Authentication and extension bypass

High **jcoglan** published **GHSA-qpg4-4w7w-2mq5** on Apr 28, 2020

Package

faye (npm, rubygems)

Affected versions	Patched versions
< 1.0.4 || 1.1.0 - 1.1.2 || 1.2.0 - 1.2.4	1.0.4, 1.1.3, 1.2.5

Description

On 20 April 2020 it was reported to me that the potential for authentication bypass exists in Faye's extension system. This vulnerability has existed in the Node.js and Ruby versions of the server since version 0.5.0, when extensions were first introduced, in July 2010. It is patched in versions 1.0.4, 1.1.3 and 1.2.5, which we are releasing today.

The vulnerability allows any client to bypass checks put in place by server-side extensions, by appending extra segments to the message channel. For example, the Faye extension docs suggest that users implement access control for subscriptions by checking incoming messages for the `/meta/subscribe` channel, for example:

```
server.addExtension({
  incoming: function(message, callback) {
    if (message.channel === '/meta/subscribe') {
      if (message.ext.authToken !== 'my super secret password') {
        message.error = 'Invalid auth token';
      }
    }
    callback(message);
  }
});
```

A bug in the server's code for recognising the special `/meta/*` channels, which trigger connection and subscription events, means that a client can bypass this check by sending a message to `/meta/subscribe/x` rather than `/meta/subscribe`:

```
{
  "channel": "/meta/subscribe/x",
  "clientId": "3jrc6602npj4gyp6bn5ap2wqzjtb2q3",
  "subscription": "/foo"
}
```

This message will not be checked by the above extension, as it checks the message's channel is exactly equal to `/meta/subscribe`. But it will still be processed as a subscription request by the server, so the client becomes subscribed to the channel `/foo` without supplying the necessary credentials.

The vulnerability is caused by the way the Faye server recognises meta channels. It will treat a message to any channel that's a prefix-match for one of the special channels `/meta/handshake`, `/meta/connect`, `/meta/subscribe`, `/meta/unsubscribe` or `/meta/disconnect`, as though it were an exact match for that channel. So, a message to `/meta/subscribe/x` is still processed as a subscription request, for example.

An authentication bypass for subscription requests is the most serious effect of this but all other meta channels are susceptible to similar manipulation.

This parsing bug in the server is fixed in versions 1.0.4, 1.1.3 and 1.2.5. These should be drop-in replacements for prior versions and you should upgrade immediately if you are running any prior version.

If you are unable to install one of these versions, you can make your extensions catch all messages the server would process by checking the channel *begins* with the expected channel name, for example:

```
server.addExtension({
  incoming: function(message, callback) {
    if (message.channel.startsWith('/meta/subscribe')) {
      // authentication logic
    }
    callback(message);
  }
});
```

Severity

High

CVE ID

CVE-2020-11020

Weaknesses

No CWEs