

Web part / Farm Solutions / SPFX Security Reviews

If you are still using farm solutions(WSP files), it's time to start looking at rewriting those into [SPFX](#) or getting the latest cloud version for SharePoint Online.

If you have extended your SharePoint environment , and you want to migrate your extensions to the SharePoint Add-in model to make your transition to SharePoint Online easier. See the [Transforming Farm Solutions](#)

also see the client-side [migration guidance](#) article for more information on replacing traditional methods such as Script Editor ,Jquery ,DataTables etc.

If you are on SharePoint Server 2016 or later this is the preferred and supported model.

See [supported extensibility platforms - overview](#)

How to do a security review

The MCSA code review tool was retired on 1st March 2022 [Microsoft Security Code Analysis – a tool that seamlessly empowers customers to enable security controls in your CI/CD pipeline -Blog Post](#)

Please refer to the [OWASP Source Code Analysis Tools](#) for alternative options in Azure DevOps. Consider implementing tools such as [HCL AppScan CodeSweep - GitHub Action](#) which supports JS (Vue, React, Node, Angular, JQuery) or [SolarLint](#).

SonarLint is a free and open source IDE extension that identifies and helps you fix quality and security issues as you code.

There is also a very helpful [blog](#) explaining more of the options.

For customers planning to migrate to GitHub, you can check out [GitHub Advanced Security](#).

Process to share with the security team / 3rd Party tools

- Share the build folder with the security team to scan or Code and scan as you go through your IDE.
- Review the output / warnings and errors.
- You should also do a peer review of your code for best practices.
- Once the review and revisions have passed run the build and then deploy the solution.

3rd Party - Rencore

Rencore has a paid tool named SPcaf which can also help with analysis of SharePoint solutions.

[Rencore_Website](#)

Update a SPFX project

To find the outdated packages in your client-side project, including SharePoint Framework and other packages your project depends on, run the following command in a console in the same directory as your project.

- Run to get a list of outdated packages

```
npm outdated
```

- install manually each outdated package

```
npm install mypackage@newversion --save  
gulp clean
```

- Delete the folder **node_modules**
- Re-install the latest module from packages.json

```
npm install
```

- Rebuild the solution using gulp

```
gulp build
```

- Update Yeoman Generator

```
npm outdated -g
```

- Update the SP Generator

```
npm install @microsoft/generator-sharepoint@latest -g
```

When you run the npm outdated command in a project that targets both SharePoint Online and SharePoint Server 2016, it shows you the latest versions of the SharePoint Framework packages. These versions, however, work only with SharePoint Online. If you update your solution to use these latest packages, it no longer works with SharePoint Server 2016.

Read how to [setup your development environment](#)

SP Version	SPFX version	Node version	NPM	TypeScript	React	Hyperlink
SharePoint Online	Latest	LTS v12, LTS v14	v5, v6	v3.9	v16.13.1	link
SharePoint Server 2019	v1.4.1	LTS v6, LTS v8	v3, v4	v2.4	v15	link
SharePoint Server 2016 SP2	v1.1.0	LTS v6	v3, v4	v2.4	v15	link

See official MS updated pages [here](#)

Deploy an app package file

- Create app catalog
- Upload package file
- Deploy solution

See this useful [tutorial](#) for more info.