УДК 004.4 **Информационные технологии**

Убеев Валерий Геннадьевич, *специалист, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, Россия, г. Санкт-Петербург*

КАК ИСПРАВИТЬ МЕЖСАЙТОВУЮ УЯЗВИМОСТЬ WEB-SOCKET

Аннотация: В статье описывается что такое веб-сокет и как он работает. Рассмотрим межсайтовую уязвимость веб-сокета, как определить межсайтовую уязвимость и как подходить к вопросу защиты.

Ключевые слова: Веб-сокет, уязвимость, Handshake, браузер, перехват.

Annotation: The article describes what a web socket is and how it works. Let's look at a web socket cross-site vulnerability, how to identify a cross-site vulnerability, and how to approach the issue of protection.

Keywords: Web socket, vulnerability, Handshake, browser, interception.

Что такое веб-сокет?

Приложения реального времени работают в мгновенных временных рамках; воспринимая, анализируя и воздействуя на потоковые данные по мере их поступления. С приложениями реального времени вам нужна информация с ваших серверов, как только она станет доступной.

WebSocket — это сетевой протокол, основанный на tcp протоколе, предназначенный для использования более современного соединения для получения и отправки текущих сообщений в реальном времени по мере необходимости.

Сегодня все больше и больше конечных пользователей и бизнес-лидеров требуют взаимодействия в режиме реального времени, чтобы оставаться конкурентоспособными.

До появления WebSocket существовала сеть «реального времени», но ее было трудно реализовать, поскольку сеть построена на парадигме протокола HTTP. HTTP не имеет состояния: нет связи между двумя запросами, последовательно выполняемыми по одному и тому же соединению. Протокол без сохранения состояния не требует, чтобы сервер сохранял информацию или статус о каждом пользователе в течение нескольких запросов.

Давайте посмотрим, как это работает, как злоумышленники могут их использовать и как защититься от этих методов.

Как работает веб-сокет?

Сокеты соединяют две службы, образованные сочетанием IP и порта. Они действуют как уникальный программный способ получения и отправки данных и являются ключевыми для сетевых приложений во всем мире.

Соединения WebSocket начинаются через HTTP и часто имеют длительный срок службы. Сообщения могут быть отправлены в любом направлении в любое время и не являются транзакционными по своей природе [5].

Handshake через веб-сокет

Чтобы установить соединение, браузер и сервер выполняют handshake WebSocket через HTTP. Жизненный цикл взаимодействия WebSocket между клиентом и сервером включает следующие этапы:

- Клиент инициирует соединение с сервером, отправляя HTTP(S) запрос handshake WebSocket.
- Сервер отвечает на handshake, ответом который состоит из ответа HTTP с кодом состояния 101 (и ряда конкретных заголовков HTTP), чтобы принять handshake.
- Затем сервер может отправлять сообщения напрямую клиенту без инициирования запросов клиентом.
- Клиент может сразу (или через некоторое время) закрыть соединение WebSocket [6].

Запрос браузера

GET /message HTTP /1.1

Host: message - websocket.com

Sec – WebSocket - Version: 13

Sec - WebSocket-Key: wDqustmeNBJhikhdL6PW7w==

Connection: keep-alive, Upgrade

Cookie: session= <sessionID>

Upgrade: websocket

Ответ сервера

HTTP/1.1 101 Switching Protocols

Connection: Upgrade

Upgrade: websocket

Sec-WebSocket-Accept: 0FFP+2mnFIn/h+4BP36k9urzYGk=

Заголовки handshake WebSocket

Заголовки обеспечивают пояснения о соединении. Заголовки Connection и Upgrade в запросе и ответе указывают, что это handshake WebSocket. Заголовок запроса Sec-WebSocket-Version указывает версию протокола WebSocket, которую желает использовать клиент. Обычно это 13.

Заголовок запроса Sec-WebSocket-Key содержит случайное значение в кодировке Base64, которое должно генерироваться случайным образом в каждом запросе handshake. Заголовок ответа Sec-WebSocket-Accept содержит хэш значения, переданного в заголовке запроса Sec-WebSocket-Key [4].

Межсайтовая уязвимость WebSocket

Одним из ключевых моментов, на который следует обратить внимание, является межсайтовая уязвимость WebSocket (перехват WebSocket из разных источников). Он заключается в подделке межсайтовых запросов (CSRF) при рукопожатии WebSocket. Сам протокол WebSocket не предписывает какой-либо конкретный способ, которым серверы могут аутентифицировать клиентов во время рукопожатия WebSocket через HTTP.

Например, в приведенном ниже примере запрос handshake WebSocket полагается исключительно на файлы cookie HTTP для обработки сеанса и не

содержит никаких токенов в параметрах запроса. Этот запрос, вероятно, находится под угрозой CSRF.

GET /message HTTP /1.1

Host: message-websocket.com

Sec-WebSocket-Version: 13

Sec-WebSocket-Key: wDqustmeNBJhikhdL6PW7w==

Connection: keep-alive, Upgrade

Cookie: session=KOsEJNufwl4Rd9BDNrVmwvBF9rEjieE2

Upgrade: websocket

Следовательно, первым шагом к выполнению атаки является просмотр рукопожатий WebSocket и определение того, имеют ли они защиту от CSRF [2].

Как определить межсайтовую уязвимость WebSocket

В уязвимом приложении для лайв-чата, созданном с помощью WebSocket, злоумышленник использует сервер эксплойтов для размещения полезной нагрузки HTML/JavaScript. При этом используется запрос handshake WebSocket, поскольку единственный токен сеанса передается в файле cookie. Таким образом злоумышленник может извлечь историю чата жертвы [3].

Шаг 1: Получите доступ к функции лайв-чата и отправьте первое текстовое сообщение.

Шаг 2: Захватите запрос handshake WebSocket с помощью инструмента прокси-сервера HTTP и наблюдайте за запросом. Единственный присутствующий токен сеанса находится в заголовке файла cookie.

Примечание: Заголовок Sec-WebSocket-Key содержит случайное значение для предотвращения появления ошибок.

Шаг 3: Разместите полезную нагрузку HTML с URL-адресом handshake WebSocket, чтобы эксфильтровать историю чата жертвы с помощью взаимодействия с внешней службой.

Шаг 4. Как только жертва перейдут по URL-адресу через фишинговое электронное письмо или случайную рекламу на веб-странице, субъект угрозы может получить историю чата жертвы и эксфильтровать ее через взаимодействие

с внешней службой. Тело запроса содержит полное содержимое сообщения чата в формате JSON [7].

Как исправить и предотвратить межсайтовый перехват WebSocket.

Чтобы защититься от злоумышленников, убедитесь, что все приложения подключаются к серверу WebSocket через безопасное соединение TLS. Вам также необходимо правильно настроить сервер для проверки и подтверждения содержимого заголовка «origin», полученного от клиента по HTTP в его запросе на инициацию handshake WebSocket.

Атрибуты того же сайта в файле cookie сеанса также могут быть полезны. Это будет указывать браузеру не отправлять файл cookie при любом запросе из другого источника [1].

И последнее, но не менее важное: используйте токены CSRF для каждого запроса, обрабатывающего WebSockets. Токены CSRF включают в себя непредсказуемое значение, которое однозначно привязано к сеансу пользователя на странице. Сгенерируйте токены CSRF на стороне сервера. Вы можете сделать это один раз за сеанс пользователя или для каждого запроса. Таким образом, вы можете воспользоваться преимуществами WebSocket, перекрыв путь злоумышленникам.

Библиографический список:

1. Волкогонов В. Н., Гельфанд А. М., Деревянко В. С. Актуальность автоматизированных систем управления //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019). – 2019. – С. 262-266.

2. Гельфанд А. М. и др. Разработка модели распространения самомодифицирующегося кода в защищаемой информационной системе //Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. – 2018. – №. 8. – С. 91-97.

3. Орлов Г. А., Красов А. В., Гельфанд А. М. Применение Big Data при анализе больших данных в компьютерных сетях //Наукоемкие технологии в

космических исследованиях Земли. – 2020. – Т. 12. – №. 4. – С. 76-84.

4. Сахаров Д. В. и др. Использование математических методов прогнозирования для оценки нагрузки на вычислительную мощность IOT-сети //Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». – 2020. – №. 2. – С. 86-94.

5. Красов А. В. и др. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СРЕДСТВ ПРЕДОТВРАЩЕНИЙ ВТОРЖЕНИЙ И АНОМАЛИЙ СЕТЕВОЙ ИНФРАСТРУКТУРЫ.

6. Цветков А. Ю. АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ АТАК ТИПА ПЕРЕПОЛНЕНИЯ БУФЕРА НА ОПЕРАЦИОННЫЕ СИСТЕМЫ СЕМЕЙСТВА MICROSOFT //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019). – 2019. – С. 751-756.

7. Катасонов А. И., Цветков А. Ю. АНАЛИЗ МЕХАНИЗМОВ РАЗГРАНИЧЕНИЯ ДОСТУПА В СИСТЕМАХ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020). – 2020. – С. 563-568.