Ошибки путаницы привилегий: подделка межсайтовых запросов в веб-приложениях

Подъярова Ксения Витальевна (группа: НПМбд-02-21)

Российские Университет Дружбы Народов

CSRF : Подделка межсайтовых запросов

Межсайтовая подделка запросов - CSRF (Cross-Site Request Forgery) — это вид атаки на сайт, которая производится с помощью мошеннического сайта или скрипта, который заставляет браузер пользователя выполнить нежелательное действие на доверенном сайте, на котором пользователь авторизован.

Виды CSRF

Существует несколько видов атак CSRF:

- **Классический CSRF:** Злоумышленник использует незащищенные формы и ссылки для отправки запросов.
- **CSRF с использованием JavaScript:** Злоумышленник использует JavaScript-код для отправки запросов в фоновом режиме.
- **CSRF с использованием HTTP-заголовков:** Злоумышленник может использовать HTTP-заголовки, чтобы выполнить запрос.

Как работает CSRF

Чтобы CSRF атака была возможна, должны быть соблюдены три условия:

- **Активное действие**. В приложении есть действие для вызова которого у злоумышленника есть причина.
- **Обработка сеансов на основе файлов cookie**. Выполнение действия включает в себя отправку одного или нескольких HTTP-запросов, и приложение полагается исключительно на файлы cookie сессии для идентификации пользователя, отправившего запрос.
- **Нет непредсказуемых параметров запроса**. Запросы, выполняющие действие, не содержат параметров, значения которых злоумышленник не может определить или угадать.

Как работает CSRF

Предположим, приложение содержит функцию, позволяющую пользователю изменить адрес электронной почты в своей учётной записи. Когда пользователь выполняет это действие, он делает HTTP-запрос, подобный этому:

```
POST /email/change HTTP/1.1
Host: vulnerable-website.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Cookie: session=yvthwsztyeQkAPzeQ5gHgTvlyxHfsAfE

email=wiener@normal-user.com
```

Как работает CSRF

С учётом этих условий злоумышленник может создать веб-страницу содержащую следующий HTML-код:

```
<html>
    <body>
        <form action="https://vulnerable-website.com/email/change" method="POST">
            <input type="hidden" name="email" value="pwned@evil-user.net" />
        </form>
        <script>
            document.forms[0].submit();
        </script>
    </body>
</html>
```

Как работает CSRF

Если пользователь-жертва посещает веб-страницу злоумышленника, происходит следующее:

- Страница злоумышленника инициирует HTTP-запрос к уязвимому веб-сайту.
- Если пользователь авторизовался на уязвимом веб-сайте, его браузер автоматически включит файл сессии cookie в запрос (при условии, что не используются SameSite cookie
- Уязвимый веб сайт обработает запрос обычным образом, расценит его как выполненный пользователем-жертвой и изменит его адрес электронной почты.

Как доставляют CSRF эксплойт

Как правило, злоумышленник размещает вредоносный HTML-код на контролируемом им веб-сайте, а затем побуждает жертву посетить этот веб-сайт.

В предыдущем примере, если запрос на изменение электронной почты можно выполнить с помощью метода GET, то автономная атака будет выглядеть так:

```
<img src="https://vulnerable-website.com/email/change?email=pwned@evil-user.net">
```

Последствия CSRF-атак

- Неавторизованные действия - изменение паролей, перевод средств, изменение настроек учетной записи.
- Ущерб репутации - психологический эффект для пользователей и компании.

Основные способы защиты от CSRF атак

Основные способы защиты от CSRF атак:

1. **Использование токенов CSRF:** Веб-приложение генерирует уникальный токен, который должен быть включен в каждый HTTP-запрос.

2. **Проверка referer:** Проверка, что запрос отправлен с правильного домена.

3. **Внедрение механизмов авторизации:** Использование HTTP-аутентификации для защиты от CSRF-атак.

Anti-CSRF токены

Токены – это способ защиты со стороны сервера. Сервер генерирует случайный уникальный токен для браузера пользователя и проверяет его для каждого запроса.

Токен должен удовлетворять следующим условиям:

- быть уникальным в пределах каждой операции;
- использоваться один раз;
- иметь размер устойчивый к подбору;
- генерироваться криптографически стойким генератором псевдослучайных чисел;
- иметь ограниченное время жизни.

Использование флага Same-Site в cookies

Этот флаг помечает cookies для определенного домена.

Таким образом проверяется источник запроса, и его не получится выполнить с мошеннического сайта.

Этот флаг поддерживает большинство браузеров. Его стоит использовать как часть общей стратегии защиты от CSRF атак.

Заключение

Ошибки путаницы привилегий, такие как CSRF, являются серьезной угрозой для безопасности веб-приложений. Важно уделять внимание защите от этой атаки и использовать соответствующие меры безопасности для предотвращения ее возникновения.