# Introduction to Scanning Open Source Components for Vulnerabilities

In this module, you'll scan open-source components for security vulnerabilities and assess their license ratings.

It's important to have an inventory of the open-source components that your project uses. It's important to understand which libraries have vulnerability issues, when those vulnerabilities were addressed, and what versions you can use. Based on this information, you might even choose to use a different library or write your own.

You also need to understand what licenses these libraries use. Some licenses require you to make public any code that uses that library if you've made changes to the library's code. This requirement is problematic when your source code is not open source as well. You'll need to check with your own legal team to determine which licenses you can use.

In this module, you'll use WhiteSource Bolt, by WhiteSource, to scan open-source components for known vulnerabilities and license ratings.

In this module, you will:

* Learn which tools you can use to inspect open-source software packages for security and license ratings
* Access package and license ratings for open-source components by using WhiteSource Bolt
* Analyze the results of the scan so you can take the appropriate action

## Prerequisites

The modules in this learning path form a progression.

To follow the progression from the beginning, be sure to first complete the Evolve your DevOps practices  learning path.

We also recommend you start at the beginning of this learning path, Build applications with Azure DevOps .

If you want to go through just this module, you need to set up a development environment on your Windows, macOS, or Linux system. You need:

* An Azure DevOps organization
* A GitHub  account
* Visual Studio Code
* .NET Core 3.1 SDK
* Git

You can get started with Microsoft Azure DevOps for free.

This environment lets you complete the exercises in this and future modules. You can also use it to apply your new skills to your own projects.

**Note**

Keep in mind that you can use Azure DevOps to build and deploy almost any kind of application written in any language. In this module, you'll be working with a .NET Core application written in C#.

You don't need to be an expert in .NET or C# to complete this module. You can apply the patterns you learn here to your own projects that use your favorite programming languages and frameworks.

## Meet the team

You met the Space Game web team at Tailspin Toys in previous modules. As a refresher, here's who you'll work with in this module.



Andy is the development lead.



Amita is in QA.



Tim is in operations.



Mara just joined as a developer and reports to Andy.

Mara has prior experience with DevOps and is helping the team adopt a more automated process by using Azure DevOps.

# Choose a tool for scanning open-source dependencies

* 6 minutes

In this section, you'll examine a few tools that scan for known vulnerabilities and for license ratings in the open-source components that your application uses.

Let's check in with the team at their weekly meeting and see how things are progressing.

## Weekly meeting

**Andy:** Tim, before we start, how did your meeting go with the security consultant?

**Tim:** I was nervous going into it, but it went really well considering the topic. What impressed him most was how easily we were able to produce such a thorough report. Although our source code is far from clean, we can identify potential security holes and prioritize the order in which we address them. Plus, we can identify issues before the code goes to QA or production.

**Mara:** That's great! So what's the plan?

**Tim:** That's what I wanted to talk about. One other issue came up, and that's around how we use open source. The security consultant found that other teams were using software that was licensed under GPL 3.0.

**Andy:** I've heard of GPL. That stands for General Public License. What's the issue?

**Tim:** Well, it turns out that if you modify certain open-source libraries, even if it's to fix a bug or make a minor enhancement, you might be required to make those changes publicly available. Those changes could include intellectual property we want to protect. There's this big conversation happening with our legal team to identify what's next. I'm worried that this might affect our team as well.

**Amita:** I wasn't aware that this project used much open-source software.

**Andy:** It's indeed surprising how much open-source we use. We use frameworks such as Bootstrap and jQuery to make it easier to work with JavaScript and CSS. In fact, some components include other components that we haven't checked for security vulnerabilities, or haven't checked how they're licensed. We'd better make sure of where we stand before we move forward.

**Amita:** Licenses can be hard to read and interpret. Do we need a lawyer to help?

**Tim:** The legal team wants to meet with us next week. But first they want us to investigate which open-source components we're using and how those components are licensed. Andy, Mara, you're the most familiar with how the code is structured. Would you mind doing some investigation for the team?

**Andy:** Sure, we got this. We'll take a look after this meeting and report back with what we find.

## How can I access security and license ratings in open-source software?

Andy and Mara return to their desks.

**Andy:** I was at a user conference a few months ago where they talked about licenses and security in open-source software. I can bring up the slides from the presentation. Maybe we can identify a tool or process we can use.

Andy and Mara review the slides and investigate further. They find a few options.

### Black Duck

Black Duck  by Synopsys scans your open-source dependencies against their own database. They keep that database up-to-date with the latest vulnerabilities and give you timely information on fixes, workarounds, and on the exploits. They cover over 80 programming languages, and get their information from many sources.

### GitLab

GitLab  is a CI/CD pipeline tool that enables you to scan the dependencies in code that comes from GitLab repositories. It supports JavaScript, Ruby, Python, PHP, and Java.

Their Azure Pipelines extension  enables you to add a GitLab repository as a source.

### Sonatype Nexus

Sonatype Nexus  scans your dependencies for known vulnerabilities, licenses, and old or unsupported components.

Their Azure Pipelines extension  enables you to upload your artifacts to Sonatype Nexus for scanning.

### Veracode

Veracode  is a code scanning tool that integrates with Azure Pipelines to help you find vulnerabilities early in the development process.

Their Azure Pipelines extension  stops the build process if there's a severe security risk and gives you guidance on finding and fixing the vulnerability.

### WhiteSource Bolt

WhiteSource Bolt  is a tool for scanning open-source dependencies for vulnerabilities and licensing. It supports more than 200 programming languages and gives guidance on fixing the vulnerabilities. There are reports for vulnerabilities, licensing, and inventory.

Their Azure Pipelines extension  allows for extensive policy configuration as well as viewing the results from the Azure Pipelines portal.

## Which option should I choose?

Andy and Mara are both in new territory. All the options have something to offer. They decide to start with WhiteSource Bolt, which looks like a good, general-purpose tool. When they have time, they might investigate some of the other tools they've learned about.

Which tool you pick will always come down to your requirements and which option provides the best insights into your use of open source.