**Installing Sbomgen binary**

1. <https://docs.aws.amazon.com/inspector/latest/user/sbom-generator.html#install-sbomgen> Download the latest Sbomgen zip file from the correct URL for your architecture:

Linux AMD64: <https://amazon-inspector-sbomgen.s3.amazonaws.com/latest/linux/amd64/inspector-sbomgen.zip>

Linux ARM64: <https://amazon-inspector-sbomgen.s3.amazonaws.com/latest/linux/arm64/inspector-sbomgen.zip>

Alternatively, you can download [previous versions of the Amazon Inspector SBOM Generator zip file](https://docs.aws.amazon.com/inspector/latest/user/sbom-generator-versions.html).

1. Unzip the download using the following command:

unzip inspector-sbomgen.zip

1. Check for the following files in the extracted directory:
   * inspector-sbomgen – This is the tool you will execute to generate SBOMs.
   * README.txt – This is the documentation for using Sbomgen.
   * LICENSE.txt – This file contains the software license for Sbomgen.
   * licenses – This folder contains license info for third party packages used by Sbomgen.
   * checksums.txt – This file provides hashes of the Sbomgen tool.
   * sbom.json – This is a CycloneDX SBOM for the Sbomgen tool.
   * WhatsNew.txt – This file contains a summarized change log, so you can view major changes and improvements between Sbomgen versions quickly.
2. (Optional) Verify the authenticity and integrity of the tool using the following command:

sha256sum < inspector-sbomgen

* + Compare the results with the contents of the checksums.txt file.

1. Grant executable permissions to the tool using the following command:

chmod +x inspector-sbomgen

cp inspector-sbomgen /usr/local/bin

1. Verify that Sbomgen is successfully installed using the following command:

inspector-sbomgen --version

You should see the output similar to the following:

Version: 1.X.X

**Using Sbomgen**

inspector-sbomgen list-examples

inspector-sbomgen container --image alpine:latest -o sbom\_path.json

1. Create a directory to keep the code:

mkdir python-django; cd python-django

1. Create the file vulnerable\_django\_app.py. The code itself doesn’t matter, but the dependencies:

import django  
from django.conf import settings  
from django.conf.urls import url  
from django.http import HttpResponse  
from django.core.handlers.wsgi import WSGIHandler  
  
# Minimal Django settings  
settings.configure(  
 DEBUG=True,  
 SECRET\_KEY='a-very-bad-secret-key',  
 ROOT\_URLCONF=\_\_name\_\_,  
 ALLOWED\_HOSTS=['\*'], # Allow all hosts  
)  
def index(request):  
 return HttpResponse("Hello, Django!")  
urlpatterns = [  
 url(r'^$', index),  
]  
# Create the WSGI application  
application = WSGIHandler()  
if \_\_name\_\_ == "\_\_main\_\_":  
 from django.core.management import execute\_from\_command\_line  
 execute\_from\_command\_line(['manage.py', 'runserver', '0.0.0.0:8000'])

1. Create a requirements.txt file:

Django==1.11.29

1. Create a virtual env to install the needed requirements and activate it:

python3 -m venv my-python-env && source my-python-env/bin/activate

1. Install the requirements:

pip3 install -r requirements.txt

1. Now, we can generate the SBOM of the directory:

inspector-sbomgen directory --path python-django/ -o ~/SBOM-python-django.json