УДК 004.8

ИССЛЕДОВАНИЕ ПРОЦЕССА ФАЗЗИНГА SQL-ИНЪЕКЦИЙ ВЕБ-ПРИЛОЖЕНИЙ НА ОСНОВЕ ДИНАМИЧЕСКОЙ СЕТИ БАЙЕСА

Т. В. Азарнова, П. В. Полухин

Воронежский государственный университет

Поступила в редакцию 07.02.2014 г.

Аннотация. В статье представлены результаты научно-практического исследования, направленного на решение задач разработки и анализа вероятностной модели фаззинга SQL-инъекций веб-приложений. Модель представлена в виде динамической сети Байеса. Динамические сети Байеса – это хорошо апробированный инструмент моделирования динамических процессов, протекающих в условиях риска и неопределенности, позволяющий существенно повысить управляемость и эффективность моделируемых процессов. В работе рассматриваются вопросы синтеза и семантики анализируемой динамической сети Байеса, описываются приближенные алгоритмы вероятностного вывода, фильтрации и прогнозирования, адаптированные к процессам фаззинга..

Ключевые слова: фаззинг SQL-инъекций для веб-приложений, динамическая байесовская сеть, приближенные алгоритмы вероятностного вывода, фильтрации и прогнозирования для динамических байесовских сетей.

Annotation. Results of the scientific and practical research directed on the solution of problems of development and the analysis of probabilistic model of a fuzzing of SQL injections of web applications are presented in article. The model is presented in the form of dynamic Bayesian network. Dynamic Bayesian networks is the well approved instrument of modeling dynamic processes proceeding in the conditions of risk and uncertainty, allowing significantly to increase controllability and efficiency of modeled processes. In work are described questions of synthesis and semantics of an analyzed dynamic Bayesian network, approximate algorithms of a probabilistic inference, filtration and prediction, adapted for fuzzing processes.

Keywords: fuzzing sql injections, dynamic bayesian network, approximate inference algorithms, dynamic bayesian network filtration and prediction.

ВВЕДЕНИЕ

Современный этап развития бизнеса и информационно-коммуникационных технологий характеризуется стремлением компаний повысить свою конкурентоспособность за счет разработки веб-приложений и веб-сервисов. Веб-приложения обеспечивают разработчикам и конечным пользователям существенные преимущества, связанные с возможностью адаптации, непрерывного обновления и доработки приложения со стороны разработчиков (в том числе устранения проблем, связанных с безопасностью), и с возможностью конечных пользователей автоматически получать обновления (патчи), поскольку приложение находится на централизованном сервере.

Современные инструменты разработки веб-приложений стремительно развиваются, расширяют свою функциональность, ста-

новятся более удобными и эффективными, интеграция данных инструментов с базами данных упрощает механизм централизованного хранения и оперативного доступа к информации [7]. Несмотря на широкие возможности инструментальных средств разработки веб-приложений, развитие данных инструментов в области решения вопросов безопасности является актуальным направлением исследований. Создать безопасное и защищенное веб-приложение довольно сложно, необходимо тщательное тестирование, приложение может содержать уязвимости. Одним из эффективных современных способов поиска возможных уязвимостей является фаззинг. Фаззинг веб-приложений способствует выявлению уязвимостей в самих приложениях и компонентах инфраструктуры (веб-сервер, сервер баз данных и др.), в которые встроено приложение. Сущность фаззинга заключается в создании специальных наборов входных данных, при получении которыхможет произойти сбой приложения или обнаружены недокументированные возможности, анализ которых приведет к раскрытию конфиденциальных данных и нарушению целостности всего приложения. Моделирование процесса фаззинга в виде байесовской сети и проведение на базе данной модели процедур вероятностного вывода, фильтрации и прогнозирования позволяет осуществить интеллектуальный ситуационный анализ объекта исследования и организовать целенаправленнуюкорректировку алгоритма фаззинга в соответствии с результатами анализа. Адаптивная корректировка фаззинга направлена на повышение его результативности и эффективности. В работе рассматриваются вопросы синтеза и семантики динамической байесовской сети фаззинга SQL-инъекций веб-приложений, описываются приближенные алгоритмы вероятностного вывода, фильтрации и прогнозирования, адаптированные к процессам фаззинга.

ХАРАКТЕРИСТИКА ПРЕДМЕТНОЙ ОБЛАСТИ: КЛАССИФИКАЦИЯ SQL-ИНЪЕКЦИЙ И МЕТОДОВ ФАЗЗИНГА ВЕБ-ПРИЛОЖЕНИЙ

Согласно международному проекту по безопасности веб-приложений (OWASP), одной из наиболее критичных уязвимостей в данной области является SQL-инъекция (SQL-Injection). SQLинъекция – это специальная процедура (атака), заключающаясяво вставке вредоносного кода в строки, которые передаются СУБДдля синтаксического анализа и выполнения. Опасность SQL-инъекциисвязана с тем, что их внедрение может открыть доступ к конфиденциальной служебной информации компании, хранящейся в системе управления базами данных. СУБД выполняют все получаемые синтаксически правильные запросы, поэтому любая процедура, создающая инструкции SQL, должна контролироваться на предмет уязвимости к вставке небезопасного кода. Даже параметризованные данные могут стать предметом манипуляций [1, 5].

Изучение современных методов SQL-инъекций, показывает, что основная форма атаки SQL-инъекций состоит в прямой вставке кода в пользовательские входные переменные, которые объединяются с SQL командами и выполняются. Менее явная атака внедряет небезопасный код в строки, предназначенные для хранения в таблице или в виде метаданных, впоследствии сохраненные строки объединяются с динамической командой SQL, происходит выполнение небезопасного кода [8, 9].

SQL-инъекции можно классифицировать в зависимости от векторов атаки. К первой группе относятся атаки типа – inband (unionquery) – это типичный вариант инъекции SQL-кода, основывается на том, что злоумышленник и приложение взаимодействуют по одному каналу (протоколу), например HTTP. Атакующий использует данный канал (протокол) для внедрения SQL запросов, с целью извлечения данных и метаданных из сервера баз данных. При этом приложение отображает полученные данные непосредственно на веб-странице, которые и анализирует злоумышленник [2].

Во вторую группу входят атаки типа – boolean-basedblind, которые находят свое применение в тех ситуациям, когда приложение скрывает сообщения об ошибках и уведомления. Механизм реализации основан на переопределении логики запроса, тем самым одна часть запроса должна возвращать значение true или false.

Третья группа представлена атаками типа errorbased. Атаки данного типа находят свое применение в тех случаях, когда приложение некорректно обрабатывает ошибки и исключения, возникающие при работе с базой данных, и сообщение об этом отображается пользователю. Существует достаточное число встроенных в СУБД функций, которые при ошибке отображают часть данных или метаданных [4].

В четвертую группу включаются атаки stackedqueries. Применяются, когда приложение допускает выполнение последовательных SQL запросов. Атака реализуется путем разделения уязвимого параметра и внедряемого SQL запроса символом «;» однако данная атака может быть применена не для всех СУБД и веб-технологий. Например, MSSQLразрешает выполнение нескольких запросов только при обращении из ASP.NET или PHP, но не разрешает из JSP.

Пятая группа представлена атаками timebasedblind. Используется подзапрос с временной задержкой, что приводит к паузам в работе СУДБ, позволяя при этом извлекать данные из сервера, сравнивая время ответа на оригинальный запрос и запрос с внедренным кодом. Используя данную технику можнопроверять логические условия, например наличие роли DBA у текущего пользователя, а также посимвольно извлекать данные из таблиц.

К шестой группе относятся атаки outofband. Используется два канала взаимодействия. Функционал современных СУБД позволяет отправлять данные на почту по протоколу SMTP или POP, взаимодействовать с файловой системой по FTP, SMB, использовать DNS. Устанавливается соединение с СУДБ, и отправляются SQL команды по одному каналу, в ответ СУБД посылает результаты ответа по другому каналу, например, по электронной почте или путем выполнения команды MicrosoftSQLServerxp_cmdshell.

Профессиональные разработчики веб-приложений используют специальные инструменты защиты создаваемых продуктов. Одним из таких достаточно эффективных инструментов является межсетевой экран безопасности веб-приложений (WAF). WAF позволяет защитить приложения от ряда атак, за счет мониторинга HTTP трафика и анализа событий в реальном режиме времени. Но и для инструментов WAF существует ряд методов обхода защиты [5]:

– HTTPParameterPollution (HPP) – подстановка нескольких HTTP параметров с одинаковыми именами. Это может вызвать у приложения недокументированное поведение, используя данный метод, не исключена возможность, что удастся обойти механизмы фильтрации входных данных и даже изменить значения внутренних переменных приложения.

– HTTPParameterFragmentation (HPF) – разделение значения HTTP параметра с помощью закодированного разделителя. Используя данный прием, можно фрагментировать код SQL инъекции и распределить его между двумя HTTP параметрами. Прием особенно эффективен для тестирования приложений, использующих динамическую генерацию SQLкоманд, на основе данных пользователей.

– DNSExfiltration позволяет расширить механизм Outofbandза счёт использования DNS запросов, направленных как на получения доступа к ресурсам СУБД, так и к файловой системе сервера.

Методы фаззингаSQL-инъекций веб-приложений логически можно разделить на двегруппы: черного и серого ящика. Метод черного ящика является простым и быстрым в реализации, это позволяет увеличить скорость тестирования. В тоже время метод обладает рядом ограничений и не позволяет накапливать структурированные данные о результатах тестирования, на каждой итерации генерируются новые случайные данные, несвязанные с предыдущей итерацией, это

делает трудным и практически невозможным обнаружение сложных уязвимостей. Метод серого ящика позволяет накапливать статистику, анализировать ее, генерировать новые данные по результатам проведенного анализа и оценивать эффективность реализации метода на каждой итерации. Однако реализация сложного и адаптируемого к результатам анализа метода неизбежно ведет к увеличению времени реализации и объема системных ресурсов на этапе проведения тестирования. Актуальными направлением современных исследований в области развития подхода серого ящика является создание ресурсосберегающих технологий тестирования, представленных виде моделей, методов и алгоритмов. Предложенная в рамках работы динамическая байесовская сеть процесса фаззинга SQL-инъекций веб-приложений реализована в рамках описанного выше направления исследований.

ОПИСАНИЕ ДИНАМИЧЕСКОЙ БАЙЕСОВСКОЙ СЕТИ ФАЗЗИНГА SQL-ИНЪЕКЦИЙ ВЕБ-ПРИЛОЖЕНИЙ

Байесовская сеть представляет собой ориентированный граф без циклов, вершинами которого являются дискретные и (или) непрерывные случайные величины. При интерпретации дуг считается, что стрелка, ведущая от вершины x к вершине y, означает, что вершина x является родительской вершиной для вершины y $\left(x \in \text{Parents}(y)\right)$ и оказывает непосредственное влияние на y. При изображении байесовской сети родительские вершины изображаются на более высоком уровне, чем вершины потомки. Каждая вершина x характеризуется распределением условных вероятностей $P\left(x_i \middle| \text{Parents}(x_i)\right)$. Байесовская сеть служит правильным представлением проблемной области, если любая вершина на ней после задания родительских вершин становится условно независимой от других вершин, лежащих на более высоких уровнях [3]. Если зависимость между элементами z и y не является непосредственной и осуществляется посредством родительского для y элемента x, то x будет находиться между y и z, отсекая зависимость между ними и моделируя ситуация условной независимости. Распределение условных вероятностей позволяет задать полное совместное распределение для всех переменных:

$$P\left(x_1, x_2, ..., x_n\right) = \\ = P\left(X_1 = x_1, X_2 = x_2, ..., X_n = x_n\right) = . \\ = \prod_{i=1}^{n} P\left(x_i \middle| \text{Parents}(x_i)\right) \qquad (1)$$

Основной задачей, которую позволяют решать байесовские сети является задача вероятностного вывода. Сущность данной задачи заключается в вычислении распределения апостериорных вероятностей для множества переменных запроса $X = \{X_1, X_2, ..., X_n\}$, если дано некоторое событие e, при котором присвоены определенные значения множеству переменных свидетельства $E = \{E_1, E_2, ..., E_m\}$, $E_1 = e_1, E_2 = e_2, ..., E_n = e_n$ (множество всех вершин байесовской сети можно представить в виде объединения трех множеств $X \cup E \cup Y$, где $Y = \{Y_1, Y_2, ..., Y_p\}$ – множество скрытых переменных). Искомое апостериорное распределение вероятностей $P\left(X \middle| E\right)$ можно вычислить $P\left(X \middle| E\right) = \alpha \sum_{Y} P\left(X, E, Y\right)$, где α – нормирующий множитель. Для вычисления $P\left(X \middle| E\right)$ можно использовать точные алгоритмы перебора, устранения переменной и кластеризации. Для многосвязных байесовских сетей точные алгоритмы вероятностного вывода реализовать очень сложно, поэтому на практике применяют приближенные алгоритмы. Остановимся кратко на приближенных алгоритмах Монте-Карло – рандомизированных алгоритмах выборки. Данные алгоритмы дают приближенные ответы, точность которых зависит от количества сформированных выборок. Одним из представителей таких алгоритмов является метод непосредственной выборки с исключением. В данном алгоритме в топологическом порядке формируются выборки по каждой переменной в байесовской сети, причем распределение вероятностей, из которого берется выборка кон-

кретной переменной, должно определяться значениями уже полученных родительских переменных. После формирования множества выборок, из этого множества исключаются выборки не соответствующие свидетельству $E=\{E_1,E_2,...,E_m\}$. Обозначим: N_E – количество выборок, соответствующих свидетельству E, $N_E(X)$.– количество выборок, соответствующих свидетельству и имеющих заданные значения переменных запроса. При таких обозначениях $P(X|E)$ можно оценить как частоту $P(X|E)=\frac{N_E(X)}{N_E}$, качество оценки определяется объемом выборки. Доля выборок, согласованных со свидетельством, экспоненциально уменьшается по мере увеличения переменных свидетельства – это является основным недостатком данного метода. В качестве второго алгоритма рассмотрим алгоритм оценки веса с учетом правдоподобия. В данном алгоритме значения переменных свидетельства $E=\{E_1,E_2,...,E_m\}$ фиксируются и формируются выборки только для переменных $X=\{X_1,...,X_n\}$ и $X=\{Y_1,...,Y_p\}$. Обозначим $N(X,Y,E)$ – количество полученных выборок. Не все события являются равноправными, поэтому при вычислении результата каждое событие взвешивается с учетом правдоподобия того, насколько оно согласовано со свидетельством. Правдоподобие вычисляется путем произведения условных вероятностей для каждой переменной свидетельства, если заданы ее родительские переменные:

$$w(X,Y,E)=\prod_{i=1}^{m}P(E_i|\text{Parents}(E_i)). \quad (2)$$

Результирующая вероятность вычисляется следующим образом:

$$P(X|E)=\alpha\sum_{Y}N(X,Y,E)w(X,Y,E). \quad (3)$$

Преимуществом данного алгоритма является то, что в отличие от предыдущего алгоритма, он использует все сгенерированные выборки, но при большой размерности задачи почти все сформированные выборки имеют низкие веса, основной вес тяжести приходится на небольшое количество выборок согласованных со свидетельством. В качестве третьего алгоритма рассмотрим алгоритм Монте-Карло с применением цепи Маркова. В данном алгоритме считается, что байесовская сеть находится в конкретном текущем состоянии, каждое новое состояние порождается путем случайного формирования выборки значения для одной из переменных X_t. Выборка формируется в соответствии с Марковским покрытием переменной X_i, которое состоит из ее родительских переменных, дочерних переменных и родительских переменных для дочерних переменных. Переменные свидетельства остаются фиксированными на протяжении всей процедуры случайного изменения состояний.

Динамические байесовские сети представляют последовательность байесовских сетей, взятых в хронологическом порядке [10]. Каждый элемент последовательности (временной срез) описывает состояние байесовской сети на определенный момент времени. Последовательность состояний начинается в момент времени $t=0$. Будем обозначать X_t, E_t – соответственно множество ненаблюдаемых и наблюдаемых (свидетельств) переменных для момента времени t. В дальнейшем будем считать, что свидетельства начинают появляться в момент времени $t=1$.

Символами $X_{t_1:t_2}, E_{t_1:t_2}$ будем обозначать последовательности переменных на временных срезах с момента времени t_1 до момента времени t_2. При задании динамических байесовских сетей фактически необходимо задать таблицы условных вероятностей для каждой переменной каждого временного среза. Считается, что внутри временного среза таблицы условных вероятностей задаются один раз для некоторого репрезентативного среза, например, для нулевого момента времени $P(X_0)$, и не изменяются при переходе от среза к срезу. В динамических байесовских сетях возникает проблема, состоящая в том, что каждая переменная состояния или свидетельства может быть связана с неограниченным количеством переменных на предшествующих временных срезах. Данная проблема решается принятием предположения о марко-

вости, в соответствии с которым текущее состояние зависит от конечного числа предыдущих состояний. В дальнейшем будем рассматривать только так называемые Марковские процессы первого порядка, в рамках которых текущее состояние переменной состояния зависит только от непосредственно предшествующего состояния

$$P\left(X_t \middle| X_{0:t-1}\right) = P\left(X_t \middle| X_{t-1}\right), \qquad (4)$$

а состояние переменной свидетельства зависит только от текущего состояния

$$P\left(E_t \middle| X_{0:t}, E_{0:t-1}\right) = P\left(E_t \middle| X_t\right). \qquad (5)$$

Вероятностное распределение $P\left(X_t \middle| X_{t-1}\right)$ называется моделью перехода, а вероятностное распределение $P\left(E_t \middle| X_t\right)$ – моделью восприятия [6].

С учетом введенных предположений и вероятностных распределений полное совместное распределение вероятностей задается следующим образом

$$P\left(X_0, X_1, ..., X_t, E_1, ..., E_t\right) = \\ = P\left(X_0\right)\prod_{i=1}^{t} P\left(X_i \middle| X_{i-1}\right)P\left(E_i \middle| X_i\right). \qquad (6)$$

На рис. 1 приведен фрагмент (два временных среза) динамической байесовской сети процесса фаззинга SQL-инъекций веб-приложений. Табл. 1 содержит список обозначений для данной байесовской сети.

Каждый временной срез приведенной на рис. 1 динамической байесовской сети интерпретируется как эксперимент, проведенный в определенный момент времени. Момент времени является в данном случае понятием интервальным, эксперимент длится некоторый период времени, который воспринимается как единый момент эксперимента. В работе будут рассматриваться две интерпретации понятия эксперимент. В рамках первой интерпретации эксперимент – это этап работы по тестированию одного веб-приложения. На каждом этапе тестирования применяются определенные инструменты тестирования, устанавливаются характеристики веб-приложения (наблюдаемые переменные, свидетельства) и формируется информация о наличии уязвимостей (переменные состояния). Изменения во времени в данном случае связаны с накоплением информации о тестируемом веб-приложении.

Таблица 1

Характеристика узлов динамической байесовской сети

Наименование узла	Характеристика
fin_prt_dbms	Определение типа и версии СУБД, установленной на сервере.
bl,tbl,bbl,ebl,ob	РазличныеподходыэксплуатацииSQL-инъекции:Blind, Time Based blind, Boolean Based Blind, Error Based Blind, Out Of Band.
http_par_frag dens_ex, http_par_pl	Механизмы обхода межсетевых экранов безопасности веб-приложений (WAF).
encod	Механизмы кодирования и шифрования, предназначенные для обфускации и обхода механизмов фильтрации.
cmd_exec	Исполнение нативных команд операционной системы.
priv_esc	Эскалация привилегий и расширение полномочий.
data	Получение доступа к данным, хранящимся в СУБД.
net	Получение доступа к компонентам сети из командного интерфейса СУБД.
db_struct	Получение структуры таблиц и баз данных СУБД.
fupl	Возможность удаленной загрузки файлов, через внутренние механизмы СУБД.
dos	Нарушение работы СУБД (отказ в обслуживании).
author, authen, conf,avail, integ,	Нарушение механизмов аутентификации, авторизации, целостности, конфиденциальностии доступности.

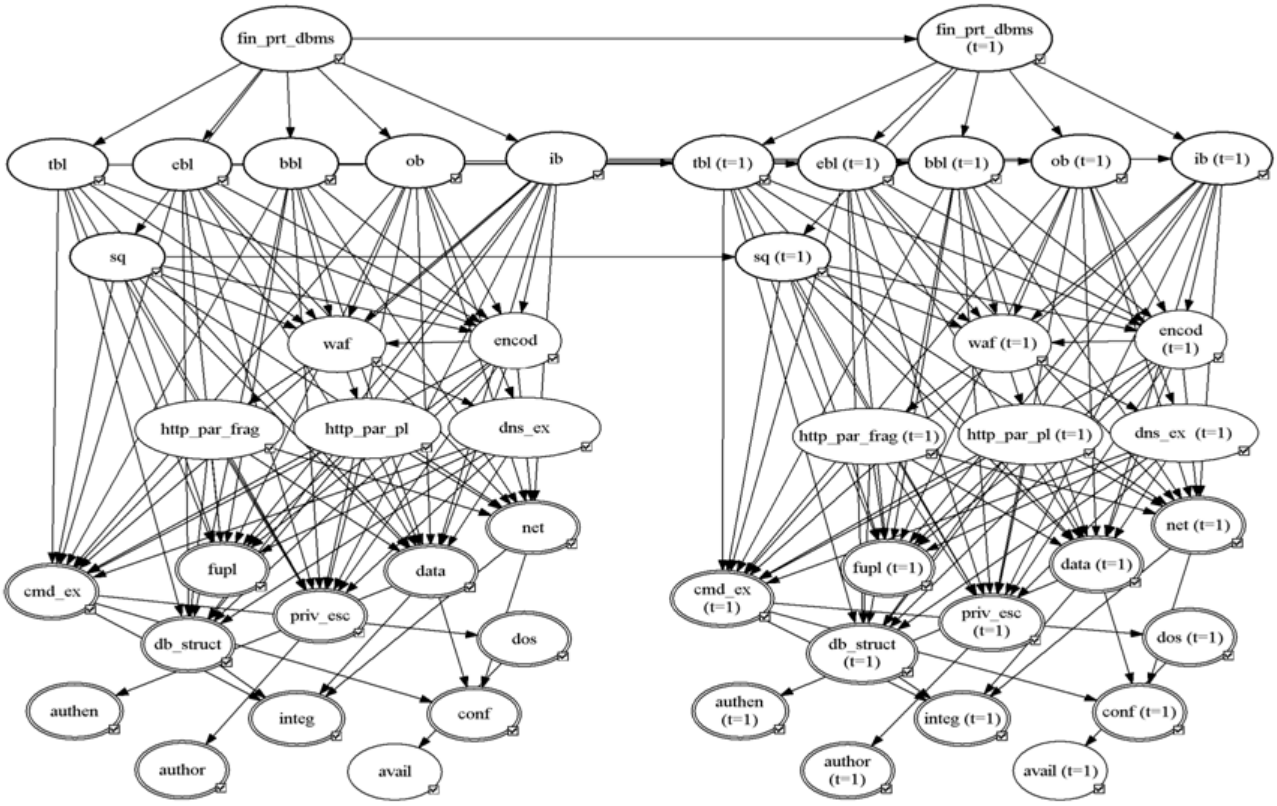

Рис. 1. Фрагмент динамической байесовской сети процесса фаззинга SQL-инъекций веб-приложений

Вероятностное распределение для нулевого момента времени $P(X_0)$, модель перехода и модель восприятия строятся на основе статистики по этапам тестирования веб-приложений. В рамках второй интерпретации эксперимент – это процедура полного тестирования одного приложения из группы однородных веб-приложений. Для разбиения веб-приложений на однородные классы используется специальная многокритериальная процедура классификации, ориентированная на специфику веб-приложений. Изменения во времени при втором подходе связаны с тем, что со временем «восприятие информации» оказывает влияние и на разработчиков веб-приложений, и на методики генерации инъекций. Вероятностное распределение для нулевого момента времени $P(X_0)$, модель перехода и модель восприятия строятся на основе статистики комплексного тестирования последовательности однородных веб-приложений.

Динамические байесовские сети позволяют решать целый ряд задач вероятностного вывода. Основные из этих задач кратко изложены в табл. 2.

Дадим характеристику данных задач применительно к байесовской сети фаззинга SQL-инъекций веб-приложений. На рис. 1 наблюдаемые переменные (свидетельства) выделены жирной линией, а переменные состояния выделены двойной сплошной линией. Задачу фильтрации сформулируем и для первого, и для второго подходов интерпретации байесовских сетей процедуры фаззинга. В рамках первой интерпретации решение задачи фильтрации позволяет оценить вероятностное распределение присутствия различных уязвимостей в веб-приложении после свидетельств на нескольких предыдущих этапах тестирования. В рамках второй интерпретации решение задачи фильтрации определяет вероятностное распределение присутствия различных уязвимостей при следующем комплексном тестировании веб-приложения, если есть свидетельства данного тестирования и свидетельства тестирования аналогичных веб-приложений в предыдущие моменты времени. Задачи прогнозирования и сглаживания являются более актуальными для второй интерпретации. Задача прогнозирования

Таблица 2

Основные задачи вероятностного вывода для динамических 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 путем рекурсивного вычисления вероятностного распределения в момент времени $t+k+1$ 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: $P(X_k \mid E_{1:t}) = \alpha P(X_k \mid E_{1:k}) P(E_{k+1:t} \mid X_k)$ $$P(E_{k+1:t} \mid X_k) =$$ $$= \sum_{X_{k+1}} P(E_{k+1:t} \mid X_k, X_{k+1}) P(X_{k+1} \mid X_k) =$$ $$= \sum_{X_{k+1}} P(E_{k+1} \mid X_{k+1}) P(E_{k+2:t} \mid X_{k+1}) P(X_{k+1} \mid X_k)$$

позволяет сделать выводы о присутствии уязвимостей при тестировании определенного класса веб-приложений в будущем, если известны свидетельства прошлого тестирования веб-приложений данного класса. Задача сглаживания позволяет сделать выводы о вероятностном распределении переменных, характеризующих присутствие определенных уязвимостей при тестировании веб-приложений в прошлые моменты времени, если получена последовательность свидетельств вплоть до текущего момента времени.

При решении задач фильтрации, прогнозирования и сглаживания в работе используются приближенные методы вероятностного вывода, основанные на подходе взвешивания с учетом правдоподобия. Остановимся более подробно на основных аспектах применения данного алгоритма применительно к задачам фильтрации. При адаптации описанного выше метода взвешивания с учетом правдоподобия к специфике динамических байесовских сетей осуществляется развертывание сети. Под развертыванием сети (под решение определенной задачи) понимается представление динамической байесовской сети в виде статической байесовской сети за счет повторения временных срезов до тех пор, пока в полученной сети будут учтены все наблюдения необходимые для решения поставленной задачи. Ме-

тод взвешивания с учетом правдоподобия выполняется путем генерирования выборок значений переменных состояния развернутой байесовской сети, которые взвешиваются с учетом правдоподобия их соответствия наблюдаемым переменным свидетельства. Каждая выборка формируется отдельно, переходя последовательно в топологическом порядке от одной переменной состояния к следующей вдоль всей сети. При применении алгоритма взвешивания с учетом правдоподобия к развернутой сети возникают проблемы, связанные с возрастанием временных и пространственных требований при каждом обновлении. В направлении решения ресурсных проблем можно использовать некоторые модификации алгоритма взвешивания с учетом правдоподобия, в данной работе используется алгоритм фильтрации частиц. В алгоритме фильтрации частиц, представленном на рис. 2, на нулевом срезе из вероятностного распределения $P(X_0)$ одновременно генерируется N выборок. Обозначим $N(X_t|E_{1:t})$ – количество выборок для состояния X_t после получения свидетельств $E_{1:t}$. При переходе от временного среза t к временному срезу $t+1$ обновление множества выборок строится через модель перехода $P(X_{t+1}|X_t)$. Количество выборок для состояния X_{t+1}, получаемых с помощью модели перехода, определяется следующим образом $N(X_{t+1}|E_{1:t}) = \sum_{X_t} P(X_{t+1}|X_t) N(X_t|E_{1:t})$.

Каждая выборка взвешивается с учетом правдоподобия по отношению к новым свидетельствам, ей присваивается вес $P(E_{t+1}|X_{t+1})$, суммарный вес выборок равен $w(X_{t+1}|E_{1:t+1}) = P(E_{t+1}|X_{t+1}) P(X_{t+1}|E_{1:t})$. Выборки, которые имеют малый вес отбрасываются. Из новой популяции выборок формируется N выборок, имеющих наибольшие веса.

ЗАКЛЮЧЕНИЕ

Результаты проведенного исследования показывают, что технология фаззинга является эффективным, уже достаточно хорошо апробированным инструментом выявления уязвимостей веб-приложений и взаимодействующих с ними компонентов (СУБД и веб-сервера). Предложенный автором алгоритм позволяет расширить возможности данной технологии за счет использования

Particle_Filtering (e, N, dbn)

inputs: e – полученное свидетельство

N – количество выборок, которые должны сопровождаться

dbn – динамическая сеть байеса с распределением априорных вероятностей $P(X_0)$, моделью перехода $P(X_1|X_0)$ и моделью восприятия $P(E_1|X_1)$

static: S – вектор выборок размерностью N, инициализированный из $P(X_0)$

local variable: W – вектор весов размерностью N

for $i = 1$ to N do

$S[i] \leftarrow$ выборка из $P(X_1|X_0 = S[i])$

$W[i] \leftarrow P(e|X_1 = S[i])$

$S \leftarrow Weighted - Sample - With - Replace(N, S, W)$

return S

Рис. 2. Алгоритм фильтрации частиц

вероятностной модели процесса фаззинга, представленной в виде динамической сети Байеса. Задачи логического вывода, реализуемые аппаратом динамических байесовских сетей, предоставляют возможность получать актуальную информацию об апостериорных вероятностных распределениях случайных величин, характеризующих наличие определенных уязвимостей веб-приложений и взаимодействующих с ними компонент. Это позволяет рационально адаптировать алгоритм анализа веб-приложения при использовании метода серого ящика в качестве инструмента тестирования. В рамках проведенного исследования автором была создана программная реализация предложенного алгоритма и проведена апробация в практических исследованиях по тестированию целого ряда однородных веб-приложений. По результатам практической апробации можно сделать обобщающий вывод о высокой ресурсной эффективности внедрения алгоритма в инструменты фаззинга.

СПИСОК ЛИТЕРАТУРЫ

1. *Godefroid P.* Automated whitebox fuzz testing / P. Godefroid, M. Levin. – NDSS. – 2008. – 350 p.

2. *Godefroid P.* Automating Software Testing Using Program Analysis / P. Godefroid, Peli de Halleux // IEEE SOFTWARE. – 2008 – P. 30 – 37.

3. *Korb A.* Bayesian Artificial Intelligence / K. Korb, A. Nicholson / Chapman & Hall /CRC Press UK, 2004. – 244 p.

4. *Oehlert P.* Violating Assumotions with Fuzzing / P. Oehlert // IEEE Security &Privace. – 2005. – № 2. – P. 58-67.

5. *Zalewski M.* The Tangled Web. A Guide to Securing Modern Web Applications / M. Zalewski. NostarchPress, 2012. – 477 p.

6. *Бабиков В.* Баесовские сети доверия и некоторые аспекты идентификации систем / В.Бабиков. – СПб. : УТЭОСС-2012. – http: // uteoss2012.ipu.ru/procdngs/0353.pdf

7. *Визгунов А.* Современные подходы к обеспечению безопасности в области дистанционного банковского обслуживания / А. Визгунов // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – № 2, 2011. – С. 49-58.

8. *Вялых А.* Динамика уязвимостей в современных защищенных информационных системах / А. Вялых, С. Вялых // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – № 2, 2011. – С. 59–63.

9. *Колесников Д.* Методика оценки защищенности специального программного обеспечения при проведении испытаний автоматизированных систем / Д. Колесников, А. Петров, В. Храмов // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – № 1, 2010. – С. 74–79.

10. *Тулупьев А.* Байесовские сети, логико-вероятностный подход / А. Тулупьев, С. Николенко, А. Сироткин. – СПб. : Наука, 2006. – 728 с.

Азарнова Т. В. – д.т.н., профессор кафедры «Математических методов исследования операций», Воронежский государственный университет. E-mail: ivdas_92@mail.ru

Asarnova T.V. – professor of chair «Mathematical methods and operational research», D.Sc. Technic, Voronezh State University.

Полухин П. В. – аспирант кафедры «Математических методов исследования операций», Воронежский государственный университет. E-mail: alfa_force@mail.ru

Polukhin P. V. – graduate student of chair «Mathematical methods and operational research», Voronezh State University.