ЗАХИСТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ОБЛАДНАННЯ / SOFTWARE & HARDWARE ARCHITECTURE SECURITY

DOI: 10.18372/2225-5036.23.11821

ТЕХНОЛОГИЯ ТЕСТИРОВАНИЯ DOM XSS УЯЗВИМОСТИ

Александр Коваленко

Центральноукраинский национальный технический университет, Украина

КОВАЛЕНКО Александр Владимирович, к.т.н.

Год и место рождения: 1982 год, г. Кировоград, Украина.
Образование: Кировоградский национальный технический университет (с 2017 года – Центральноукраинский национальный технический университет), 2004 год.
Должность: доцент кафедры кибербезопасности и программного обеспечения с 2013 года.
Научные интересы: программирование и информационная безопасность.
Публикации: более 140 научных публикаций, среди которых монографии, учебники, учебные пособия, научные статьи.
E-mail: clashav@gmail.com

Аннотация. *В работе представлены результаты исследования и алгоритмы тестирования на уязвимость к одним из наиболее распространенных видов атак на Web-приложения – межсайтовому скриптингу – XSS (Cross Site Scripting) – DOM XSS. Межсайтовый скриптинг это ошибка валидации пользовательских данных, которая позволяет передать JavaScript код на исполнение в браузер пользователя. Атаки такого рода часто также называют HTML-инъекциями, ведь механизм их внедрения очень схож с SQL-инъекциями, но в отличие от последних, внедряемый код исполняется в браузере пользователя. Аргументировано выбран подход математического моделирования на основе GERT-сетей. Проведенные исследования показали, что GERT (Graphical Evaluation and Review Technique) – является методом изучения и анализа стохастических сетей, используемых для описания логической взаимосвязи между частями проекта или этапами процесса. Главной целью GERT является оценка логики сети и продолжительность активности и получения заключения о необходимости выполнения некоторых активностей. Разработана технология тестирования Web-приложений и соответствующий комплекс математических моделей. В основу математического моделирования положен подход GERT-сетевого синтеза. В результате разработаны математические модели технологии тестирования DOM XSS уязвимости. Математическая модель технологии тестирования DOM XSS уязвимости отличается от известных, учетом выполнения или анализа DOM структуры. Разработанную технологию можно использовать при тестировании на уязвимость Web-приложения.*

Ключевые слова: *тестирование, DOM XSS уязвимости, GERT-сети, уязвимости безопасности.*

Введение

В настоящее время большой спрос на Web-приложения и Web-услуги обуславливает большой интерес злоумышленников к их возможным уязвимостям. При этом основные угрозы в направлении серверных компонент трансформируется в атаки, направленные против обычных пользователей.

Проведенный анализ материалов Open Web Application Security Project (OWASP TOP-10) показал, что одним из наиболее опасных видов атак (уязвимостей) является межсайтовый скриптинг – XSS (Cross Site Scripting).

Анализ литературы показал, что межсайтовый скриптинг это ошибка валидации пользовательских данных, которая позволяет передать JavaScript код на исполнение в браузер пользователя. Атаки такого рода часто также называют HTML-инъекциями, ведь механизм их внедрения очень схож с SQL-инъекциями, но в отличие от последних, внедряемый код исполняется в браузере пользователя.

Из работ [1-8, 15-20] известно, что под XSS обычно подразумевается моментальный и отложенный межсайтовый скриптинг. При моментальном XSS злонамеренный код (Javascript) возвращается атакуемым сервером немедленно в качестве ответа на HTTP запрос. Отложенный XSS означает, что злонамеренный код сохраняется на атакуемой системе и позднее может быть внедрен в HTML страницу уязвимой системы. Такая классификация предполагает, что фундаментальное свойство XSS состоит в том,

что злонамеренный код отсылается из браузера на сервер и возвращается в этот же браузер (моментальный XSS) или любой другой браузер (отложенный XSS).

В ряде интернет-статей подробно описаны основные механизмы возникновения подобного рода угроз, а также пути возможного блокирования. Однако, чтобы идентифицировать эти угрозы и возможные последствия их распространения в процессе безопасного управления IT-проектами, а также предложить оптимальные пути решения этой проблемы, существует необходимость математической формализации процесса их инициализации и распространения.

Особенно актуальной задачей в этом направлении представляется моделирование DOM (Document Object Model) XSS уязвимости. Связано это с тем, что уязвимость DOM XSS представляет собой подвид XSS, в случае которой результат атаки находится не в ответе сервера и, соответственно, не в HTML коде, а в DOM структуре HTML страницы. Результаты атак посредством таких уязвимостей можно обнаружить только в процессе выполнения или анализе DOM структуры. Сам механизм атаки, а именно инъекция Javascript кода в уязвимый сегмент, остается неизменным.

Целью данной работы является формирование технологии тестирования на уязвимость к одним из наиболее распространенных видов атак на Web-приложения – DOM XSS.

Алгоритм анализа DOM XSS уязвимости

Для математической формализации алгоритма анализа DOM XSS уязвимости воспользуемся основными положениями сетевого GERT-моделирования, подробно описанными в работах [9-12].

Графическое представление алгоритма анализа DOM XSS уязвимости представлено на рис. 1. В соответствии с этим алгоритмом основные этапы можно описать следующим образом:

1) Из кода анализируемой страницы извлекаются все теги <script> и формируется список тегов для анализа.

2) Выполняется анализ содержимого тега. При этом, если теги не содержат код, а ссылаются на удалённый файл, выполняется обращение к файлу и получение кода из него. В содержимом файла находятся потенциальные небезопасные участки кода (sink), которые используют входные данные клиента (source).

Примерами источников могут быть: document.URL; document.documentURI; location.href; location.search; location.*; window.name; document.re ferrer.

Примеры sink: document.write; (element).innerHTML; eval; setTimout / setInterval; execScript.

3) Если в коде тега используется source, выполняется атака с определенным маркером, который можно отследить в DOM структуре страницы после исполнения кода (например, инъекция определенного текстового содержимого в DOM).

4) Выполняется проверка содержимого DOM. Если в результате атаки маркер находится в DOM, можно сделать вывод о наличии DOM уязвимости.

5) Шаги 2 – 4 выполняются для каждого тега script на странице.

Для построения формальной модели алгоритма анализа уязвимости Web-приложений к DOM XSS выбрана стохастическая GERT- сеть.

Проведенные исследования показали, что GERT (*Graphical Evaluation and Review Technique*) – является методом изучения и анализа стохастических сетей, используемых для описания логической взаимосвязи между частями проекта или этапами процесса [9-12]. Главной целью GERT является оценка логики сети и продолжительность активности и получения заключения о необходимости выполнения некоторых активностей.

Сети GERT состоят из узлов типа AND, INCLUSIVE-OR и EXCLUSIVE-OR, и веток с двумя и более параметрами. Ветка, имеет направление, имеет узел начала и узел конца. Параметры ветви содержат: вероятность прохождения ветви (P_a) при условии, что узел, который является источником ветви, был реализован; время (t_a) прохождения ветви, если она будет реализована.

Время t_a может быть случайной величиной. Если ветвь не является частью реализации сети, то есть во время выполнения процесса активность, связанная с ветвью, не происходит, то $t_a = 0$.

Узел в стохастической сети GERT состоит из функции входа (контрибутивной функции) и функции выхода (дистрибутивной функции). Каждая из функций описывается определенным логическим отношением относительно связанных ветвей.

Рис.1. Структурная схема алгоритма анализа DOM XSS уязвимости

В целом, проведенные исследования показали, что GERT-моделирование является эффективным способом определения заранее неизвестных законов и функций распределения случайных величин при известном алгоритме функционирования (процесса). Именно поэтому, в качестве инструмента математического моделирования нами было выбрано GERT-моделирование.

GERT-модель технологии тестирования DOM XSS уязвимости

Построим, в соответствии с представленным описанием сетевую GERT-модель технологии тестирования DOM XSS уязвимости. Графическое изображение GERT-модели представлено на рис. 2.

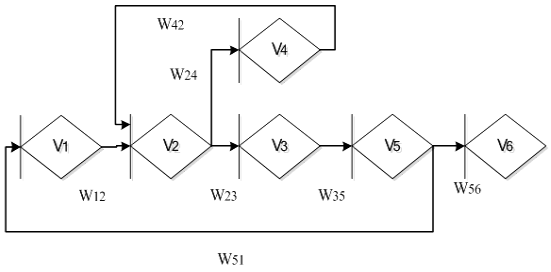

Рис. 2. GERT-модель технологии тестирования DOM XSS

В представленной сети узлы графа интерпретируются состояниями компьютерной системы в процессе функционирования DOM структуры, а ветви графа – вероятностно-временными характеристиками переходов между состояниями. В частности ветвь (1,2) характеризует время получения и анализа содержимого тега. Ветвь (2,3) отображает временные характеристики выполнения атаки в случае наличия «source» структуры. Ветвь (2,4) задает случайное время обращения к содержимому удаленного файла (поиск «sink»). Ветвь (4,2) характеризует возврат на выполнение атаки. Ветвь (3,5) описывает продолжение атаки, в частности проверку содержимого DOM. Далее ветвь (5,6) характеризует время принятия решения об уязвимости, в то же время ветвь (5,1) отображает временные характеристики перехода к новому тегу.

Характеристики ветвей модели представлены в табл. 1.

Характеристики ветвей модели Таблица 1

№ п/п	Ветвь	*W*-функция	Вероятность	Производящая функция моментов
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	(5,1)	W_{51}	$1-p4$	$\lambda_5/(\lambda_5-s)$
7	(4,2)	W_{42}	$p3$	$\lambda_3/(\lambda_3-s)$

Эквивалентная *W*-функция времени выполнения технологии тестирования DOM XSS уязвимости равна:

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

где $1-p_4=q_1$.

Особенность рассматриваемого процесса заключается в разнородности анализируемых и обрабатываемых данных. При этом возможны различные случаи организации обратной связи. На рис. 2 эти циклы зафиксированы в виде переходов $W_{12}\to W_{24}\to W_{42}$, $W_{12}\to W_{23}\to W_{35}\to W_{51}$.

Для *GERT*-сетей с циклами не существует простых методов нахождения особых точек функции $\Phi_E(z)$ замены действительных переменных $(z=-i\varsigma)$, где ς – действительная переменная. Это объясняется тем, что для нахождения особых точек необходимо решать нелинейные уравнения, и чем сложнее структура *GERT*-сети, тем сложнее и исходное уравнение. Поэтому в ходе моделирования предлагается прибегнуть к подобной замене.

Выполняя комплексное преобразование $z=-s$, получим:

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

где:

$$u=-p_1p_2^2p_4\lambda_1\lambda_2^2\lambda_4,$$
$$v=p_1p_2^2p_4\lambda_1\lambda_2^2\lambda_4(\lambda_5+2\lambda_3),$$
$$b=-p_1p_2^2p_4\lambda_1\lambda_2^2\lambda_4\lambda_3(2\lambda_5-\lambda_3),$$
$$k=-p_1p_2^2\lambda_1\lambda_2^2\lambda_3^2\lambda_4\lambda_5\left(p_4+p_3^2q_1\right),$$
$$c=\lambda_1+2\lambda_2+2\lambda_3+\lambda_5,$$
$$d=-(2\lambda_3\lambda_5+\lambda_1\lambda_5+2\lambda_2\lambda_5+\lambda_3^2+$$
$$+2\lambda_1\lambda_3+4\lambda_2\lambda_3+2\lambda_1\lambda_2+\lambda_2^2),$$
$$g=\lambda_3^2\lambda_5+4\lambda_1\lambda_2\lambda_5+4\lambda_2\lambda_3\lambda_5+\lambda_2^2+\lambda_3^2\lambda_1+$$
$$+2\lambda_3^2\lambda_2+4\lambda_1\lambda_2\lambda_3+2\lambda_2^2\lambda_3+\lambda_2^2\lambda_1,$$
$$h=-(\lambda_1\lambda_3^2\lambda_5+2\lambda_2\lambda_3^2\lambda_5+4\lambda_1\lambda_2\lambda_3\lambda_5+2\lambda_2^2\lambda_3\lambda_5+$$
$$+\lambda_2^2\lambda_3^2+2\lambda_1\lambda_2^2\lambda_3-p_1p_2^2q_1\lambda_1\lambda_2\lambda_5),$$
$$w=\lambda_1\lambda_2\lambda_3^2\lambda_5+\lambda_2^2\lambda_3^2\lambda_5+2\lambda_1\lambda_2^2\lambda_3\lambda_5+$$
$$+\lambda_1\lambda_2^2\lambda_3-2p_1p_2^2q_1\lambda_1\lambda_2\lambda_3\lambda_5,$$
$$m=p_1p_2^2q_1\lambda_1\lambda_2\lambda_3^2\lambda_5+p_1p_2^2p_3q_1\lambda_1\lambda_2^2\lambda_3^2\lambda_5-\lambda_1\lambda_2^2\lambda_3\lambda_5.$$

Плотность распределения вероятностей времени выполнения алгоритма анализа DOM XSS уязвимости:

$$\phi(x)=\frac{1}{2\pi i}\int_{-i\infty}^{i\infty}e^{zx}\frac{uz^3+vz^2+bz+k}{(\lambda_4+z)\left(z^6+cz^5+dz^4+\right.}dz, \quad (3)$$
$$\left.+gz^3+hz^2+wz+m\right)$$

где операция интегрирования выполняется с помощью интеграла Бромвича-Вагнера [13].

Способ интегрирования зависит от того, имеет ли функция $\Phi(z)$ только простые полюсы, или полюсы некоторого порядка. В том случае, когда

функция $\Phi(z)$ имеет только простые полюсы, выражение $\mathring{a}^{zx}\Phi(z)$ можно представить в виде:

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

где:

$$\gamma_6=\lambda_4+c,$$
$$\gamma_5=c\lambda_4+d,$$
$$\gamma_4=d\lambda_4+g,$$
$$\gamma_3=g\lambda_4+h,$$
$$\gamma_2=h\lambda_4+w,$$
$$\gamma_1=w\lambda_4+m,$$
$$\gamma_0=m\lambda_4.$$

Тогда плотность распределения времени выполнения алгоритма анализа DOM XSS уязвимости равна:

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

Функция $\Phi(z)$ кроме решений, определяемых корнями уравнения $z^6+cz^5+dz^4+gz^3+hz^2+wz+m=0$, может иметь и полюс второго или третьего порядка тех случаях, когда значение λ_4 равно значению корней z_2, z_3, z_4, z_5, z_6, z_7. В этих случаях плотность распределения времени передачи сообщения $\varphi(x)$ находится по формуле нахождения вычетов r_{-1} от полюсов z_k порядка n:

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

Выражение (6) представляет собой дробно-рациональную функцию относительно Z со степенью знаменателя большей, чем степень числителя. Поэтому для него выполняется условия леммы Жордана [13]. Функция $\Phi(z)$ имеет полюсы в точках $z_1=-\lambda_4$. Многочлен $z^6+cz^5+dz^4+gz^3+hz^2+wz+m$ порождает еще семь полюсов. Решение уравнения

$$z^6+cz^5+dz^4+gz^3+hz^2+wz+m=0, \quad (7)$$

может быть найдено любым методом, например, по формулам Виета [13]. В результате вычисляются особые точки z_2, z_3, z_4, z_5, z_6, z_7.

Таким образом, на основе экспоненциальной *GERT*-сети разработана математическая модель алгоритма анализа DOM XSS уязвимости, которая отличается от известных, учетом выполнения или анализа DOM структуры.

Модель может быть использована для исследования процессов в компьютеризированных системах, при разработке новых средств и протоколов защиты данных. Применение экспоненциальных стохастических моделей *GERT* даст возможность использования результатов, полученных в аналитическом виде (функции, плотности распределения) для проведения сравнительного анализа и исследований, более сложных компьютерных систем математическими методами.

Исследования GERT-модели технологии тестирования DOM XSS уязвимости

Рассмотрим пример XSS атаки через DOM (аналогичный алгоритм простого использования клиентского скрипта для (небезопасной) переадресации браузера к другому ресурсу).

Найдем плотности распределения $\varphi(x)$ вероятностей времени выполнения алгоритма при условии, что Z выбираются как корни уравнения $(\lambda_4+z)(z^6+cz^5+dz^4+gz^3+hz^2+wz+m)=0$, условные вероятности и интенсивности в ветвях *GERT*-сети имеют значения: $p_1=0{,}999$, $p_2=0{,}6$, $p_3=0{,}4$, $p_4=0{,}99$, $\lambda_1=0{,}9999$, $\lambda_2=0{,}79$, $\lambda_3=0{,}29$. С учетом приведенных признаков *GERT*-сети, в соответствии с выражением (2), а так же используя специализированный математический пакет *Mathcad,* получим, что в знаменателе выражения (3) сформирован полином:

$$x^6+4.07x^5-6.66x^4+6.529x^3- \\ -1.592x^2+0.617x-0.164=0. \quad (8)$$

Корни этого полинома (и соответственно функция $\Phi(z)$) равны:

$$x1\approx -5.505381393772 08,\ (P(x1)\approx 0,\ iter=1);$$
$$x2\approx -0.04984635\ 17249773+i\cdot 0.33125906\ 4468874,\ (P(x2)\approx -0.000162-i\cdot 0.00073,\ iter=3);$$
$$x3\approx -0.04956650\ 29547472-i\cdot 0.33124693\ 1512067,\ (P(x3)\approx -0.000307+i\cdot 0.00013,\ iter=3);$$
$$x4\approx 0.28764249\ 382953-i\cdot 0.00014096\ 1065526,\ (P(x4)\approx 0.000129-i\cdot 0.000118,\ iter=3);$$
$$x5\approx 0.62354397\ 1678568-i\cdot 0.73145420\ 7007899,\ (P(x5)\approx 0,\ iter=3);$$
$$x6\approx 0.623607782943707+i\cdot 0.731584150633824,\ (P(x6)\approx 0,\ iter=2).$$

Кроме этого существует простой полюс в точке $z_1=-\lambda_4$ ($z_1=-0.9$) ($x7\approx -0.9$). Исследуем зависимость функции $\Phi(z)$ от интенсивности λ_2, являющейся одним из основных показателей в рассматриваемом алгоритме (интенсивность λ_2 характеризует выполнения атаки в случае наличия «source» структуры).

На рис. 3 представлена кривая графика зависимости функции $\Phi(z)$ от интенсивности λ_2 в рассматриваемых выше условиях. Как видно из рисунка

случайная величина λ_2 распределена в соответствии с показательным законом.

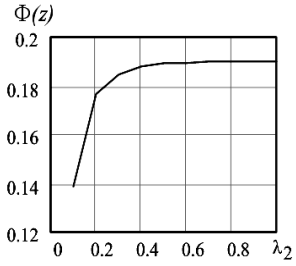

Рис. 3. График зависимости функции $\Phi(z)$ от интенсивности λ_2

Используя полученные значения, найдем $\varphi(x)$. В соответствии с формулой (5) $\varphi(x)$ равна:

$$\phi(x)=\sum_{k=1}^{6}\operatorname{Res}_{z=z_k}\left[e^{zx}\Phi(z)\right]=$$
$$=\frac{e^{(a+\delta i)x}\left(u(a+\delta i)^3+v(a+\delta i)^2+b(a+\delta i)+k\right)}{\left(\begin{array}{l}7(a+\delta i)^6+6g_6(a+\delta i)^5+5g_5(a+\delta i)^4+\\+4g_4(a+\delta i)^3+3g_3(a+\delta i)^2+2g_2(a+\delta i)+g_1\end{array}\right)}- \quad (9)$$
$$-\frac{e^{(a-bi)x}\left(u(a-\delta i)^3+v(a-\delta i)^2+b(a-\delta i)+k\right)}{\left(\begin{array}{l}7(a-\delta i)^6+6g_6(a-\delta i)^5+5g_5(a-\delta i)^4+\\+4g_4(a-\delta i)^3+3g_3(a-\delta i)^2+2g_2(a-\delta i)+g_1\end{array}\right)}.$$

Из [13] известно, что сумма значений любой дробно-рациональной функции

$$f(z)=\frac{d_m z^m+d_{m-1}z^{m-1}+\ldots d_1 z+d_0}{\ell_m z^m+\ell_{m-1}z^{m-1}+\ldots+\ell_1 z+\ell_0},$$
$$d_m\neq 0,\ \ell_m\neq 0$$

исследуемой при значениях комплексных сопряженных аргументов, может быть представлена в виде:

$$\frac{(\tau+i\beta)}{(\gamma+i\beta)}+\frac{(\tau-i\theta)}{(\gamma-i\theta)},$$

где $\tau, \beta, \gamma, \theta$ – некоторые коэффициенты.

Используя выражения Эйлера [13], получим:

$$\phi(x)=\sum_{k=1}^{6}\operatorname{Res}_{z=z_k}\left(e^{zx}\Phi(z)\right)=$$
$$=e^{(a+\delta i)x}\frac{\tau+i\beta}{\gamma+i\theta}+e^{(a-\delta i)x}\frac{\tau-i\beta}{\gamma-i\theta}= \quad (10)$$
$$=\frac{2e^{ax}}{\gamma^2+\theta^2}\left((\tau\gamma+\beta\theta)\cos(\delta x)+(\tau\gamma-\beta\theta)\sin(\delta x)\right),$$

где

$$\tau=a^3u-3a\delta^2u+a^2v-\delta^2v+ab+k,$$
$$\beta=3a^2\delta u-\delta^3u+2a\delta v+\delta b,$$

$$\gamma=7a^6-10a^4\delta^2+105a^2\delta^4-7\delta^6+6g_6a^5-$$
$$-60g_6a^3\delta^2+30g_6a\delta^4+5g_5a^4-$$
$$-30g_5a^2\delta^2+5g_5\delta^4+4g_4a^3-12g_4a\delta^2+$$
$$+3g_3a^2-3g_3\delta^2+2g_2a+g_1,$$

$$\theta=49a^5\delta-140a^3\delta^3+49a\delta^5+$$
$$+30g_6a^4\delta-60g_6a^2\delta^3+6g_6\delta^5+20g_5a^3\delta-$$
$$-20g_5a\delta^3+12g_4a^3\delta-4g_4\delta^3+6g_3a\delta+2g_2\delta.$$

На рис. 4 представлены кривые плотности распределения $\varphi(x)$ вероятностей времени выполнения алгоритма анализа DOM XSS уязвимости для приведенных выше условий (в качестве входных данных использовались корни полинома (8)).

При этом рис. 4а соответствует случаю когда в качестве входных данных $(a+\delta i)$ использовалось значение x7.

Рис. 4б соответствует случаю когда в качестве входных данных использовалось значение x1.

Рис. 4в соответствует случаю когда в качестве входных данных использовалось значение x3.

Рис. 4д соответствует случаю когда в качестве входных данных использовалось значение x4.

Рис. 4е соответствует случаю когда в качестве входных данных использовалось значение x5.

Рис. 4ж соответствует случаю когда в качестве входных данных использовалось значение x6.

Внешний вид кривых графиков рис. 4 дает основания предположить, что не все найденные выше решения (корни полинома (8)) применимы при математическом и имитационном моделировании в качестве входных данных. Так значения x1, x6 и x7 невозможно в дальнейшем использовать при анализе и моделировании. В то же время внешний вид графиков, полученных для значений x3 и x5 дает основания предположить, что случайная величина времени выполнения алгоритма анализа DOM XSS уязвимости имеет гамма-распределение.

Проверим эту гипотезу по критерию χ^2 Пирсона [14]:

$$\chi^2=N^*\sum_{i=1}^{k}(P_i^*-P_i)^2/P_i,$$

где k – число разрядов (интервалов) статистического ряда; P_i^* и P_i – «статистическая» и теоретическая вероятности события.

В результате эксперимента были получены теоретические значения χ^2 и табличное значение $\overline{\chi^2}$, обратное правосторонней вероятности распределения χ^2. Проведенная проверка показала, что выдвинутую гипотезу можно считать правдоподобной или, по крайней мере, не противоречащей полученным при математическом моделировании результатам.

Это подтверждается тем, что при достаточно большом значении доверительной вероятности $Q=0{,}95$ для всех рассматриваемых x2 и x5 соответствующие значения χ^2 ($\chi_1^2=19{,}3$, $\chi_2^2=15{,}1$) $<<\overline{\chi^2}=101{,}9$ позволяют признать расхождения между «статистическими» (P_i^*) и теоретическими (P_i) вероятностями наступления события несущественными.

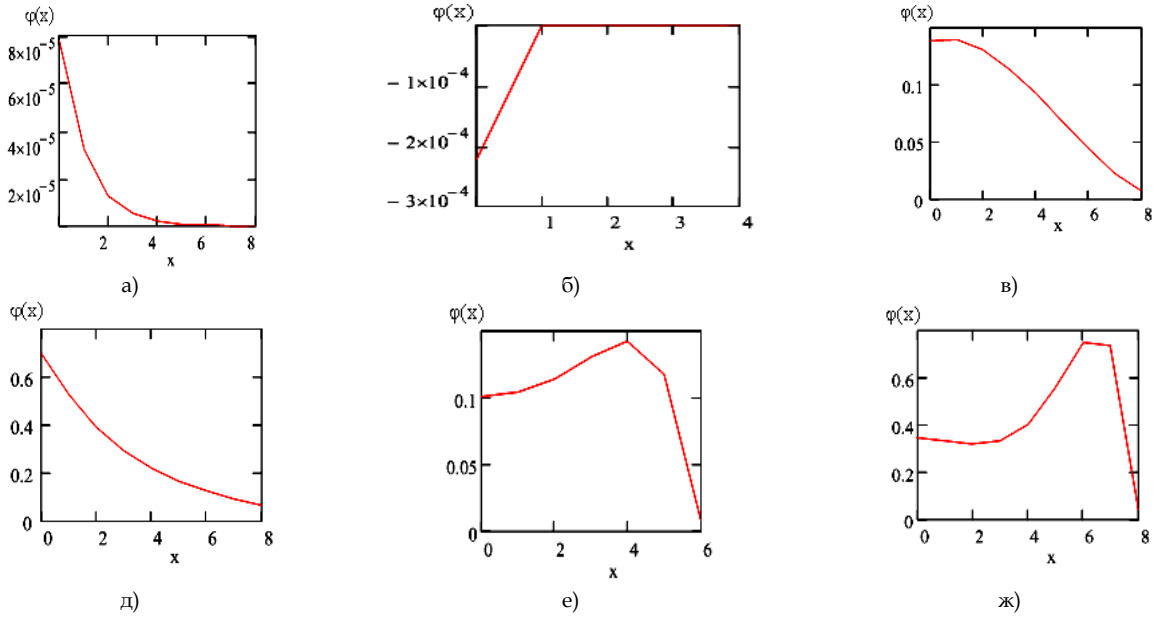

Рис. 4. График плотности распределения $\varphi(x)$ вероятности времени выполнения алгоритма анализа DOM XSS уязвимости: а) соответствует случаю когда в качестве входных данных $(a+\delta i)$ использовалось значение x7; б) соответствует случаю когда в качестве входных данных использовалось значение x1; в) соответствует случаю когда в качестве входных данных использовалось значение x3; д) соответствует случаю когда в качестве входных данных использовалось значение x4; е) соответствует случаю когда в качестве входных данных использовалось значение x5; ж) соответствует случаю когда в качестве входных данных использовалось значение x6.

Выводы

В работе разработан комплекс математических моделей технологии тестирования WEB-приложений. В основу математического моделирования положен подход GERT-сетевого синтеза. В результате разработаны математические модели технологии тестирования DOM XSS уязвимости.

Математическая модель технологии тестирования DOM XSS уязвимости отличается от известных, учетом выполнения или анализа DOM структуры, что дает возможность провести аналитическую оценку временных затрат тестирования указанной уязвимости в условиях реализации стратегии разработки безопасного программного обеспечения.

В ходе исследования представленных моделей было определено, что случайная величина времени выполнения рассматриваемых технологий тестирования в целом соответствует гамма-распределению. Проверка этой гипотезы произведена по критерию χ^2 Пирсона.

Литература

[1] About The Open Web Application Security Project – OWASP. [Електронний ресурс]. Режим доступу: https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project.

[2] OWASP Top 10 – 2017 RC1. [Електронний ресурс]. Режим доступу: https://github.com/OWASP/Top10/blob/master/2017/OWASP%20Top%2010%20-%202017%20RC1-English.pdf.

[3] Positive Research 2016. [Електронний ресурс]. Режим доступу: https://www.ptsecurity.com/upload/ptru/analytics/Positive-Research-2016-us.pdf.

[4] OSSTMM 3 – The Open Source Security Testing Methodology Manual. Contemporary Security Testing And Analysis. [Електронний ресурс]. Режим доступу: http://www.isecom.org/mirror/OSSTMM.3.pdf.

[5] Testing for DOM-based Cross-site scripting (OTG-CLIENT-001) – OWASP. [Електронний ресурс]. Режим доступу: https://www.owasp.org/index.php/Testing_for_DOM-based_Cross_site_scripting_(OTG-CLIENT-001).

[6] Testing for SQL Injection (OTG-INPVAL-005) – OWASP. [Електронний ресурс]. Режим доступу: https://www.owasp.org/index.php/103.

[7] W. Cohen, P. Ravikumar, S. Fienberg, «A Comparison of String Metrics for Matching Names and Records». [Online]. Available at: https://www.cs.cmu.edu/afs/cs/Web/People/wcohen/postscript/kdd-2003-match-ws.pdf.

[8] K. Dreßler, Axel-Cyrille Ngonga Ngomo, «On the Efficient Execution of Bounded Jaro-Winkler Distances», *Semantic Web-Interoperability, Usability, Applicability an IOS Press Journal*. [Online]. Available at: http://www.semantic-web-journal.net/system/files/s wj944.pdf

[9] A.A.B Pritsker, «GERT: Graphical Evaluation and Review Technique. Part I. Fundamentals», *The Journal of Industrial Engineering*, pp. 267-274, 1966.

[10] A.A.B Pritsker, «Modeling and analysis using Q-GERT networks», *Wiley: Distributed by Halsted Press*, New York, 435 p., 1979.

[11] С.Г.Семенов, С.Ю. Гавриленко, Кассем Халіфе, «Gert-модель прогнозування параметрів функціональної безпеки технічних систем», *Зб. наукових праць. Системи обробки інформації*, Х, ХУПС, вип. 2(139), с. 50-52, 2016.

[12] S. Semenov, V. Zmiyevskaya, K. Khalife, «Development of Gert model of management system by using test

cases», *Journal of Qafqaz university-mathematics and computer science*, Vol.(4), № 1 C. 52-59, 2016.

[13] Г. Эдвардс, «Последняя теорема Ферма. Генетическое введение в алгебраическую теорию чисел», М., Мир, 486 с., 1980.

[14] В.Е. Гмурман, «Теория вероятностей и математическая статистика», М.: Высшая школа, 479 с., 2003.

[15] А.А. Смирнов, А.В. Коваленко, «Методы качественного анализа и количественной оценки рисков разработки программного обеспечения», *Збірник наукових праць «Системи обробки інформації»*, випуск 5(142), Х, ХУПС, с. 153-157, 2016.

[16] А.А. Смирнов, А.В. Коваленко, Н.Н. Якименко, А.П. Доренский , «Проблемы анализа и оценки рисков информационной деятельности», *Збірник наукових праць «Системи обробки інформації»*., випуск 3(140), Х, ХУПС, с. 40-42, 2016.

[17] А.А. Смирнов, А.В. Коваленко, Н.Н. Якименко, А.П. Доренский, «Метод качественного анализа рисков разработки программного обеспечения», *Наука і техніка Повітряних Сил Збройних Сил України*, випуск 2(23), Харків, ХУПС, с. 150-158, 2016.

[18] А.А. Смирнов, А.В. Коваленко, Н.Н. Якименко, А.П. Доренский, «Метод количественной оценки рисков разработки программного обеспечения», *Збірник наукових праць Харківського університету Повітряних Сил*, випуск 2 (47), Харків, ХУПС, с. 128-133, 2016.

[19] А.В Коваленко, А.А. Смирнов, «Использование псевдобулевых методов бивалентного программирования для управления рисками разработки программного обеспечения», *Системи управління, навігації та зв'язку*, випуск 1 (37), Полтава, ПолтНТУ, с. 98-103, 2016.

[20] А.В. Коваленко, «Метод управления рисками разработки программного обеспечения», *Системи управління, навігації та зв'язку*, випуск 2 (38), Полтава, ПолтНТУ, с. 93-100, 2016.

УДК 004.41:004.056 (045)

Коваленко О.В. Технологія тестування DOM XSS уразливості

Анотація. *В роботі представлені результати дослідження та алгоритми тестування на вразливість до одних з найбільш поширених видів атак на Web-додатки - Міжсайтовий Скриптінг - XSS (Cross Site Scripting) - DOM XSS. Міжсайтовий скриптинг це помилка валідації призначених для користувача даних, яка дозволяє передати JavaScript код на виконання в браузер користувача. Атаки такого роду часто також називають HTML-ін'єкціями, адже механізм їх впровадження дуже схожий з SQL-ін'єкціями, але на відміну від останніх, впроваджуваний код виповнюється в браузері користувача. Аргументовано обраний підхід математичного моделювання на основі GERT-мереж. Проведені дослідження показали, що GERT (Graphical Evaluation and Review Technique) - є методом вивчення та аналізу стохастичних мереж, які використовуються для опису логічного взаємозв'язку між частинами проекту або етапами процесу. Головною метою GERT є оцінка логіки мережі і тривалість активності і отримання висновку про необхідність виконання деяких активностей. Розроблено технологію тестування Web-додатків і відповідний комплекс математичних моделей. В основу математичного моделювання покладено підхід GERT-мережевого синтезу. В результаті розроблено математичні моделі технології тестування DOM XSS уразливості. Математична модель технології тестування DOM XSS уразливості відрізняється від відомих, урахуванням виконання або аналізу DOM структури. Розроблену технологію можна використовувати при тестуванні на вразливість Web-додатку.*

Ключові слова: *тестування, DOM XSS уразливості, GERT-мережі, уразливості безпеки.*

Kovalenko O. DOM XSS vulnerability testing technology

Abstract. *The paper presents research results and vulnerability testing algorithms for one of the most common types of attacks on Web-based applications - cross site scripting - XSS (Cross Site Scripting) - DOM XSS. Cross-site scripting is the error of validating user data, which allows you to pass JavaScript code to execution in the user's browser. Attacks of this kind are often also called HTML injections, because the implementation mechanism is very similar to SQL injections, but unlike the latter, the implemented code is executed in the user's browser. The approach of mathematical modeling on the basis of GERT-networks is argued. Studies have shown that GERT (Graphical Evaluation and Review Technique) is a method of studying and analyzing stochastic networks used to describe the logical relationship between parts of a project or process steps. The main goal of GERT is to evaluate the logic of the network and the duration of activity and to obtain an opinion on the need to perform certain activities. The technology of testing Web-applications and the corresponding complex of mathematical models is developed. The basis of mathematical modeling is the approach of GERT-network synthesis. As a result, mathematical models of DOM XSS testing technology have been developed. The mathematical model of the testing technology of the DOM XSS vulnerability differs from the known, taking into account the execution or analysis of the DOM structure. The developed technology can be used in testing for the vulnerability of a Web application.*

Key words: *testing, DOM XSS vulnerabilities, GERT-network, security vulnerabilities.*

Отримано 26 *червня 2017 року*, затверджено редколегією 29 *липня 2017 року*