS (Cross Site Scripting).

Аналіз літератури [1–4] показав, що міжсайтовий скриптинг – це помилка валідації призначених для користувача даних, яка дозволяє передати JavaScript код на виконання в браузер користувача. У широких колах фахівців такі атаки часто також називають HTML ін'єкціями, тому що механізм їх впровадження дуже схожий з SQL-ін'єкціями, але, на відміну від останніх, впроваджуваний код виповнюється в браузері користувача.

З робіт [3, 4] відомо, що під XSS зазвичай мається на увазі моментальний і відкладений міжсайтовий скриптинг. При моментальному XSS зі шкідливим кодом (Javascript) повертається атакується сервером негайно як відповідь на HTTP запит. Відкладений XSS означає, що ця шкідлива програма зберігається на атакованій системі і пізніше може бути впроваджена в HTML сторінку вразливої системи. Така класифікація передбачає, що фундаментальна властивість XSS полягає в тому, що ця шкідлива програма відсилається з браузера на сервер і повертається в цей же браузер (моментальний XSS) або будь-який інший браузер (відкладений XSS). Існують механізми виникнення подібного роду загроз, а також шляхи можливого їх блокування. Однак, щоб ідентифікувати ці загрози і можливі наслідки їх поширення в процесі безпечного управління IT-проектами, а також запропонувати оптимальні шляхи вирішення цієї проблеми, існує необхідність математичної формалізації процесу їх ініціалізації і поширення.

У ряді робіт здійснювалися спроби математичної формалізації процесу пошуку і усунення вразливостей подібного роду. Так, у роботах [1, 5, 6] представлені узагальнені матеріали механізмів і процедур безпечного програмування, які мають на меті зниження ризиків уразливості. У роботах [7–9] представлені математичні моделі, які описують алгоритми аналізу Web-додатків (у тому числі й алгоритм однієї з найбільш поширених уразливостей – DOM (Document Object Model) XSS уразливості). Однак представлені моделі не враховують останні тенденції XSS уразливості, а саме відмінність їх типів («stored XSS», «reflected XSS» і DOM Based XSS) і необхідність їх виявлення.

Саме тому особливо актуальним завданням у цьому напрямку є моделювання алгоритму виявлення DOM XSS уразливості з урахуванням комплексу трьох їх можливих типів. Також на основі аналізу методології тестування вразливості Web-додатків до DOM XSS можна розробити алгоритм аналізу вразливості Web-додатків до SQL ін'єкцій

Метою роботи є розробка математичних моделей технології тестування DOM XSS уразливості та вразливості до SQL ін'єкцій.

Проведені дослідження показали, що вразливість DOM XSS є підвидом XSS, в разі якої результат атаки знаходиться не у відповіді сервера і, відповідно, не в HTML коді, а в DOM структурі HTML сторінки. При цьому в режимі «stored XSS» здійснюється передача і зберігання XSS на сервері. Надалі на цю сторінку перенаправляються користувачі. У режимі «reflected» XSS повертається в тілі відповіді від сервера на конкретний запит із самої XSS. Результати атак за допомогою таких уразливостей можна виявити тільки в процесі виконання або аналізі DOM структури. Сам механізм атаки, а саме ін'єкція Javascript коду у вразливий сегмент, залишається незмінним.

Слід зауважити, що одним із найменш математично формалізованих і досліджуваних типів XSS є DOM Based XSS. Можливо, це пов'язано з тим, що навіть сучасними сканерами їх не часто можна виявити і відповідно представити чіткий алгоритм виконання операцій аналізу вразливості.

Для математичної формалізації алгоритму виявлення комплексу DOM XSS уразливостей різних типів скористаємося основними положеннями мережевого GERT-моделювання, докладно описаними в роботах [5, 7].

Основні етапи алгоритму виявлення комплексу DOM XSS уразливостей можна описати таким чином:

1) З коду аналізованої сторінки витягуються всі теги <script> і формується список тегів для аналізу.

2) Виконується аналіз вмісту тега. При цьому, якщо теги не містять код, а посилаються на віддалений файл, виконується звернення до файлу та отримання коду з нього. У вмісті файлу знаходяться потенційні небезпечні ділянки коду (sink), які використовують вхідні дані клієнта (source).

3) Якщо в коді тега використовується source, виконується атака з певним маркером, який можна відстежити в DOM структурі сторінки після виконання коду (наприклад ін'єкція певного текстового вмісту в DOM).

4) Виконується перевірка вмісту DOM. Якщо в результаті атаки маркер знаходиться в DOM, можна зробити висновок про наявність DOM уразливості.

5) Після впровадження даних вручну і аналізу результатів виконуваності на перших чотирьох етапах виконується аудит коду (може бути здійснений дистанційно).

6) Кроки 2–5 виконуються для кожного тега <script> на сторінці.

Для побудови формальної моделі алгоритму виявлення комплексу DOM XSS уразливостей обрано стохастичну GERT-мережу.

Мережі GERT складаються з вузлів типу AND, INCLUSIVE-OR і EXCLUSIVE-OR і гілок з двома й більше параметрами. Гілка має напрямок, вузол початку і вузол кінця. Параметри гілки містять:

1) ймовірність проходження гілки (P_a) за умови, що вузол, який є джерелом гілки, був реалізований;

2) час (t_a) проходження гілки, якщо вона буде реалізована.

Час t_a може бути випадковою величиною. Якщо гілка не є частиною реалізації мережі, тобто під час виконання процесу активність, пов'язана з гілкою, не відбувається, то $t_a = 0$.

Вузол у стохастичній мережі GERT складається з функції входу (контрибутивної

функції) і функції виходу (дистрибутивної функції). Кожна з функцій описується певним логічним відношенням щодо пов'язаних гілок.

В цілому, проведені дослідження показали, що GERT-моделювання є ефективним способом визначення заздалегідь невідомих законів і функцій розподілу випадкових величин при відомому алгоритмі функціонування (процесу). Саме тому як інструмент математичного моделювання нами було вибрано GERT-моделювання.

У розглянутій мережі вузли графа інтерпретуються станами комп'ютерної системи в процесі функціонування DOM структури, а гілки графа – ймовірнісно-часовими характеристиками переходів між станами. Характеристики гілок моделі представлені в табл. 1.

Таблиця 1

Характеристики гілок моделі

№ п/п	Гілка	W-функція	Ймовірність	Твірна функція моментів
1	(1, 2)	W_{12}	p1	$\lambda_1/(\lambda_1-s)$
2	(2, 3)	W_{23}	p2	$\lambda_2/(\lambda_2-s)$
3	(2, 4)	W_{24}	p3	$\lambda_3/(\lambda_3-s)$
4	(3,5)	W_{35}	p2	$\lambda_2/(\lambda_2-s)$
5	(5,6)	W_{56}	p4	$\lambda_4/(\lambda_4-s)$
6	(6,1)	W_{51}	1 – p4	$\lambda_5/(\lambda_5-s)$
7	(4,2)	W_{42}	p3	$\lambda_3/(\lambda_3-s)$
8	(5,7)	W42	p4	$\lambda_4/(\lambda_4-s)$

Еквівалентна W-функція часу виконання алгоритму тестування комплексу DOM XSS різних типів (у тому числі DOM Based XSS) вразливостей дорівнює:

$$W_E(s)=\frac{W_{12}W_{23}W_{35}W_{56}+W_{12}W_{24}W_{42}W_{23}W_{35}W_{57}}{1-W_{12}W_{23}W_{35}W_{51}-W_{12}W_{24}W_{42}W_{23}W_{35}W_{56}W_{61}}=$$
$$=\frac{p_1p_2^2\lambda_1\lambda_2^2\left(p_4\lambda_4(\lambda_3-s)^2(\lambda_5-s)+p_3^2q_1\lambda_3^2\lambda_5(\lambda_4-s)\right)}{(\lambda_4-s)\begin{pmatrix}(\lambda_1-s)(\lambda_2-s)^2(\lambda_3-s)^2(\lambda_5-s)-\\-p_1\lambda_1p_2^2\lambda_2^2q_1\lambda_5(\lambda_3-s)^2-p_1p_2^2p_3^2p_4\lambda_1\lambda_2^2q_1\lambda_3^2\lambda_4\lambda_5\end{pmatrix}}, \quad (1)$$

де $1-p_4=q_1$.

Особливість цього процесу полягає в різнорідності аналізованих і оброблюваних даних. При цьому можливі різні випадки організації зворотного зв'язку.

Для GERT-мереж з циклами не існує простих методів знаходження особливих точок функції $\Phi_E(z)$ заміни дійсних змінних $(z=-i\varsigma)$, де ς – дійсна змінна. Це пояснюється тим, що для знаходження особливих точок необхідно вирішувати нелінійні рівняння, і чим складніша структура GERT-мережі, тим складніше і вихідне рівняння. Тому в ході моделювання пропонується вдатися до подібної заміни. Виконуючи комплексне перетворення $z=-s$, отримаємо:

$$\Phi(z)=\frac{uz^3+vz^2+bz+k}{(\lambda_4+z)\left(z^6+cz^5+dz^4+gz^3+hz^2+wz+m\right)}, \quad (2)$$

де $u=-p_1p_2^2p_4\lambda_1\lambda_2^2\lambda_4$,

$v=p_1p_2^2p_4\lambda_1\lambda_2^2\lambda_4(\lambda_5+2\lambda_3)$,

$b=-p_1p_2^2p_4\lambda_1\lambda_2^2\lambda_4\lambda_3(2\lambda_5-\lambda_3)$,

$k=-p_1p_2^2\lambda_1\lambda_2^2\lambda_3^2\lambda_4\lambda_5\left(p_4+p_3^2q_1\right)$,

$c=\lambda_1+2\lambda_2+2\lambda_3+\lambda_4+\lambda_5$,

$$d=-\begin{pmatrix}2\lambda_3\lambda_5\lambda_4+\lambda_1\lambda_5\lambda_4+2\lambda_2\lambda_5\lambda_4+\\+\lambda_3^2+2\lambda_1\lambda_3+4\lambda_2\lambda_3+2\lambda_1\lambda_2+\lambda_2^2\end{pmatrix},$$

$$g=\begin{pmatrix}\lambda_3^2\lambda_4\lambda_5+4\lambda_1\lambda_2\lambda_4\lambda_5+4\lambda_2\lambda_3\lambda_4\lambda_5+\\\lambda_2^2+\lambda_3^2\lambda_1+2\lambda_3^2\lambda_2+4\lambda_1\lambda_2\lambda_3\lambda_4+\\+2\lambda_2^2\lambda_3+\lambda_2^2\lambda_1+\lambda_3^2\lambda_4+\lambda_2^2\lambda_4\end{pmatrix},$$

$$h=-\begin{pmatrix}\lambda_1\lambda_3^2\lambda_4\lambda_5+2\lambda_2\lambda_3^2\lambda_4\lambda_5+\\+4\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5+2\lambda_2^2\lambda_3\lambda_4\lambda_5+\\+\lambda_2^2\lambda_3^2\lambda_4+2\lambda_1\lambda_2^2\lambda_3\lambda_4-\\-p_1p_2^2p\ q_1\lambda_1\lambda_2\lambda_4\lambda_5\end{pmatrix},$$

$$w=\lambda_1\lambda_2\lambda_3^2\lambda_4\lambda_5+\lambda_2^2\lambda_3^2\lambda_4\lambda_5+$$
$$+2\lambda_1\lambda_2^2\lambda_3\lambda_4\lambda_5+\lambda_1\lambda_2^2\lambda_4\lambda_3-\ ,$$
$$-2p_1p_2^2p_4q_1\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5$$
$$m=p_1p_2^2p_4q_1\lambda_1\lambda_2\lambda_3^2\lambda_4\lambda_5+$$
$$+p_1p_2^2p_3p_4q_1\lambda_1\lambda_2^2\lambda_3^2\lambda_4\lambda_5-\lambda_1\lambda_2^2\lambda_3\lambda_4\lambda_5.$$

Щільність розподілу ймовірностей часу виконання алгоритму аналізу DOM XSS вразливості

$$\varphi(x)=\frac{1}{2\pi i}\int_{-i\infty}^{i\infty} e^{zx}\frac{uz^3+vz^2+bz+k}{\left(z^6+cz^5+dz^4+gz^3+hz^2+wz+m\right)}dz, \quad (3)$$

де операція інтегрування виконується за допомогою інтегралу Бромвича–Вагнера [5].

Спосіб інтегрування залежить від того, чи має функція $\Phi(z)$ лише прості си, чи полюси деякого порядку.

У тому випадку, коли функція $\Phi(z)$ має лише прості полюси, вираз $e^{zx}\Phi(z)$ можна представити у вигляді

$$e^{zx}\Phi(z)=\frac{e^{zx}\left(uz^3+vz^2+bz+k\right)}{z^7+\gamma_6 z^6+\gamma_5 z^5+\gamma_4 z^4+\gamma_3 z^3+\gamma_2 z^2+\gamma_1 z+\gamma_0}=\frac{\mu(z)}{\psi(z)}, \quad (4)$$

де $\gamma_6=c$, $\gamma_5=c+d$, $\gamma_4=d+g$, $\gamma_3=g+h$, $\gamma_2=h+w$, $\gamma_1=w+m$, $\gamma_0=m$.

Тоді щільність розподілу часу виконання алгоритму аналізу DOM XSS вразливості всіх типів дорівнює

$$\varphi(x)=\sum_{k=1}^{7}\operatorname{Res}\left[e^{zx}\Phi(z)\right]=\sum_{k=1}^{7}\frac{\mu(z_k)}{\psi(z_k)}=\sum_{k=1}^{7}\frac{e^{zx}\left(uz^3+vz^2+bz+k\right)}{7z_k^6+6\gamma_6 z_k^5+5\gamma_5 z_k^4+4\gamma_4 z_k^3+3\gamma_3 z_k^2+2\gamma_2 z_k+\gamma_1}. \quad (5)$$

Функція $\Phi(z)$, крім рішень, які визначаються корінням рівняння $z^6+cz^5+dz^4+gz^3+hz^2+wz+m=0$, може мати і полюс другого або третього порядку. Тоді щільність розподілу часу передачі повідомлення $\varphi(x)$ знаходиться за формулою знаходження відрахувань r_{-1} від полюсів z_k порядку n:

$$r_{-1}=\frac{1}{(n-1)!}\lim_{z\to z_k}\frac{d^{n-1}\left((z-z_k)^n e^{zx}\Phi(z)\right)}{dz^{n-1}}. \quad (6)$$

Вираз (6) являє собою дробово-раціональну функцію відносно z зі ступенем знаменника більшим, ніж ступінь чисельника. Тому для нього виконується умова леми Жордана [5].

Многочлен $z^6+cz^5+dz^4+gz^3+hz^2+wz+m$ породжує сім полюсів. Рішення рівняння $z^6+cz^5+dz^4+gz^3+hz^2+wz+m=0$ може бути знайдено будь-яким методом, наприклад, за формулами Вієта [5]. В результаті обчислюються особливі точки z_1, z_2, z_3, z_4, z_5, z_6.

Таким чином, на основі експоненційної GERT-мережі розроблено математичну модель технології тестування комплексу DOM XSS уразливостей усіх типів («stored XSS», «reflected XSS» і DOM Based XSS), яка відрізняється від відомих урахуванням їх специфіки та необхідності автоматичного аудиту DOM Based XSS окремо.

Розроблена модель може бути використана для дослідження Інтернет-Web-додатків у мережевих структурах, а також при розробці нових засобів і протоколів захисту даних у комп'ютерних системах і мережах.

Застосування експоненційних стохастичних моделей GERT дасть можливість використання результатів, отриманих в аналітичному вигляді (функції, щільності розподілу), для проведення порівняльного аналізу і досліджень більш складних комп'ютерних систем математичними методами.

Відмітною особливістю алгоритму аналізу вразливості Web-додатків до SQL ін'єкцій є облік тільки вразливості, яка є в GET параметрах URL і використовує тільки сліпий метод ін'єкції SQL коду, що використовує особливість використання булевих операторів у SQL запитах (Boolean blind SQL injection) [10].

Етапи алгоритму аналізу вразливості Web-додатка до SQL ін'єкцій можна описати таким чином:

1. З введеного URL посилання виходить список GET параметрів.

2. Виконується перевірка стабільності Web-сторінки. Для цього виконуються два послідовні запити в Web-сторінку і обчислюється відстань між вмістом HTML коду сторін-

ки за допомогою критерію Джаро–Вінклера [5]. Якщо значення критерію менше певного порогового значення, виконувати подальший аналіз неможливо.

3. У параметр GET запиту виконується ін'єкція SQL коду, який не змінює результат запиту до бази даних і зберігається результуючий HTML код.

4. У параметрі GET запиту виконується ін'єкція SQL коду, який змінює результат запиту до бази даних, що призводить або до отримання повного набору даних з таблиці, або до відсутності результату, після чого зберігається результуючий HTML код.

5. За допомогою критерію Джаро–Вінклера виконується порівняння результатів ін'єкції SQL коду. Якщо значення критерію менше певного порогового значення, то в даному GET параметрі є можлива вразливість до SQL ін'єкції.

6. Кроки 2–5 повторюються для всіх параметрів GET запиту наданого URL.

Побудуємо відповідно до представленого описом мережеву GERT-модель технології тестування уразливості до SQL ін'єкцій. Графічне зображення GERT-моделі представлено на рис. 1.

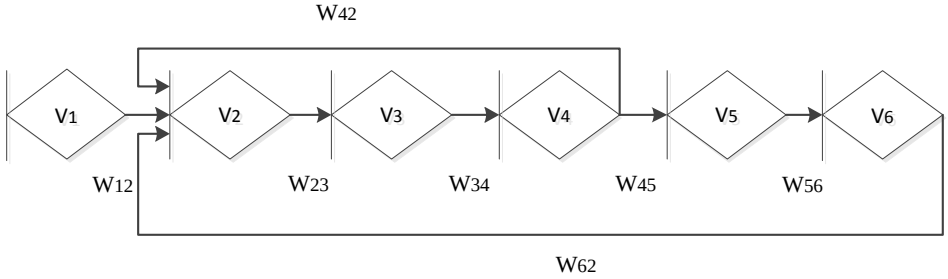

Рис. 1. GERT-модель технології тестування вразливостей до SQL ін’єкцій

У представленій мережі вузли графа інтерпретуються станами комп'ютерної системи в процесі тестування уразливості до SQL ін'єкцій, а гілки графа – ймовірнісно-часовими характеристиками переходів між станами. Зокрема, гілка (1, 2) характеризує час отримання і аналізу GET-параметрів з введеного URL посилання. Гілка (2, 3) відображає час відправлення первинних і вторинних запитів у Web-сторінки. Гілка (3, 4) задає випадковий час порівняння сторінок (час обчислення відстані між вмістом HTML коду сторінки за допомогою критерію Джаро–Вінклера). Гілка (4, 5) характеризує час, за який виконується ін'єкція SQL коду, який не змінює результат запиту до бази даних, а також який змінює результат запиту до бази даних відповідно. Далі гілка (5, 6) характеризує час порівняння результатів ін'єкції SQL коду. Гілка (4, 2) характеризує часові характеристики повернення системи в початковий стан, коли значення критерію Джаро–Вінклера менше певного порогового значення, в той же час гілка (6, 2) відображає часові характеристики переходу до нової перевірки в разі, якщо значення критерію Джаро–Вінклера більше певного порогового значення.

Еквівалентна W-функція часу виконання технології тестування вразливостей до SQL ін’єкцій дорівнює

$$W_E(s)=\frac{W_{12}W_{23}W_{34}W_{45}W_{56}}{1-W_{12}W_{23}W_{34}W_{42}-W_{12}W_{23}W_{34}W_{45}W_{56}W_{62}}=$$

$$=\frac{p_1p_2p_3p_4p_5\lambda_1\lambda_2\lambda_3^2\lambda_4(\lambda_3-s)(\lambda_5-s)(\lambda_6-s)}{\left[\begin{array}{l}(\lambda_1-s)(\lambda_2-s)(\lambda_3-s)^2(\lambda_4-s)(\lambda_5-s)(\lambda_6-s)-\\ -\left(\begin{array}{l}p_1p_2p_3\lambda_1\lambda_2\lambda_3\times\\ \times\left(q_1\lambda_5\left(\lambda_3\lambda_4-\lambda_4s-\lambda_3s-s^2\right)(\lambda_6-s)-p_4p_5p_6\lambda_3\lambda_4\lambda_6(\lambda_5-s)\right)\end{array}\right)\end{array}\right]}, \quad (8)$$

де $1-p_4=q_1$.

Аналогічно алгоритму тестування комплексу DOM XSS різних типів виконується комплексне перетворення $z = -s$.

Щільність розподілу ймовірностей часу виконання технології тестування вразливостей до SQL ін'єкцій дорівнює

$$\varphi(x) = \frac{1}{2\pi i}\int_{-i\infty}^{i\infty} e^{zx}\frac{vz^2 + bz + k}{\left(z^7 + rz^6 + cz^5 + dz^4 + gz^3 + hz^2 + wz + m\right)}dz, \qquad (9)$$

де операція інтегрування виконується за допомогою інтегралу Бромвича–Вагнера [5].

Тоді вираз $e^{zx}\Phi(z)$ можна представити у вигляді

$$\begin{aligned} &e^{zx}\Phi(z) = \\ &= \frac{e^{zx}\left(vz^2 + bz + k\right)}{z^7 + rz^6 + cz^5 + dz^4 + gz^3 + hz^2 + wz + m} = \\ &= \frac{\mu(z)}{\psi(z)} \end{aligned} \qquad (10)$$

Тоді щільність розподілу часу виконання алгоритму тестування вразливості до SQL ін'єкцій дорівнює

$$\varphi(x) = \sum_{k=1}^{7} \operatorname{Res}\left[e^{zx}\Phi(z)\right] = \sum_{k=1}^{7}\frac{\mu(z_k)}{\psi(z_k)} = \sum_{k=1}^{7}\frac{e^{zx}\left(vz^2 + bz + k\right)}{7z_k^6 + 6rz_k^5 + 5cz_k^4 + 4dz_k^3 + 3gz_k^2 + 2hz_k + w}. \qquad (11)$$

Сім полюсів породжує многочлен $rz^6 + cz^5 + dz^4 + gz^3 + hz^2 + wz + m$. Рішення рівняння $rz^6 + cz^5 + dz^4 + gz^3 + hz^2 + wz + m = 0$ може бути знайдено будь-яким методом, наприклад, за формулою Вієта [5]. В результаті обчислюються особливі точки z_1, z_2, z_3, z_4, z_5, z_6, z_7.

Таким чином, на основі експоненційної GERT-мережі розроблено математичну модель технології тестування уразливості до SQL ін'єкцій, яка відрізняється від відомих вдосконаленим способом визначення відстані між результатами ін'єкції. Використання в запропонованому способі критерію Джаро–Вінклера для порівняння результатів ін'єкції SQL коду і введення порогового значення дозволить підвищити точність результатів тестування уразливості до SQL ін'єкцій.

Висновки. У роботі розроблено математичні моделі технології тестування DOM XSS уразливості і технології тестування уразливості до SQL ін'єкцій. В основу математичного моделювання покладено підхід GERT-мережевого синтезу. В результаті розроблено математичну модель процесу тестування Web-додатків.

Математична модель технології тестування комплексу DOM XSS уразливостей відрізняється від відомих урахуванням специфіки комплексного аналізу різних типів XSS уразливості («stored XSS», «reflected XSS» і DOM Based XSS), а також включенням в алгоритм процедур автоматичного аудиту DOM Based XSS окремо. Це дає можливість провести аналітичне оцінювання часових витрат тестування зазначених уразливостей в умовах реалізації стратегії розробки безпечного програмного забезпечення.

Математична модель технології тестування уразливості до SQL ін'єкцій відрізняється від відомих вдосконаленим способом визначення відстані між результатами ін'єкції. Використання в запропонованому способі критерію Джаро–Вінклера для порівняння результатів ін'єкції SQL коду і введення порогового значення дозволить підвищити точність результатів тестування безпеки програмного забезпечення.

Подальші дослідження спрямовані на можливість застосування GERT-моделей технології тестування комплексу DOM XSS уразливостей і технології тестування вразливості до SQL ін'єкцій на реальних прикладах, а також на розробку імітаційної моделі технології тестування безпеки на основі положень теорії масштабування імітаційних моделей.

Список літератури

1. Семенов С. Г., Швачич Г. Г., Карпова Т. П., Волнянський В. В. Застосування багатопроцесорних систем для удосконалення технологічних процесів. *Системи обробки інформації.* 2016. № 3 (140). С. 221–226.
2. Коваленко А. В., Смирнов А. А., Якименко Н. Н., Доренский А. П. Проблемы анализа и оценки рисков информационной деятельности. *Системи обробки інформації*. 2016. № 3 (140). С. 40–42.
3. Spring Framework. URL: http://projects.spring.io/spring-framework/
4. Fowler M. Inversion of control containers and the dependency injection pattern. URL: https://martinfowler.com/articles/injection.html
5. Липаев В. В. Надежность и функциональная безопасность комплексов программ реального времени. Москва, 2013. 176 с.
6. Коваленко А. В., Смирнов А. А. Использование псевдобулевых методов бивалентного программирования для управления рисками разработки программного обеспечения. *Системи управління, навігації та зв'язку.* 2016. № 1 (37). С. 98–103.
7. Ковалев В. П. ГЕРТ-сетевой анализ мультиверсионных архитектур программного обеспечения. *Успехи современного естествознания.* 2011. № 9. С. 161–164.
8. Kovalenko O., Smirnov O., Kovalenko A., Smirnov S., Vialkova V. The mathematical model of the testing technology for DOM XSS vulnerabilities. *Scientific & practical cyber security journal (SPCSJ)*. 2018. Vol. 2. Iss. 1. P. 22–28. URL: https://journal.scsa.ge/issues/2018/03/997
9. Коваленко А. В., Смирнов А. А. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения. *Системи обробки інформації.* 2016. № 5 (142). С. 153–157.
10. Коваленко А. В. Технология тестирования уязвимости к SQL инъекциям. *Системи управління, навігації та зв'язку.* 2017. № 5 (45). С. 66–71.

References

1. Semenov, S. G., Shvachich, G. G., Karpova, T. P., Volnyanskiy, V. V. (2016) Application of multiprocessor systems for the improvement of technological processes. *Systemy obrobky informatsii*, No. 3 (140), pp. 221–226 [in Ukrainian].
2. Kovalenko, O. V., Smirnov, O. A., Yakimenko, N. N., Dorenskyi, O. P. (2016) The problems of risk analysis and assessment in information activities. *Systemy obrobky informatsii*, No. 3 (140), pp. 40–42 [in Russian].
3. Spring Framework. URL: http://projects.spring.io/spring-framework/
4. Fowler, M. Inversion of control containers and the dependency injection pattern. URL: https://martinfowler.com/articles/injection.html
5. Lipaev, V. V. (2013) Reliability and functional safety of real-time program complexes. Moscow, 176 p. [in Russian].
6. Kovalenko, O. V., Smirnov, O. A. (2016) Application of pseudo-Boolean methods of bivalent programming for software development risk management. *Systemy upravlinnia, navihatsii ta zviazku*, No. 1 (37), pp. 98–103 [in Russian].
7. Kovalev, V. P. (2011) GERT-network analysis of multi-version software architectures. *Uspekhi sovremennogo yestestvoznaniya,* No. 9, pp. 161–164 [in Russian].
8. Kovalenko, O., Smirnov, O., Kovalenko, A., Smirnov, S., Vialkova, V. (2018) The mathematical model of the testing technology for DOM XSS vulnerabilities. *Scientific & practical cyber security journal (SPCSJ).* Vol. 2, iss. 1, pp. 22–28. URL: https://journal.scsa.ge/issues/2018/03/997
9. Kovalenko, O. V., Smirnov, O. A. (2016) Methods of qualitative analysis and quantification of software development risks. *Systemy obrobky informatsii*, No. 5 (142), pp. 153–157 [in Russian].
10. Kovalenko, O. V. (2017) Technology for testing vulnerability to SQL injections. *Systemy upravlinnia, navihatsii ta zviazku*, No. 5 (45), pp. 66–71 [in Russian].

O. V. Kovalenko, *Ph.D., associate professor*
Central Ukrainian National Technical University
Universytetskyi ave., 8, Kropyvnytskyi, 25006, Ukraine
e-mail: clashav@gmail.com

MATHEMATICAL MODELS OF THE TECHNOLOGY FOR TESTING DOM XSS VULNERABILITY AND SQL INJECTIONS VULNERABILITY

The analysis of various types of statistical materials from known organizations has shown that cross-site scripting – XSS (Cross Site Scripting) is one of the most dangerous types of attacks (vulnerabilities). However, in order to identify these threats and the possible consequences of their spread in the process of safe management of IT projects and to propose the best ways to solve this problem, there is a need for mathematical formalization of the process of their initialization and dissemination. In a number of papers, attempts have been made to mathematically formalize the process of finding and eliminating vulnerabilities of this kind. However, the presented models do not take into account the latest trends in XSS vulnerability, namely the difference between their types ("stored XSS", "reflected XSS" and DOM Based XSS) and the need for their detection. The aim of the work is to develop mathematical models of the technology for testing DOM XSS vulnerability and SQL injections vulnerability.

Mathematical models for testing DOM XSS complex of vulnerabilities and the technology for testing to SQL injections vulnerability have been developed. GERT-network synthesis approach is the basis of mathematical modeling. Mathematical model for testing DOM XSS complex of vulnerabilities differs from the known ones by taking into account the specifics of complex analysis of various types of XSS vulnerabilities (stored XSS, reflected XSS and DOM Based XSS) and separate inclusion of DOM Based XSS automatic audit procedures in the algorithm. This makes possible to conduct an analytical assessment of the time spent while testing these vulnerabilities in the context of implementing a strategy for developing safe software. Mathematical model for testing the technology of SQL injections vulnerability differs from the known ones by an improved method for determining the distance between injection results. The use of Jaro–Winkler criterion in the proposed method to compare the results of injecting SQL code and the introduction of a threshold value will increase the accuracy of the results of software security testing.

Keywords: *testing technology DOM XSS, SQL injections, GERT-network, Web security testing, mathematical models.*