DOI 10.37882/2223-2966.2026.02-2.32

ПОДХОД К ДЕТЕКТИРОВАНИЮ СЛЕДОВ ЭКСПЛУАТАЦИИ УЯЗВИМОСТЕЙ ВЕБ-ПРИЛОЖЕНИЙ КЛАССА INJECTIONS НА ПРИМЕРЕ УЯЗВИМОСТЕЙ ХРАНИМОГО МЕЖСАЙТОВОГО СКРИПТИНГА И УДАЛЕННОГО ВКЛЮЧЕНИЯ ФАЙЛОВ

AN APPROACH TO DETECTING TRACES OF EXPLOITING VULNERABILITIES IN WEB APPLICATIONS OF THE INJECTIONS CLASS USING THE EXAMPLE OF VULNERABILITIES IN STORED CROSS-SITE SCRIPTING AND REMOTE FILE INCLUSION

G. Shipulin

Summary. The article is devoted to issues related to the identification of traces of exploitation of vulnerabilities in web applications of the Injections class according to the updated OWASP Top 10 classification of 2025 using the example of vulnerabilities in stored cross-site scripting and remote file inclusion. The proposed approach to detecting post-exploitation (traces of exploitation) vulnerabilities is based on comparing the current state of a web application with its image of a reference state and subsequent analysis of changes in web application data (files and rows of target database tables) and includes five procedures: forming an image of a reference state; determining changes in the current state; analyzing and evaluating changes in the current state. states; updating data on changes to the current and previous states; updating the image of the reference state of the system. To test the proposed approach in relation to detecting traces of exploitation of vulnerabilities in stored cross-site scripting and remote inclusion of files, a software tool was developed and tested to confirm its applicability. The limited application of the proposed approach is determined by the peculiarities of exploiting certain vulnerabilities of the Injections class, as a result of which no changes are made to the structure and content of the attacked web application.

Keywords: vulnerabilities, web applications, stored cross-site scripting, detection of post-exploitation of web vulnerabilities, traces of exploitation of web vulnerabilities, OWASP, information security.

Шипулин Георгий Фаризович
кандидат юридических наук, доцент РТУ МИРЭА;
доцент, Московский политехнический университет
podumai_nad@mail.ru

Аннотация. Статья посвящена рассмотрению вопросов, связанных с выявлением следов эксплуатации уязвимостей веб-приложений класса Injections согласно обновленной классификации OWASP Top 10 2025 года на примере уязвимостей хранимого межсайтового скриптинга и удаленного включения файлов. Предложенный подход к детектированию постэксплуатации (следов эксплуатации) уязвимостей основан на сравнении текущего состояния веб-приложения с его образом эталонного состояния и последующим анализом изменений данных веб-приложения (файлов и строк целевых таблиц базы данных) и включает пять процедур: формирование образа эталонного состояния; определение изменений текущего состояния; анализ и оценка изменений текущего состояния; обновление данных об изменениях текущего и предыдущего состояний; обновление образа эталонного состояния системы. Для апробации предложенного подхода в отношении детектирования следов эксплуатации уязвимостей хранимого межсайтового скриптинга и удаленного включения файлов было разработано и протестировано программное средство, подтверждающее его применимость. Ограниченность применения предложенного подхода определяется особенностями эксплуатации некоторых уязвимостей класса Injections, в результате которых не вносится изменений в структуру и содержимое атакованного веб-приложения.

Ключевые слова: уязвимости, веб-приложения, хранимый межсайтовый скриптинг, детектирование постэксплуатации веб-уязвимостей, следы эксплуатации веб-уязвимостей, OWASP, информационная безопасность.

Проблема своевременного обнаружения успешно эксплуатируемых уязвимостей веб-приложений, к которым относятся также уязвимости хранимого межсайтового скриптинга и удаленного включения файлов, является актуальной ввиду большого количества совершаемых атак на веб-приложения. Так, согласно отчету компании «Webmonitorx» за 2025 г. было зафиксировано более 870 миллионов попыток эксплуатации разных уязвимостей, из которых порядка четверти относились к межсайтовому скриптингу [1].

Уязвимости удаленного включения файлов (Remote File Inclusion — RFI) и хранимого межсайтового скриптинга (Stored XSS) относятся к категории A05:2025 —

Injections обновленной в конце 2025 года классификации OWASP Top 10 2025, что также отражает широкую распространенность эксплуатации и опасность данных уязвимостей веб-приложений. Детальная техническая спецификация каждого из рассматриваемых видов уязвимостей описана в системе категоризации уязвимостей аппаратного и программного обеспечения Common Weakness Enumeration (CWE). [2,3]

Таким образом, не только задачи определения уязвимых сущностей веб-приложения [4], но и постэксплуатации (следов эксплуатации) уязвимостей данного класса являются актуальными.

Предлагаемый в исследовании подход к детектированию следов эксплуатации уязвимостей класса Injections основан на сравнении текущего состояния веб-приложения с его эталонным состоянием посредством анализа изменений данных веб-приложения (файлов и строк целевых таблиц базы данных). Также предполагается многоуровневая нормализация данных, подсчет весов **признаков (следов) эксплуатации** и динамическое подавление легитимных изменений через адаптивный механизм учета легитимных изменений состояния системы.

Преимуществами данного подхода является минимизация ложноположительных срабатываний и фокусирование только на неизвестных изменениях с адаптацией к легитимной динамике изменения контента без необходимости пересоздания эталонного образа. *Ограниченность применения определяется особенностями эксплуатации уязвимостей класса Injections, в результате которых не вносится изменений в структуру и содержимое атакованного веб-приложения*.

Сам подход к детектированию постэксплуатации уязвимостей класса Injections определяется следующими четырьмя взаимосвязанными процедурами и одной отдельной процедурой обновления образа эталонного состояния системы:

1. формирование образа эталонного состояния;
2. определение изменений текущего состояния;
3. анализ и оценка изменений (дельт) текущего состояния;
4. обновление данных об изменениях текущего и предыдущего состояний.

Процедуры определения изменений текущего состояния, анализа и оценки изменений (дельт) текущего состояния, обновления данных об изменениях текущего и предыдущего состояний выполняются последовательно непрерывным циклом.

Процедура формирования образа эталонного состояния системы выполняется однократно перед выполнением всех последующих процедур. Она заключается в последовательном обращении к файлам веб-приложения и таблицам баз данных, предварительно заданных в виде целевого набора, с вычислением их хэш-значений на основе алгоритма SHA256.

Рекурсивный обход файловой системы выполняется от заданного корневого каталога веб-приложения с учетом списка исключений, при этом для каждого файла вычисляется его хэш и фиксируются его атрибуты (путь, размер, время модификации). Взаимодействие с целевыми таблицами базы данных выполняется следующим образом: для каждой записи вычисляется хэш по значимым полям и формируется набор пар вида: «первичный ключ — хэш», что позволяет сопоставлять изменения между состояниями.

Таким образом, формируется образ эталонного состояния системы, включающий реестр хэшей файлов веб-приложения и реестр хэшей строк целевых таблиц базы данных, используемый в дальнейшем для определения и анализа изменений в текущем состоянии системы.

Процедура определения изменений текущего состояния выполняется повторный обход каталога веб-приложения и целевых таблиц базы данных с вычислением хэшей по аналогии с предыдущей процедурой. Далее выполняется сопоставление текущего и эталонного (при первом выполнении процедуры) или предыдущего (при последующих выполнениях процедуры) состояний на основе вычисленных хэшей, в результате чего определяются новые, удаленные или модифицированные файлы и записи в таблицах базы данных. В результате сформированный набор (список) отличий с полем веса признака эксплуатации (с заданным нулевым значением) в каждой записи передается в качестве входных данных следующей процедуре.

Из сформированного набора в рамках **процедуры анализа и оценки изменений (дельт) текущего состояния** выполняется последовательное обращение к соответствующим файлам и записям таблиц базы данных с опциональным декодированием информации (строк) и сканирование посредством набора регулярных выражений. При каждом обнаружении сигнатур эксплуатации уязвимостей, в частности, межсайтового скриптинга и удаленного включения файлов, соответствующим регулярным выражением в анализируемом файле или записи таблицы базы данных значение поля веса инкрементируется соответствующей записи сформированного набора на заранее заданное в регулярном выражении значение.

При прохождении порогового значения веса считается, что анализируемый объект (файл или запись таблицы базы данных) содержит признаки постэксплуатации уязвимостей.

Процедура обновления данных об изменениях текущего и предыдущего состояний состоит в адаптивном учете легитимных изменений состояния системы, а именно в формировании/обновлении образа текущего состояния системы на основе сформированного в процедуре определения изменений текущего состояния, насыщенного недостающими данными из образа эталонного состояния системы.

Процедура обновления образа эталонного состояния системы аналогична процедуре формирования образа эталонного состояния.

Для реализации данного подхода в отношении детектирования постэксплуатации уязвимостей Stored XSS и RFI было разработано программное средство на языке программирования Python3 [5]. Выбор используемого языка программирования обоснован кроссплатформенностью и широкими возможностями обработки текстовых данных и работы с файловыми системами и базами данных.

Архитектура разработанного программного средства включает следующие модули:

- интерфейса управления;
- сканирования файлов;
- сканирования базы данных;
- нормализации и анализа данных и состояний;
- формирования отчетов.

Функциональные возможности, порядок работы и взаимодействие модулей определяются описанными ранее процедурами.

Само программное средство включает два файла: скрипт postexp_guard.py и конфигурационный файл config.json. В файле postexp_guard.py реализованы все модули программного средства, включая интерфейс управления, сканирование файловой системы и базы данных, декодирование и анализ их содержимого, а также формирование отчетов о результатах выполнения. Файл config.json содержит конфигурации режимов работы и параметров сканирования, в том числе директорию каталога веб-приложения, перечень исключаемых каталогов и файлов, параметры подключения к системе управления базами данных, целевой набор таблиц и полей для анализа, пороговые значения весов, а также набор регулярных выражений.

Скрипт postexp_guard.py запускается через командную строку интерпретатора операционной системы и поддерживает две основные команды:

- «init» — запуск создания образа эталонного состояния системы;
- «scan» — запуск сканирования (проверки).

Дополнительно реализован флаг «-promote» команды «init», позволяющий инициировать контролируемое обновление образа эталонного состояния системы.

Для проверки работоспособности разработанного программного средства, реализующего предложенный подход по отношению к уязвимостям хранимого межсайтового скриптинга и удаленного включения файлов, был проведен эксперимент.

В рамках эксперимента было развернуто на базе виртуальной машины под управлением операционной системы Linux Ubuntu 22.04 веб-приложение dvwa, предназначенное для выработки практических навыков тестирования на проникновение веб-приложений [6].

Для формирования образа эталонного состояния веб-приложения были настроен конфигурационный файл config.json и запущена утилита postexp_guard.py с командой «init» (рис. 1).

Далее была выполнена эксплуатация уязвимости хранимого межсайтового скриптинга восемью разными векторами [7] с разными уровнями сложностями, выставленными в настройках самого веб-приложения dvwa (определяются реализованными механизмами безопасности) на веб-странице http://IP-адрес/DVWA/vulnerabilities/xss_s/ данного веб-приложения.

После чего было выполнено сканирование с целью выявления следов эксплуатации уязвимостей хранимого межсайтового скриптинга посредством запуска утилиты postexp_guard.py с командой «scan» (рис. 2).

Для проверки работоспособности разработанного программного средства в отношении детектирования признаков эксплуатации уязвимостей удаленного включения файлов была выполнена эксплуатация данного уязвимости тремя разными векторами с разными уровнями сложностями, выставленными в настройках самого веб-приложения dvwa (определяются реализованными механизмами безопасности) на веб-странице

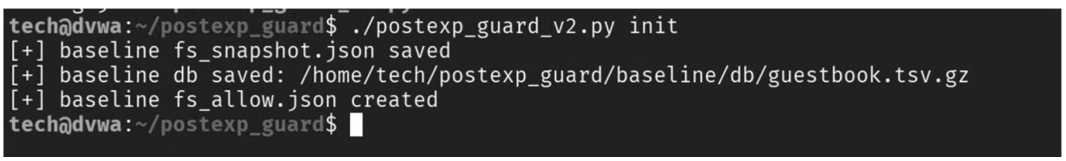

Рис. 1. Создание образа эталонного состояния веб-приложения dvwa (источник: разработка автора)

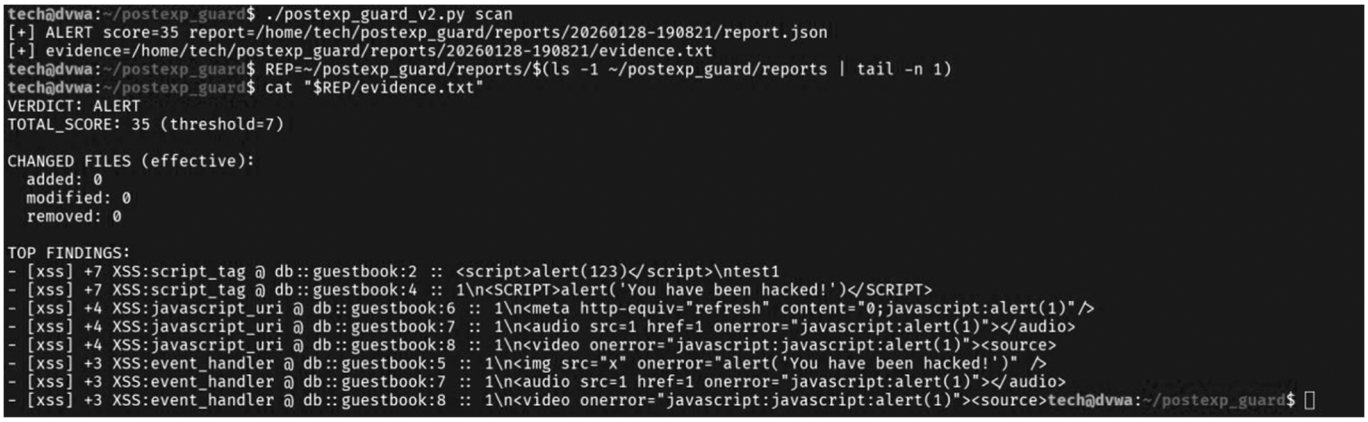

Рис. 2. Вывод результатов выполнения разработанного программного средства (источник: разработка автора)

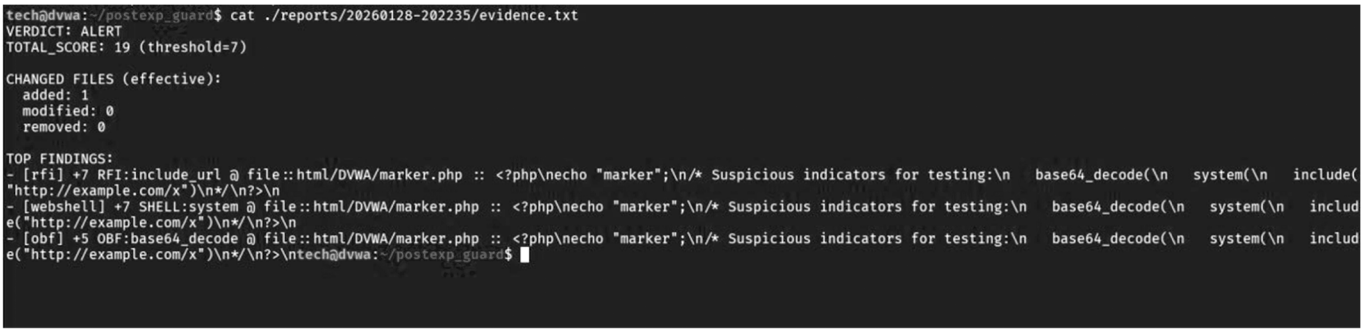

Рис. 3. Вывод результатов выполнения разработанного программного средства (источник: разработка автора)

http://IP-адрес/DVWA/vulnerabilities/upload/ данного веб-приложения.

Также по аналогии было выполнено сканирование с целью выявления следов эксплуатации уязвимостей удаленного включения файлов посредством запуска утилиты postexp_guard.py с командой «scan» (рис. 3).

Как видно, программное средство отработало корректно и определило все следы эксплуатации уязвимостей хранимого межсайтового скриптинга и удаленного включения файлов, что подтверждает применимость предложенного подхода.

Таким образом, был описан предлагаемый подход к детектированию следов эксплуатации уязвимостей веб-приложений класса Injections на уровне процедур, определены его преимущества и ограничения. Описана архитектура разработанного программного средства, реализующего данный подход в отношении следов эксплуатации уязвимостей хранимого межсайтового скриптинга и удаленного включения файлов, а также представлены результаты его применения.

ЛИТЕРАТУРА

1. Отчет о веб-атаках на онлайн-ресурсы российских компаний за 3 квартала 2025 года // Webmonitorx URL: https://webmonitorx.ru/trends/issledovaniya/otchet-o-veb-atakakh-na-onlayn-resursy-rossiyskikh-kompaniy-za-3-kvartala-2025-goda/ (дата обращения: 20.12.2025).
2. OWASP Top 10:2025 // owasp.org URL: https://owasp.org/Top10/2025/ (дата обращения: 25.12.2025).
3. CWE CATEGORY: OWASP Top Ten 2021 Category A03:2021 — Injection // cwe.mitre.org URL: https://cwe.mitre.org/data/definitions/1347.html (дата обращения: 26.12.2025).
4. Шипулин, Г.Ф. Построение системы поиска уязвимых к межсайтовому скриптингу сущностей веб-приложений / Г.Ф. Шипулин, А.Д. Шабалин, С.В. Спевакова // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. — 2025. — № 4-2. — С. 144–146. — DOI 10.37882/2223-2966.2025.04-2.34.
5. Our Documentation // Python.org URL: https://www.python.org/doc/ (дата обращения: 04.01.2026).
6. GitHub: DVWA // GitHub URL: https://github.com/digininja/DVWA (дата обращения: 15.01.2026).
7. GitHub: XSS Vectors Cheat Sheet // GitHub URL: https://gist.github.com/kurobeats/9a613c9ab68914312cbb415134795b45 (дата обращения: 16.01.2026).

Журнал «Современная наука: актуальные проблемы теории и практики»