

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Информационная безопасность

Работа 4

Анализ уязвимостей веб-приложения с помощью OWASP ZAP

Группа: Р3416

Выполнил:

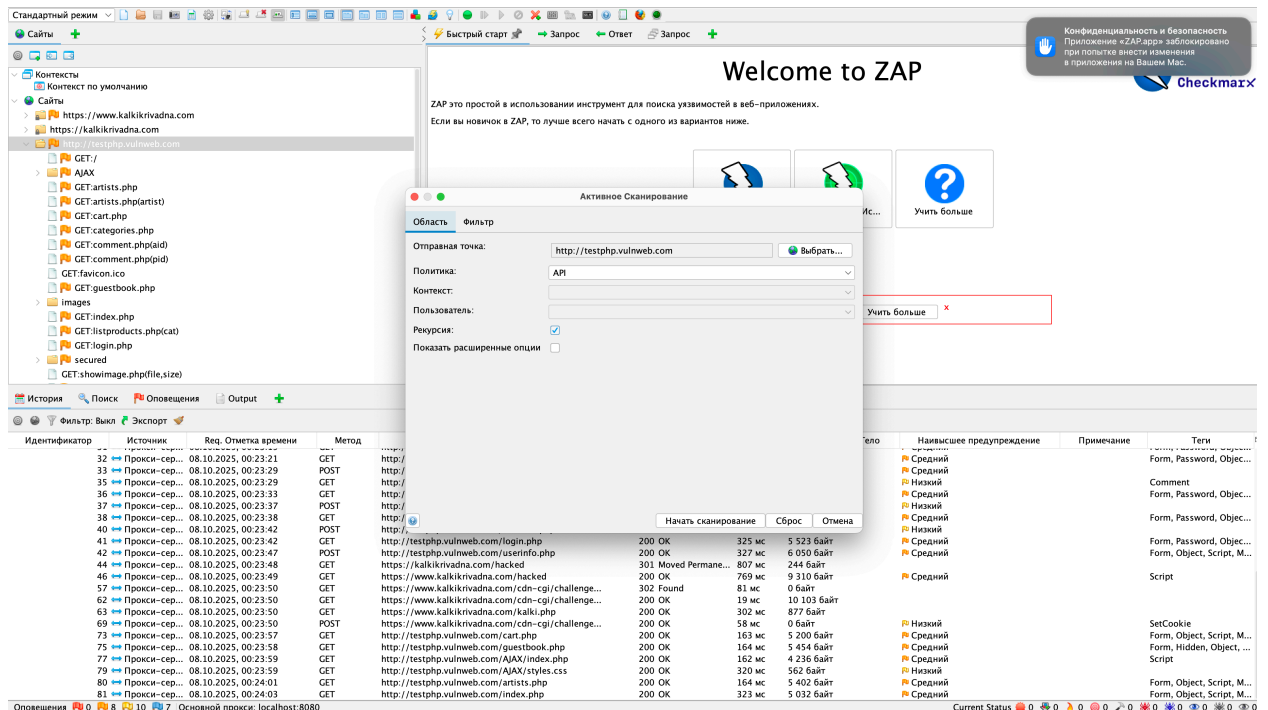
Сиразетдинов Азат Ниязович

г. Санкт-Петербург

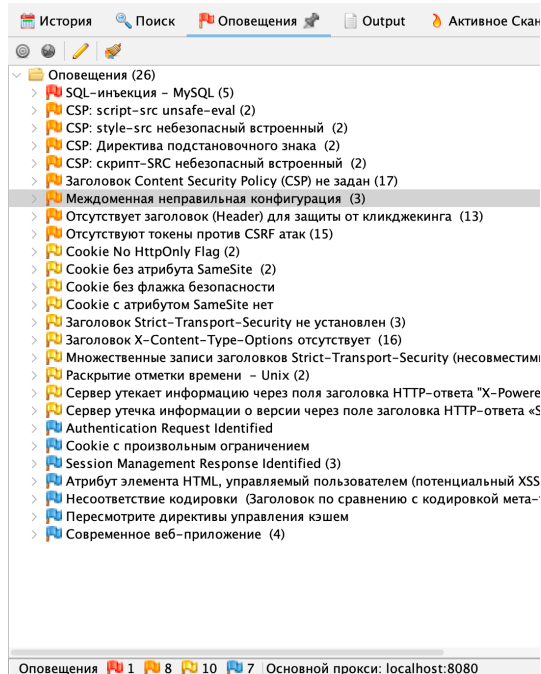
2025

Выполнение

Выполняем сканирование с помощью OWASP ZAP. Для сканирования был выбран сайт <http://testphp.vulnweb.com/>



В процессе сканирования были выявлены уязвимости: 1 high, 8 medium, 10 low и 7 informational



Уязвимость 1: SQL инъекция

Уровень риска

Высокий(High). Количество экземпляров – 5


Причина

Пользовательский ввод попадает в SQL запрос без экранирования. Злоумышленник может удалять/изменять/получать данные из таблицы используя фразы SQL.

Обнаружение

Чтобы обнаружить наличие SQL инъекции можно передать невалидный SQL скрипт. Одним из вариантов является одинарная кавычка. SQL вернет ошибку, что кавычка не имеет закрывающую кавычку

Пример

SQL-инъекция – MySQL	
URL-адрес:	http://testphp.vulnweb.com/artists.php?artist=%27
Риск:	 High
Достоверность:	Medium
Параметр:	artist
Атака:	'
Доказательства:	You have an error in your SQL syntax
CWE ID:	89
WASC ID:	19
Источник:	Активная (40018 – SQL-инъекция)
Input Vector:	URL Query String
Описание:	SQL injection may be possible.
Дополнительно:	RDBMS [MySQL] likely, given error message regular expression [^QYou have an error in your SQL syntax\E] matched by the HTML results. The vulnerability was detected by manipulating the parameter to cause a database error message to be returned and recognised.
Решение:	Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side. If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?'
Ссылка:	https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

При запросе <http://testphp.vulnweb.com/artists.php?artist=%27> параметр artist не экранируется при запросе в базу данных, вследствие чего одинарная кавычка вызывает ошибку SQL синтаксиса

Уязвимость 2: XSLT Инъекция

Уровень риска

Средний (Medium). Количество экземпляров – 2

Причина

Текстовый ввод попадает в XSLT-процессор без дальнейшего форматирования. Это позволяет злоумышленнику добавлять XSL трансформации и получить содержимое системных файлов, делать запросы на внутренние порты, вызывать удаленное исполнение кода

Обнаружение

Для обнаружения XSLT инъекции можно передать невалидную XSL трансформацию. Например, двойные кавычки, одинарные кавычки и угловые скобки

Пример

HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Tue, 07 Oct 2025 22:03:31 GMT
Content-Type: image/jpeg
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
content-length: 286

Warning: fopen(<xsl:value-of select="document('http://testphp.vulnweb.com:22')"/>): failed to open stream: No such file or directory in /hj/var/www/showimage.php on line 13

Warning: fpassthru() expects parameter 1 to be resource, boolean given in /hj/var/www/showimage.php on line 19

XSLT Инъекция
URL-адрес: http://testphp.vulnweb.com/showimage.php?file=%3Cxsl%3Avalue-of+select%3D%22document%28%27http%3A%2F%2Ftestphp.vulnweb.com%3A22%27%29%22%2F%3E
Риск: Medium
Достоверность: Medium
Параметр: file
Атака: <xsl:value-of select="document('http://testphp.vulnweb.com:22')"/>
Доказательства: failed to open stream
CWE ID: 91
WASC ID: 23
Источник: Активная (90017 – XSLT Инъекция)
Input Vector: URL Query String
Описание: Внедрение с использованием преобразований XSL может быть возможным и может позволить злоумышленнику читать системную информацию, читать и записывать файлы или выполнять произвольный код.
Дополнительно: Возможно сканирование портов.

При запросе <http://testphp.vulnweb.com/showimage.php?file=%3Cxsl%3Avalue-of+select%3D%22document%28%27http%3A%2F%2Ftestphp.vulnweb.com%3A22%27%29%22%2F%3E> мы передаем XSL трансформацию

<xsl:value-of select="document('http://testphp.vulnweb.com:22')"/> что вызывает ошибку XSLT-процессора

Уязвимость 3: Отсутствует заголовок (Header) для защиты от кликджекинга

Уровень риска

Средний (Medium). Количество экземпляров – 21

Причина

Сайт не защищен от кликджекинга. Это позволяет злоумышленнику использовать невидимые iframe для совершения действий на уязвимом сайте

Обнаружение

Для обнаружения нужно проверить наличие заголовка X-Frame-Options (позволит ограничить появление сайта в iframe) или Set-Cookie (запретит использование cookie файлов на других доменах)

Пример

Заголовок: Текст

Тело: Текст

HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Tue, 07 Oct 2025 21:22:43 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
content-length: 4958

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<!-- InstanceBeginEditable name="document_title_rgn" -->
<title>Home of Acunetix Art</title>
<!-- InstanceEndEditable -->

Классификация +

Отсутствует заголовок (Header) для защиты от кликджекинга

URL-адрес: http://testphp.vulnweb.com/
Риск: Medium
Достоверность: Medium
Параметр: x-frame-options
Атака:
Доказательства:
CWE ID: 1021
WASC ID: 15
Источник: Пассивный (10020 – Заголовок против кликджекинга)
Alert Reference: 10020-1
Input Vector:
Описание:
The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.
Дополнительно:

При запросе на <http://testphp.vulnweb.com/> отсутствуют заголовки, защищающие от кликджекинга