

Product ▾


Solutions ▾

Resources ▾

Open Source ▾


Enterprise ▾


Pricing





Sign in


Sign up


 mttaggart / quasar Public


 Notifications


 Fork 3


 Star 24


 Code


 Issues


 Pull requests


 Actions


 Projects


 Security


 Insights

 main ▾







 Go to file








 Code


▾


 mttaggart Add image


dbebfda · 2 years ago

 4 Commits


 .gitignore	Initial commit	2 years ago
 LICENSE	Initial commit	2 years ago
 README.md	Add image	2 years ago
 index.js	Add source	2 years ago
 package.json	Add source	2 years ago
 quasar.png	Add image	2 years ago
 yarn.lock	Add source	2 years ago

 README

 MIT license



quASAR



quASAR: ASAR manipulation made easy

This project is a proof-of-concept for manipulating ASAR files for code injection in Electron apps.

This capability works across all platforms, and compiled binaries are available on the [releases](#) page.

Usage


```
quasar [options]
```


Options:


- i, --input **<inputFile>** asar file to mutate (default: "app.asar")
- c, --command **<command>** **command** to insert (default: "calc.exe")


About


quASAR: ASAR manipulation made easy


 Readme

 MIT license

 Activity


 24 stars

 2 watching

 3 forks

[Report repository](#)

Releases 1

 **v0.1.0** Latest
on Sep 7, 2022

Packages

No packages published

Languages

JavaScript

100.0%

Page 1 of 2

```
-w --write      write evil files directly to application (
-h, --help     display help for command
```

quasar requires a .asar file as a target. It can either be located elsewhere on the filesystem or, as is default, an app.asar file local to the current directory.

You will be presented with a list of injectable .js files in the archive. Select one by number, and the command provided by -c will be injected.

Without -w, the resulting app.asar and app.asar.unpacked will be created in a new evil directory within the current directory. However, if -w is provided, the ASAR files will be written back to the original path, and the original files will have .bak appended to their filenames.

