Shivam Bathla

269 Followers

A06:2021-Vulnerable and Outdated Components

Let’s discuss about the #6 vulnerability OWASP Top 10 2021 list…

Shivam Bathla Just now · 4 min read

. . .

Introduction

When you design an application, I am sure you use some out-of-the-box libraries or packages to make your task easier, don’t you?

Consider a simple python code to send a GET request. Chances are that it would be using requests library or something similar. If you wish to work with images, you might use Pillow.

If you are into web development, you might be using Bootstrap, jQuery, VueJS, ReactJS, Angular, and other popular libraries and frameworks.

If the components you are using to build your applications become outdated or have a serious vulnerability, you would be impacted by that and so would your customers and application users.

And this is what we will talk about in this post.

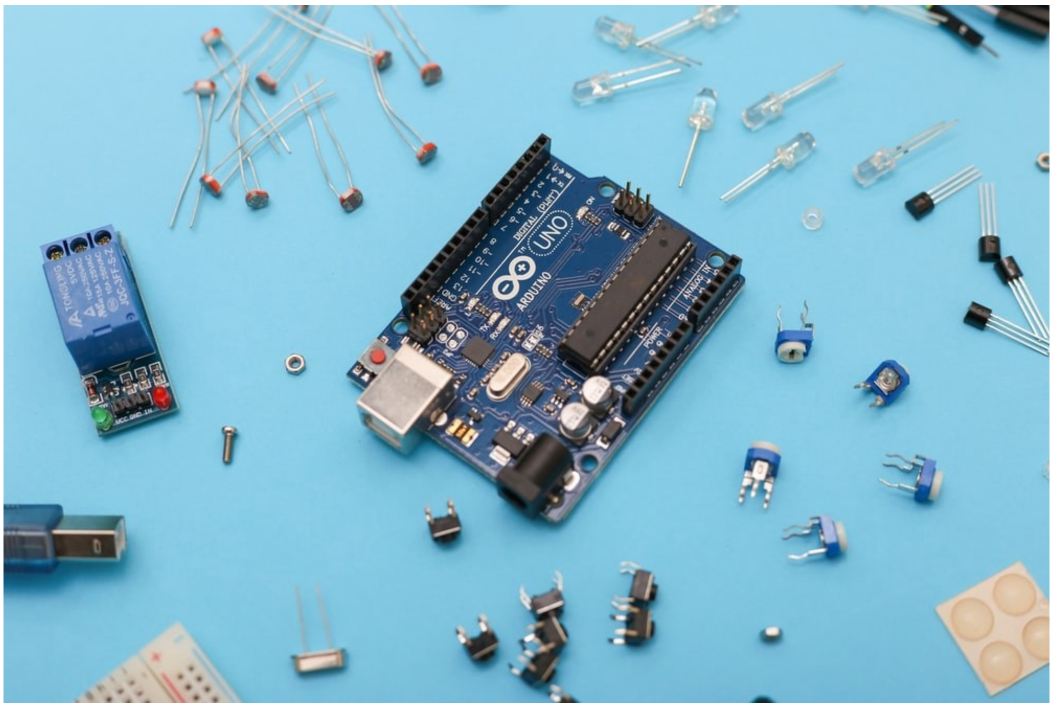
Photo by Sahand Babali on Unsplash

. . .

Vulnerable and Outdated Components

This issue was also there in the 2017 Top 10 list and has secured a better position: #6, while it was on #9 position in 2017 and was named **Using Components with Known Vulnerabilities**.

As you can already imagine from the discussion above, it's quite easy to have this issue in your applications. Why?

Because believe it or not, managing your dependencies is a great deal of work. It's not as simple as running the update command or downloading the updated dependencies & packages. But it much more than this: your apps might break with the latest changes, some features might get deprecated, functions might renamed, the dependencies might be abandoned, the fix might not work on your system without breaking some other dependencies, creating a havoc. You get the idea right…

This process of updating things and making sure that they remain the latest sounds simple but it's quite a lot of work and sometimes it's not that straight forward, unless you are willing to put in more time and update your code to get it to work well with the latest and greatest updates. At the very least, this is not always feasible and in the worst case, it would be your worst nightmare, if you understand what I am talking about!

Some easy things to look out for are:

- Vulnerable components (OS or software packages, applications, runtime environments) in the client and server-side code.
- Insecure software configuration
- Old/unpatched dependencies in the dependency chain of the components being used.

You get the idea right. These issues are a superset of Supply Chain Attacks and thus these must be taken quite seriously and the newer position (#6) must already be indicative of that.

. . .

The Fix

Some of the measures you can take to get rid of such issues are:

1. Maintain an inventory of components you are using and ensure that they are kept up-to-date.
2. Remove unused dependencies and components to reduce the attack surface and your liabilities (yes, code is indeed a great liability!)
3. Install the components via trusted channels and make sure to validate their integrity. Also, it's better to use signed packages (if available).
4. Be on the lookout for any security patches for the components you are relying on. If the packages you use are not maintained, then either make sure you apply patches yourself or use some alternate component that is well maintained and has a big user-base and supporting community. If possible, then this should actually be done right from the start — choose your dependencies and components wisely!

These are some of the practical ways you can address and hopefully mitigate this issue. You might have already noticed that these are preventive measures and some measures only take place after a vulnerability has been discovered. So the best you can do is stay up-to-date and follow good practices list above.

. . .

Playground

Here are some of the labs that you can try your hands-on and get some experience on exploiting these issues:

1. **OWASP Top 10: Vulnerable Components:** https://attackdefense.com/listing?labtype=webapp-owasp-top10&subtype=webapp-owasp-top10-vuln-components
2. Export Injection Flaws (using vulnerable version of a PDF generation library)
3. Vulnerable SVG converter library

Even some of the CVE-based labs we have would fall under this category because the components of the applications: wordpress plugins, CMS, or extensions would be included. So you can give those a try as well!

. . .

Closing Thoughts

Before I conclude this post, I would just say that this list by OWASP is still a draft and therefore don't just follow it blindly, let it settle for a while and let things be confirmed. But on the flip side, I think it's quite good list and things *might* not change much. That's why I am trying to cover this list this early :D

I hope you enjoyed this post and I will see you next time while covering the other vulnerabilities from the new OWASP Top 10 list.

Happy Hacking y’all!

Vulnerability Penetration Testing Owasp Top 10 Cybersecurity Infosec