УДК 519.6 DOI: 10.14529/ctcr170205

УПРАВЛЕНИЕ ПРОЦЕССОМ ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЙ МЕТОДОМ ФАЗЗИНГА НА ОСНОВЕ ДИНАМИЧЕСКИХ БАЙЕСОВСКИХ СЕТЕЙ

С.А. Баркалов[1], Т.В. Азарнова[2], П.В. Полухин[2]

[1] Воронежский государственный технический университет, г. Воронеж,
[2] Воронежский государственный университет, г. Воронеж

В настоящий период времени ведутся интенсивные исследования в области создания результативных технологий тестирования веб-приложений на наличие уязвимостей; одной из таких технологий, позволяющих проводить комплексное тестирование на всех этапах жизненного цикла приложения, является тестирование методом фаззинга. Актуальным направлением развития данной технологии является разработка математического и программного обеспечения, реализующего интеллектуальные компоненты фаззинга, внедрение которых позволит существенно повысить его результативность и ресурсную эффективность. В статье представлена концептуальная модель применения аппарата динамических байесовских сетей для управления тестированием веб-приложений методом фаззинга. В рамках построенной концептуальной модели разработаны динамические байесовские модели для основных OWASP – классов уязвимостей веб-приложений и соответствующее алгоритмическое и программное обеспечение для проведения тестирования.

Ключевые слова: OWASP – классы уязвимостей веб-приложений, управление тестированием веб-приложений, динамическая байесовская сеть, алгоритмы обучения и вероятностного вывода.

Введение

Информационные системы и технологии, реализующие проектирование, создание, обновление и настройку взаимодействия Web-приложений в совокупности можно рассматривать как многофункциональную платформу. Компоненты платформы непрерывно развиваются, но темпы их развития носят достаточно несогласованный характер. Отсутствует единый системный механизм управления развитием платформы, что приводит к проблемам, связанным с безопасностью функционирования веб-приложений. Для анализа и поиска путей решения данных проблем М. Керфи, Д. Гровс создали открытый проект безопасности веб-приложений OWASP. Задачи OWASP связаны со структуризацией, подробной характеристикой ошибок, анализом причин их возникновения, оценкой критичности и масштаба распространения и разработкой новых технологий обнаружения ошибок. Основными механизмами обнаружения ошибок безопасности являются механизмы тестирования. Выделяются механизмы тестирования: функциональности, отказоустойчивости, интерфейсов и системных компонентов, совместимости, портативности и способности к взаимодействию. Среди современных эффективных технологий тестирования безопасности веб-приложений можно выделить технологию тестирования методом фаззинга [11]. В работе будет использоваться широко распространенное определение фаззинга, представленное М. Саттоном и П. Амини на конференции Black Hat: фаззинг – метод обнаружения ошибок в программном обеспечении, заключающийся в подаче на вход исследуемого объекта заведомо некорректных данных с целью вызова события сбоя или ошибки. Фаззинг позволяет прогнозировать наличие ошибок и проводить анализ того, какие именно входные данные могут их вызвать. Выделяют порождающий и мутационный фаззинг [5]. При использовании порождающего фаззинга входные данные генерируются случайным образом, наборы тестов не связаны друг с другом. Апробация данного метода показывает, что он недостаточно эффективен для тестирования сложных современных приложений. Мутационный метод позволяет изменять и приспосабливать входные данные с учетом специфики функционирования целевого приложения. Данный метод более сложен в реализации, но он позволяет расширить возможности поиска и находить ранее недокументированные ошибки. Актуальным направлением исследований является разработка технологий, способных управлять процессом тестирования и комбинировать методы порождающего и мутационного фаззинга. Для создания подобных технологий требуются научно-обоснованные меха-

низмы, учитывающие стохастический характер фаззинга, и опирающиеся на результаты концептуального, имитационного, стохастического моделирования и интеллектуального анализа данных [6]. В статье предложена технология управления тестированием веб-приложений методом фаззинга, базирующаяся на интеллектуальном аппарате динамических байесовских сетей [3]. Построена концептуальная модель применения аппарата динамических байесовских сетей для управления процессом тестирования веб-приложений методом фаззинга. В рамках описанной концептуальной модели реализованы инструменты управления процессом тестирования для десяти основных OWASP – классов ошибок безопасности веб-приложений [1]. Инструменты доведены до алгоритмической и программной реализации и апробированы в рамках специально спроектированного вычислительного эксперимента. Применение динамических байесовских моделей в управлении процессом тестирования позволит: спроектировать данный процесс в виде единой иерархической структуры, смоделировать потоки информации, управляющие воздействия и механизмы взаимодействия между подпроцессами тестирования; формализовать отношения между элементами иерархической структуры на языке условно-вероятностных зависимостей. Предложенные инструментальные средства расширяют возможности фаззинга, способствуют выходу его за границы случайного тестирования и созданию механизмов прогнозирования и самообучения.

Концептуальная модель применения аппарата динамических байесовских сетей для управления процессом тестирования веб-приложений методом фаззинга

Управление тестированием методом фаззинга представляет собой комплекс мероприятий по организации и управлению процессами и компонентами тестирования. Основные задачи управления тестированием: координация активных действий; выявление зависимостей и связей между наборами тестов; определение, измерение и отслеживание показателей качества. Структуризация управления процессом тестирования методом фаззинга приведена на рис. 1.

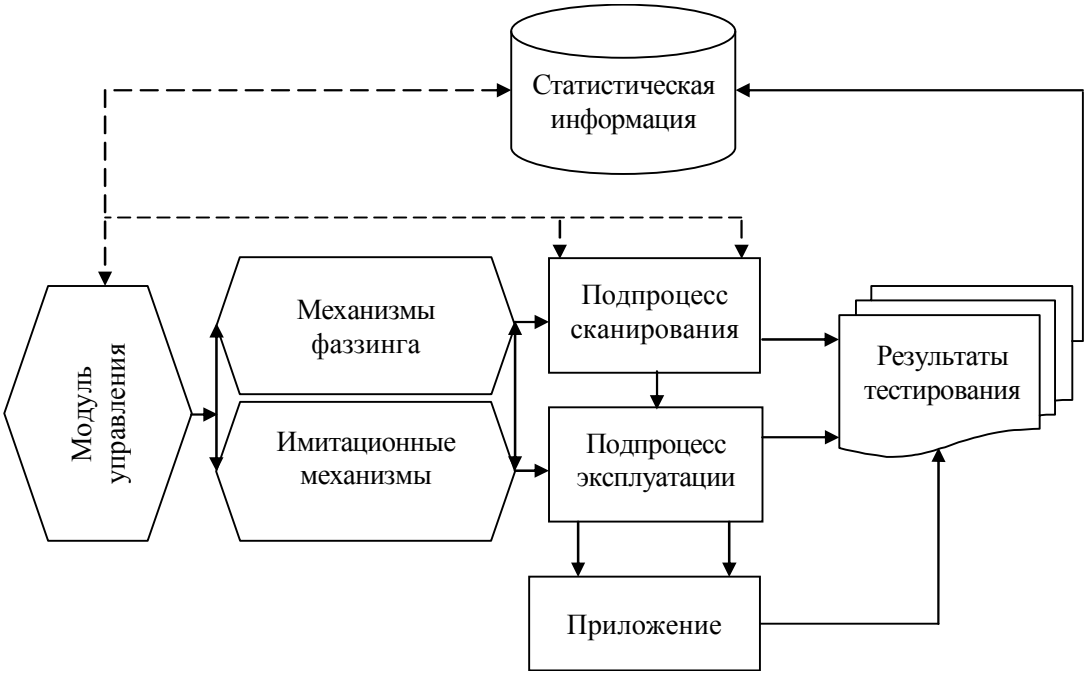

Рис. 1. Структуризация управления процессом тестирования методом фаззинга

Предложенная в рамках исследования концептуальная модель применения аппарата динамических байесовских сетей для управления процессом тестирования веб-приложений методом фаззинга (рис. 2) предназначена для того, чтобы показать, как все аспекты управления могут быть реализованы формализованными средствами динамических байесовских сетей. Концептуальная модель отражает подготовку аналитической базы веб-приложений, необходимой для тестирования: выработку требований к критерию однородности веб-приложений в рамках первоначальной процедуры семантического анализа и определения структуры веб-приложений; оценку степени однородности компонентов (библиотек) входящих в состав веб-приложений.

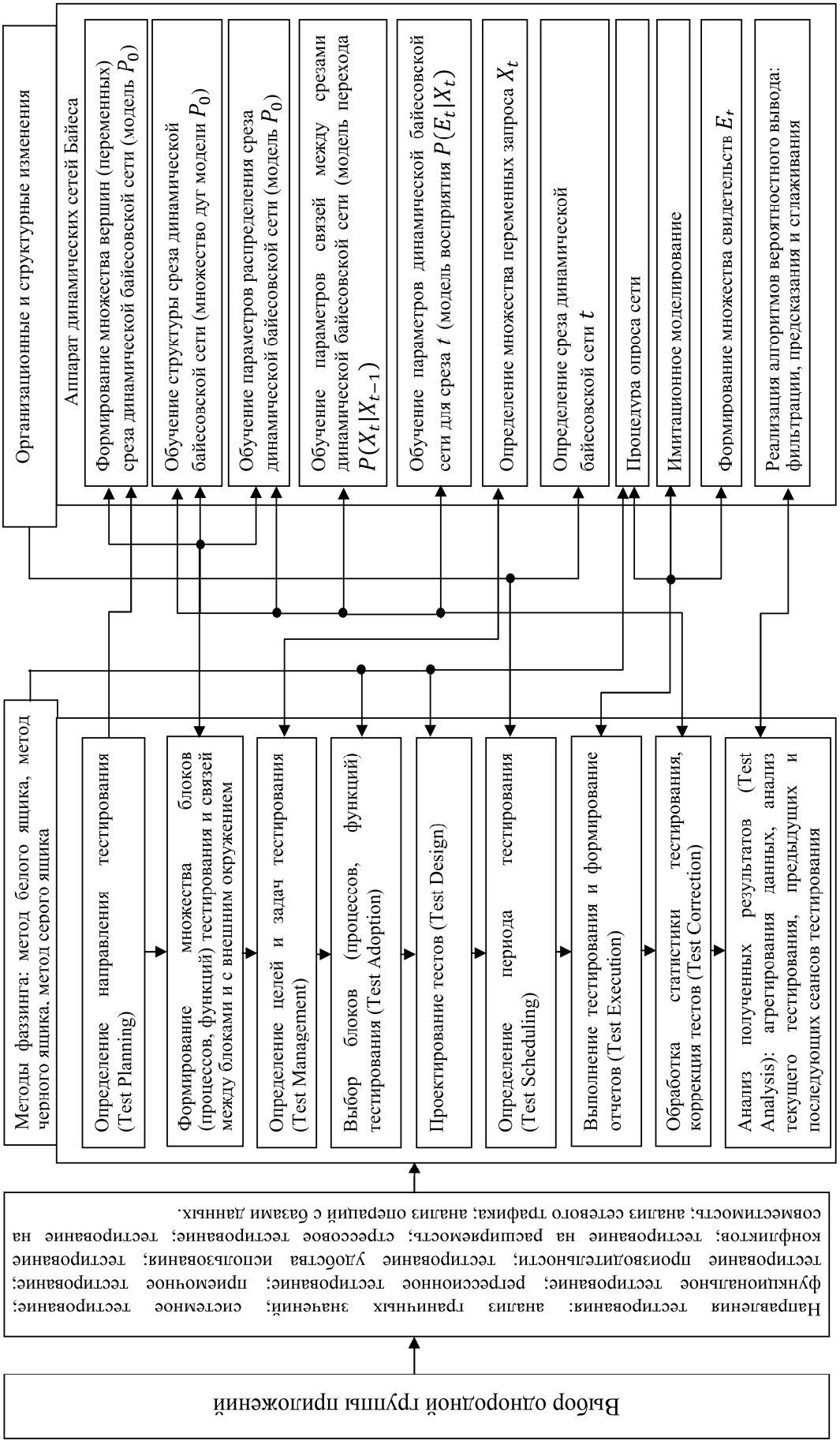

Рис. 2. Концептуальная модель применения аппарата динамических байесовских сетей в управлении процессом тестирования

Реализуется системный подход к управлению, определяются: компоненты тестирования, внутренние информационные и логические связи между компонентами тестирования в пространстве и во времени, связи между компонентами тестирования и окружающей средой, подпроцессы тестирования, инструменты тестирования, результаты тестирования. Инструментальные средства байесовских сетей позволяют осуществить данный системный подход, формализовать вероятностный характер результатов тестирования, вписать в общую иерархическую вычислительную структуру представленные выше системные элементы и пространственные и временные связи между ними, и формируют возможность на базе агрегированного представления проводить ретроспективный анализ и прогнозировать вероятности интересующих исследователя результатов тестирования.

Алгоритмическое обеспечение процесса управления тестированием на основе аппарата динамических байесовских сетей

Кратко остановимся на основных аспектах работы с динамическими байесовскими моделями для решения поставленной в исследовании задачи. Байесовская модель [8, 10] – графическая стохастическая модель, представленная орграфом не содержащим циклов, вершины которого являются случайными величинами (дискретными и (или) непрерывными), а дуги, ведущие от вершины x к вершине y, означают что $x \in Parents(y)$, т. е. x является родительской для y. Вершинам x ставится в соответствие условное распределение вероятностей $P(x_i|Parents(x_i))$. Динамические байесовские сети представляют собой последовательности взятых в хронологическом порядке байесовских сетей, соединенных логико-вероятностными связями. Вершины динамических сетей (переменные) описывают пространственно-временные состояния исследуемого процесса. Множество вершин $Z_t = \{X_t, E_t, Y_t\}$ представляется с помощью трех подмножеств: Y_t – скрытые переменные, X_t – переменные наблюдения, E_t – переменные свидетельств. Для динамических байесовских сетей управления процессом тестирования методом фаззинга переменные представляют собой описанные в концептуальной модели элементы формализованного представления процедуры тестирования. Временные срезы динамической байесовской сети интерпретируются как проведенные в некоторый момент времени эксперименты по тестированию. В исследовании разработаны динамические байесовские сети для управления процессом тестирования десяти наиболее известных уязвимостей, представленных консорциумом OWASP. На рис. 3 и 4 представлены фрагменты графов динамических байесовских сетей управления процессом тестирования методом фаззинга SQL-инъекций и межсайтового скриптинга (XSS) веб-приложений. Обозначения для данных моделей приведены в табл. 1, 2.

Для построения байесовских сетей нами используется алгоритм Перла [2, 9]. Сущность алгоритма заключается в процедуре добавления дуг из множества родительских вершин для каждого узла сети, так что вершина будет условно независимой от всех предшествующих узлов, не являющихся его родителями. Улучшить структуру построенных динамических байесовских сетей можно за счет применения специальных алгоритмов обучения структуры сети, но в рамках проведенного исследования обучение не проводилось.

Предполагается, что процессы управления являются Марковскими процессами первого рода [7], для которых:

$$P(X_t|X_{0:t-1}) = P(X_t|X_{t-1}),\ \ P(E_t|X_{0:t}, E_{0:t-1}) = P(E_t|X_t).$$

Полное совместное распределение вероятностей для рассматриваемых динамических байесовских сетей имеет вид

$$P(X_0, X_1, \dots, X_t, E_1, \dots, E_t) = P(X_0)\prod_{i=1}^{t} P(X_i|X_{i-1})P(E_i|X_i),$$

где X_t, E_t – соответственно множество ненаблюдаемых и наблюдаемых (свидетельств) переменных для момента времени, $P(X_0)$ – начальное распределение вероятностей, $P(X_i|X_{i-1})$ – модель перехода, $P(E_i|X_i)$ – модель восприятия.

Как отражено в концептуальной модели основные задачи управления тестированием формализуются в виде задач фильтрации, предсказания и сглаживания для динамических байесовских сетей (табл. 3).

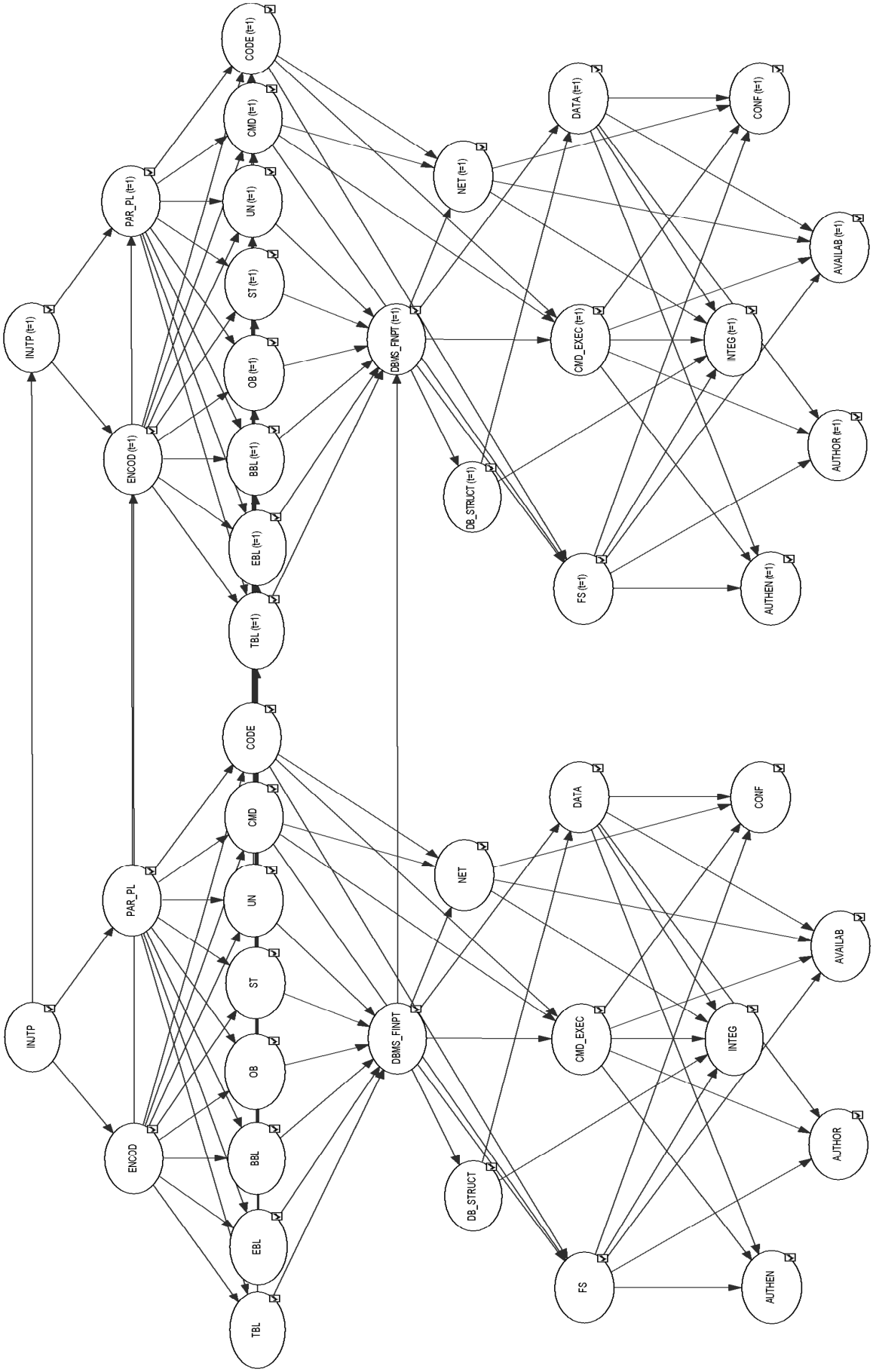

Рис. 3. Фрагмент динамической байесовской модели для управления процессом тестирования SQL-инъекций методом фаззинга

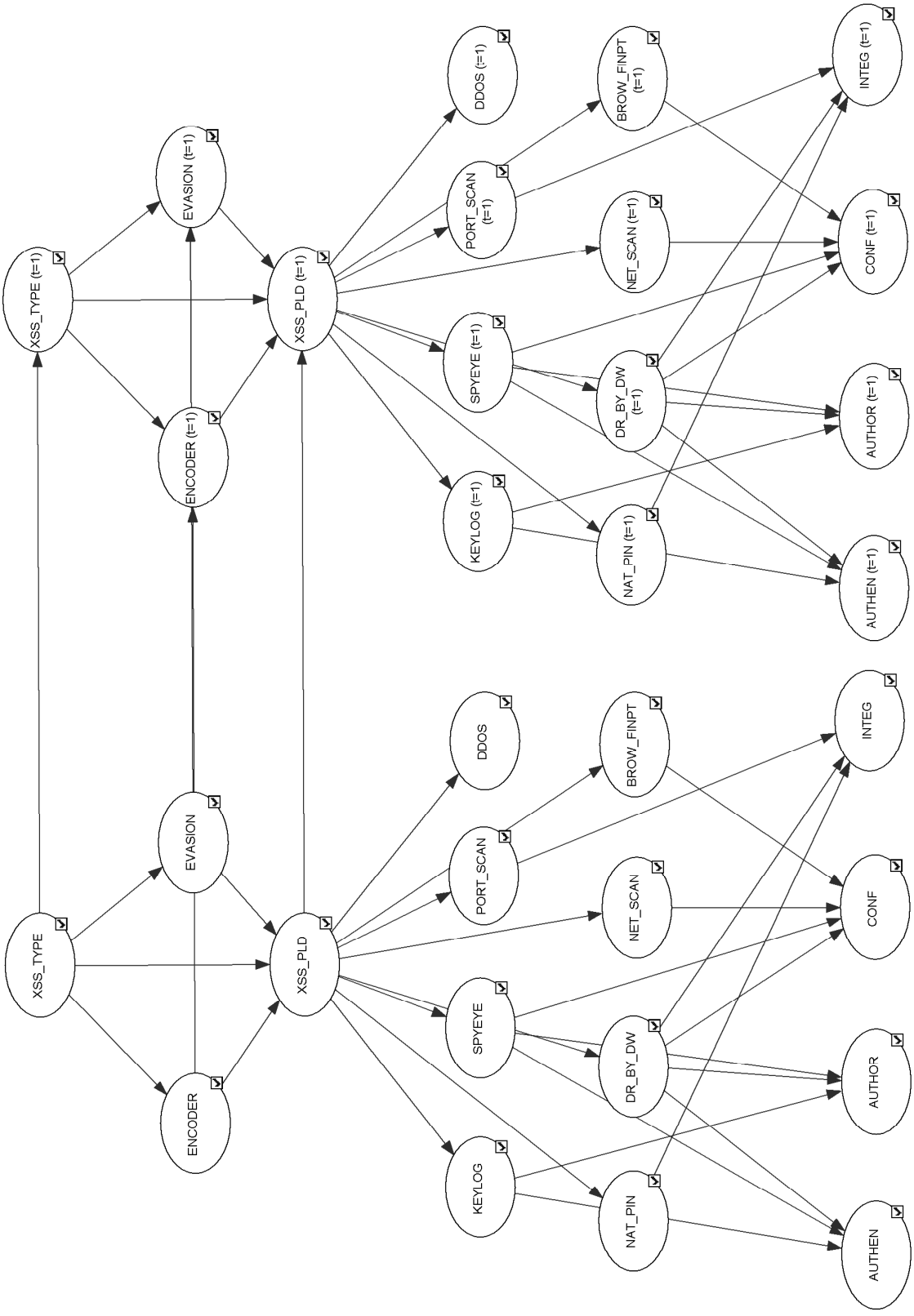

Рис. 4. Фрагмент динамической байесовской модели управления процессом тестирования межсайтового скриптинга методом фаззинга

Таблица 1

Обозначения к рис. 3

Название узла	Характеристика
INJTP	Определение типа инъекции: SQL, команд, кода
ENCOD, PAR_PL	Механизмы кодирования обхода межсетевых экранов веб-приложений (WAF)
UN, BBL, TBL, EBL, SQ, OB	Различные типы инъекций: Time Based blind, Boolean Based Blind, Error Based Blind, Out Of Band, Union injection, Stacked Time
CMD, CODE	Инъекции команд и кода
DBMS_FINPT	Определение типа и версии СУДБ, установленной на сервере
CMD_EXEC	Исполнение команд ОС и команд внутри инъекции кода и SQL инъекции
NET	Получение доступа к компонентам сети из командного интерфейса СУБД
DB_STRUCT	Получение структуры таблиц и баз данных СУБД
DATA	Получения данных хранящихся в таблицах базы данных
FS	Возможность удаленной загрузки файлов через внутренние механизмы СУБД
CONF, AUTHEN, AUTHOR, INTEG, AVAILAB	Нарушение механизмов аутентификации, авторизации, целостности, конфиденциальности и доступности

Таблица 2

Обозначения к рис. 4

Название узла	Характеристика
XSS_TYPE	Виды XSS направленные на внедрение JavaScript кода на страницу пользователя или в хранилище данных
ENCODER	Механизмы кодирования XSS для обхода программных фильтров
EVASION	Механизмы запутывания XSS для обхода программных фильтров
XSS_PLD	Тип полезной нагрузки, используемый XSS (html тэги, обработчики событий)
KEYLOG	Механизмы запоминания комбинаций клавиш, нажатых пользователем
SPY_EYE	Методика получения снимка страницы веб-браузера пользователя через XSS
DDOS	Механизм использования браузера пользователя в качестве составного элемента атаки отказа в обслуживания на сторонние ресурсы
PORT_SCAN	Механизм сканирования открытых портов на компьютере пользователя (запущенные службы и процессы)
NET_SCAN	Сканирование локальной сети, внутри которой находится пользователь, построение топологии сети
NAT_PIN	Механизм обхода сетевых правил маршрутизатора NAT, проникновение в локальную сеть
DR_BY_DW	Перенаправление пользователя на ресурсы, содержащие вредоносные программы и вирусы
BROW_FINPT	Получение отпечатка браузера пользователя (установленные плагины, компоненты, расширения)
AUTHEN, AUTHOR, INTEG, CONF	Нарушение механизмов аутентификации, авторизации, целостности и конфиденциальности данных

Таблица 3

Характеристика задач фильтрации, предсказания и сглаживания байесовских сетей

Задача	Характеристика
Фильтрация	Задача вычисления апостериорных вероятностей $P(X_t \mid E_{1:t})$ переменных текущего состояния при условии наличия всех свидетельств $E_{1:t}$, начиная с начального момента $t=1$ и до текущего момента времени t. Данная задача решается рекурсивным способом, распределение вероятностей для текущего момента времени проектируется вперед от t к $t+1$, далее, используя новое свидетельство для момента времени $t+1$, распределение вероятностей обновляется: $P(X_{t+1} \mid E_{1:t+1}) = P(X_{t+1} \mid E_{1:t}, E_{t+1}) = \alpha P(E_{t+1} \mid X_{t+1}) P(X_{t+1} \mid E_{t:1}) =$ $= \alpha P(E_{t+1} \mid X_{t+1}) \sum_{X_t} P(X_{t+1} \mid X_t) P(X_t \mid E_{1:t})$
Предсказание	Задача вычисления распределения апостериорных вероятностей $P(X_{t+k} \mid E_{1:t})$ значений переменных в будущем состоянии, если даны все свидетельства, полученные к данному моменту времени. Задача решается рекурсивным вычислением вероятностного распределения в момент $t+k+1$ на основании предсказания для $t+k$: $P(X_{t+k+1} \mid E_{1:t}) = \sum_{X_{t+k}} P(X_{t+k+1} \mid X_t) P(X_{t+k} \mid E_{1:t})$
Сглаживание (ретроспективный анализ)	Задача вычисления апостериорных вероятностей значений переменных $P(X_k \mid E_{1:t})$, относящихся к прошлому состоянию, если даны все свидетельства вплоть до нынешнего. Вычисление осуществляется следующим образом: $P(X_k \mid E_{1:t}) = \alpha P(X_k \mid E_{1:k}) P(E_{k+1:t} \mid X_k)$, $P(E_{k+1:t} \mid X_k) = \sum_{X_{k+1}} P(E_{k+1:t} \mid X_k, X_{k+1}) P(X_{k+1} \mid X_k) =$ $= \sum_{X_{k+1}} P(E_{k+1} \mid X_{k+1}) P(E_{k+2:t} \mid X_{k+1}) P(X_{k+1} \mid X_k)$

Для решения приведенных в таблице задач вероятностного вывода в исследовании используется алгоритм фильтрации частиц. Данный метод осуществляется за счет генерации выборок переменных развернутой по временным срезам байесовской сети, которые взвешиваются с позиции их правдоподобия по отношению к наблюдаемым свидетельствам. Выборки формируются, переходя последовательно в топологическом порядке от одной переменной состояния к следующей вдоль всей сети. Приведем укрупненную схему алгоритма фильтрации частиц, используемого в исследовании:

Шаг 1. На начальном срезе из распределения $P(X_0)$ *одновременно генерируется* N *выборок.*

Шаг 2. Вводится множества свидетельств для всех срезов сети $E_1, E_2, ..., E_T$.

Шаг перехода от временного среза t *к временному срезу* $t+1$. *Через модель перехода* $P(X_{t+1} \mid X_t)$ *осуществляется обновление множества выборок:* $N(X_{t+1} \mid E_{1:t}) = \sum_{X_t} P(X_{t+1} \mid X_t) N(X_t \mid E_{1:t})$

($N(X_t \mid E_{1:t})$ *– количество выборок для состояния* X_t *после получения свидетельств* $E_{1:t}$*)* → *выборки взвешиваются с учетом правдоподобия по отношению к новым свидетельствам* E_{t+1}, *им присваивается вес* $P(E_{t+1} \mid \mathrm{X}_{t+1})$ → *вычисляется суммарный вес выборок в состоянии* X_{t+1}

после получения свидетельств E_{t+1}: $w\left(X_{t+1}\middle|E_{1:t+1}\right)=P\left(E_{t+1}\middle|X_{t+1}\right)P\left(X_{t+1}\middle|E_{1:t}\right)$ → *отбрасываются выборки с малым весом* → *формируются новые* N *выборок, каждая выборка тиражируется пропорционально её весу.*

Для сокращения переменных в выборках в алгоритме используется подход, построенный на основании теоремы Рао – Блэкуэлла. Теорема Рао – Блекуэлла [2, 4], представляет собой утверждение математической статистики, позволяющее улучшить статистические свойства оценок параметров распределения за счет использования достаточных статистик. Основная идея алгоритма фильтрации частиц с применением теоремы Рао – Блекуэлла заключается в разделении множества скрытых переменных Z_t на X_t и R_t, при этом модель перехода определяется следующим выражением

$$P\left(Z_t|Z_{t-1}\right)=P\left(X_t|R_{t-1:t},X_{t-1}\right)P\left(R_t|R_{t-1}\right).$$

Условное распределение вероятностей $P(X_{0:t}\mid Y_{1:t},R_{0:t})$ может быть найдено аналитическим способом, а оценка $P(R_{0:t}\mid Y_{0:t})$ определена из следующего рекурсивного выражения

$$P\left(R_{0:t}|Y_{0:t}\right)=\frac{P\left(Y_t|Y_{1:t-1},R_{0:t}\right)P\left(R_t|R_{t-1}\right)P\left(R_{0:t-1}|Y_{1:t-1}\right)}{P\left(Y_t|Y_{1:t-1}\right)}.$$

Ниже приведен алгоритм фильтрации частиц с применением теоремы Рао – Блекуэлла.

1. Последовательная выборка по значимости

– Инициализация

Цикл $i = 1\ldots N$

Порождение выборки $R_0^{\,i}\sim P(R_0)$

Вычисление весов $W_0^i=P(Y_0\mid R_0)$

Нормализация весов $\tilde{W}_0^i=\dfrac{W_0^i}{\sum_{j=1}^{n}W_0^j}$

– Итерация

Цикл $i = 1\ldots N$

Порождение выборки

$R_t^i\sim Q(R_t|R_{0:t-1},Y_{1:t})$ и $R_{0:t}^i=(R_{t-1}^i,R_t^i)$

Вычисление весов

$$W_t^i=\frac{P(R_{0:t}^i\mid Y_{1:t})}{Q(R_t^i\mid R_{0:t-1}^i,Y_{1:t})P(R_{0:t-1}^i,Y_{1:t-1})}$$

Цикл $i = 1\ldots N$

Нормализация весов $\tilde{W}_t^i=\dfrac{W_t^i}{\sum_{j=1}^{n}W_t^j}$

2. Выборка

Перемножаем выборку $R_{0:t}^i$ *с весом* $\tilde{W}_t^i$ *распределения по значимости*

3. Применение метода Монте-Карло с применением цепей Маркова

Применение цепей Маркова в виде модели перехода $P(R_t^i\mid R_{0:t-1}^i)$ *для получения* $R_{0:t}^i$

В рамках исследования разработан инструментарий тестирования веб-приложений на основе предложенной математической модели, а также параллельные алгоритмы: фильтрации частиц и фильтрации частиц с применением теоремы Рао – Блекуэлла. На рис. 5 произведено сравнение этих двух алгоритмов с точки зрения временных затрат и ресурсной эффективности этих двух алгоритмов вероятностного вывода в процессе их функционирования в рамках концептуальной модели управления тестированиям веб-приложений. Анализируя данные сравнения двух алгоритмов, видно, что алгоритмы вероятностного вывода на основе теоремы Рао – Блекуэлла доказывают свою высокую ресурсную и вычислительную эффективность.

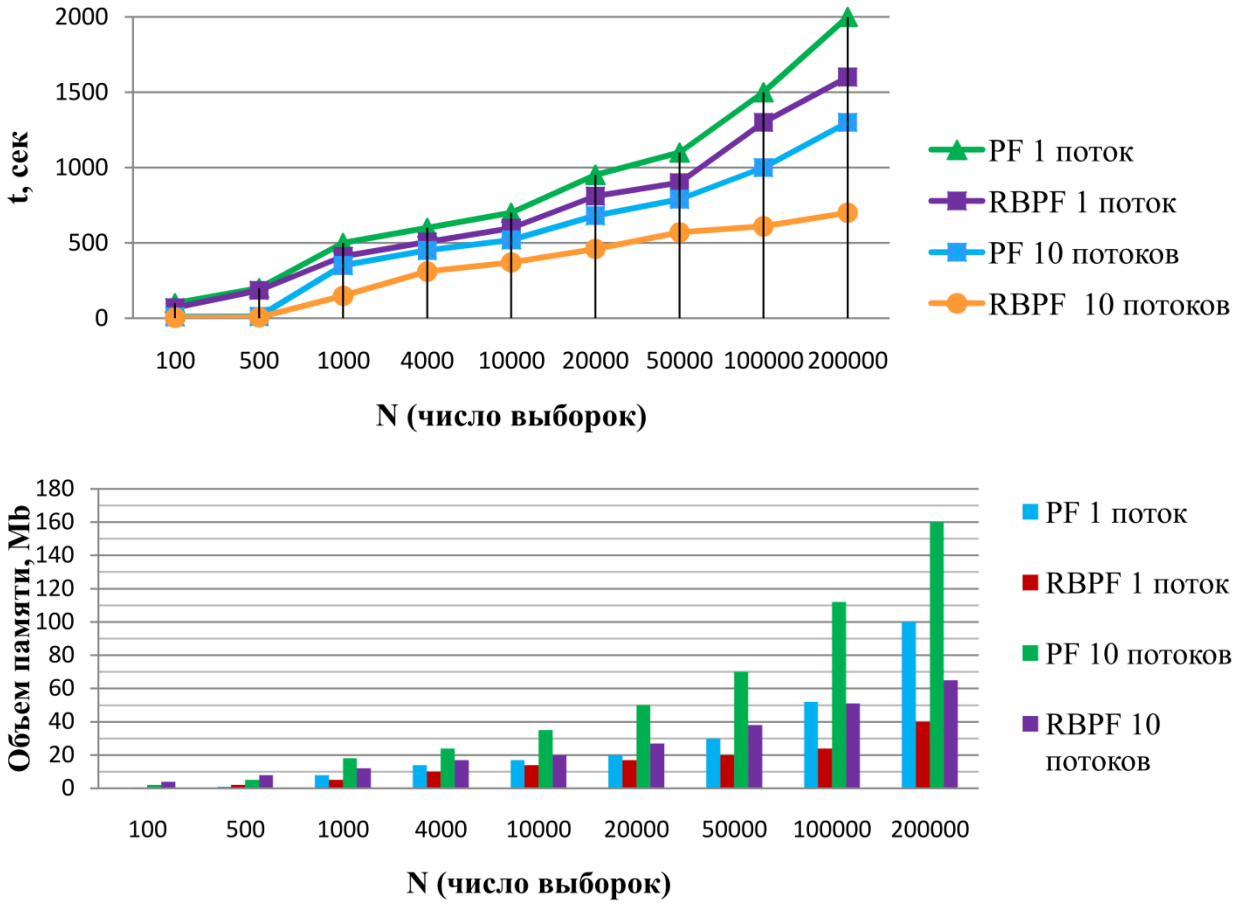

Рис. 5. Сравнение алгоритмов: фильтрации частиц и фильтрации частиц с применением теоремы Рао – Блэкуэлла (PF – алгоритм фильтрации частиц, RBPF – алгоритм фильтрации частиц с применением теоремы Рао – Блекуэлла)

Снижение общего числа выборок, улучшение статистических показателей оценки выборок с точки зрения среднеквадратичного отклонения в рамках реализация алгоритма фильтрации частиц с применением теоремы Рао-Блекуэлла позволяет использовать данный алгоритм для динамических байесовских сетей со сложно топологической структурой и большим доменом возможных значений, а также повысить ресурсную эффективность алгоритмов на основе теоремы Рао – Блекуэлла.

Заключение

В рамках исследования была разработана структура, аппаратные средства и проведен вычислительный эксперимент по тестированию основных классов уязвимостей веб-приложений на основе предложенного алгоритмического и программного обеспечения. На рис. 6, 7 и в табл. 4 представлены результаты сравнительного вычислительного эксперимента на основе случайного тестирования методом черного ящика (методика 1) и тестирование на основе разработанных алгоритмических и программных решений (методика 2). Тестирование по второй методике позволяет существенно снизить временные и ресурсные затраты на проведение комплексного тестирования без снижения качества детектирования программных ошибок.

Разработанные инструментальные средства в виде моделей, методов, алгоритмов и программного обеспечения могут найти применение в решении ряда конкретных задач, связанных с разработкой, тестированием и сопровождением веб-приложений, адаптированных под специфику выявления ошибок различных программных платформ построения веб-приложений.

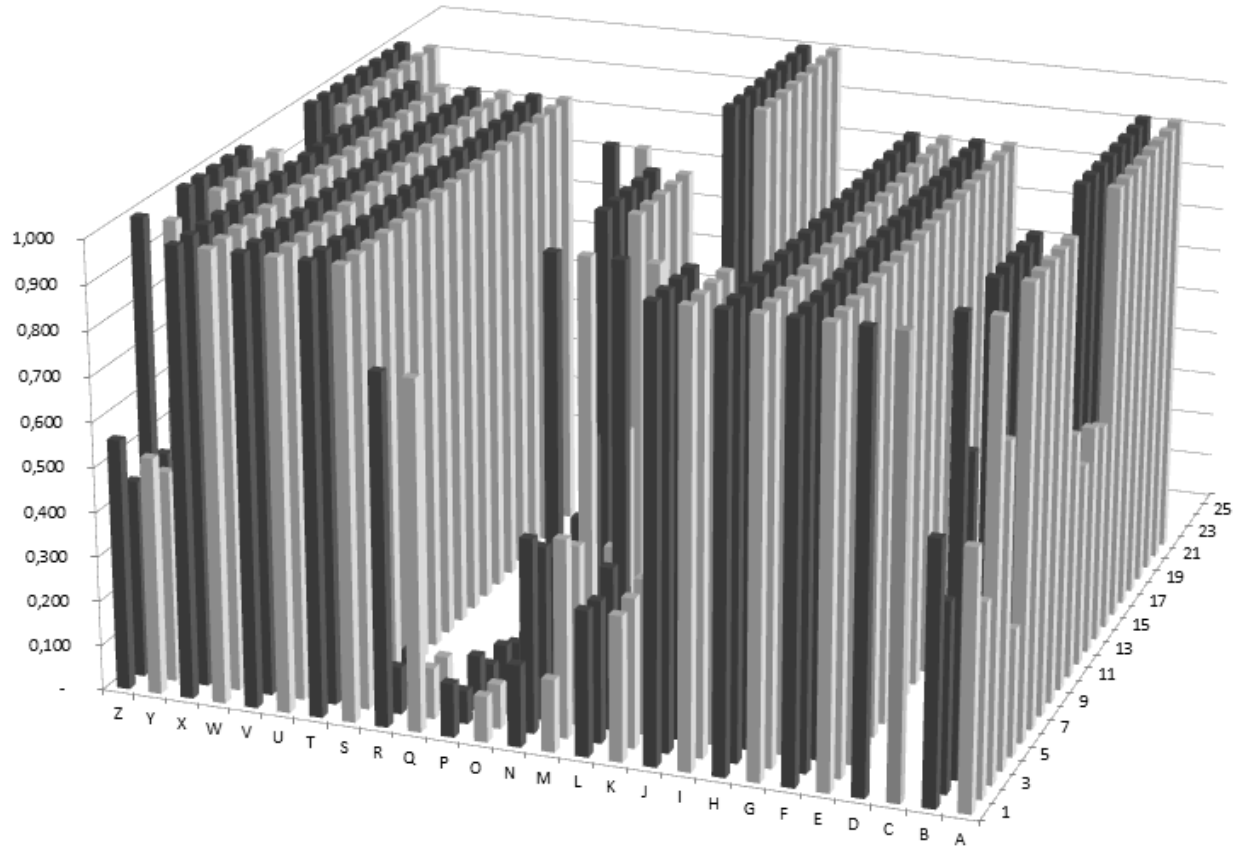

Рис. 6. Сравнительный анализ результатов по двум методикам тестирования. Обозначения, соответственно, по методике 1 и 2: A, B – BooleanBasedBlind_t; C, D – CmdExecution_t; E, F – Code_t; G, H – Command_t; I, J – DbmsFingerprint_t; K, L – Encoder_t; M, N – ErrorBlind_t; O, P – HttpParameterPolution_t; Q, R – InjectType_t; S, T – OutOfBand_t; U, V – StackedTime_t; W, X – TimeBlind_t; Y, Z – UnionInjection_t

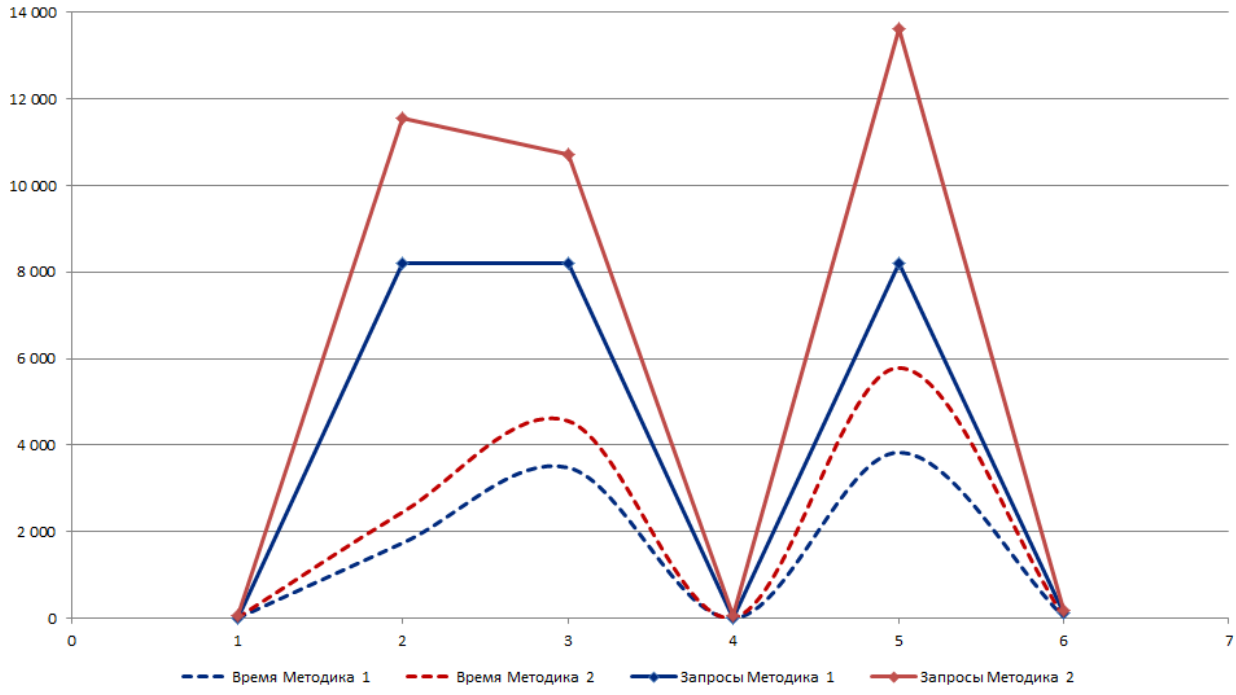

Рис. 7. Сравнительный анализ методик по числу запросов и времени выполнения при тестировании SQL инъекций

Таблица 4

Сравнительные данные по методикам тестирования

№ п/п	Название метода	Запросы Методика 1	Запросы Методика 2	Время Методика 1	Время Методика 2
1	UnionInjection	4	64	0,850	13,600
2	BooleanBasedBlind	8192	11 553	1740,800	2454,970
3	TimeBlind	8192	10 728	3481,600	4559,230
4	ErrorBlind	1	61	0,213	12,963
5	StackedTime	8192	13 616	3829,760	5786,715
6	OutOfBand	128	188	27,200	39,950

Литература

1. Азарнова, Т.В. Расширение функциональных возможностей фаззинга веб-приложений на основе динамических сетей Байеса / Т.В. Азарнова, П.В. Полухин // Научно-техническая информация. Серия 2. Информ. процессы и системы. – 2014. – № 9. – С. 12–19.

2. Кельберт, М.Я. Вероятность и статистика в примерах и задачах. Т. 1: Основные понятия теории вероятностей и математической статистики / М.Я. Кельберт, Ю.М. Сухов. – М.: МЦНМО, 2007. – 456 с.

3. Масленников, Е.Д. Предсказания на основе байесовских сетей доверия: алгоритм и программная реализация / Е.Д. Масленников, В.Б. Сулимов // Вычислительные методы и программирование. – 2010. – № 11. – С. 94–107.

4. Микаэльян, С.В. Методы фильтрации на основе многоточечной аппроксимации плотности вероятности оценки в задаче определения параметров движения цели при помощи измерителя с нелинейной характеристикой / С.В. Микаэльян // Наука и Образование. – 2011. – № 10. – С. 2–14.

5. Полухин, П.В. Интеграция динамических байесовских сетей в процесс тестирования веб-приложений для выявления уязвимостей межсайтингового скриптинг / П.В. Полухин // Научное обозрение. – 2014. – № 9. – С. 414–422.

6. Соболь, И.М. Численные методы Монте-Карло. – М.: Наука, 1973. – 312 с.

7. Тихонов, В.И. Марковские процессы / В.И. Тихонов, М.А. Миронов. – М.: Сов. радио, 1977. – 488 с.

8. Торопова, А.В. Подходы к диагностике согласованности данных в байесовских сетях доверия / А.В. Торопова // Труды СПИИРАН. – 2015. – № 43. – С. 156–178.

9. Тулупьев, А.Л. Байесовские сети доверия: логико-вероятностный вывод в ациклических направленных графах / А.Л. Тулупьев, А.В. Сироткин, С.И. Николенко. – СПб.: Изд-во С.-Петерб. ун-та, 2009. – 400 с.

10. Russel, S. Artificial Intelligence: A Modern Approach / S. Russel, P. Norvig. – Boston: Prentice Hall, 2009 – 1095 p.

11. Sutton, M. Fuzzing: Brute Force Vulnerability Discovery / M. Sutton, A. Greene, P. Amini. – Addison Wesley, 2007. – 527 p.

Баркалов Сергей Алексеевич, д-р техн. наук, профессор, зав. кафедрой управления строительством, декан факультета экономики, менеджмента и информационных технологий, Воронежский государственный технический университет, г. Воронеж; sbarkalov@nm.ru.

Азарнова Татьяна Васильевна, д-р техн. наук, профессор, зав. кафедрой математических методов исследования операций, Воронежский государственный университет, г. Воронеж; ivdas_92@mail.ru.

Полухин Павел Валерьевич, канд. техн. наук, кафедра математических методов исследования операций, Воронежский государственный университет, г. Воронеж; alfa_force@mail.ru.

Поступила в редакцию 15 марта 2017 г.

DOI: 10.14529/ctcr170205

MANAGEMENT OF THE PROCESS OF WEB APPLICATIONS TESTING BY THE FUZZING METHOD BASED ON DYNAMIC BAYESOV NETWORKS

S.A. Barkalov[1], *sbarkalov@nm.ru,*
T.V. Azarnova[2], *ivdas_92@mail.ru,*
P.V. Polukhin[2], *alfa_force@mail.ru*
[1] *Voronezh State Technical University, Voronezh, Russian Federation,*
[2] *Voronezh State University, Voronezh, Russian Federation*

Nowadays, intensive research is being conducted in the field of developing effective technologies for testing web applications for vulnerabilities, one of such technologies that allowing to hold complex testing at all stages of the application life cycle is fuzzing testing. The actual direction of development this technology is the development of mathematical and software that realizes intellectual components of fuzzing, the implementation of which will significantly improve its effectiveness and resource efficiency. In article the conceptual model of the application dynamic Bayesian networks to control web application testing by fuzzing is provided. Within the framework of the constructed conceptual model, dynamic Bayesian models for the main OWASP – vulnerability classes of Web applications and corresponding algorithmic and software for testing were developed.

Keywords: OWASP – classes of vulnerabilities of web applications, control of testing web applications, dynamic Bayesian network, algorithms of training and a probable output.

References

1. Azarnova T.V., Polukhin P.V. [Expanding the Functionality of Fuzzing of Web Applications Based on Dynamic Bayesian Networks]. *Scientific and Technical Information. Series 2. Inform. Processes and Systems*, 2014, no. 9, pp. 12–19. (in Russ.)

2. Kelbert M.Ya., Sukhov Yu.M. *Veroyatnost' i statistika v primerakh i zadachah. Tom 1: Osnovnye ponyatiya teorii veroyatnostey i matematicheskoy statistiki* [Probability and Statistics in Examples and Problems. Vol. 1: Basic Concepts of Probability Theory and Mathematical Statistics]. Moscow, MTSNMO Publ., 2007. 456 p.

3. Maslennikov E.D., Sulimov V.B. [Predictions Based on Bayesian Networks of Trust: Algorithm and Software Implementation]. *Computational Methods and Programming*, 2010, no. 11, pp. 94–107. (in Russ.)

4. Mikaelyan S.V. [Filtering Methods Based on Multipoint Approximation of the Probability Density of Estimation in The Problem of Determining the Parameters of Target Motion Using a Meter with a Nonlinear Characteristic]. *Science and Education*, 2011, no. 10, pp. 2–14. (in Russ.)

5. Polukhin P.V. [Integration of Dynamic Bayesian Networks into the Process of Testing Web Applications to Identify Vulnerabilities of Cross-Site Scripting]. *Scientific Review*, 2014, no. 9, pp. 414–422. (in Russ.)

6. Sobol I.M. *Chislennye metody Monte-Karlo* [Numerical Monte Carlo Methods]. Moscow, Nauka Publ., 1973. 312 p.

7. Tikhonov V.I., Mironov M.A. *Markovskie processy* [Markov Processes]. Moscow, Sov. Radio Publ., 1977. 488 p.

8. Toropova A.V. [Approaches to the Diagnosis of Data Consistency in Bayesian Networks of Trust]. *Proceedings of SPIIRAS*, 2015, no. 43, pp. 156–178. (in Russ.)

9. Tulup'ev A.L., Sirotkin A.V., Nikolenko S.I. *Bayesovskie seti doveriya: logiko-veroyatnostnyy vyvod v atsiklicheskikh napravlennykh grafakh* [Bayesian Confidence Networks: Logical-Probabilistic Conclusion in Acyclic Directed Graphs]. St. Petersburg, St. Petersburg Univ. Publ., 2009. 400 p.

10. Russel S., Norvig P. *Artificial Intelligence: A Modern Approach*. Boston, Prentice Hall, 2009. 1095 p.

11. Sutton M., Greene A., Amini P. *Fuzzing: Brute Force Vulnerability Discovery*. Addison Wesley, 2007. 527 p.

Received 15 March 2017

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Баркалов, С.А. Управление процессом тестирования веб-приложений методом фаззинга на основе динамических байесовских сетей / С.А. Баркалов, Т.В. Азарнова, П.В. Полухин // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2017. – Т. 17, № 2. – С. 51–64. DOI: 10.14529/ctcr170205

FOR CITATION

Barkalov S.A., Azarnova T.V., Polukhin P.V. Management of the Process of Web Applications Testing by the Fuzzing Method Based on Dynamic Bayesov Networks. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*, 2017, vol. 17, no. 2, pp. 51–64. (in Russ.) DOI: 10.14529/ctcr170205