|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ | | | | | | | | | | | | | | | | | | | | |
| Федеральное государственное автономное образовательное  учреждение высшего образования | | | | | | | | | | | | | | | | | | | | |
| **Дальневосточный федеральный университет** | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
| **ШКОЛА ЕСТЕСТВЕННЫХ НАУК** | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
| **Кафедра информационной безопасности** | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
| **О Т Ч Е Т** | | | | | | | | | | | | | | | | | | | | |
| о прохождении учебной практики (учебно-лабораторного практикума) | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | |  |  | Выполнил студент  гр. С8117-10.05.01ммзи | | | | | | | | |
|  |  |  | | | | | | Волошин Н.Д. | | |
|  | | | | | | | | | |  |  | (подпись) | | | | | |  | | |
|  | | | | | | | | | | | | | | | | | | | | |
| Отчет защищен с оценкой | | | | | | | | | |  |  | Руководитель практики | | | | | | |  | |
|  | | | | | | | | | |  |  | Старший преподаватель кафедры информационной безопасности ШЕН | | | | | | | | |
|  | | | |  | С.С. Зотов | | | | |  |  |  | | | | |  | С.С. Зотов | | |
| (подпись) | | | |  | (И.О. Фамилия) | | | | |  |  | (подпись) | | | | |  | (И.О. Фамилия) | | |
| « | 26 | » | июня | | | | | | 2021 г. |  |  |  | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
| Регистрационный № | | | | | | | |  | |  |  | Практика пройдена в срок | | | | | | | | |
| « | 26 | » | июня | | | | | | 2021 г. |  |  | с | « | 22 | » | февраля | | | | 2021 г. |
|  | | | | | | | | | |  |  | по | « | 26 | » | июня | | | | 2021 г. |
|  | | | | | |  | Е.В. Третьяк | | |  |  | на предприятии | | | | | | | | |
| (подпись) | | | | | |  | (И.О. Фамилия) | | |  |  |
|  | | | | | | | | | |  |  | Кафедра информационной | | | | | | | | |
|  | | | | | | | | | |  |  | безопасности ШЕН ДВФУ | | | | | | | | |
|  | | | | | | | | | |  |  |  | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
| г. Владивосток | | | | | | | | | | | | | | | | | | | | |
| 2021 | | | | | | | | | | | | | | | | | | | | |

**Содержание**

Задание на практику 3

Введение 4

1 Описание веб-приложений 5

2 Реализация А1, A5, A8 9

Заключение 16

Список использованных источников 17

**Задание на практику**

- Проведение исследования в области защиты от взлома веб-приложений.

- Написание отчета по практике о проделанной работе.

**Введение**

Учебная практика (учебно-лабораторный практикум) проходил на кафедре информационной безопасности ШЕН ДВФУ в период с 22 февраля 2021 года по 26 июня 2021 года.

Целью прохождения практики является приобретение практических и теоретических навыков по специальности, а также навыков оформления проведенного исследования в отчетной форме.

Задачи практики:

1. Исследовать нормативную базу по информационной и компьютерной безопасности в области уязвимостей веб-приложений.
2. Реализовать 9 веб-приложений c уязвимостям из OWASP Top 10.
3. На основе полученных знаний написать отчет по практике о проделанной работе.

# **1 Описание веб-приложений**

**1.1 A1. Injections**

По этой уязвимости реализовывалось два веб-приложения: первое веб-приложение содержит NOSQL уязвимость, второе – SQL уязвимость.

В обоих случаях реализовывалась форма отправки данных на клиенте с именем пользователя и паролем и соответствующая логика, допускающая соответствующую инъекцию, на сервере.

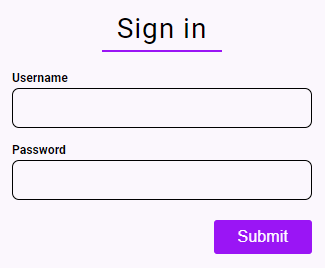


Рисунок 1 – Форма отправки данных

При успешном проведении атаки пользователю высвечивается флаг.

**1.2 A2. Broken auth**

Данный случай показывает, как плохие требования к паролю могут привести к соответствующим последствиям. Клиент состоит из двух частей: из формы отправки данных (форма входа) и формы регистрации.

В форме регистрации есть требование к паролю, которое подразумевает, что пароль должен состоять только из 6 цифр, не больше и не меньше.

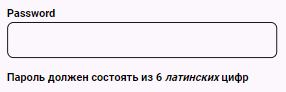


Рисунок 2

Данный случай легко решается перебором всех возможных комбинаций пароля.

**1.3 A3. Cross Site Scripting (XSS)**

Этот случай нацелен на проведение XSS-атаки [1]. На клиенте расположена форма отзывов к продукту, где поле комментария уязвимо к XSS-атаке. Пользователь отправляет данные с JavaScript(JS)-кодом, на сервере этот код никак не обрабатывается, и всё сохраняется в базу данных. Если любой пользователь зайдёт на страничку с этими отзывами, он получит отзыв с JS кодом, который запустится на его клиенте.

**1.4 A4. Insecure dor**

Клиент веб приложение состоит из трёх частей: форма входа, отображение данных пользователя и страница 404.

Случай заключается в том, что пользователь по URL может просмотреть данные всех людей, которые пользуются данным приложением, если же такого пользователя не существует, то показывается страница 404.

**1.5 A5. Miscofiguration**

Пользователю необходимо суметь авторизироваться в системе, проанализировав какая серьёзная уязвимость скрывается на сервере.

Клиент состоит из двух частей: поле поиска файлов и формы авторизации.

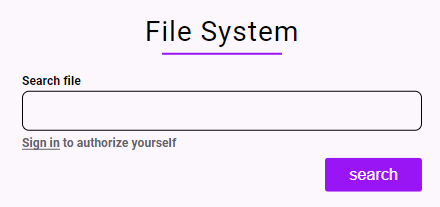


Рисунок 3 – Поисковая система файлов

**1.6 A6. Sensetive data exposure**

Целиком и полностью весь случай состоит из двух клиентов и сервера. Случай подразумевает клиент пользователя, которому необходимо войти в систему и клиент жертвы.

Клиент жертвы работает следующим образом: каждые 5 секунд делается POST запрос на авторизацию с валидными данными, уязвимость заключается в том, что данные передаются по HyperText Transfer Protocol (HTTP).

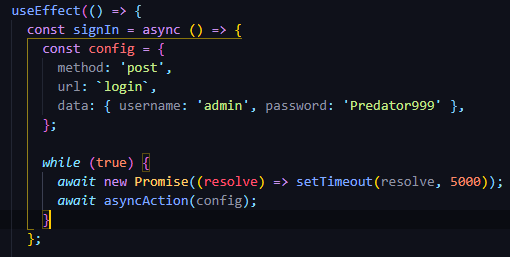


Рисунок 4 – Код автоматической авторизации

Пользователю необходимо прослушать траффик внутри сети Wi-Fi и авторизироваться, узнав данные жертвы.

**1.7 A7. Access controls**

Веб приложение показывает список пользователей, зарегистрированных в системе веб приложения с дезактивированной кнопкой удаления пользователя. Пользователю необходимо сделать запрос на удаление пользователя из списка. Уязвимость заключается в том, что на сервере не проверяются права пользователя, удаляющего пользователя из базы данных.

**1.8 A8. Cross-Site Request Forgery (CSRF)**

Данное приложение направленно на проведение CSRF-атаки [2].

**2 Реализация A1, A5, A8**

Моей задачей было реализовать сервер (API) для проведения NOSQL-инъекции(A1), а также полноценно реализовать A5 и A8, а также полностью обернуть A1, A5 и А8 в Docker-контейнеры.

**2.1 A1**

API состоит из двух endpoit’ов – это авторизация и регистрация. Регистрация нужна для удобного добавления пользователей в систему через такой клиент как Postman.

NOSQL-инъекция находится в поле пароля, допустим: { $ne: “” }, что в купе с именем пользователя admin будет означать следующее при отправки данных на сервер: найти пользователя с именем admin и паролем, который не равен пустой строке.

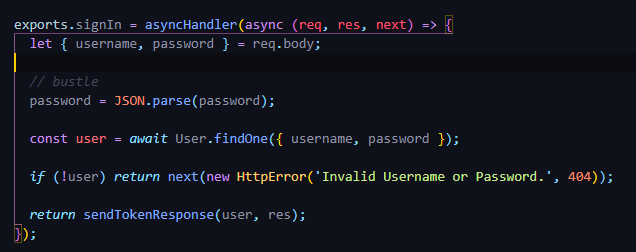


Рисунок 6 – Уязвимый код аутентификации

Чтобы сервер не подвергся такой атаке, достаточно найти пользователя по его имени в базе данных, и если пользователь существует, сравнить хеши паролей – пароля введённого пользователем и пароля из базы данных.

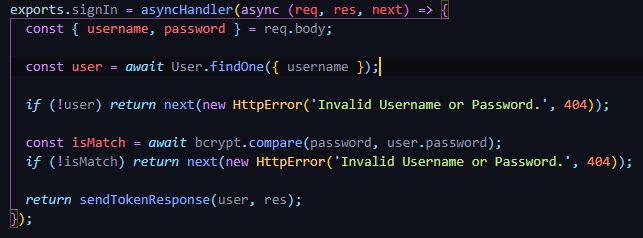


Рисунок 7 – Правильный код аутентификации

**2.2 A5**

Данный случай получился достаточно необычным, он включает в себя две уязвимости: Misconfiguration и Code-injection. Как только пользователь заходит в веб приложение, его встречает строка поиска файлов в системе, проблема заключается в том, что только авторизированный пользователь может ей воспользоваться.



Рисунок 8 – Поле поиска файлов

Также на клиенте располагается форма авторизации, которая никак не поможет нашему пользователю. Возвращаясь к строке поиска, пользователь может ею воспользоваться, но будет получать ошибки, и в некоторых случаях будет получать достаточно странные ошибки.



Рисунок 9 – Отображение ошибки

Загвоздка заключается в том, что пользователь введёт в строке поиска, будет интерпретироваться как код, который будет запускаться на сервере. Допустим, пользователь это понял, а также понял какой язык программирования используется на сервере, посмотрев во вкладке сеть, с помощью какого фреймворка написан сервер.



Рисунок 10 – Фреймворк сервера

Пользователь может просмотреть все файлы, которые содержит сервер. Пользователь сразу на первом уровне вложенности файлов увидит папку под названием с «secret-files», где будут находится два текстовых файла «JWT.txt» и «administrators.txt», из которых он может извлечь следующую информацию: ключ с помощью которого генерируются Json Web Token (JWT), и где токены хранятся, а также администраторы, которые зарегистрированы в данном веб-приложении.

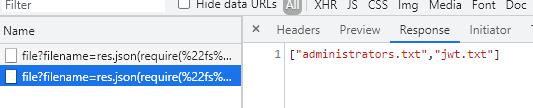


Рисунок 11 – Файлы в папке secret-files

Пользователь должен с генерировать JWT (это можно сделать с помощью скрипта, который также находится на сервере), а также прикрепить JWT к браузеру, и если же JWT – валиден, то у пользователя будет доступ к файловой системе.



Рисунок 12 – код генерации JWT

**2.3 A8**

Данные случай, как и A5, выделяется своей необычностью: он включает в себя клиент пользователя, клиент одного из злоумышленников и сервер. Клиент пользователя состоит из двух частей: форма входа (с данными) и новостной страницей.

Форма входа необходима для того, чтобы понять пользователю, как идентифицирует нас сервер – идентификация происходит по JWT, который хранится в cookie. Новостная страница состоит из следующих элементов: баланс пользователя, форма перевода средств, новость, комментарий и форма добавления комментария к новости.

Форма перевода средств деактивировна, но из этой формы можно узнать какие данные нужны для перевода средств, а также API endpoint.

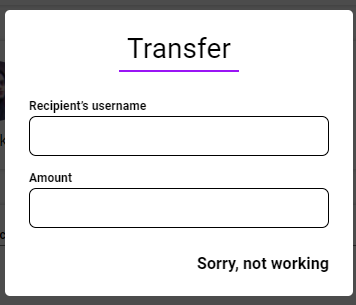


Рисунок 13 – Форма перевода средств

В комментарии, который уже прикреплён к новости, находится ссылка на клиент злоумышленника, перейдя на который, с пользователя спишется $100 с баланса – это будет означать, что над пользователем провели CSRF-атаку.

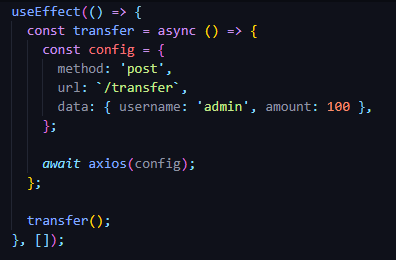


Рисунок 14 – Код на клиента злоумышленника

Данное веб-приложение реализовано с помощью веб-сокетов, которые позволяют увидеть результат сразу после успешного проведения CSRF-атаки – нет необходимости перегружать страницу, чтобы узнать, что с балансом пользователя.

Задача пользователя состоим в том, чтобы создать клиент, который произведёт CSRF-атаку над жертвой, перешедшей по ссылке, а также в добавлении комментария со ссылкой (здесь трудностей нет).

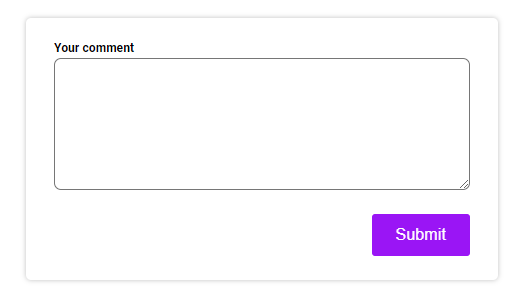


Рисунок 15 – Поле добавления комментария

**2.4 Технологии**

В качестве технологий при создании A1(API), A5, A5 использовались следующие технологии и языки программирования:

На стороне клиента использовался такой язык программирования как JavaScript, а также следующие технологии:

* react;
* axios;
* formik;
* js-cookie;
* node-sass;
* socket.io-client.

На стороне сервера использовался язык программирования Node.js и следующие технологии:

* bcryptjs;
* cookie-parser;
* cors;
* express;
* express-validator;
* jsonwebtoken;
* mongodb;
* socket.io.

База данных по А1, A5, A8 располагается в облаке на Mongo DB Atlas.

Docker compose файлы всех веб-приложений располагаются в следующем репозитории: github.com/nukuutos/owasp. Все Docker Images располагаются на Docker Hub.

**Заключение**

Для достижения данной цели, в процессе прохождения учебной практики (учебно-лабораторного практикума) исследовал нормативную базу по информационной и компьютерной безопасности в области уязвимостей веб-приложений, реализовал веб-приложения со следующими уязвимостями: A1, A5, A8.

Также были изучены требования к написанию отчета по практике. В результате прохождения практики был составлен отчет по практике, соответствующий предъявленным требованиям.

В ходе прохождения практики все задачи были выполнены, а цель достигнута.

**Список используемых источников**

1. XSS-атака [Электронный ресурс]. – Режим доступа: https://owasp.org/www-community/attacks/xss/
2. CSRF-атака [Электронный ресурс]. – Режим доступа: <https://owasp.org/www-community/attacks/csrf>