



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Информационная безопасность» (ИУ8)

ОТЧЁТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Тип практики: учебно-технологическая

Название предприятия: ООО «Стройгрупп»

Студент:


Тимошенко Юлия Геннадьевна, группа ИУ8-43 (2 курс)

 02.08.21
(подпись, дата)

Руководитель от предприятия:

Директор Шевченко Василий Анатольевич



 02.08.21
(подпись, дата)

Руководитель от кафедры:

доцент кафедры ИУ8 Зайцева Анастасия Владленовна

(подпись, дата)

Оценка: 5



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)
КАФЕДРА «Информационная безопасность» (ИУ8)

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ

Тип практики: учебно-технологическая

учебно-технологическая

Название предприятия ООО «Стройгрупп»

Сроки практики: с 19.07.2021 по 01.08.2021

Специальность / направление: 10.05.03 «Информационная безопасность автоматизированных систем»

Специализация / профиль: 10.05.03_01 «Анализ безопасности информационных систем»

За время прохождения практики студенту надлежит согласно программе практики:

- Исследовать механизмы защиты от CSRF атак с помощью CSRF токенов и Referer заголовков.

Студент:

Тимошенко Юлия Геннадьевна, группа ИУ8-43 (2 курс)

Руководитель от предприятия:

Директор Шевченко Василий Анатольевич

Руководитель от кафедры:

доцент кафедры ИУ8 Зайцева Анастасия Владленовна



Ю.Г. Тимошенко 02.08.21
(подпись, дата)

В.А. Шевченко 02.08.21
(подпись, дата)

(подпись, дата)

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ОСНОВНАЯ ЧАСТЬ	5
1 Характеристика организации	5
2 Понятие CSRF атаки.....	5
2.1 Последствия CSRF атаки	6
2.2 Механизм атаки.....	6
2.3 Исполнение CSRF атаки.....	7
3 Предотвращение CSRF атаки с помощью токенов	8
3.1 Создание CSRF токена	8
3.2 Передача токенов	8
3.3 Проверка токена в зависимости от метода запроса.....	10
3.4 Проверка токена в зависимости от его присутствия	10
3.5 Токен не привязан к сессии пользователя	10
3.6 Токен продублирован в данных cookie.....	11
4 Защита с помощью заголовка Referer	12
4.1 Проверка Referer в зависимости от присутствия заголовка	12
4.2 Простая проверка Referer	12
ЗАКЛЮЧЕНИЕ	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	14

ВВЕДЕНИЕ

Целью данной практики является изучение CSRF атак и способов защиты от нее с помощью токенов и заголовков Referer.

Для осуществления цели нужно выполнить следующие задачи:

1. Изучить понятие CSRF
2. Рассмотреть указанные методы защиты

Данное задание было выдано по следующим причинам:

- Соответствует специальности обучения,
- Проблема данных атак актуальна,
- Компания в ближайшее время планирует запуск сайта и разбор этих атак поможет избежать проблем в дальнейшем,
- Сотрудники компании сталкивались с данным видом атаки.

ОСНОВНАЯ ЧАСТЬ

1 Характеристика организации

Дата регистрации 26 мая 2015 года. Организация зарегистрирована 6 лет назад, что говорит о стабильной деятельности и поднадзорности государственным органам. Основные виды деятельности общества: строительство зданий и сооружений; производство строительных металлических конструкций, окон, дверей; разборка и снос зданий; производство земляных работ; производство электромонтажных работ; монтаж отопительных систем и систем кондиционирования; производство штукатурных, столярных, плотничных, стекольных, кровельных и прочих отделочных работ; производство гидроизоляционных работ и другие.

Компания является участником системы государственных закупок в качестве поставщика.

По данным ФНС, в прошлом отчетном периоде выручка составила 369,93 млн.руб.; чистая прибыль организации составила 40,8 млн руб., были уплачены налоги на сумму 25,98 млн руб., задолженностей по пеням и штрафам нет.

Фирма молодая, перспективная, ритмично развивающаяся, среднесписочная численность работников составляет 38 человек. Основное направление работы – это обслуживание объектов нефтяной и газовой промышленности. Места работы компании постоянно пополняется новыми регионами России и новыми объектами. Так, например, в августе 2021 года Фирма выиграла тендер по строительным работам в детском лагере «Артек» (Крым).

2 Понятие CSRF атаки

Межсайтовая подделка запроса (CSRF) – уязвимость сетевой безопасности, которая позволяет злоумышленнику тайно вызвать действия от лица пользователя. Она позволяет злоумышленнику частично обойти правило ограничения домена, предназначенное для предотвращения взаимодействия различных сайтов друг с другом. Для большинства сайтов браузеры автоматически добавляют к запросам данные, связанные с ними. Таким образом

если пользователь авторизован на каком-то сайте, без специальных методов, использующихся для защиты от данного вида атак, нельзя обнаружить запрос, отправленный самим пользователем или злоумышленником от его лица [1].

2.1 Последствия CSRF атаки

В случае успеха CSRF атаки жертва непреднамеренно выполняет какое-либо действие. Таким действием может быть перевод денежных средств, смена пароля и другое. Кроме того, злоумышленник может получить полный контроль над учетной записью пользователя. Если жертва имеет привилегированную роль в приложении, то злоумышленник может получить доступ ко всем данным и функциям [2].

2.2 Механизм атаки

Для осуществления CSRF атаки необходимы:

- Потенциально выгодное действие;
- Данные cookie. Действие включает несколько HTTP запросов, а приложение опирается только на сессионные файлы cookie для идентификации пользователя, отправившего запросы;
- Отсутствие непредсказуемых параметров запроса. Запросы, которые представляют действие, не должны содержать параметры, значения которых злоумышленник не может определить или угадать. Например, функция смены пароля защищена, если для ее исполнения требуется знать значение существующего пароля.

Например, когда пользователь меняет адрес электронной почты в своей учетной записи (при такой возможности), возникает HTTP запрос, представленный на рисунке 1 [3].

```
POST /password/change HTTP/1.1
Host: page.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 20
Cookie: session=45HYkhe49YI1092tFg4eYPj90lKder5X;
password=newpassword
```

Рисунок 1 HTTP запрос о смене пароля

Это соответствует условиям для CSRF:

В свою очередь злоумышленник может сменить пароль от имени жертвы при следующих условиях:

- Приложение использует данные cookie о сессии чтобы идентифицировать пользователя;
- Не используются средства защиты от CSRF атак;
- Злоумышленник может определить значения необходимых для данного действия параметров.

Сайт злоумышленника может содержать HTML:

Злоумышленник может разместить на своем сайте HTML, внутри которого будет запрос к уязвимому сайту. Тогда этот запрос отправится от имени жертвы. Такой HTML приведен на рисунке 2 [4].

```
<html>
<body>
  <form action="https://page.ru/password/change" method="POST">
    <input type="hidden" name="password" value="newpassword " />
  </form>
  <script>
    document.forms[0].submit();
  </script>
</body>
</html>
```

Рисунок 2 HTML, содержащий запрос смены пароля

Последствия при посещении жертвой сайта злоумышленника:

- Будет отправлен HTTP-запрос на уязвимый сайт;
- Если пользователь авторизован на данном сайте, браузер автоматически добавит данные cookie к запросу;
- Уязвимый сайт ответит на запрос, как если бы он был отправлен самим жертвой-пользователем.

2.3 Исполнение CSRF атаки

Самый распространенный способ — помещение злоумышленником вредоносного HTML на собственный сайт. Затем необходимо побудить жертву

зайти на данную страницу. Чаще всего это делается через рассылку личных сообщений по электронной почте или в социальных сетях.

Если какой-либо запрос может быть выполнен с помощью метода GET, то атака может быть осуществлена с помощью одного адреса URL на уязвимом сайте. В таком случае в URL входит советующая команда.

3 Предотвращение CSRF атаки с помощью токенов

Самый надежный способ защититься от CSRF атаки — это добавить CSRF токен (уникальный секретный ключ) к соответствующим запросам. Такой токен должен быть:

- Непредсказуемым,
- Связанным с сессией пользователя,
- Строго проверяющимся перед выполнением каждого действия.

3.1 Создание CSRF токена

CSRF токены должны обладать высокой энтропией, то есть быть непредсказуемыми, а также свойствами, которые есть у сессионных токенов.

Для создания токена следует использовать криптографически стойкий генератор псевдослучайных чисел с меткой времени.

Для более сильной защиты можно сгенерировать индивидуальные токены путем объединения сгенерированных псевдослучайных чисел и пользовательской энтропией с последующим взятием хэша. Таким образом проанализировать токены на основе какого-либо образца будет труднее.

3.2 Передача токенов

Такие токены должны оставаться секретными и обрабатываться безопасно на протяжении всего жизненного цикла. На рисунке 3 показано как вводить токен.

```
<input type="hidden" name="csrf-token"
value="Ht9glYn7vBjk6lAdrsZ2dfOq98Hpu78s" />
```

Рисунок 3 Объявление CSRF-токена

Для дополнительной безопасности поле, содержащее CSRF токен должно быть как можно раньше в HTML, лучше всего перед всеми открытыми полями

ввода и перед данными, которые контролирует пользователь. Это снижает риск получения злоумышленником частей HTML документа.

Существует также другой подход, при котором токен располагается в строке запроса URL, он является менее безопасным:

- Прописывается в разных местах на стороне клиента и сервера,
- Может быть передан третьим лицам,
- Может быть отображен на экране браузера пользователя.

Некоторые приложения передают CSRF токены через настраиваемый заголовок запроса, что обеспечивает дополнительную защиту. А именно не позволяет предугадать или захватить токен другого пользователя, потому что браузеры, как правило, запрещают пересылать настраиваемые заголовки между доменами. Для многих ситуаций этот способ является слишком сложным, так как используется XHR.

CSRF токены не должны передаваться внутри cookies.

После создания токен должен храниться на стороне сервера в данных сеанса пользователя. При получении запроса серверное приложение должно проверить наличие в запросе токена, соответствующего значению, которое было сохранено в сеансе пользователя. Если токен в запросе неправильный или отсутствует, запрос требуется отклонить. Данная проверка выполняется независимо от метода HTTP или типа содержимого запроса.

Наиболее интересные CSRF уязвимости возникают вследствие ошибок, совершенных в подтверждении CSRF токенов. На рисунке 4 представлен запрос, включающий токен для защиты.

```
POST /password/change HTTP/1.1
Host: page.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 58
Cookie: session=45HYkhe49YI1092tFg4eYPj90lKder5X;
csrf=F7ov4hTquYmzFytug8olhZBloNvZZVko&password=newpassword
```

Рисунок 4 HTTP запрос о смене пароля при наличии CSRF токена

Токен должен запрашиваться при любом из методов POST, PUT, PATCH или DELETE. То есть должно закладываться поле для токена.

Это должно предотвратить CSRF атаки, так как это нарушает обязательные условия CSRF уязвимости: приложение больше не опирается только на файлы cookies, а запрос содержит параметры, значения которых злоумышленник не может определить. Однако, существуют различные методы, в первую очередь связанные с ошибками в проверке или хранении токенов, с помощью которых можно обойти защиту.

3.3 Проверка токена в зависимости от метода запроса

Некоторые приложения правильно подтверждают токен, если используется POST метод, но ошибаются в случае GET метода.

В такой ситуации, злоумышленник может переключить на GET метод, чтобы пройти подтверждение и провести атаку. Пример представлен на рисунке 5.

```
GET /password/change?password=newpassword HTTP/1.1
Host: page.ru
Cookie: session= F7ov4hTquYmzFytug8olhZBloNv6ZVko
```

Рисунок 5 HTTP запрос о смене пароля с использованием метода GET

Таким образом для того, чтобы избежать атак с помощью этого метода, нужно запретить менять данные GET запросами, а допускать только получение данных [2].

3.4 Проверка токена в зависимости от его присутствия

Некоторые приложения проводят проверку токена правильно, но пропускают подтверждение если токен исключен.

В такой ситуации, злоумышленник может удалить целый параметр, содержащий токен, чтобы получить подтверждение.

3.5 Токен не привязан к сессии пользователя

Некоторые приложения не проверяют принадлежность токена к сеансу пользователя. Они сохраняют глобальный набор токенов и принимают любой из них.

В этом случае злоумышленник может войти в приложение, используя свой собственный аккаунт. Таким образом он обеспечит прохождение проверки токена и проведет успешную атаку.

Существуют случаи, когда приложения привязывают токен к cookie, но не к тем же, что используются для отслеживания сессий. Это происходит, например, при использовании двух различных структур для сессии и для защиты от CSRF атаки.

Этот случай сложнее для проведения атаки, но по-прежнему имеет уязвимость. Если сайт позволяет каким-либо образом установить cookie в браузере жертвы, атака возможна. Злоумышленник может войти в собственную учетную запись, тем самым получив действительный токен и связанные с ним cookie, а затем поместить их в браузер жертвы.

Установка cookie не обязана быть внутри того же веб-приложения, что и CSRF уязвимость.

3.6 Токен продублирован в данных cookie

Также некоторые приложения не поддерживают серверную запись полученных токенов, вместо этого дублируя каждый внутри данных cookie и параметре запроса. При получении запроса проводится проверка соответствия токена в запросе и значения в cookie. Такая защита получила название "double submit", ее большое преимущество в простой реализации. Рисунок 6 демонстрирует HTTP запрос в данном случае.

```
POST /password/change HTTP/1.1
Host: page.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 58
Cookie: session=45HYkhe49YI1092tFg4eYPj90lKder5X;
csrf= F7ov4hTquYmzFytug8o1hZBloNvZZVko
csrf=F7ov4hTquYmzFytug8o1hZBloNvZZVko&password=newpassword
```

Рисунок 6 HTTP запрос о смене пароля с дублированием CSRF токена

Здесь задача злоумышленника состоит в передаче своих данных cookie в браузер жертвы. Однако это возможно только при наличии функций настроек cookie.

4 Защита с помощью заголовка Referer

Для защиты от CSRF атак, вместо использования CSRF токенов, некоторые приложения применяют заголовок HTTP Referer. Он позволяет получить URL предыдущей страницы. В таком случае производится проверка того, что запрос идет от собственного домена. Этот метод менее эффективен и часто обходится злоумышленниками.

4.1 Проверка Referer в зависимости от присутствия заголовка

Некоторые приложения проводят соответствующую проверку Referer заголовка при его наличии в запросе, но пропускают ее, если заголовок опущен.

В таком случае злоумышленник может провести атаку таким образом, что в результирующем запросе заголовок Referer будет отброшен. Наиболее простой способ исполнить это использовать тег meta на соответствующей HTML странице.

4.2 Простая проверка Referer

Некоторые приложения проводят проверки заголовка Referer примитивным способом, который можно обойти. Например, проверка того, что домен начинается с ожидаемого значения. Тогда злоумышленнику достаточно поместить это значение в свой домен в качестве поддомена.

Аналогично, если приложение просто проверяет что Referer содержит его собственное имя, злоумышленнику нужно расположить это имя где-либо в URL.

Но этот подход часто не работает. Это объясняется тем, что в целях обеспечения конфиденциальности многие браузеры по умолчанию удаляют строку запроса из заголовка Referer. Так как данный заголовок может раскрыть информацию об истории посещенных пользователем страниц.

ЗАКЛЮЧЕНИЕ

В ходе данной практики были изучены CSRF атака и некоторые методы по ее предотвращению. CSRF атака нацелена на проведение действий от лица пользователя-жертвы, выгодных для злоумышленника. Однако есть методы защиты от нее с разными степенями надежности. Одним из них является заголовок HTTP Referer. Он позволяет проконтролировать источник, то есть проверить, откуда выполнен переход на страницу. Наиболее популярным и надежным методом защиты является CSRF токен. Для того, чтобы злоумышленник не мог обойти защиту, необходимо соблюдать все требования и не допустить некоторых ошибок.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Cross Site Request Forgery [Электронный ресурс]. – Режим доступа: <https://infostart.ru/public/1142804>
2. William Zeller, Edward W. Felten, Cross-Site Request Forgeries: Exploitation and Prevention [Электронный ресурс]. – Режим доступа: <https://narfu.ru/agtu/www.agtu.ru/fad08f5ab5ca9486942a52596ba6582elit.html>
. — (Дата обращения: 25.07.2021)
3. Документация HTTP [Электронный ресурс]. – Режим доступа: <https://httpwg.org/specs/>
4. Документация HTML [Электронный ресурс]. – Режим доступа: <https://devdocs.io/html/>