УДК 004.415.53:005.6

А.В. Коваленко

Центральноукраинский национальный технический университет, Кропивницкий

ТЕХНОЛОГИЯ ТЕСТИРОВАНИЯ УЯЗВИМОСТИ К SQL ИНЪЕКЦИЯМ

Представлены результаты исследования и алгоритмы тестирования на уязвимость к одним из наиболее распространенных видов атак на Web-приложения – SQL инъекции. Аргументировано выбран подход математического моделирования на основе GERT-сетей. Разработан комплекс математических моделей технологии тестирования Web-приложений. В основу математического моделирования положен подход GERT-сетевого синтеза. В результате разработаны математические модели технологии тестирования уязвимости к SQL инъекциям. Математическая модель технологии тестирования уязвимости к SQL инъекциям отличается от известных, усовершенствованным способом определения расстояния между результатами инъекции.

Ключевые слова: *уязвимости к SQL инъекциям, GERT-сети, уязвимости безопасности*

Введение

Проведенные исследования [1-8, 15-20] показали, что на основе анализа методологии тестирования уязвимости Web-приложений к DOM XSS и материалов Open Web Application Security Project, можно разработать алгоритм анализа уязвимости Web-приложений к SQL инъекциям. Отличительной особенностью данного алгоритма является учет только уязвимости, которая имеется в GET параметрах URL и использует только слепой метод инъекции SQL кода, использующего особенность использования булевых операторов в SQL запросах (Boolean blind SQL injection).

1. Алгоритм анализа уязвимости к SQL инъекциям

Структурная схема алгоритма анализа уязвимости Web-приложения к SQL инъекциям представлена на рис. 1.

В соответствии с представленным алгоритмом, его этапы можно описать следующим образом:

1. Из введенного URL ссылки получается список GET параметров.

2. Выполняется проверка стабильности Web-страницы. Для этого выполняется два последовательных запроса в Web-страницы и вычисляется расстояние между содержанием HTML кода страницы с помощью критерия Джаро-Винклера [8]. Если значение критерия меньше определенного порогового значения, выполнять дальнейший анализ невозможно.

3. В параметр GET запроса выполняется инъекция SQL кода, который не меняет результат запроса к базе данных и сохраняется результирующий HTML код.

4. В параметре GET запроса выполняется инъекция SQL кода, который меняет результат запроса к базе данных, приводит или к получению полного

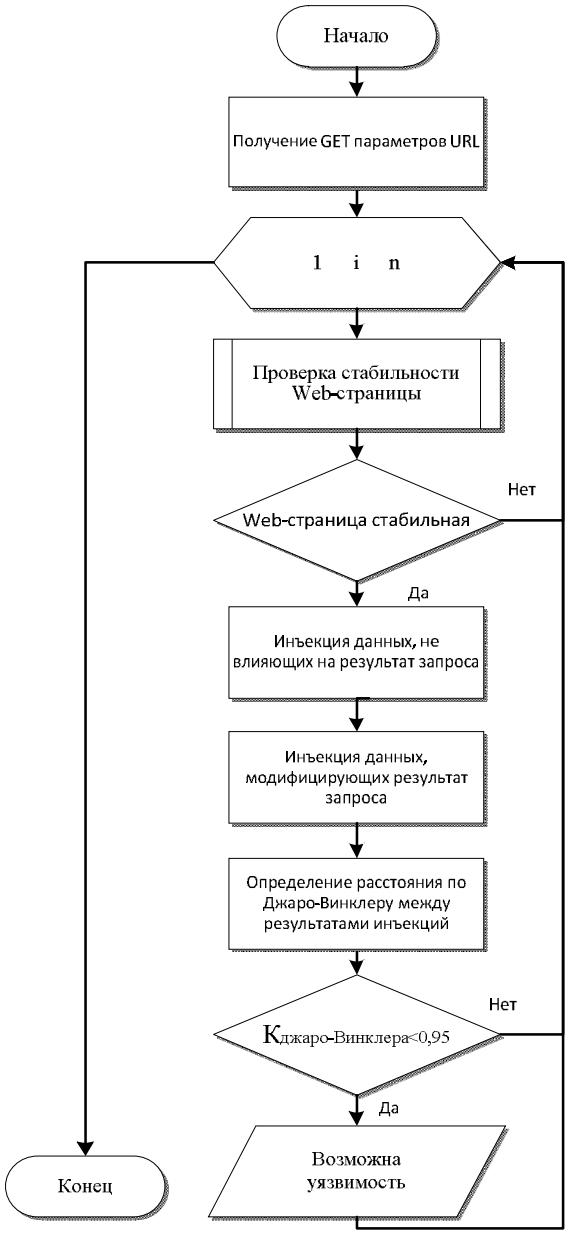

Рис. 1. Структурная схема алгоритма анализа уязвимости Web-приложения к SQL инъекциям

© А.В. Коваленко

набора данных из таблицы, или к отсутствию результата, после чего сохраняется результирующий HTML код.

5. С помощью критерия Джаро-Винклера выполняется сравнение результатов инъекции SQL кода. Если значение критерия меньше определенного порогового значения, то в данном GET параметре является возможная уязвимость к SQL инъекции.

6. Шаги 2 – 5 повторяются для всех параметров GET запроса предоставленного URL.

Для построения формальной модели алгоритма анализа уязвимости Web-приложений к к SQL инъекциям 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:

1) вероятность прохождения ветви (P_a) при условии, что узел, который является источником ветви, был реализован;

2) 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. 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. На основании представленного алгоритма разработаем GERT-модель технологии тестирования уязвимости к SQL инъекциям.

2. GERT-модель технологии тестирования уязвимости к SQL инъекциям

Построим, в соответствии с представленным описанием сетевую GERT-модель технологии тестирования уязвимости к SQL инъекциям. Графическое изображение GERT-модели представлено на рис. 2. В представленной сети узлы графа интерпретируются состояниями компьютерной системы в процессе тестирования уязвимости к SQL инъекциям, а ветви графа – вероятностно-временными характеристиками переходов между состояниями. В частности ветвь (1,2) характеризует время получения и анализа GET-параметров из введенного URL ссылки. Ветвь (2,3) отображает время отправления первичных и вторичных запросов в Web-страницы. Ветвь (3,4) задает случайное время сравнения страниц (время вычисления расстояния между содержанием HTML кода страницы с помощью критерия Джаро-Винклера). Ветвь (4,5) характеризует время, за которое выполняется инъекция SQL кода, который не меняет результат запроса к базе данных, а также который меняет результат запроса к базе данных соответственно. Далее ветвь (5,6) характеризует время сравнения результатов инъекции SQL кода. Ветвь (4,2) характеризует временные характеристики возврата системы в первоначальное состояние, когда значение критерия Джаро-Винклера меньше определенного порогового значения, в то же время ветвь (6,2) отображает временные характеристики перехода к новой проверке в случае если значение критерия Джаро-Винклера больше определенного порогового значения.

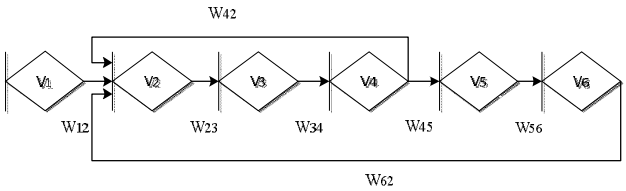

Рис. 2. GERT-модель технологии тестирования уязвимости к SQL инъекциям

Характеристики ветвей модели представлены в табл. 1.

Таблица 1

Характеристики ветвей модели технологии тестирования уязвимости к SQL инъекциям

№ п/п	Ветвь	*W*-функция	Вероятность	Производящая функция моментов
1	(1,2)	W_{12}	p_1	$\lambda_1 / (\lambda_1 - s)$
2	(2,3)	W_{23}	p_2	$\lambda_2 / (\lambda_2 - s)$
3	(3,4)	W_{34}	p_3	$\lambda_3 / (\lambda_3 - s)$
4	(4,5)	W_{45}	p_4	$\lambda_4 / (\lambda_4 - s)$
5	(5,6)	W_{56}	p_5	$\lambda_3 / (\lambda_3 - s)$
6	(4,2)	W_{42}	$q_1=1-p_4$	$\lambda_5 / (\lambda_5 - s)$
7	(6,2)	W_{62}	p_6	$\lambda_6 / (\lambda_6 - s)$

Эквивалентная *W*-функция времени выполнения технологии тестирования уязвимости к SQL инъекциям равна:

$$W_E(s) = \frac{W_{12}W_{23}W_{34}W_{45}W_{56}}{1 - W_{12}W_{23}W_{34}W_{42} - W_{12}W_{23}W_{34}W_{45}W_{56}W_{62}} = p_1p_2p_3p_4p_5\lambda_1\lambda_2\lambda_3^2\lambda_4(\lambda_3 - s)(\lambda_5 - s)(\lambda_6 - s)\times$$

$$\left(\begin{array}{c}(\lambda_1 - s)(\lambda_2 - s)(\lambda_3 - s)^2(\lambda_4 - s)(\lambda_5 - s)(\lambda_6 - s) - \\ -\left(p_1p_2p_3\lambda_1\lambda_2\lambda_3\times\left(q_1\lambda_5\left(\lambda_3\lambda_4 - \lambda_4 s - \lambda_3 s - s^2\right)(\lambda_6 - s) - p_4p_5p_6\lambda_3\lambda_4\lambda_6(\lambda_5 - s)\right)\right)\end{array}\right)^{-1}. \quad (1)$$

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

$$\Phi(z) = \left(vz^2 + bz + k\right)\Big/\left(z^7 + rz^6 + cz^5 + dz^4 + gz^3 + hz^2 + wz + m\right), \quad (2)$$

где $v = -p_1p_2p_3p_4p_5\lambda_1\lambda_2\lambda_3^2\lambda_4$, $b = p_1p_2p_3p_4p_5\lambda_1\lambda_2\lambda_3^2\lambda_4(\lambda_5 + \lambda_6)$, $k = -p_1p_2p_3p_4p_5\lambda_1\lambda_2\lambda_3^2\lambda_4\lambda_5\lambda_6$,

$r = \lambda_1 + \lambda_2 + \lambda_4 + \lambda_5 - 2\lambda_3 - \lambda_6$, $m = -\lambda_1\lambda_2\lambda_3^2\lambda_5\lambda_6\left(\lambda_4 + p_1p_2p_3q_1 - p_4p_5p_6\big/(\lambda_1\lambda_2)\right)$

$$\tilde{n} = \left(\lambda_1\lambda_4 + \lambda_2\lambda_4 + \lambda_1\lambda_5 + \lambda_2\lambda_5 + \lambda_3^2 + 2\lambda_3\lambda_6 - \lambda_4\lambda_6 - \lambda_5\lambda_6 - \lambda_1\lambda_6 - \lambda_4\lambda_5 - 2\lambda_3\lambda_4 - 2\lambda_3\lambda_5 - \lambda_1\lambda_2 - 2\lambda_1\lambda_3 - 2\lambda_2\lambda_3\right),$$

$$d = -\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5\lambda_6\left(\begin{array}{c}\frac{1}{\lambda_2\lambda_3\lambda_5} + \frac{1}{\lambda_1\lambda_3\lambda_5} + \frac{1}{\lambda_2\lambda_3\lambda_4} + \frac{1}{\lambda_1\lambda_3\lambda_4} + \frac{\lambda_3}{\lambda_1\lambda_2\lambda_4\lambda_5} + \frac{1}{\lambda_2\lambda_3\lambda_6} + \frac{1}{\lambda_1\lambda_3\lambda_6} + \frac{1}{\lambda_3\lambda_5\lambda_6} + \\ + \frac{2}{\lambda_2\lambda_5\lambda_6} + \frac{2}{\lambda_1\lambda_5\lambda_6} + \frac{1}{\lambda_3\lambda_4\lambda_6} + \frac{2}{\lambda_2\lambda_4\lambda_6} + \frac{2}{\lambda_1\lambda_4\lambda_6} - \frac{1}{\lambda_1\lambda_2\lambda_3} - \frac{2}{\lambda_1\lambda_2\lambda_5} - \frac{2}{\lambda_1\lambda_2\lambda_4} - \\ - \frac{1}{\lambda_3\lambda_4\lambda_5} - \frac{2}{\lambda_2\lambda_4\lambda_5} - \frac{2}{\lambda_1\lambda_2\lambda_6} - \frac{\lambda_3}{\lambda_1\lambda_2\lambda_5\lambda_6} - \frac{\lambda_3}{\lambda_1\lambda_2\lambda_4\lambda_6} - \frac{2}{\lambda_4\lambda_5\lambda_6} \\ - \frac{\lambda_3}{\lambda_2\lambda_4\lambda_5\lambda_6} - \frac{\lambda_3}{\lambda_1\lambda_4\lambda_5\lambda_6}\end{array}\right),$$

$$g = \left[\left(\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5\lambda_6\left(\begin{array}{c}\frac{2}{\lambda_1\lambda_2} + \frac{\lambda_3}{\lambda_1\lambda_2\lambda_5} + \frac{\lambda_3}{\lambda_1\lambda_2\lambda_4} + \frac{2}{\lambda_4\lambda_5} + \frac{\lambda_3}{\lambda_2\lambda_4\lambda_5} + \frac{\lambda_3}{\lambda_1\lambda_4\lambda_5} + \frac{\lambda_3}{\lambda_1\lambda_2\lambda_6} - \frac{1}{\lambda_2\lambda_3} - \\ - \frac{1}{\lambda_1\lambda_3} - \frac{1}{\lambda_3\lambda_5} - \frac{2}{\lambda_2\lambda_5} - \frac{2}{\lambda_1\lambda_5} - \frac{1}{\lambda_3\lambda_4} - \frac{2}{\lambda_2\lambda_4} - \frac{2}{\lambda_1\lambda_4} - \frac{1}{\lambda_3\lambda_6} - \frac{2}{\lambda_2\lambda_6} - \frac{2}{\lambda_1\lambda_6} - \\ - \frac{2}{\lambda_5\lambda_6} - \frac{\lambda_3}{\lambda_2\lambda_5\lambda_6} - \frac{\lambda_3}{\lambda_1\lambda_5\lambda_6} - \frac{2}{\lambda_4\lambda_6} - \frac{\lambda_3}{\lambda_2\lambda_4\lambda_6} - \frac{\lambda_3}{\lambda_1\lambda_4\lambda_6} - \frac{\lambda_3}{\lambda_4\lambda_5\lambda_6}\end{array}\right)\right) + p_1p_2p_3q_1\lambda_1\lambda_2\lambda_3\lambda_5\right],$$

$$h = -\left[\left(\begin{array}{c}\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5\lambda_6\times \\ \times\left(\begin{array}{c}\frac{\lambda_3}{\lambda_1\lambda_2} + \frac{\lambda_3}{\lambda_4\lambda_5} - \frac{1}{\lambda_3} - \frac{2}{\lambda_2} - \frac{2}{\lambda_1} - \frac{2}{\lambda_5} - \frac{\lambda_3}{\lambda_2\lambda_5} - \frac{\lambda_3}{\lambda_1\lambda_5} - \frac{2}{\lambda_4} - \\ - \frac{\lambda_3}{\lambda_2\lambda_4} - \frac{\lambda_3}{\lambda_1\lambda_4} - \frac{2}{\lambda_6} - \frac{\lambda_3}{\lambda_2\lambda_6} - \frac{\lambda_3}{\lambda_1\lambda_6} - \frac{\lambda_3}{\lambda_5\lambda_6} - \frac{\lambda_3}{\lambda_4\lambda_6}\end{array}\right)\end{array}\right) + \left(\begin{array}{c}p_1p_2p_3q_1\lambda_1\lambda_2\lambda_3\lambda_5\lambda_6\times \\ \times\left(\frac{1}{\lambda_5\lambda_6} + \frac{\lambda_3}{\lambda_4\lambda_5\lambda_6} - \frac{1}{\lambda_4\lambda_5}\right)\end{array}\right)\right],$$

$$w = \left[\left(\begin{array}{c}\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5\lambda_6\times \\ \times\left(2 + \frac{\lambda_3}{\lambda_2} + \frac{\lambda_3}{\lambda_1} + \frac{\lambda_3}{\lambda_5} + \frac{\lambda_3}{\lambda_4} + \frac{\lambda_3}{\lambda_6}\right)\end{array}\right) - \left(p_1p_2p_3q_1\lambda_1\lambda_2\lambda_3\lambda_5\lambda_6\times\left(-1 - \frac{\lambda_3}{\lambda_4} - \frac{\lambda_3}{\lambda_6}\right)\right) + p_4p_5p_6\lambda_3\lambda_4\lambda_6\right].$$

Плотность распределения вероятностей времени выполнения технологии тестирования уязвимости к SQL инъекциям равна:

$$\phi(x) = \left(2\pi i\right)^{-1}\times$$

$$\times\int_{-i\infty}^{i\infty}\frac{e^{zx}\left(vz^2 + bz + k\right)}{z^7 + rz^6 + cz^5 + dz^4 + gz^3 + hz^2 + wz + m}dz, \quad (3)$$

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

Тогда $\mathring{a}^{zx}\Phi(z)$ можно представить в виде:

$$e^{zx}\Phi(z) = e^{zx}\times$$

$$\times\frac{vz^2 + bz + k}{z^7 + rz^6 + cz^5 + dz^4 + gz^3 + hz^2 + wz + m} = \frac{\mu(z)}{\psi(z)}. \quad (4)$$

Тогда плотность распределения времени выполнения алгоритма тестирования уязвимости к SQL инъекциям равна:

$$\phi(x) = \sum_{k=1}^{7}\mathrm{Res}\left[e^{zx}\Phi(z)\right] = \sum_{k=1}^{7}\frac{\mu(z_k)}{\psi(z_k)} =$$

$$= \sum_{k=1}^{7}\frac{e^{zx}\left(vz^2 + bz + k\right)}{7z_k^6 + 6rz_k^5 + 5cz_k^4 + 4dz_k^3 + 3gz_k^2 + 2hz_k + w} \quad (5)$$

Многочлен

$$rz^6 + cz^5 + dz^4 + gz^3 + hz^2 + wz + m$$

порождает семь полюсов. Решение уравнения

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

может быть найдено любым методом, например, по формулам Виета [13]. В результате вычисляются особые точки z_1, z_2, z_3, z_4, z_5, z_6, z_7.

Таким образом, на основе экспоненциальной *GERT*-сети разработана математическая модель технологии тестирования уязвимости к SQL инъекциям, которая отличается от известных, усовершенствованным способом определения расстояния между результатами инъекции. Использование в предложенном способе критерия Джаро-Винклера, для сравнения результатов инъекции SQL кода и введение порогового значения позволит повысить точность результатов тестирования уязвимости к SQL инъекциям.

3. Исследования GERT-модели технологии тестирования уязвимости к SQL инъекциям

Рассмотрим пример атаки SQL инъекций. Суть таких инъекций – внедрение в данные (передаваемые через GET, POST запросы или значения Cookie) произвольного SQL кода.

Найдем плотности распределения $\phi(x)$ вероятностей времени выполнения алгоритма при условии, что z выбираются как корни уравнения $z^7 + rz^6 + cz^5 + dz^4 + gz^3 + hz^2 + wz + m = 0$, условные вероятности и интенсивности в ветвях *GERT*-сети имеют значения: $p_1 = p_2 = p_3 = p_4 = p_5 = 0,999999$, $p_6 = 0,9$, $\lambda_1 = \lambda_2 = \lambda_3 = 0,9999$, $\lambda_4 = 0,8$, $\lambda_5 = 0,1$, $\lambda_6 = 0,999999$.

С учетом приведенных признаков *GERT*-сети, в соответствии с выражением (2), а также используя математический пакет *Mathcad,* получим, что в знаменателе выражения (3) сформирован полином

$$x^7 - 0.1x^6 - 174x^5 + 2.471x^4 - 509x^3 + 128x^2 + 2.014x - 0.169 = 0. \quad (7)$$

Корни этого полинома (и соответственно функция Φ(z)) равны:

x1 ≈ −2.11254039, (P(x1) ≈ 0; iter = 1);
x2 ≈ −0.56188563, (P(x2) ≈ 0; iter = 4);
x3 ≈ −0.20818597 − i · 0.609441, (P(x3) ≈ 0; iter = 5);
x4 ≈ −0.20818588 + i · 0.609441, n(P(x4) ≈ 0; iter = 4);
x5 ≈ −0.10358122, (P(x5) ≈ 0; iter = 3);
x6 ≈ 1.6471895 − i · 0.7751076, (P(x6) ≈ 0; iter = 1);
x7 ≈ 1.6471895 + i · 0.7751076, (P(x7) ≈ 0; iter = 4).

На рис. 3 представлена кривая графика зависимости функции Φ(z) от *z* в рассматриваемых выше условиях. Как видно из рисунка случайная величина z распределена в соответствии с показательным законом.

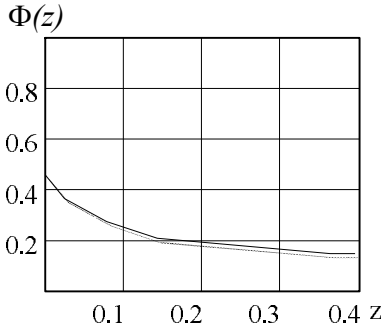

Рис. 3. График зависимости функции Φ(z) от интенсивности z

Найдем эту функцию и исследуем ее с использованием *Mathcad*. В соответствии с (5) $\phi(x)$ равна:

$$\phi(x) = \sum_{k=1}^{6} \operatorname{Res}_{z=z_k}\left[e^{zx}\Phi(z)\right] = \frac{e^{(a+\delta i)x}\left(v(a+\delta i)^2 + b(a+\delta i) + k\right)}{7u(a+\delta i)^6 + 6r(a+\delta i)^5 + 5c(a+\delta i)^4 + 4d(a+\delta i)^3 + 3g(a+\delta i)^2 + 2h(a+\delta i) + w} - \frac{e^{(a-bi)x}\left(v(a-\delta i)^2 + b(a-\delta i) + k\right)}{7u(a-\delta i)^6 + 6r(a-\delta i)^5 + 5c(a-\delta i)^4 + 4d(a-\delta i)^3 + 3g(a-\delta i)^2 + 2h(a-\delta i) + w}. \quad (8)$$

Используя выражения Эйлера [13], получим:

$$\phi(x) = \sum_{k=1}^{6} \operatorname{Res}_{z=z_k}\left(e^{zx}\hat{O}(z)\right) = e^{(a+\delta i)x}\frac{\tau + i\beta}{\gamma + i\theta} + e^{(a-\delta i)x}\frac{\tau - i\beta}{\gamma - i\theta} = \frac{2e^{ax}}{\gamma^2 + \theta^2}\left((\tau\gamma + \beta\theta)\cos(\delta x) + (\tau\gamma - \beta\theta)\sin(\delta x)\right), \quad (9)$$

где $\tau = a^2v - \delta^2 v + ab + k$, $\beta = 2a\delta v - \delta b$, $\gamma = 7ua^6 - 10ua^4\delta^2 + 105ua^2\delta^4 - 7u\delta^6 + 6ra^5 - 60ra^3\delta^2 + 30ra\delta^4 + 5ca^4 - 30ca^2\delta^2 + 5c\delta^4 + 4da^3 - 12da\delta^2 + 3ga^2 - 3g\delta^2 + 2ha + w$,

$\theta = 49ua^5\delta - 140ua^3\delta^3 + 49ua\delta^5 + 30ra^4\delta - 60ra^2\delta^3 + 6r\delta^5 + 20ca^3\delta - 20ca\delta^3 + 12da^3\delta - 4d\delta^3 + 6ga\delta + 2h\delta$.

На рис. 4 представлены кривые плотности распределения $\phi(x)$ вероятностей времени выполнения технологии тестирования уязвимости к SQL инъекциям для приведенных выше условий (в качестве входных данных использовались корни полинома (7)). При этом рис. 4, а соответствует случаю когда в качестве входных данных $(a + \delta i)$ использовалось значение x1; рис. 4, б соответственно случаю использования x2; рис. 4, в – использовалось значение x3. Рис. 4, д соответствует случаю когда в качестве входных

данных использовалось значение x4; рис. 4, е – x5; рис. 4, ж – x6; рис. 4, з – x7 соответственно.

Как и рассматриваемом выше примере, внешний вид кривых графиков рис. 4 дает основания предположить, что не все найденные выше решения (корни полинома (7)) применимы при математическом и имитационном моделировании в качестве входных данных.

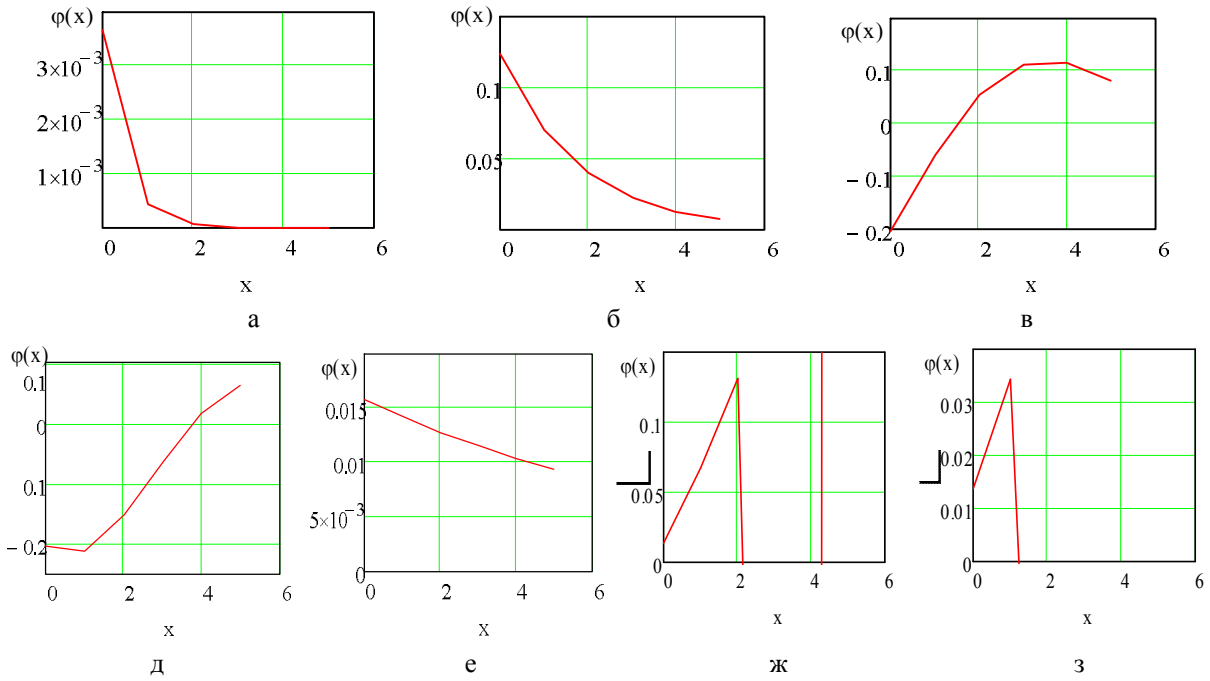

Рис. 4. Графики плотности распределения $\phi(x)$ – вероятности времени выполнения технологии тестирования уязвимости к SQL инъекциям

Так значения x3, x4, x6 и x7 невозможно в дальнейшем использовать при анализе и моделировании.

В то же время внешний вид графиков, полученных для значений x1, x2 и x5 дает основания предположить, что случайная величина времени выполнения технологии тестирования уязвимости к SQL инъекциям соответствует гамма-распределению (близкое к экспоненциальному).

Результаты проверки этой гипотезы по критерию χ^2 Пирсона [14] подтвердили ее правдоподобность.

Так при достаточно большом значении доверительной вероятности $Q = 0{,}95$ для всех рассматриваемых x1, x2 и x5 соответствующие значения χ^2 ($\chi_1^2 = 19{,}3$, $\chi_2^2 = 15{,}1$, $\chi_5^2 = 25{,}6$) << $\overline{\chi^2} = 101{,}9$.

Выводы

В работе разработан комплекс математических моделей технологии тестирования WEB-приложений. В основу математического моделирования положен подход GERT-сетевого синтеза. В результате разработана математическая модель технологии тестирования уязвимости к SQL инъекциям.

Математическая модель технологии тестирования уязвимости к SQL инъекциям отличается от известных, усовершенствованным способом определения расстояния между результатами инъекции. Использование в предложенном способе критерия Джаро-Винклера, для сравнения результатов инъекции SQL кода и введение порогового значения позволит повысить точность результатов тестирования безопасности 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.

Список литературы

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-rus.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/103Testing_for_SQL_Injection_(OTG-INPVAL-005).

7. Cohen W., Ravikumar P., Fienberg S. A Comparison of String Metrics for Matching Names and Records [Электронный ресурс] / William W. Cohen, Pradeep Ravikumar, Stephen E. Fienberg. Режим доступа: https://www.cs.cmu.edu/afs/cs/Web/People/wcohen/postscript/kdd-2003-match-ws.pdf.

8. Kevin Dreßler a , Axel-Cyrille Ngonga Ngomo On the Efficient Execution of Bounded Jaro-Winkler Distances / Semantic Web – Interoperability, Usability, Applicability an IOS Press Journal Электронный ресурс http://www.semantic-web-journal.net/system/files/swj944.pdf

9. Pritsker A. A. B., Happ W. W. GERT: Graphical Evaluation and Review Technique. Part I. Fundamentals // The Journal of Industrial Engineering (May 1966).

10. Pritsker, A. A. B. Modeling and analysis using Q-GERT networks New York: Wiley : Distributed by Halsted Press, 1979.

11. Семенов С.Г. Gert-модель прогнозування параметрів функціональної безпеки технічних систем / С.Г.Семенов, Гавриленко С.Ю., Кассем Халіфе // Зб. наукових праць. Системи обробки інформації. – Х.: ХУ ПС, 2016. – Вип. 2(139) С.50-52.

12. Semenov S.G., Zmiyevskaya V N., Kassem Khalife Development of Gert model of management system by using test cases // Journal of Qafqaz university-mathematics and computer science 2016, Vol.(4), № 1. С. 52-59

13. Эдвардс Г. Последняя теорема Ферма. Генетическое введение в алгебраическую теорию чисел / Г. Эдвардс. – М.: Мир, 1980. – 486 с.

14. Гмурман В.Е. Теория вероятностей и математическая статистика / В.Е. Гмурман. – М.: Высшая школа, 2003. – 479 с.

15. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.А. Смирнов, А.В. Коваленко // Збірник наукових праць "Системи обробки інформації". – Випуск 5(142). – Х.: ХУПС – 2016. – С. 153-157.

16. Коваленко А.В. Проблемы анализа и оценки рисков информационной деятельности / А.А. Смирнов, А.В. Коваленко, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць "Системи обробки інформації". – Випуск 3(140). – Х.: ХУПС – 2016. – С. 40-42.

17. Коваленко А.В. Метод качественного анализа рисков разработки программного обеспечения / А.А. Смирнов, А.В. Коваленко, Н.Н. Якименко, А.П. Доренский // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 2(23). – Харків: ХУПС. – 2016. – С. 150-158.

18. Коваленко А.В. Алгоритм анализа уязвимости SQL Injection для управления рисками разработки программного обеспечения / А.А. Смирнов, А.В. Коваленко, А.С. Коваленко // Збірник тез другої міжнародної науково-технічної конференції «Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2017). м. Харків.10-12 квітня 2017 р. – Харків: НТУ «ХПІ». – 2017. – С. 27.

19. Коваленко А.В. Метод управления рисками разработки программного обеспечения на основе алгоритмов анализа уязвимостей / А.А. Смирнов, А.В. Коваленко, А.С. Коваленко // Збірник тез Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології» (IS&CT). м. Кіровоград. 20-22 квітня 2017 р. – Кіровоград: КНТУ. – 2017. – С. 92.

*20. Коваленко А.В. Алгоритмы анализа DOM XSS уязвимости и уязвимости SQL Injection при управлении рисками разработки программного обеспечения / А.А. Смирнов, А.В. Коваленко, А.С. Коваленко // Збірник тез **IX** міжнародної науково-практичної конференції "Проблеми і перспективи розвитку IT-індустрії". м. Харків. 20-21 квітня 2017 р. – Харків: ХНЕУ. – 2017. – С. 61.*

Надійшла до редколегії 12.07.2017

Рецензент: д-р техн. наук, с.н.с. С.Г. Семенов, Національний технічний університет «Харківський політехнічний інститут», Харків.

ТЕХНОЛОГІЯ ТЕСТУВАННЯ ВРАЗЛИВОСТІ ДО SQL ІН'ЄКЦІЇ

О.В. Коваленко

В роботі представлені результати дослідження та алгоритми тестування на вразливість до одним з найбільш поширених видів атак на Web-додатки – SQL ін'єкції. Аргументовано обраний підхід математичного моделювання на основі GERT-мереж. Розроблено комплекс математичних моделей технології тестування Web-додатків. В основу математичного моделювання покладено підхід GERT-мережевого синтезу. В результаті розроблено математичні моделі технології тестування уразливості до SQL ін'єкцій. Математична модель технології тестування уразливості до SQL ін'єкцій відрізняється від відомих, вдосконаленим способом визначення відстані між результатами ін'єкції.

***Ключові слова:** уразливості до SQL ін'єкцій, GERT-мережі, уразливості безпеки*

TECHNOLOGY OF VULNERABILITY TESTING TO SQL INJECTIONS

O.V. Kovalenko

The paper presents research results and vulnerability testing algorithms for one of the most common types of attacks on Web applications-SQL injections. The approach of mathematical modeling on the basis of GERT-networks is chosen. A set of mathematical models for testing Web applications has been developed. The basis of mathematical modeling is the approach of GERT-network synthesis. As a result, mathematical models have been developed for testing vulnerability to SQL injections. The mathematical model of vulnerability testing technology for SQL injections differs from the known ones, an improved method for determining the distance between injection results.

***Keywords:** vulnerabilities to SQL injections, GERT-networks, security vulnerabilities.*