ОБНАРУЖЕНИЕ УЯЗВИМОСТЕЙ К SQL-ИНЪЕКЦИЯМ НА ОСНОВЕ АБСТРАКТНОЙ ИНТЕРПРЕТАЦИИ С ЦЕЛЬЮ ОЦЕНКИ КАЧЕСТВА WEB-ПРИЛОЖЕНИЙ

Оношко Д.Е.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Леванцевич В.А. – старший преподаватель каф. ПОИТ

Предлагается расширение модели обнаружения уязвимостей web-приложений к SQL-инъекциям, основанной на статическом анализе исходных кодов. На основе предлагаемого расширения строится мера оценки внутреннего качества web-приложения в рамках модели качества продукта ISO/IEC 25010:2011.

По состоянию на 2017 год по данным Открытого проекта безопасности web-приложений (OWASP) инъекции являлись наиболее значимой угрозой безопасности для web-приложений [1]. В опубликованном организацией в 2021 году отчёте [2] инъекции занимают лишь третье место, однако следует учитывать, что в исследовании 2021 года в связи был измененён подход организации к классификации угроз, что безусловно повлияло на результаты: в зависимости от специфики конкретного проекта обнаруженная инъекция может быть классифицирована и как самостоятельный случай (A03 Injection), и, нередко, как следствие ошибок, допущенных на стадии проектирования web-приложения, которые представлены новой категорией — A04 Insecure Design.

Таким образом, при оценке распространённости инъекций следует исходить из того, что данный вид угроз не теряет своей актуальности. Среди отнесённых в данную категорию уязвимостей очевидным образом наиболее значимую роль играют SQL-инъекции, поскольку в отличие от других перечисленных OWASP векторов атаки системы управления базами данных (СУБД) используются подавляющим большинством web-приложений.

Как отмечалось ранее [3], технические решения, которые позиционируются как полностью исключающие или частично способствующие предотвращению возникновения в web-приложении подобных уязвимостей, в т.ч. подготовленные выражения (prepared statements), объектно-реляционное отображение (ORM) и хранимые процедуры (stored procedures), в действительности оказываются неэффективными при неправильном применении.

Помимо очевидных организационных мер превентивного характера, связанных с обучением программистов грамотному применению этих возможностей, обязательным является также контроль фактически разработанных исходных кодов на предмет подобных ошибок.

Предложенная в [3] модель обнаружения уязвимостей в сочетании с её расширением, описанным в [4], базируется на элементах абстрактной интерпретации исходного кода web-приложения. Структура web-приложения представляется в виде ориентированного графа $G = (P, C)$, где $P = \{p_1, p_2, \ldots, p_n\}$ — множество вершин, соответствующих обобщённым процедур, которые образуют код web-приложения, а $C \subset P \times P$ — множество дуг, отражающих направления вызова обобщёнными процедурами друг друга. При этом каждой процедуре p_i ставится в соответствие упорядоченный набор оценок $\vec{m}_i = \vec{M}(p_i)$, каждое значение которого m_{ij} в простейшем случае соответствует оценке j-го формального параметра i-й обобщённой процедуры. При этом формальные параметры в зависимости от направления передачи данных по отношению к процедуре p_i подразделяются на in- и out-параметры. В случае расширения модели — упорядоченный набор значений $\vec{m}_{ij} = (m_{ij1}, m_{ij2}, \ldots)$, каждое из которых является оценкой одного из свойств параметра.

Среди таких свойств главным является возможность прямой подстановки данных, соответствующих этому параметру, в SQL-запрос. В базовом варианте модели значениями этого свойства могут являться U (подстановка приводит к уязвимости) и S (подстановка допускается). С целью сокращения доли ложноположительных результатов при применении модели, обусловленных ограниченной возможностью распознавания алгоритмов, обеспечивающих надлежащую обработку данных, средствами абстрактной интерпретации, возможно также дополнение шкалы значением UDS (подстановка допускается по мнению пользователя).

Другим возможным свойством, оценка которого позволяет повысить достоверность результатов анализа и упростить формирование диагностических сведений об обнаруженных уязвимостях, являются сведения о компоненте web-приложения, который является с точки зрения исходного кода окончательным приёмником соответствующих данных.

В современных языках программирования, применяемых для разработки web-приложений, как правило, взаимодействие с СУБД организовано посредством вызова встроенных или библиотечных

подпрограмм, исходный код которых при анализе недоступен. Параметры соответствующих таким подпрограммам обобщённых процедур получают оценки путём их назначения на основании сведений из документации к языку программирования или библиотеке, предоставляющим эти подпрограммы, до начала анализа. Эти обобщённые процедуры выступают в роли окончательного приёмника данных и вместе с этим отслеживание того, какая из них станет таковым для конкретного элемента входных данных, при использовании web-приложением различных СУБД или способов доступа к ним позволяет определить не только факт наличия уязвимости, но и конкретный компонент web-приложения, который будет являться фактическим вектором атаки.

Аналогичным образом могут быть назначены оценки и для входных данных web-приложения. В этом случае сопоставление оценок для параметров обобщённых процедур и для передаваемых им фактических параметров позволяет не только обнаружить факт уязвимости web-приложения к SQL-инъекции, но и локализовать участок исходного кода, где вероятнее всего допущена соответствующая ошибка.

Результаты такого анализа исходного кода предлагается использовать не только для обнаружения уязвимостей в web-приложении, но и в качестве исходных данных для прогнозирования сроков завершения работы над проектом и принятия управленческих решений.

Основу для такого применения закладывает международный стандарт ISO/IEC 25010:2011 [5]. Модель качества продукта, представленная в данном стандарте, включает в себя три уровня: характеристик, подхарактеристик и мер. При этом регламентированы только два верхних уровня — характеристик и подхарактеристик, — а расширение модели с учётом потребностей конкретного вида программных продуктов и/или специфики проекта осуществляется за счёт добавления мер к имеющимся подхарактеристикам.

Среди предлагаемых мер, которыми можно расширить модель качества продукта, особый интерес в рассмотренном выше контексте представляет мера «Устойчивость к SQL-инъекциям», значение которой предлагается вычислять по формуле

$$X = 1 - \frac{A}{B} \quad (1),$$

где X — рассчитываемое значение меры, A — количество in-параметров процедур взаимодействия с СУБД, имеющих оценку S и получающих фактические параметры с оценкой U, B — общее количество in-параметров процедур взаимодействия с СУБД. При этом следует учитывать только обобщённые процедуры, выступающие в роли окончательного приёмника данных.

Данную меру предлагается отнести к подхарактеристике «Целостность» характеристики «Защищённость» в модели качества продукта.

При её значении равном или близком к 0 следует говорить о чрезвычайно низком внутреннем качестве web-приложения. В модели OWASP 2021 года уязвимости в таких web-приложениях с большой вероятностью будут отнесены к категории A04 Insecure Design несмотря на то, что в них имеются уязвимости к A03 Injection. Значение же равное или близкое к 1 свидетельствует о высоком внутреннем качестве продукта.

Наблюдение за динамикой изменения значения как этой меры в отдельности, так и интегральной оценки качества, построенной на основе совокупности других мер, вычисляемых по результатам применения модели обнаружения уязвимостей к SQL-инъекциям, позволяет своевременно принимать управленческие меры. В частности снижение значения меры «Устойчивость к SQL-инъекциям» может как являться отражением включения в состав команды разработчика нового человека, не имеющего надлежащей квалификации в вопросах недопущения SQL-инъекций (что означает необходимость организовать дообучение), так и свидетельствовать о более серьёзных проблемах в команде или на проекте: неэффективность организации процесса разработки, превышение критического уровня технического долга, применение новых сторонних библиотек без должной оценки и т.п. Косвенно на возможные причины могут указать изменения значений других мер, базирующихся на измерениях в ходе анализа исходного кода web-приложения с использованием предлагаемой модели.

Список использованных источников:

1. OWASP Top 10 – 2017 [Электронный ресурс]. — Режим доступа: https://wiki.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf. — Дата доступа: 08.04.2024.

2. OWASP Top 10:2021 [Электронный ресурс]. — Режим доступа: https://owasp.org/Top10/. Дата доступа: 08.04.2024.

3. Оношко Д.Е., Бахтизин В.В. Метод оценки качества web-приложений, основанный на обнаружении уязвимостей // Цифровая трансформация. 2018. №1(2). С. 58–65.

4. Оношко Д.Е., Бахтизин В.В. Расширение модели обнаружения уязвимостей в web-приложениях, основанной на статическом анализе исходных кодов // Компьютерные системы и сети: материалы 56-й научной конференции аспирантов, магистрантов и студентов, Минск, 21 – 24 апреля 2020 г.

5. ISO/IEC 25010:2011, Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — System and software quality models.