Deployment Guide

Installing SonaType IQ Server and integrating with Fortify SSC & SCA

Fortify SSC 20.x

Author: Vikas Johari
Date: 15 February 2021
Document Version: v0.1

MICRO FOCUS

Contents

Introduction

This document is written to guide Pre-Sales and Partners to install SonaType Nexus IQ Server and integrating with Fortify 20.2.0 SSC and SCA. Best way to install SonaType Nexus IQ Server using the Docker image, I have used the same.

This document is not written to install SonaType Nexus IQ Server in a Production Environment. However, this document can be used to setup SonaType Nexus IQ Server in a controlled environment like Lab or PoC or CoE Environment.

The Hardware and Software requirements are given in the link –

https://help.sonatype.com/iqserver/product-information/system-requirements

https://www.microfocus.com/documentation/fortify-software-security-center/2020/Fortify_Sys_Reqs_20.2.0/index.htm#SSC/SSC_Reqs.htm?TocPath=Fortify%2520Software%2520Security%2520Center%2520Server%2520Requirements%257C_____0

https://www.microfocus.com/documentation/fortify-static-code-analyzer-and-tools/2020/Fortify_Sys_Reqs_20.2.0/index.htm#SCA/SCA_Reqs.htm?TocPath=Fortify%2520Static%2520Code%2520Analyzer%2520Requirements%257C_____0

Detailed SSC 20.2.0 User Guide is given in https://www.microfocus.com/documentation/fortify-software-security-center/2020/SSC_Help_20.2.0/index.htm

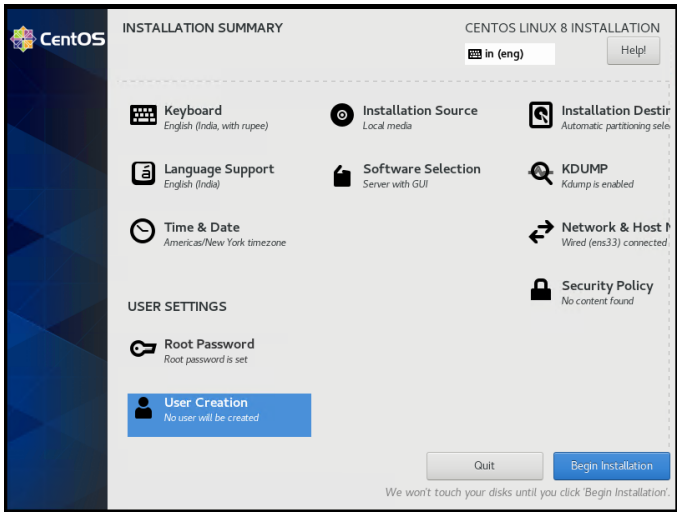
Detailed SCA 20.2.0 User Guide is given in https://www.microfocus.com/documentation/fortify-static-code-analyzer-and-tools/2020/SCA_Help_20.2.0/index.htm

I have used a two VMs with the below hardware configuration –

SonaType Nexus IQ Server:
- CPU: 2 vCPU
- RAM: 8 GB RAM
- Disk: 60 GB Thin Provisioned
- CentOS 8: Download link http://isoredirect.centos.org/centos/8/isos/x86_64/
- Internet Connection

Fortify SSC & SCA Server:
- CPU: 4 vCPU
- RAM: 16 GB RAM
- Disk: 100 GB Thin Provisioned
- Windows 2019 Server with MS SQL 2019 Server
- Internet Connection

Installing CentOS 8



Install and Configure CentOS 8 64 bit. Install all the patches.

Download and install Oracle JDK 11 i.e. jdk-11.0.10_linux-x64_bin.rpm file using –

```
# rpm -ivh jdk-11.0.10_linux-x64_bin.rpm
```

Verify the installation –

```
# java -version
java version "11.0.10" 2021-01-19 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.10+8-LTS-162)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.10+8-LTS-162, mixed mode)
# javac -version
javac 11.0.10
```

Installing and Configuring Docker on CentOS 8

Add Docker CE Repo

```
# dnf config-manager --add-
repo=https://download.docker.com/linux/centos/docker-ce.repo
```

Removing Conflicting Packages

```
# yum erase podman buildah -y
```

Installing rpm-build which is required for SCA. Just in case we have to install SCA on this server.

```
# yum install rpm-build -y
```

Install Containerd.IO
Run the below command as root –

```
# dnf install
https://download.docker.com/linux/centos/7/x86_64/stable/Packages/c
ontainerd.io-1.4.3-3.1.el7.x86_64.rpm -y
```

Its time to Install Docker

```
# dnf install docker-ce -y
```

Configure Docker Services

```
# systemctl enable docker
# systemctl start docker
# systemctl status docker
```

Verify docker is install and running

Installing Portainer (Optional)

Portainer is easy web based Container management tool.
Ref: https://documentation.portainer.io/v2.0/deploy/ceinstalldocker/
Run the below commands as root –

```
# docker volume create portainer_data
```

```
# docker run -d -p 8000:8000 -p 9000:9000 --name=portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce
```

Verify portainer is running in docker –

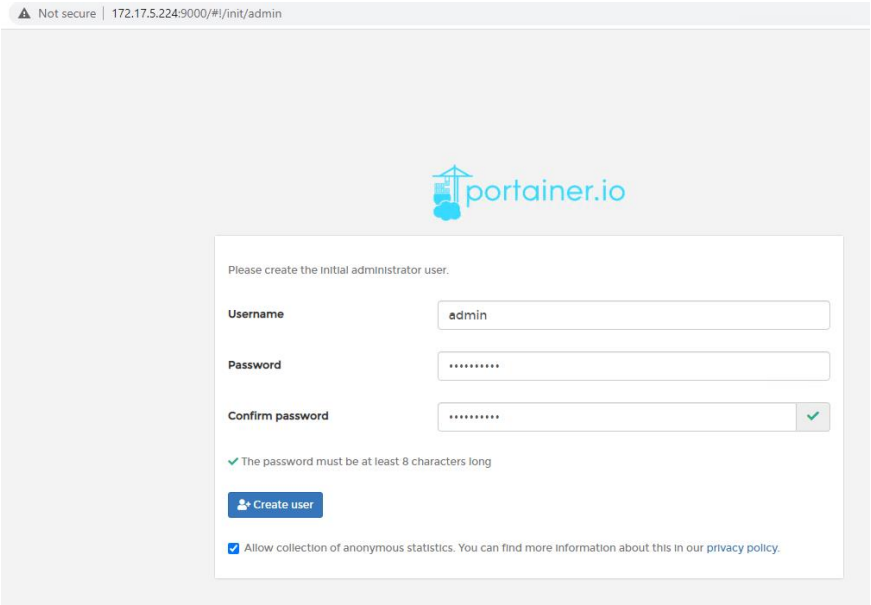
```
# docker ps
```

```
[root@iq ~]# docker ps
CONTAINER ID   IMAGE                    COMMAND        CREATED              STATUS              PORTS                                          NAMES
e557b7410534   portainer/portainer-ce   "/portainer"   About a minute ago   Up About a minute   0.0.0.0:8000->8000/tcp, 0.0.0.0:9000->9000/tcp   portainer
```

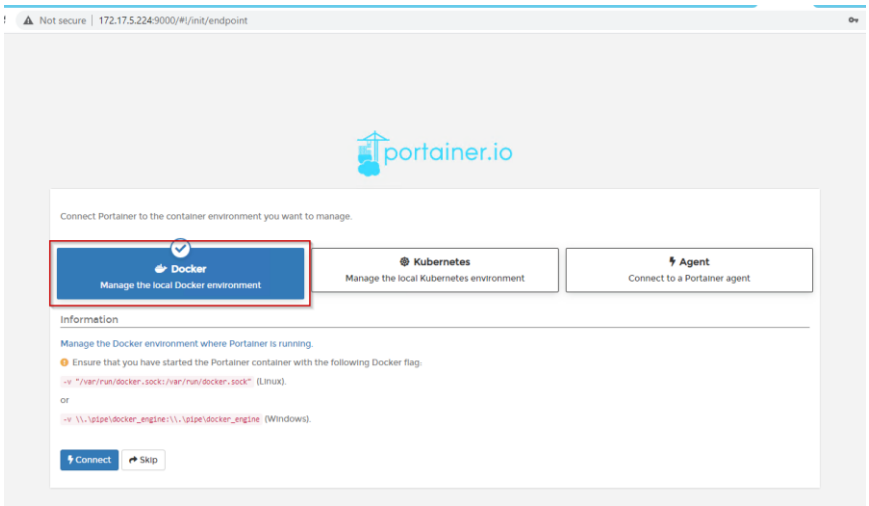
```
# netstat -anp | grep :9000
```

```
[root@iq ~]# netstat -anp | grep :9000
tcp        0      0 0.0.0.0:9000            0.0.0.0:*               LISTEN      3690/docker-proxy
```

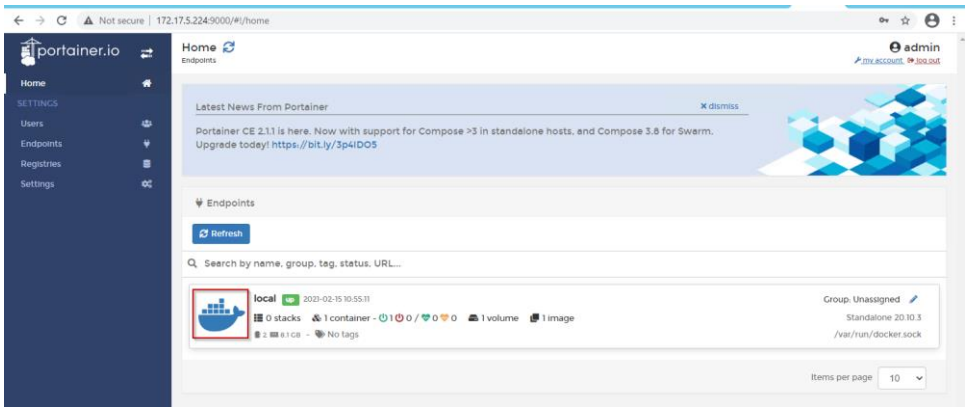
Use Browser to connect on port 9000



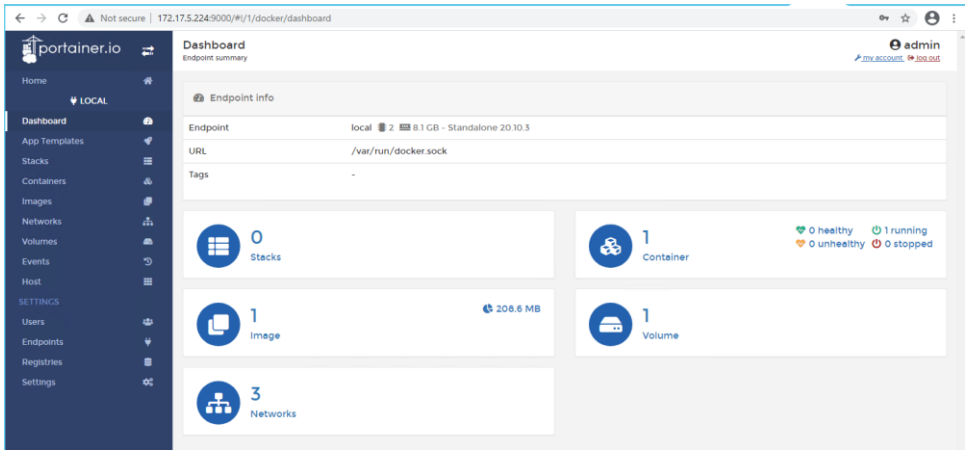
Enter a new password for admin user. Click "Create user" button.



Click on "Docker" option. Then click Connect button.



Click on Docker icon.



Looks good.

Installing SonaType IQ Server

Docker version of IQ Server is available in https://hub.docker.com/r/sonatype/nexus-iq-server
Run the below command to pull SonaType Nexus IQ Server.

```
# docker pull sonatype/nexus-iq-server
```

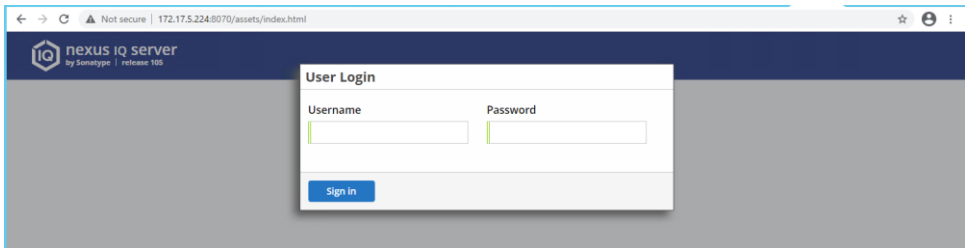
Deploy the image.

```
# docker run -d -p 8070:8070 -p 8071:8071 --name nexus-iq-server
sonatype/nexus-iq-server
```

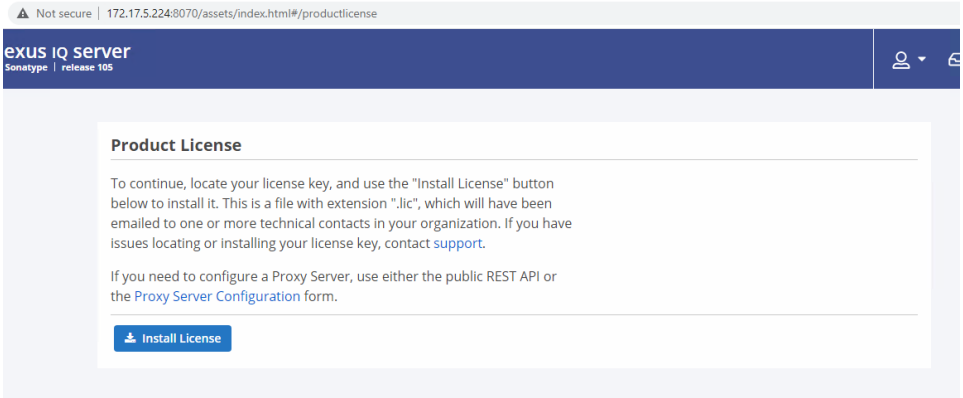
Verify using docker ps command –

```
[root@iq ~]# docker ps
CONTAINER ID   IMAGE                      COMMAND            CREATED              STATUS                        PORTS                                                                    NAMES
1a0bdef8bd04   sonatype/nexus-iq-server   "sh ./start.sh"    About a minute ago   Up About a minute (healthy)   8080/tcp, 8443/tcp, 0.0.0.0:8070-8071->8070-8071/tcp, 8778/tcp   nexus-iq-server
e557b7410534   portainer/portainer-ce     "/portainer"       13 minutes ago       Up 13 minutes                 0.0.0.0:8000->8000/tcp, 0.0.0.0:9000->9000/tcp                   portainer
[root@iq ~]#
```

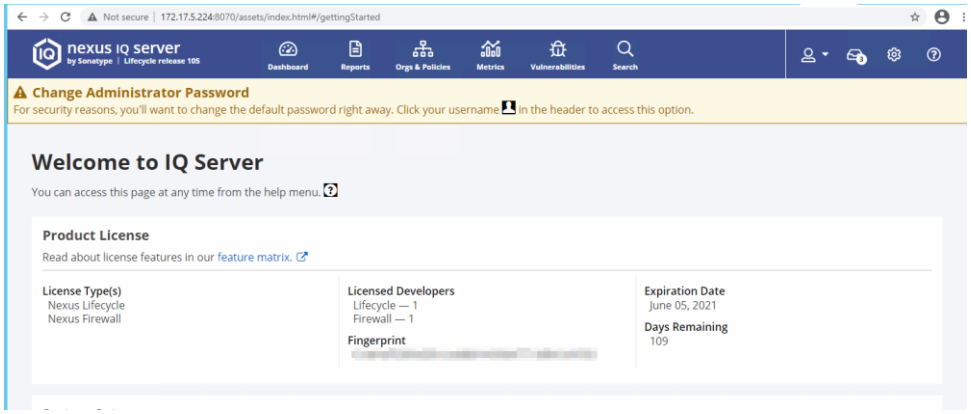
Now connect to port 8070 using browser. SonaType IQ Server runs on port 8070.



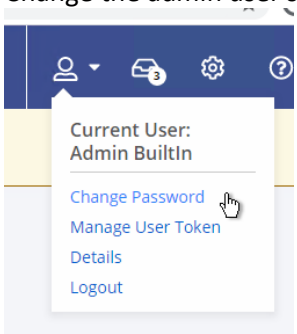
Enter the credentials, username **admin** and password **admin123**. Click Sign In.



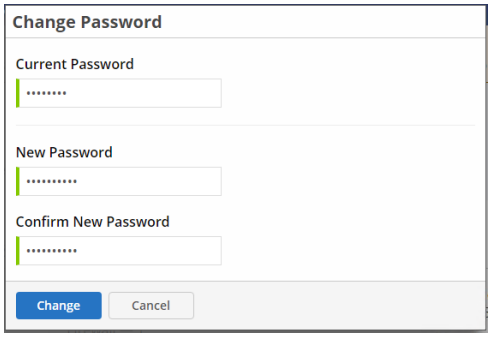
Click Install License button. Select the upload the license file.



Change the admin user's password.



Click on Change Password.



Enter the old password and new password then click on Change button.

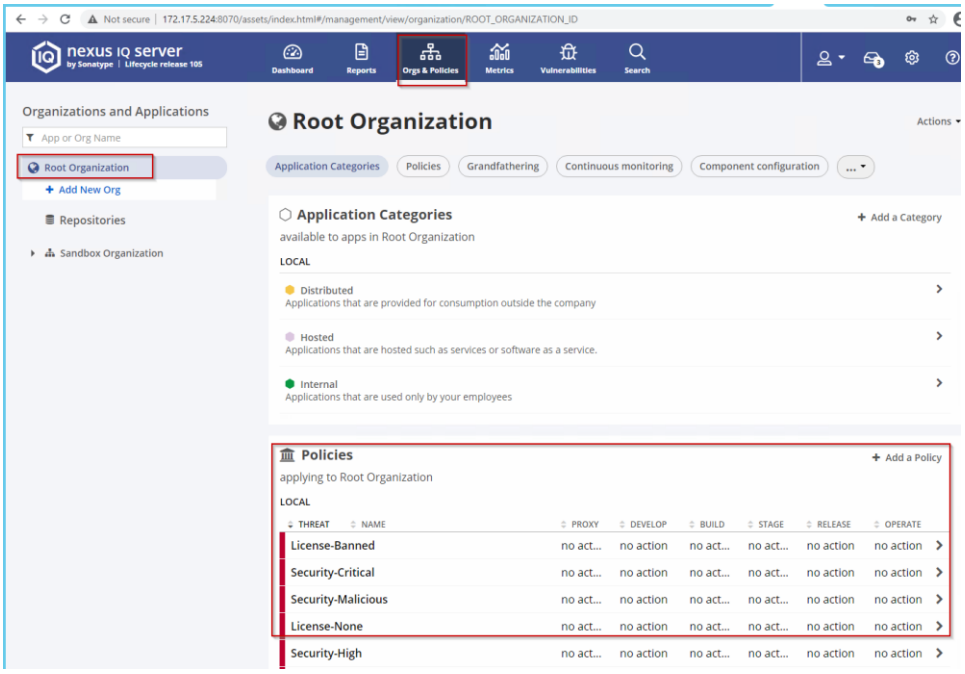
Reboot the Server.

Verify the Nexus-IQ-Server is running, if not then start the service using –

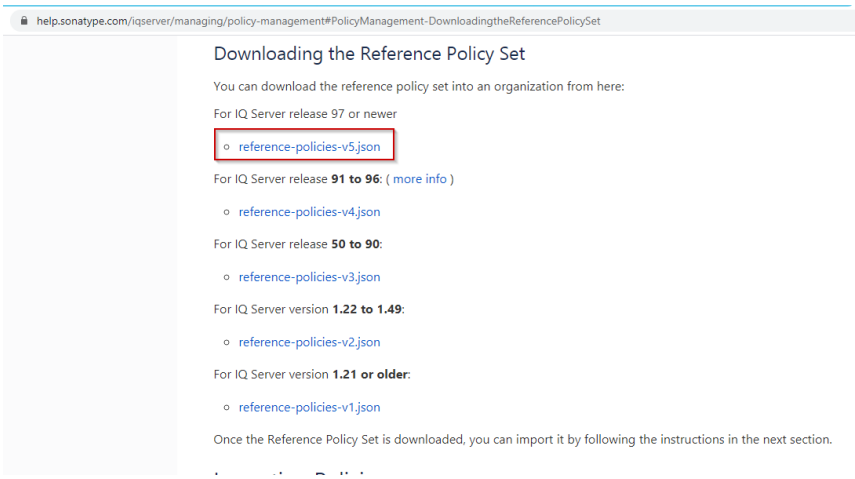
```
# docker start nexus-iq-server
```

Downloading the default policies

Click on "Orgs & Policies" -> "Root Organization", if you don't see default policies on the right panel. Then you have to download and install default policies.



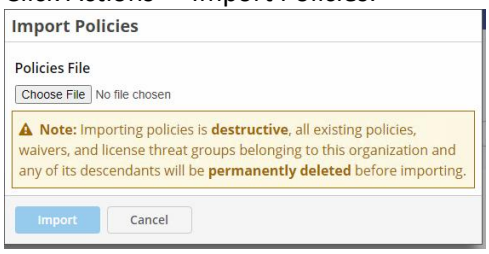
Open the URL https://help.sonatype.com/iqserver/managing/policy-management#PolicyManagement-DownloadingtheReferencePolicySet using Browser.



Download the latest json file i.e. "reference-policies-v5.json".

Make sure your IQ server can connect to internet, configure proxies if needed.

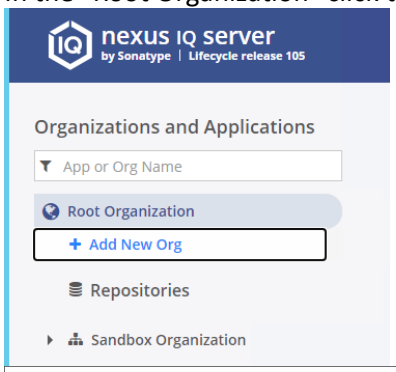
Click Actions -> Import Policies.

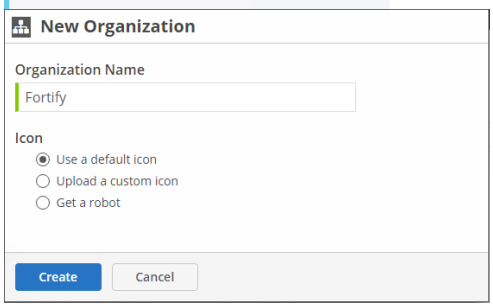


Select the downloaded reference-policies-v5.json file and click Import.

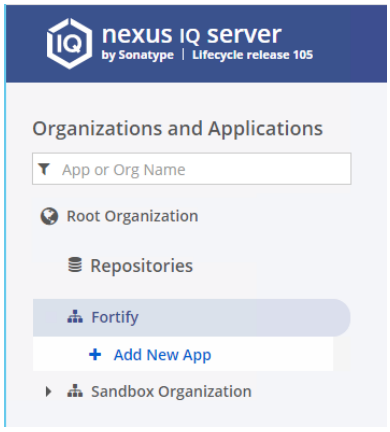
Testing the SonaType IQ Server with a Sample Code

In the "Root Organization" click to "Add New Org".

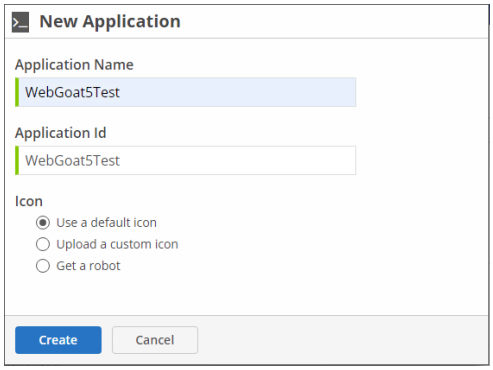




Give a name to your organization i.e. Fortify, click Create.

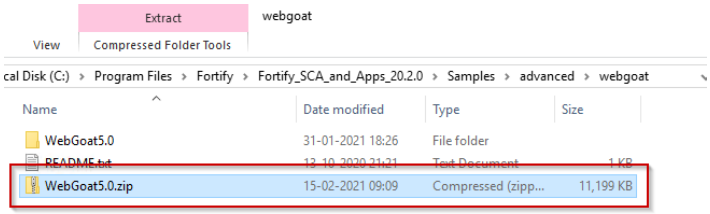


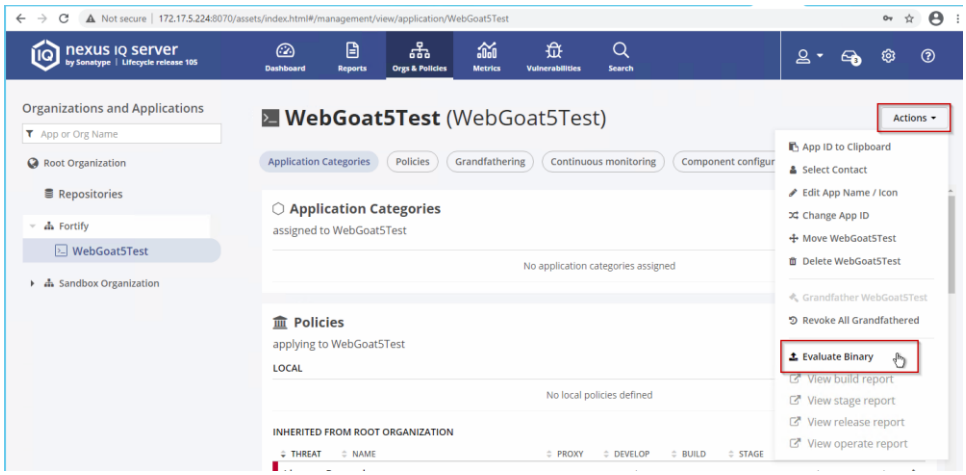
Click "Add New App".



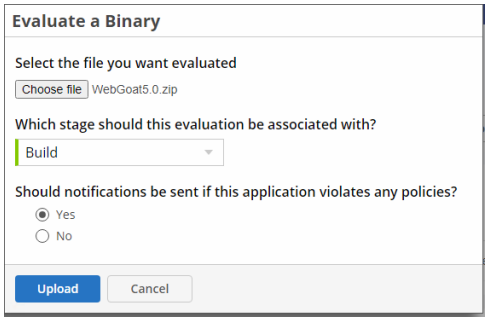
Create a Test Application name "WebGoat5Test" and its application ID will be "WebGoat5Test".

Open the folder WebGoat 5.0 folder in SCA machine and zip the entire WebGoat 5.0 source code as a Zip file.

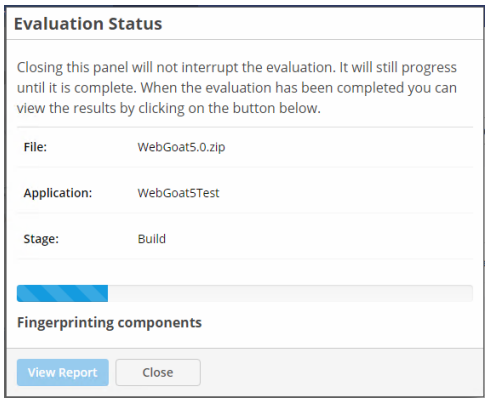




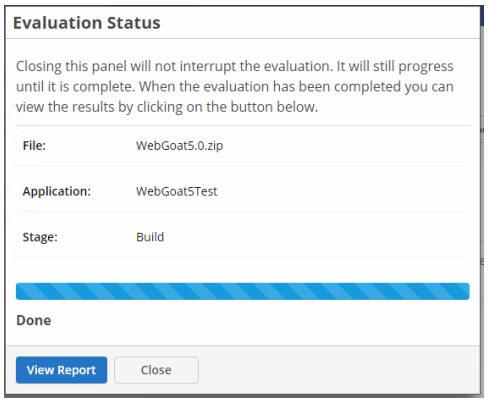
In IQ Server's WebGoat5Test application, click Actions -> Evaluate Binary.



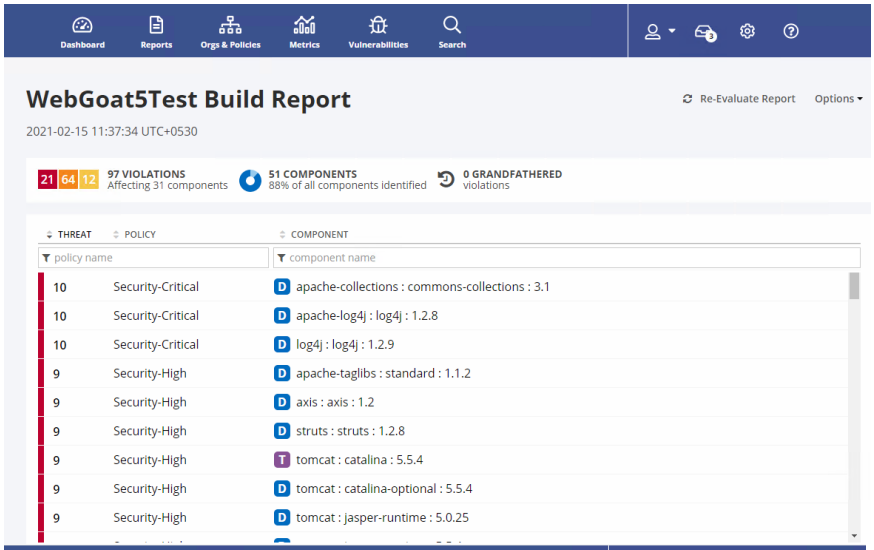
Select the WebGoat5.0.zip file and stage as Build. Click Upload.

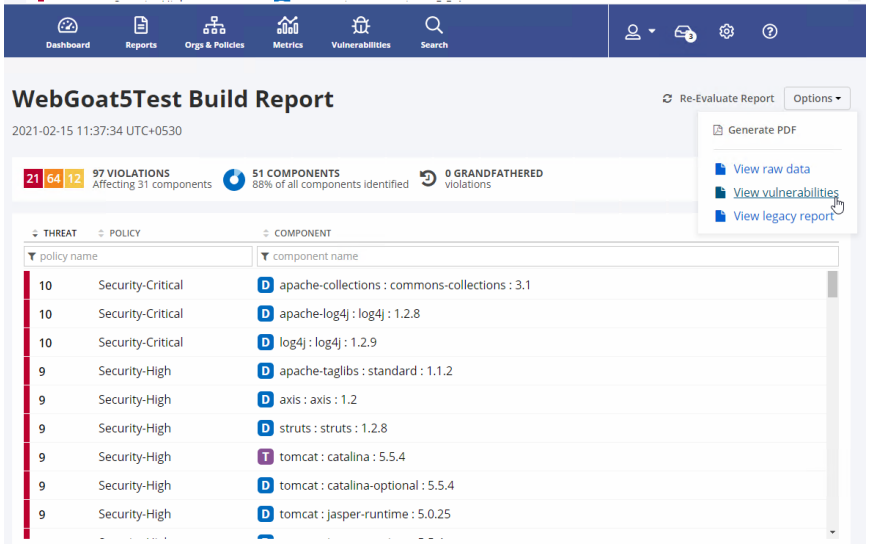


The IQ Server will start evaluation.

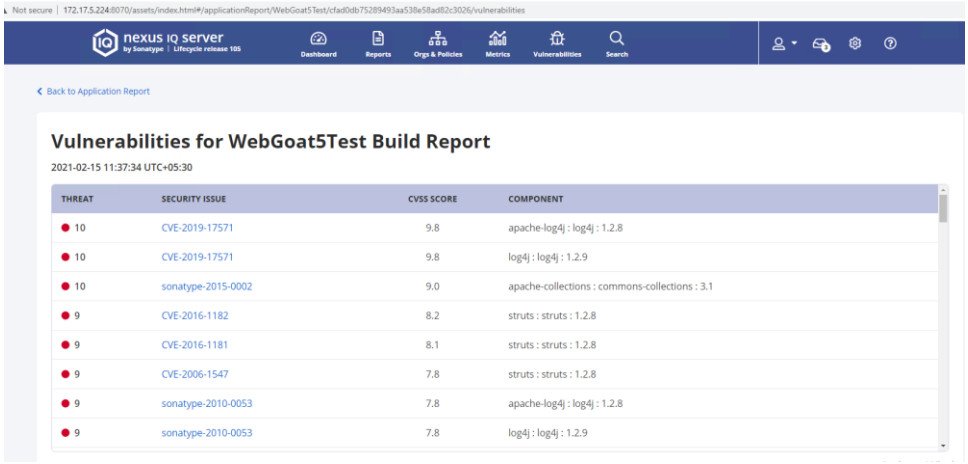


Click View Report.



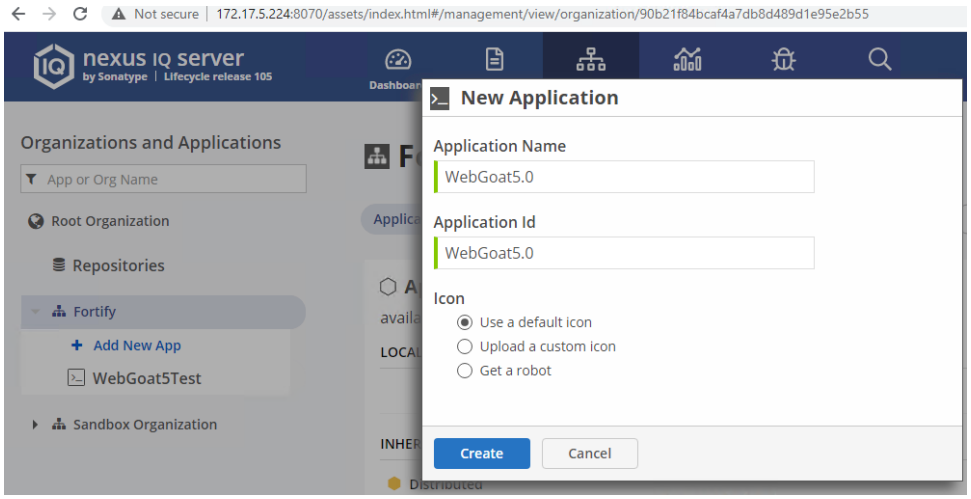


Click Options -> View Vulnerabilities.

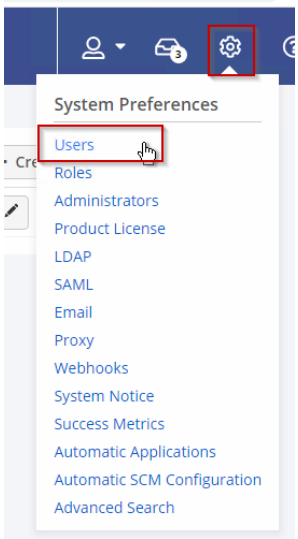


If you are able to see Vulnerabilities report then we are good for next step.

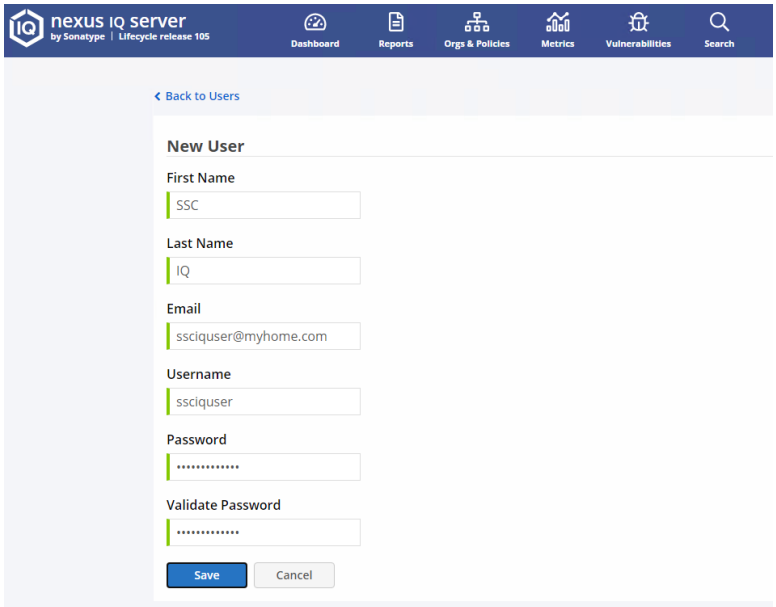
Delete the WebGoat5.0.zip file from Samples folder, this file is no longer needed.



Click on Fortify Organization, and create an Application named "WebGoat5.0" with the Application ID as "WebGoat5.0".

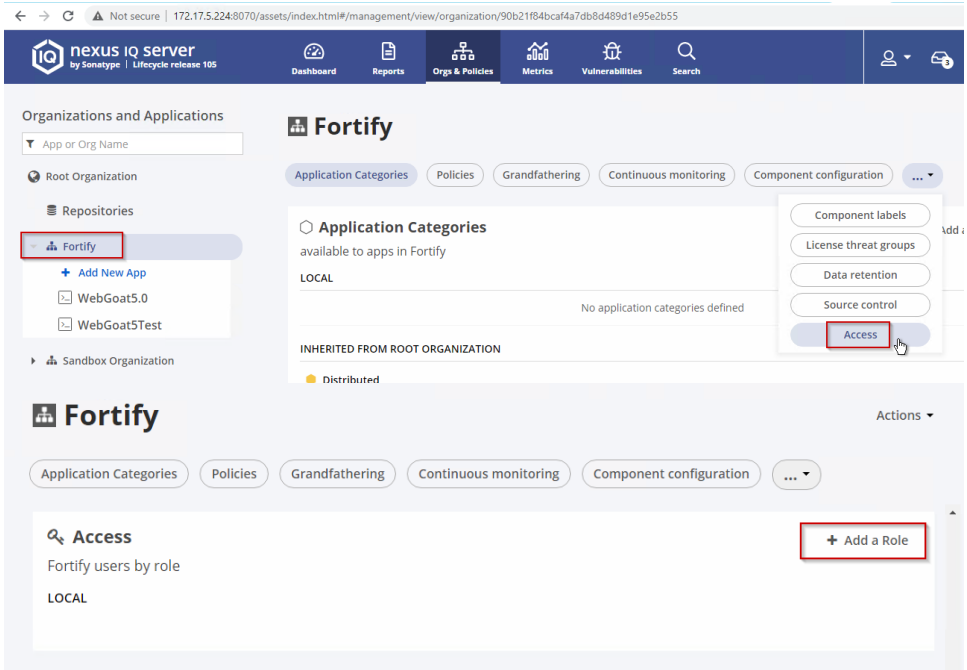


Create a user named "ssciquser" with the password "ssciquser@123" in Settings -> Users -> Create User.

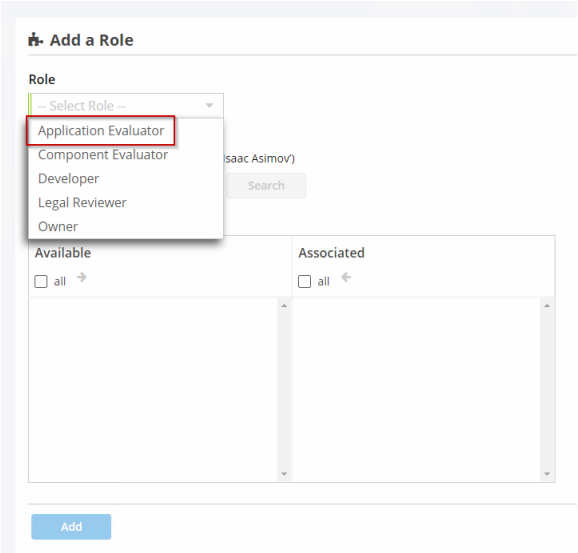


Click Save.

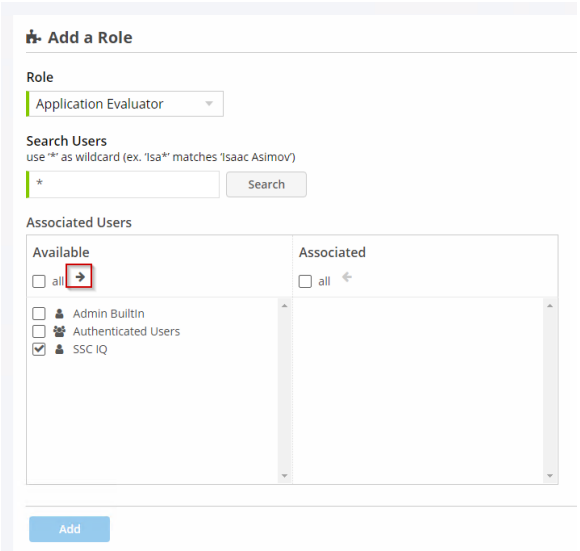
We will use this user in SCA. Lets assign this user in Fortify Organization as "Application Evaluator", "Component Evaluator" and "Developer" role.



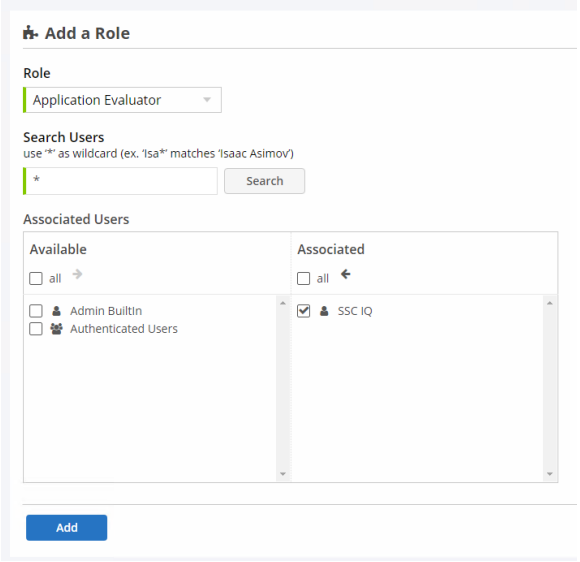
Click "Add a Role".



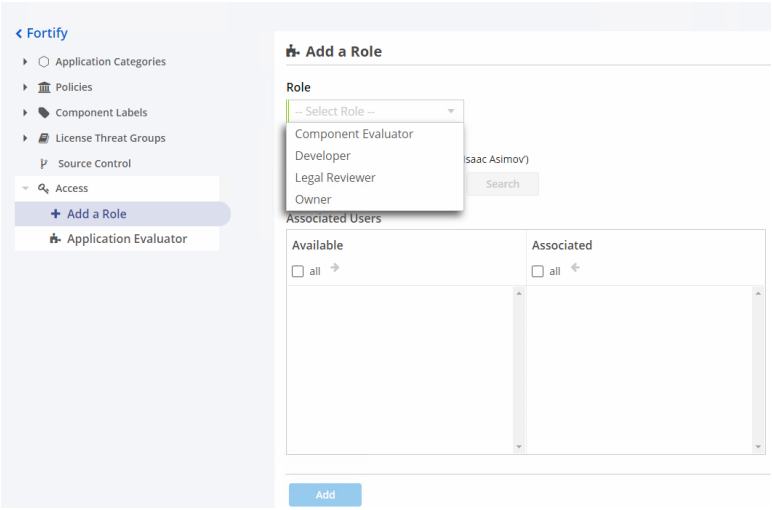
From Role drop down, select "Application Evaluator" role.



Enter "*" in Search Users and Click Search, it will display all the users. Select the check box before "SSC IQ" user and click the Right Arrow.

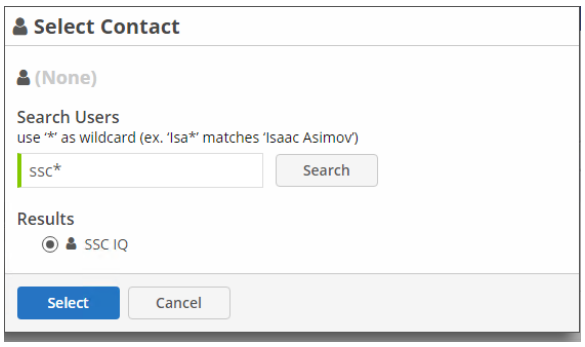


Click Add.



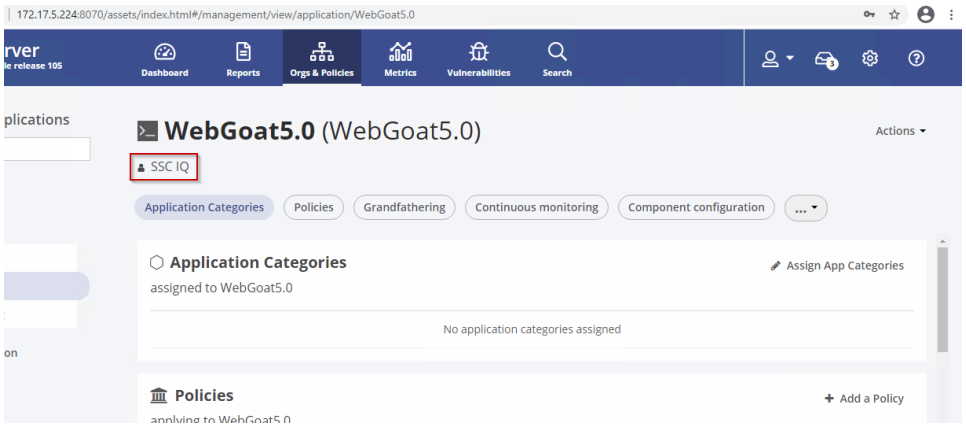
Repeat the same steps for "Component Evaluator" and "Developer" role.

To assign the point of contact for the application. Open WebGoat5.0 Application in Fortify Organization. Click Actions -> Select Contacts.

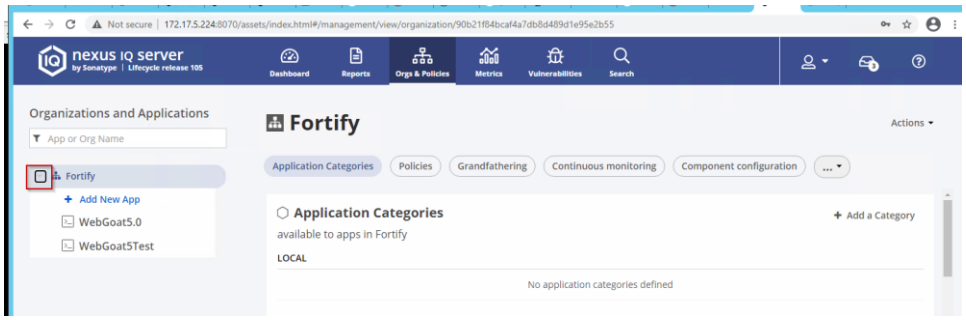


Search for "ssc*" and select "SSC IQ" user which was created for SCA integration.

Click Select.



Notice the change.
Logout from IQ Server and login as "ssciquser" with the password "ssciquser@123"

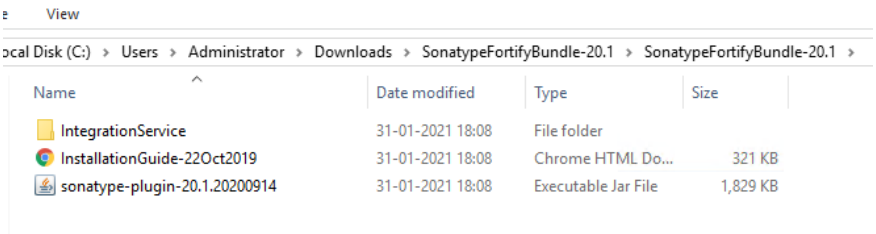


Expand the Fortify Organization and validate both applications are listed there.

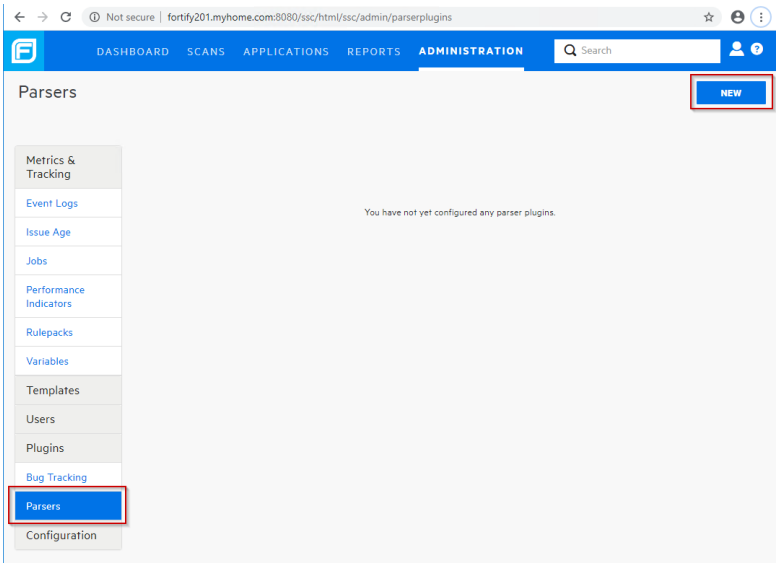
Installing SonaType SSC plugin

Download the plugin from https://marketplace.microfocus.com/fortify/content/sonatype-nexus-lifecycle-integration-with-ssc into Downloads folder.

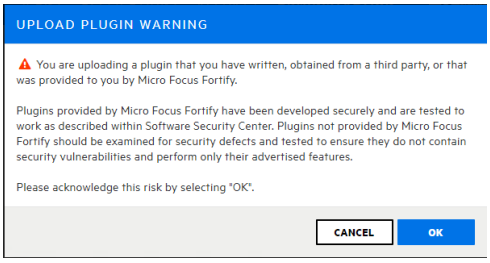
Download and extract the available latest version.



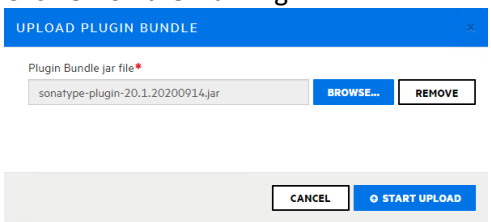
Login into SSC as admin -> Administration -> Plugins -> Parsers -> New.



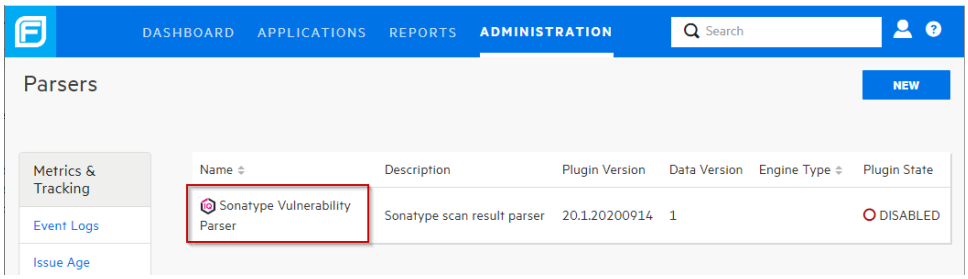
Click New.



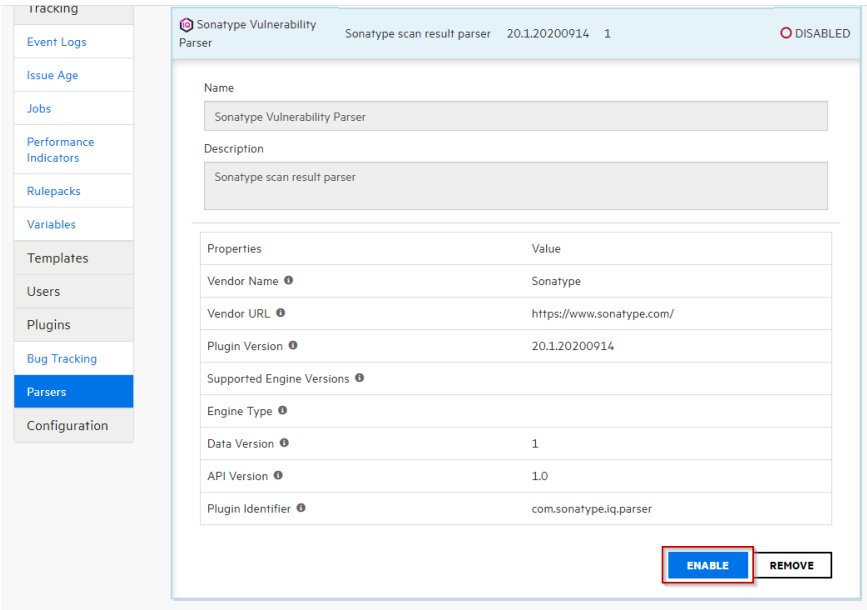
Click OK on the warning.



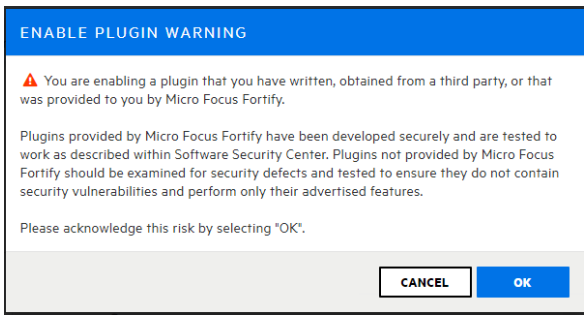
Browse and select the SonaType plugin jar file and click Start Upload.



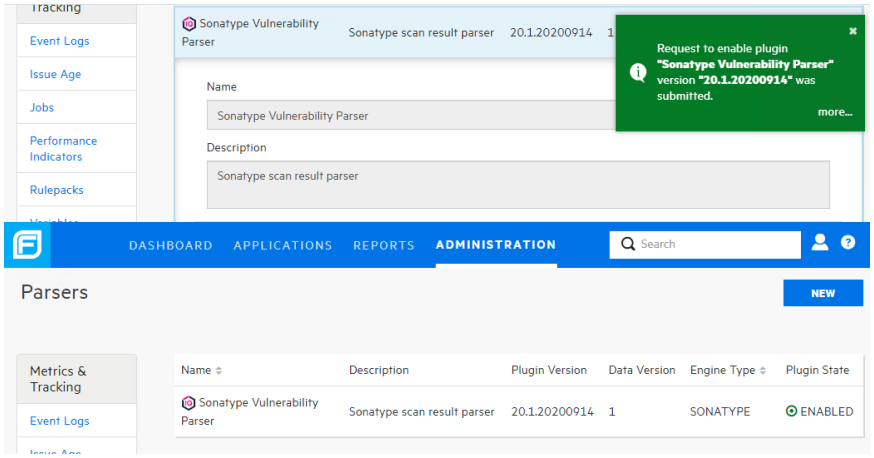
Click on the Plugin.



Click Enable.



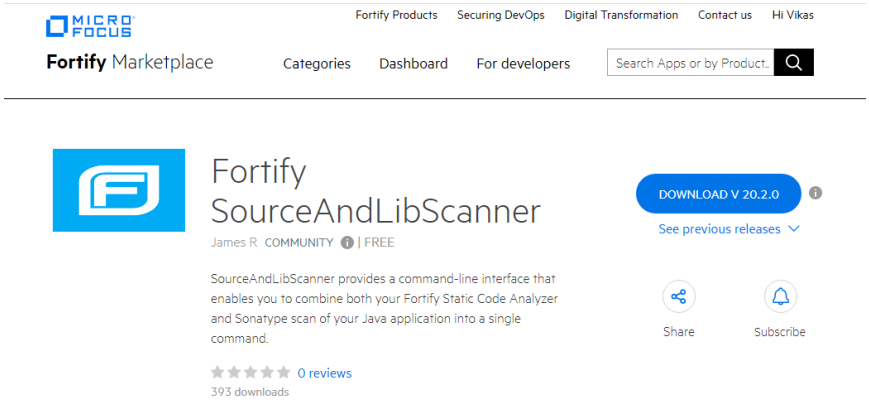
Click OK on the Warning.



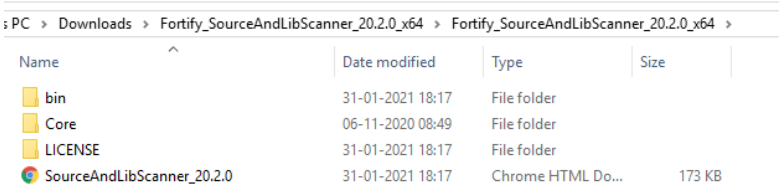
Now Plugin is Enabled.

Installing SourceAndLibScanner

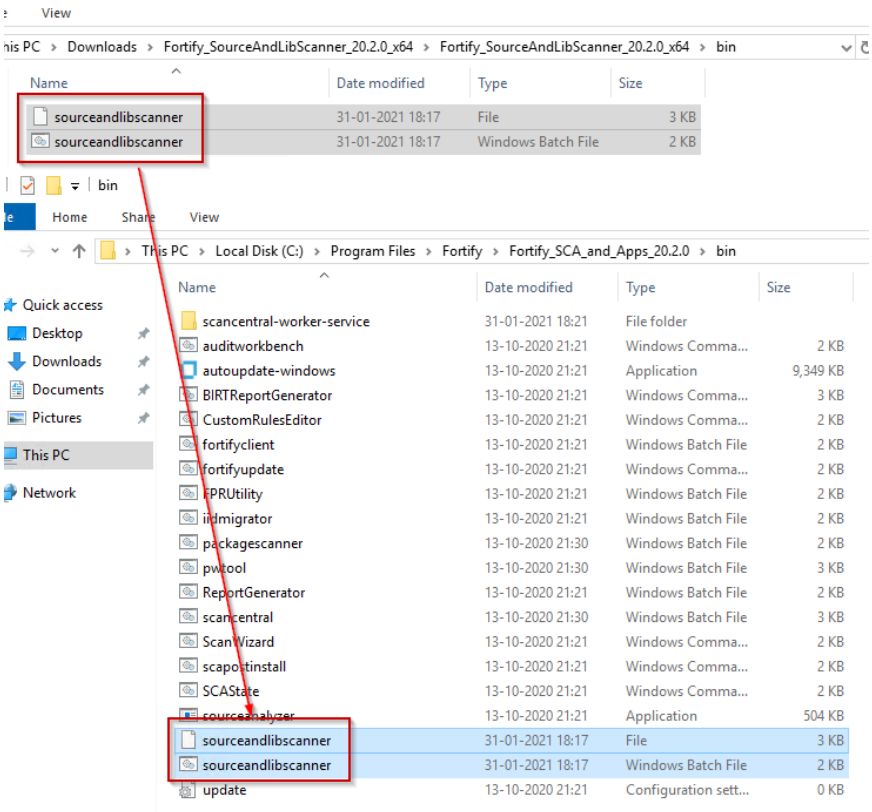
Download SourceAndLibScanner CLI Tool from https://marketplace.microfocus.com/fortify/content/fortify-sourceandlibscanner into Downloads folder.



Extract the downloaded zip file.



Copy all the files from the "`bin`" folder to "`C:\Program Files\Fortify\Fortify_SCA_and_Apps_20.2.0\bin`" folder.



Also, copy the files and folder from "`Core\lib`" to "`C:\Program Files\Fortify\Fortify_SCA_and_Apps_20.2.0\Core\lib`", do not overwrite any file if there is conflict on the files.

Using SourceAndLibScanner

- Scan your code with SCA and Sonatype (via the Open Source Component Scan Service), and then upload both results to SSC
- Scan your code with SCA and Sonatype, then upload both results to SSC and the Sonatype results to an on-premises Lifecycle product (Nexus IQ Server)
- Scan your code with SCA scans of your code OR perform Sonatype scans of your third-party components, for the Sonatype only option, you must use our -bt none option, executed in the top level dir, as we need to pass the libraries to the utility.

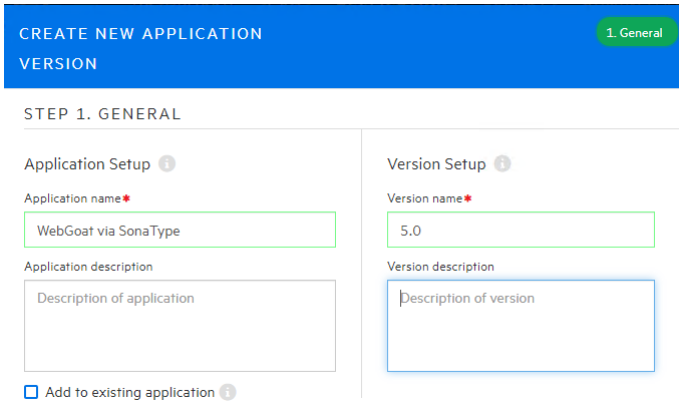
```
sourceandlibscanner -h
```

Displays the help of SourceAndLibScanner.

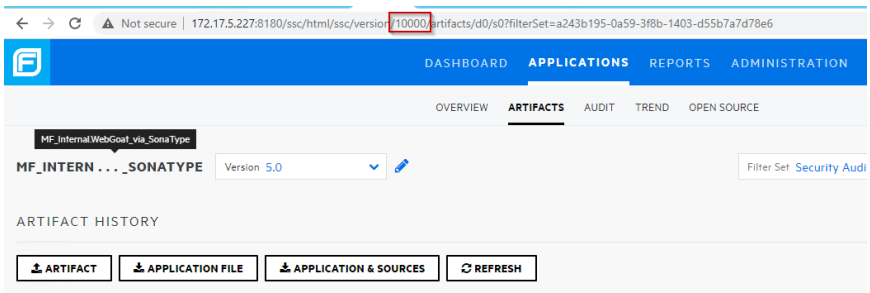
```
sourceandlibscanner -version
```

Displays the version of SourceAndLibScanner.

Scanning WebGoat via SourceAndLibScanner



Create a new Application named "MF_Internal.WebGoat_via_SonaType" and version as "5.0".



Identify the version id, as above (it is 10000 for this application) or use fortifyclient to identify the application's version id.

Generate a CIToken Use Administration -> Users -> Token Management -> New. Select Token Type as CI Token.

Note down the CIToken for Tools, we will be using this token in SCA.

Manual invocation of SCA and the Fortify / SonaType Open Source Scanning Service

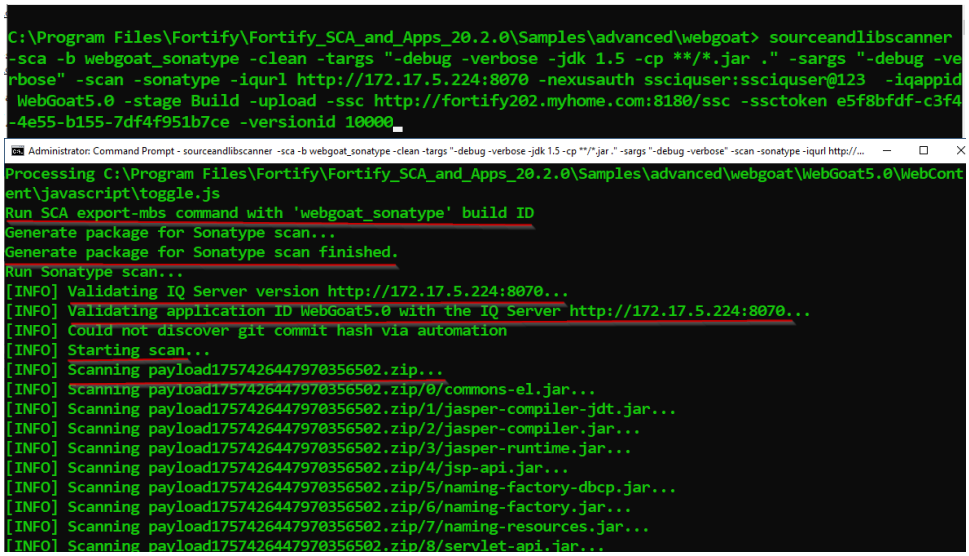
Run the below command in "`C:\Program Files\Fortify\Fortify_SCA_and_Apps_20.2.0\Samples\advanced\webgoat`" folder.

```
sourceandlibscanner -sca -b webgoat_sonatype -clean -targs "-verbose -cp -jdk 1.5 **/*.jar ." -sargs "-verbose" -scan -sonatype -libscanurl https://ds.sonatype.com -nexusauth <SonaType_Token> -upload -ssc http://fortify202.myhome.com:8080/ssc -ssctoken 6e057de0-87b8-4430-a606-da2dad049c0d -versionid 10000
```

While using IQ Server (-debug and -verbose is optional)-

```
sourceandlibscanner -sca -b webgoat_sonatype -clean -targs "-debug -verbose -jdk 1.5 -cp **/*.jar ." -sargs "-debug -verbose" -scan -sonatype -iqurl http://172.17.5.224:8070 -nexusauth ssciquser:ssciquser@123  -iqappid WebGoat5.0 -stage Build -upload -ssc http://fortify202.myhome.com:8180/ssc -ssctoken e5f8bfdf-c3f4-4e55-b155-7df4f951b7ce -versionid 10000
```

Note: Source path must be last command in targs



During the scan you can see the number of issues identified by IQ Server.

```
Administrator: Command Prompt - sourceandlibscanner -sca -b webgoat_sonatype -clean -targs "-debug -verbose -jdk 1.5 -cp **/*.jar ." -sargs "-debug -verbose" -scan -sonatype -iqurl http://...
[INFO]
[INFO]
[INFO]
[INFO]
[INFO] ******************************************************************************************
[INFO] Policy Action: None
[INFO] Stage: build
[INFO] Number of components affected: 12 critical, 13 severe, 6 moderate
[INFO] Number of open policy violations: 21 critical, 64 severe, 12 moderate
[INFO] Number of grandfathered policy violations: 0
[INFO] The detailed report can be viewed online at http://172.17.5.224:8070/ui/links/application/WebGoat5.0
/report/a035d0a418474df882cee2c861d5033e
[INFO] ******************************************************************************************
Sonatype scan finished.
Generate Sonatype based rules...
Sonatype based rules generation failed. See log for details.
Run SCA scan command with 'webgoat_sonatype' build ID
Fortify Static Code Analyzer 20.2.0.0139 (using JRE 1.8.0_181)
Analyzing 190 source file(s)
```

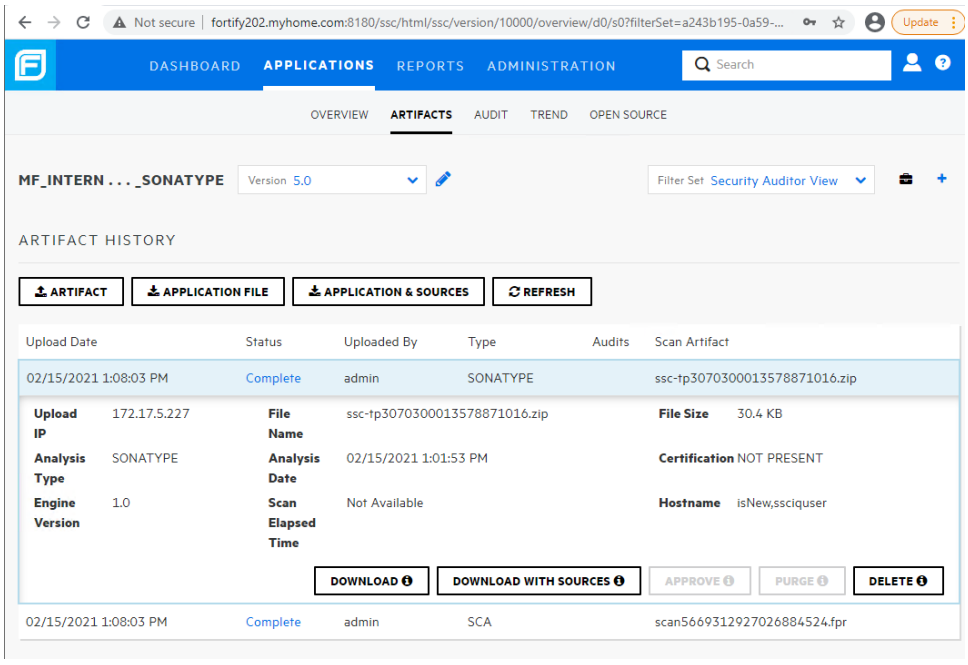
The URL is the direct link of the report.

```
Administrator: Command Prompt
/report/a035d0a418474df882cee2c861d5033e
[INFO] ******************************************************************************************
Sonatype scan finished.
Generate Sonatype based rules...
Sonatype based rules generation failed. See log for details.
Run SCA scan command with 'webgoat_sonatype' build ID
Fortify Static Code Analyzer 20.2.0.0139 (using JRE 1.8.0_181)
Analyzing 190 source file(s)
Configuration analysis complete
Rendering 1164 results
Analysis completed in 02:52
SCA execution finished.
Convert Sonatype report for uploading to SSC...
Sonatype report conversion finished.
Start uploading SCA results to SSC...
Uploading SCA results to SSC finished.
Start uploading Sonatype results to SSC...
Uploading Sonatype results to SSC finished.

C:\Program Files\Fortify\Fortify_SCA_and_Apps_20.2.0\Samples\advanced\webgoat>
```

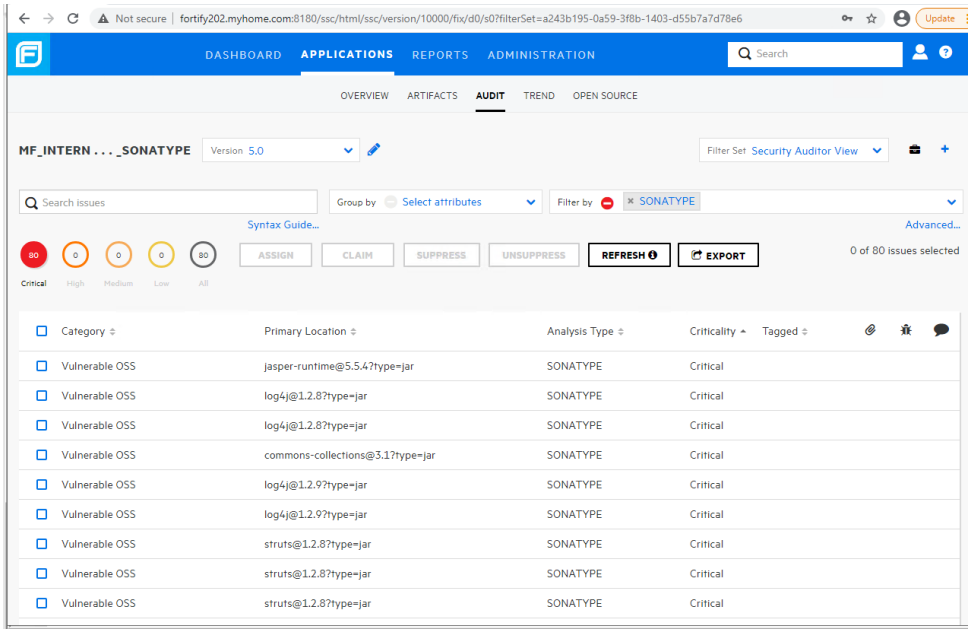
Check the on-screen information.

Open SSC and open the artifacts of the application.

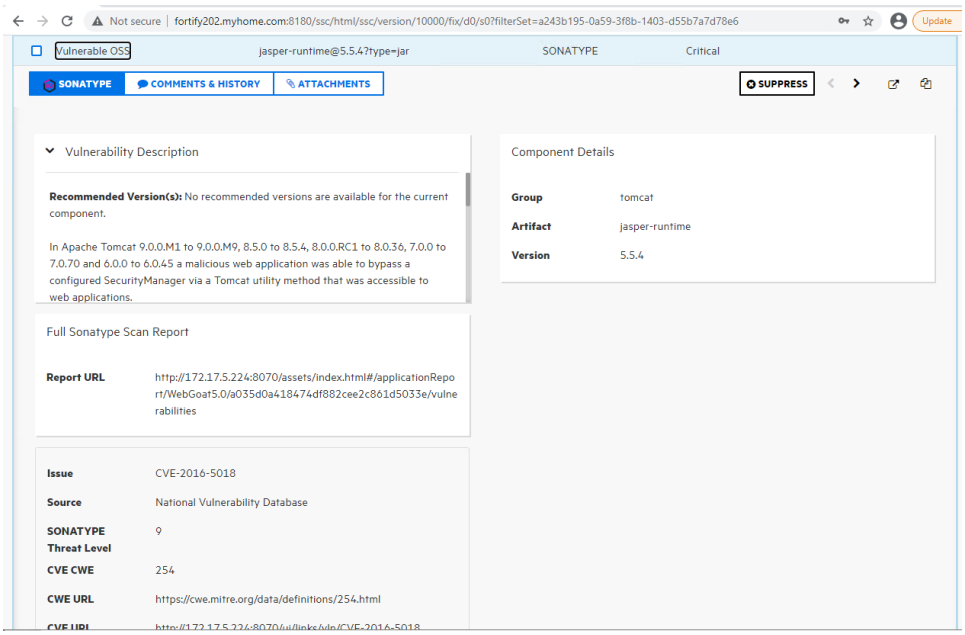


The ZIP file uploaded by SonaType will be listed in the Artifacts section, click on Audit.

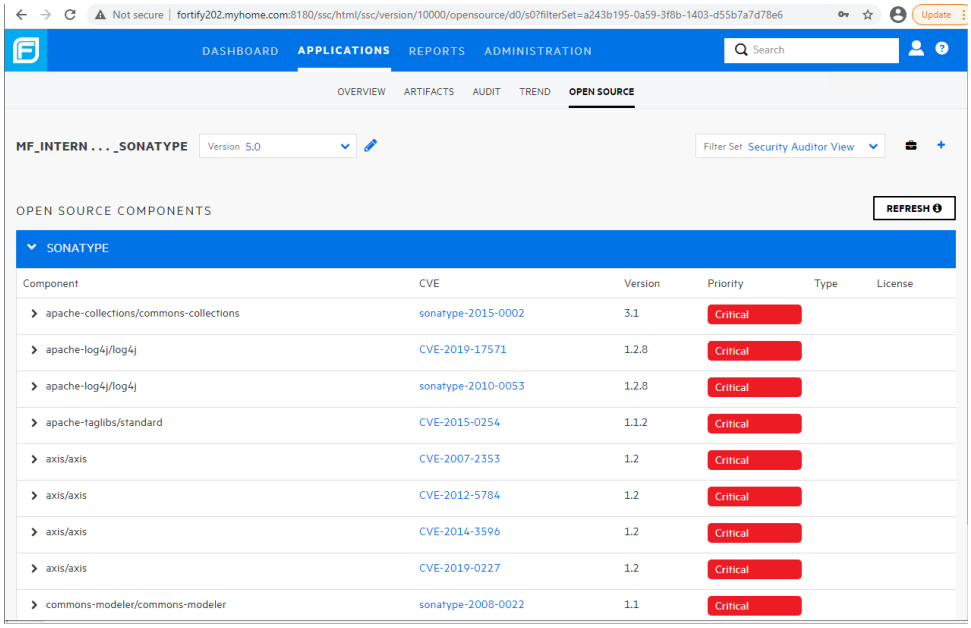
In the Filter select SONATYPE.



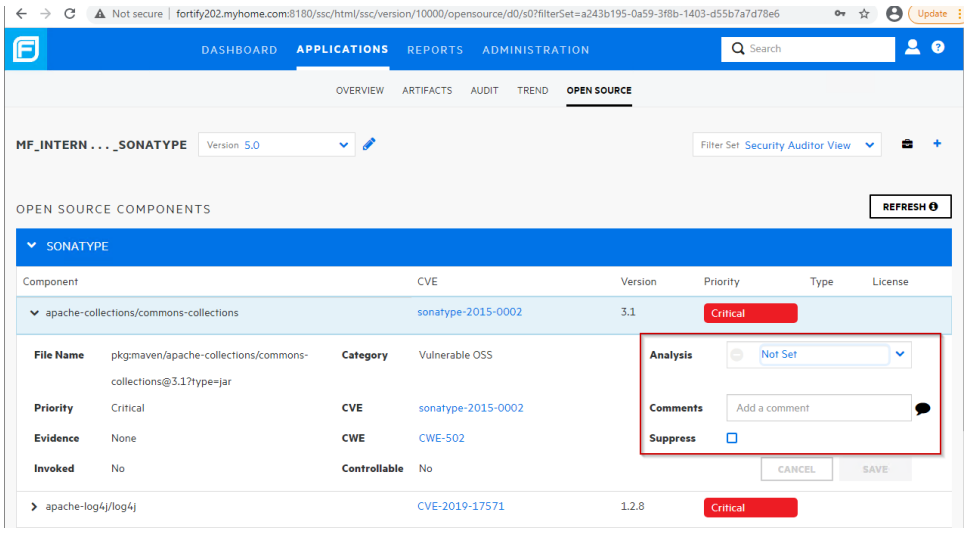
Expand the vulnerability identified by SonaType and check the details provided by SonaType.



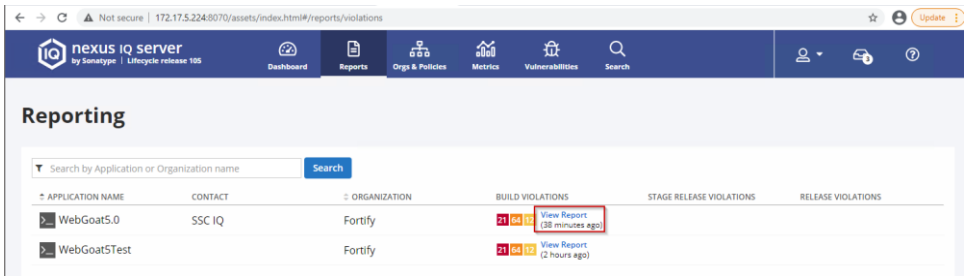
Click on OPEN SOURCE tab.



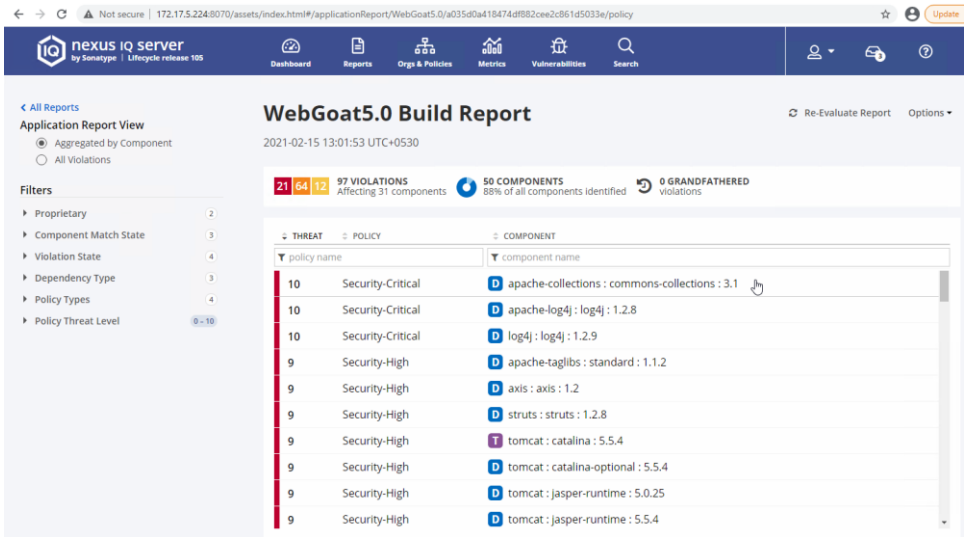
Expand the Vulnerability.



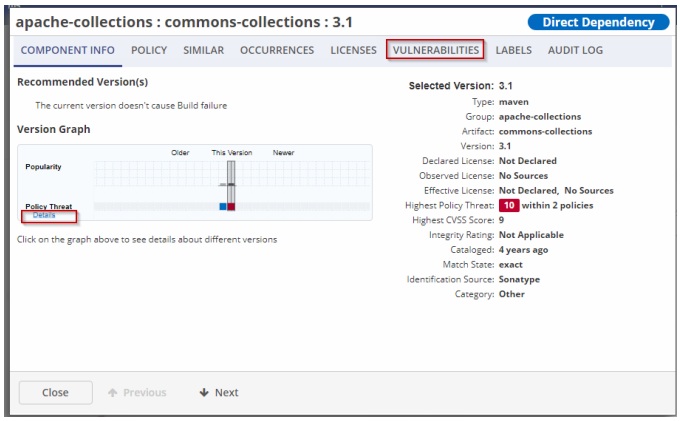
Auditing can be done on this page as well.
Login into IQ Server using user as "ssciquser" and the password "ssciquser@123" and click Reports.



Click on View Report.



Open the Component listed in the report.



Explore the various options.
The short version of the command is below, where we can use build type to `none`.

```
sourceandlibscanner -auto -bt none -scan -sonatype -iqurl
http://172.17.5.224:8070 -nexusauth ssciquser:ssciquser@123 -
iqappid WebGoat5.0 -stage build -upload -ssc
http://fortify202.myhome.com:8180/ssc -ssctoken e5f8bfdf-c3f4-4e55-
b155-7df4f951b7ce -versionid 10000
```

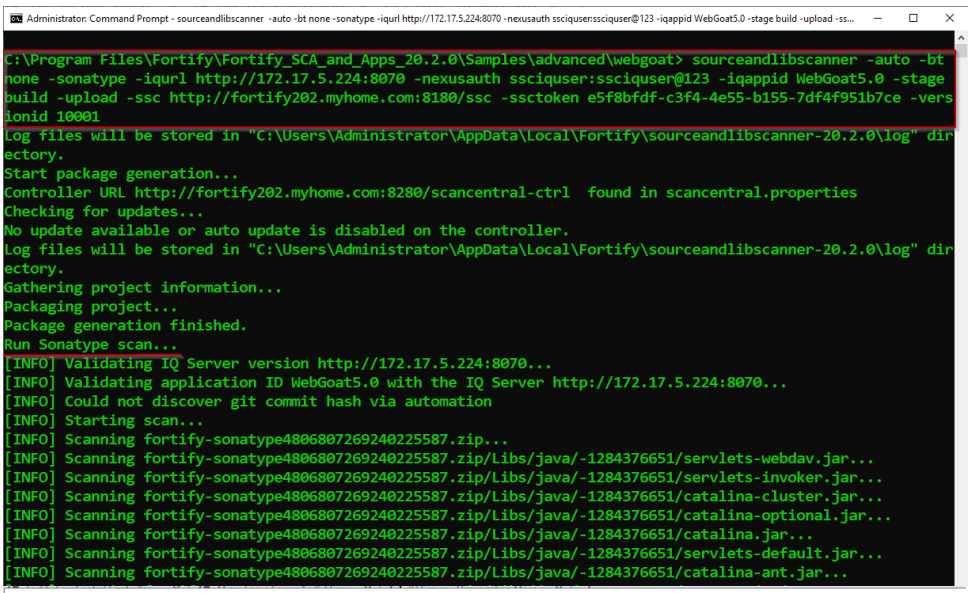
Scanning WebGoat via SonaType only

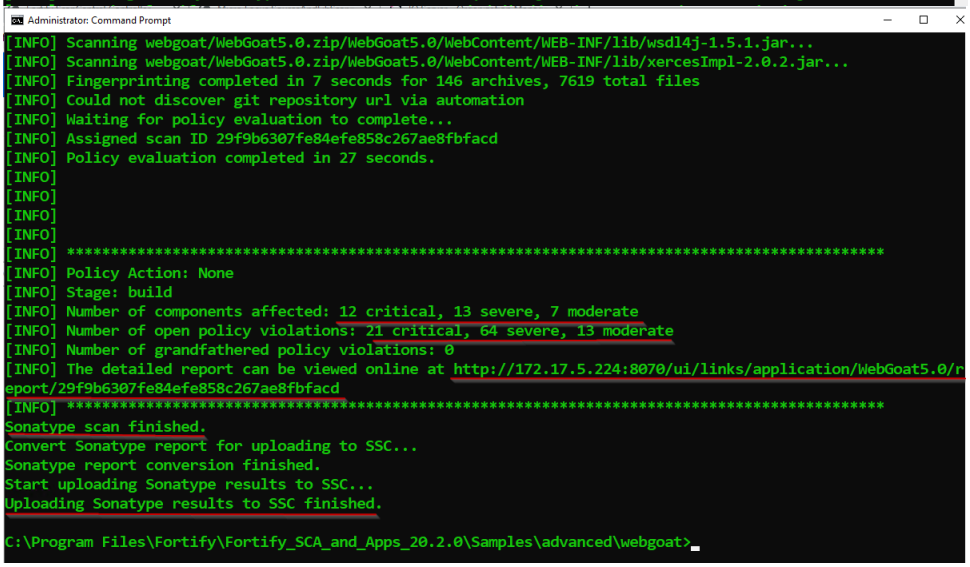
Create a new version "6.0" of Application named " MF_Internal.WebGoat_via_SonaType" and note down the version id.
Run the below command in "`C:\Program Files\Fortify\Fortify_SCA_and_Apps_20.2.0\Samples\advanced\webgoat`" folder notice it is missing "`-scan`"

```
sourceandlibscanner -sca -b webgoat_sonatype -clean -targs "-debug
-verbose -jdk 1.5 -cp **/*.jar ." -sargs "-debug -verbose" -
sonatype -iqurl http://172.17.5.224:8070 -nexusauth
ssciquser:ssciquser@123  -iqappid WebGoat5.0 -stage Build -upload -
ssc http://fortify202.myhome.com:8180/ssc -ssctoken e5f8bfdf-c3f4-
4e55-b155-7df4f951b7ce -versionid 10001
```

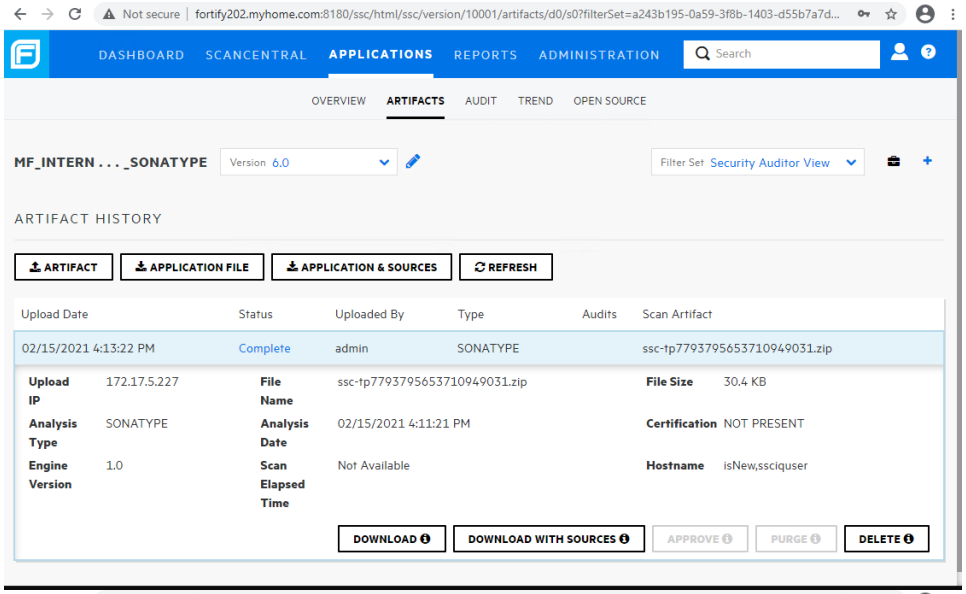
or

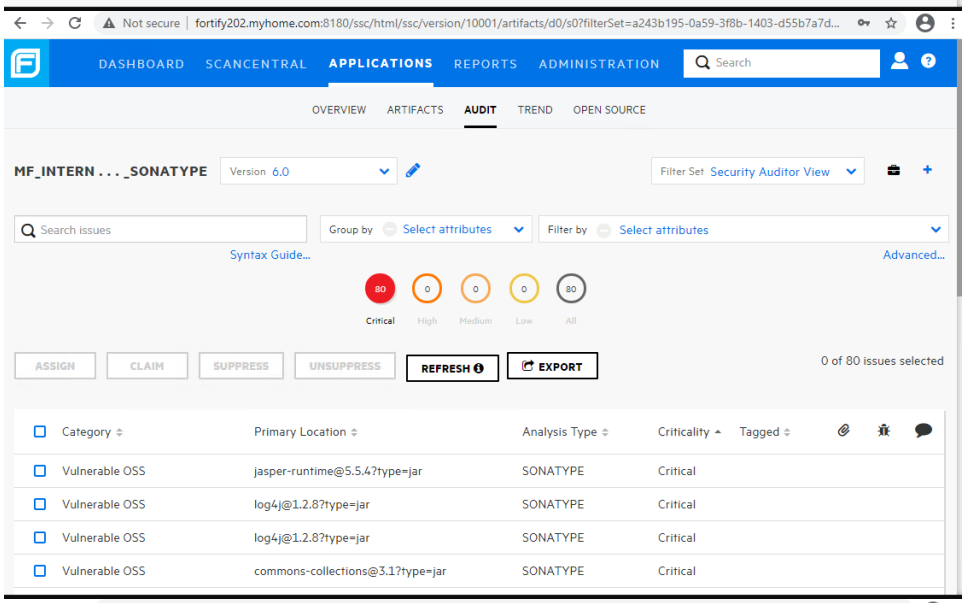
```
sourceandlibscanner -auto -bt none -sonatype -iqurl
http://172.17.5.224:8070 -nexusauth ssciquser:ssciquser@123 -
iqappid WebGoat5.0 -stage build -upload -ssc
http://fortify202.myhome.com:8180/ssc -ssctoken e5f8bfdf-c3f4-4e55-
b155-7df4f951b7ce -versionid 10001
```

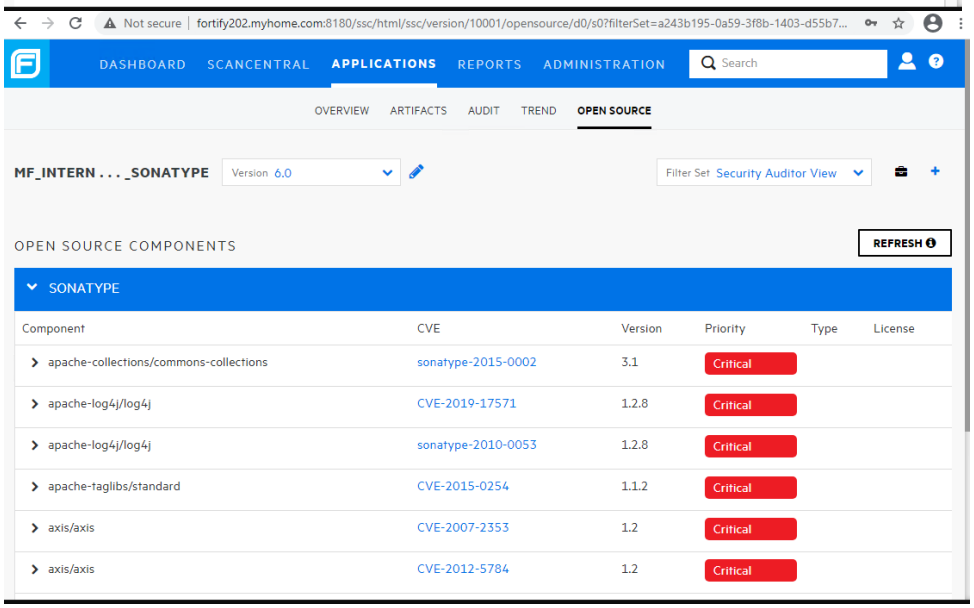




In SSC –


Not secure | fortify202.myhome.com:8180/ssc/html/ssc/version/10001/artifacts/d0/s0?filterSet=a243b195-0a59-3f8b-1403-d55b7a7d...
DASHBOARD SCANCENTRAL APPLICATIONS REPORTS ADMINISTRATION
Search
OVERVIEW ARTIFACTS AUDIT TREND OPEN SOURCE
MF_INTERN . . . _SONATYPE
Version 6.0
Filter Set Security Auditor View
ARTIFACT HISTORY
ARTIFACT
APPLICATION FILE
APPLICATION & SOURCES
REFRESH
Upload Date
Status
Uploaded By
Type
Audits
Scan Artifact
02/15/2021 4:13:22 PM
Complete
admin
SONATYPE
ssc-tp7793795653710949031.zip
Upload IP
172.17.5.227
File Name
ssc-tp7793795653710949031.zip
File Size
30.4 KB
Analysis Type
SONATYPE
Analysis Date
02/15/2021 4:11:21 PM
Certification
NOT PRESENT
Engine Version
1.0
Scan Elapsed Time
Not Available
Hostname
isNew,ssciquser
DOWNLOAD
DOWNLOAD WITH SOURCES
APPROVE
PURGE
DELETE

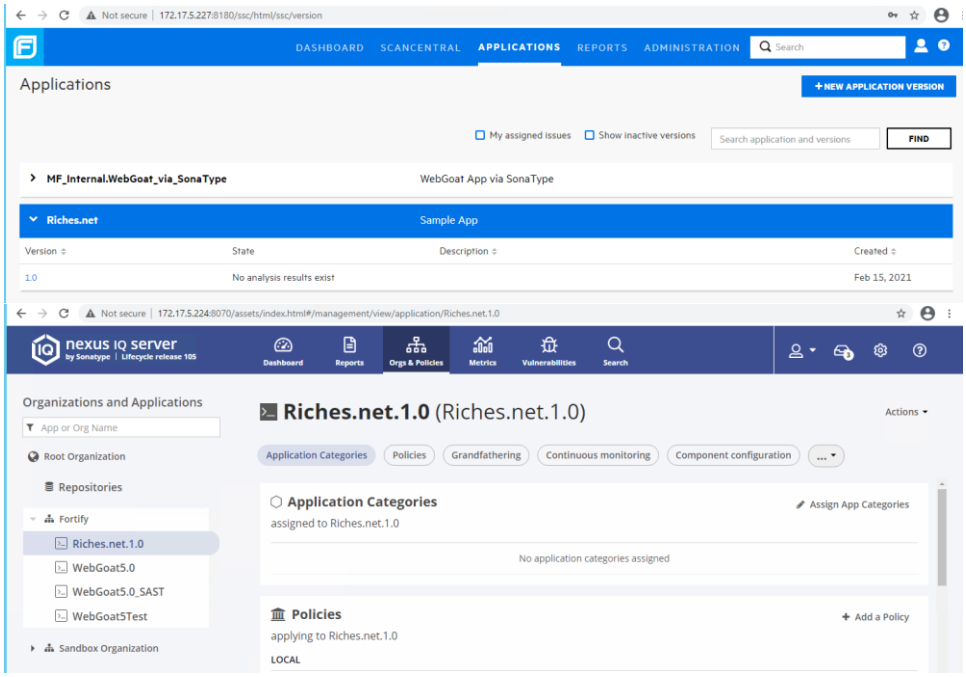

Not secure | fortify202.myhome.com:8180/ssc/html/ssc/version/10001/artifacts/d0/s0?filterSet=a243b195-0a59-3f8b-1403-d55b7a7d...
DASHBOARD SCANCENTRAL APPLICATIONS REPORTS ADMINISTRATION
Search
OVERVIEW ARTIFACTS AUDIT TREND OPEN SOURCE
MF_INTERN . . . _SONATYPE
Version 6.0
Filter Set Security Auditor View
Search issues
Group by Select attributes
Filter by Select attributes
Syntax Guide...
Advanced...
80
0
0
0
80
Critical
High
Medium
Low
All
ASSIGN
CLAIM
SUPPRESS
UNSUPPRESS
REFRESH
EXPORT
0 of 80 issues selected
Category
Primary Location
Analysis Type
Criticality
Tagged
Vulnerable OSS
jasper-runtime@5.5.4?type=jar
SONATYPE
Critical
Vulnerable OSS
log4j@1.2.8?type=jar
SONATYPE
Critical
Vulnerable OSS
log4j@1.2.8?type=jar
SONATYPE
Critical
Vulnerable OSS
commons-collections@3.1?type=jar
SONATYPE
Critical


Not secure | fortify202.myhome.com:8180/ssc/html/ssc/version/10001/opensource/d0/s0?filterSet=a243b195-0a59-3f8b-1403-d55b7...
DASHBOARD SCANCENTRAL APPLICATIONS REPORTS ADMINISTRATION
Search
OVERVIEW ARTIFACTS AUDIT TREND OPEN SOURCE
MF_INTERN . . . _SONATYPE
Version 6.0
Filter Set Security Auditor View
OPEN SOURCE COMPONENTS
REFRESH
SONATYPE
Component
CVE
Version
Priority
Type
License
apache-collections/commons-collections
sonatype-2015-0002
3.1
Critical
apache-log4j/log4j
CVE-2019-17571
1.2.8
Critical
apache-log4j/log4j
sonatype-2010-0053
1.2.8
Critical
apache-taglibs/standard
CVE-2015-0254
1.1.2
Critical
axis/axis
CVE-2007-2353
1.2
Critical
axis/axis
CVE-2012-5784
1.2
Critical

Scanning Riches.Net Code with SonaType IQ Server and SCA

Make sure Visual Studio 2019 CE / EE is installed.

Create an application in SSC as "Riches.net" version "1.0" and in IQ Server application name and ID will be "Riches.net.1.0" in Fortify Organization.



Open CMD in C:\Program Files\Fortify\Fortify_SCA_and_Apps_20.2.0\Samples\advanced\riches.net folder. Run the below commands one by one –

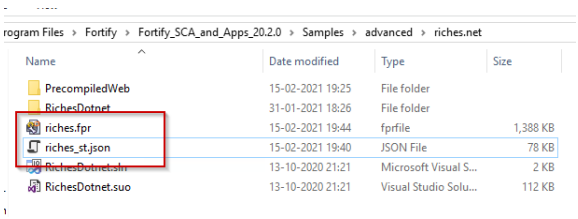
```
MKDIR "C:\hp_la_chouffe\rules\scratch\RichesDotnet\scratch\RichesDotnet"

"C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\Tools\VsDevCmd.bat"

sourceandlibscanner -auto -bt msbuild -bf RichesDotnet.sln -scan -sonatype -iqurl http://172.17.5.224:8070 -nexusauth ssciquser:ssciquser@123 -iqappid Riches.net.1.0 -stage build -f riches.fpr -r riches_st.json
```

if everything goes well then you will see two files –

1. riches.fpr containing SCA results
2. riches_st.json containing SonaType IQ Server results



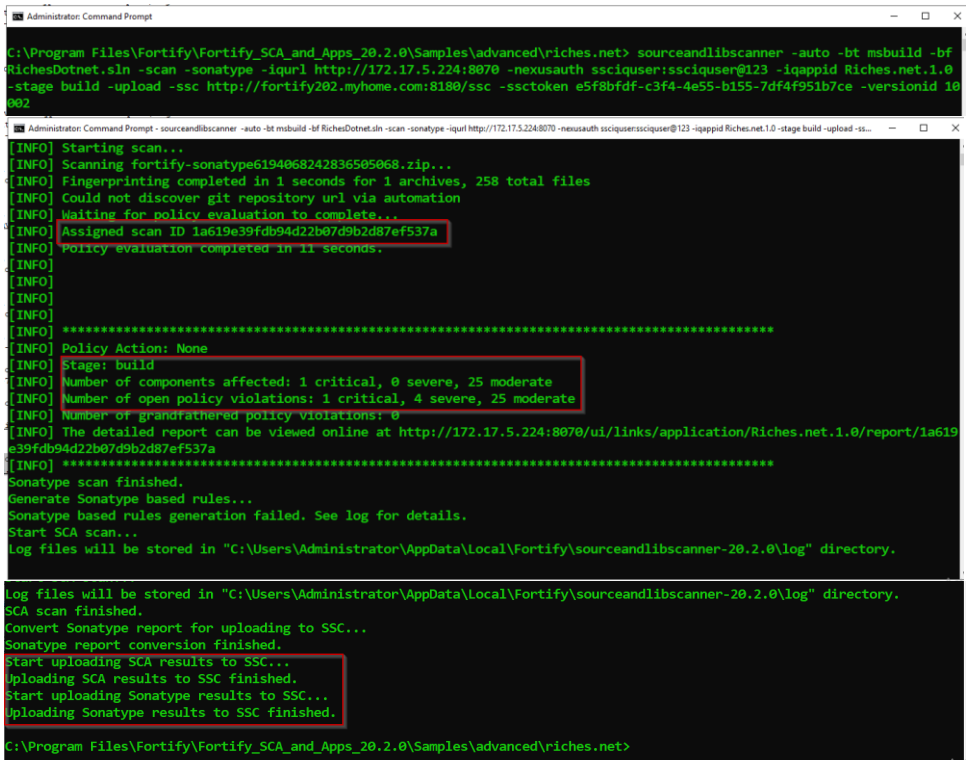
Open these files into respective applications and verify the contents.

Now either delete both files or move it some other folder i.e. Downloads.

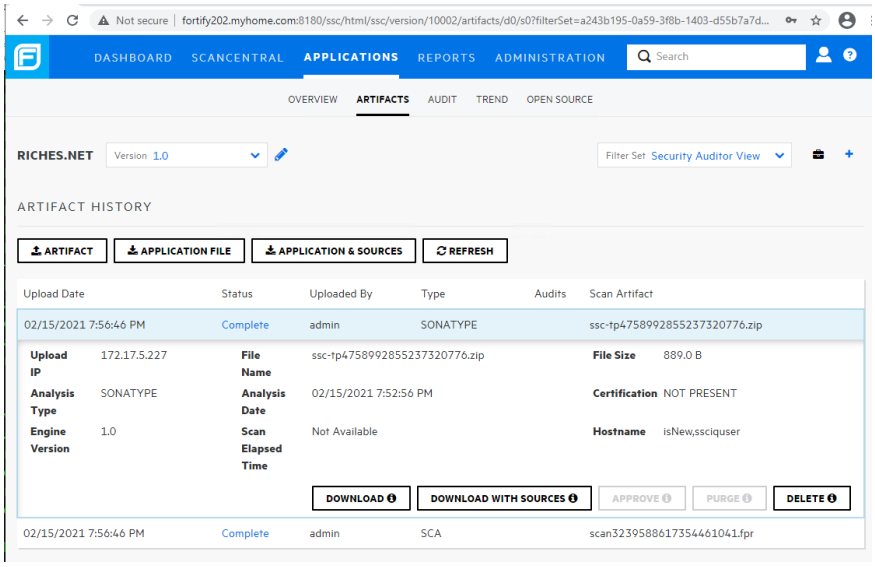
Identify the version id of the Riches.net application in SSC or use fortifyclient.

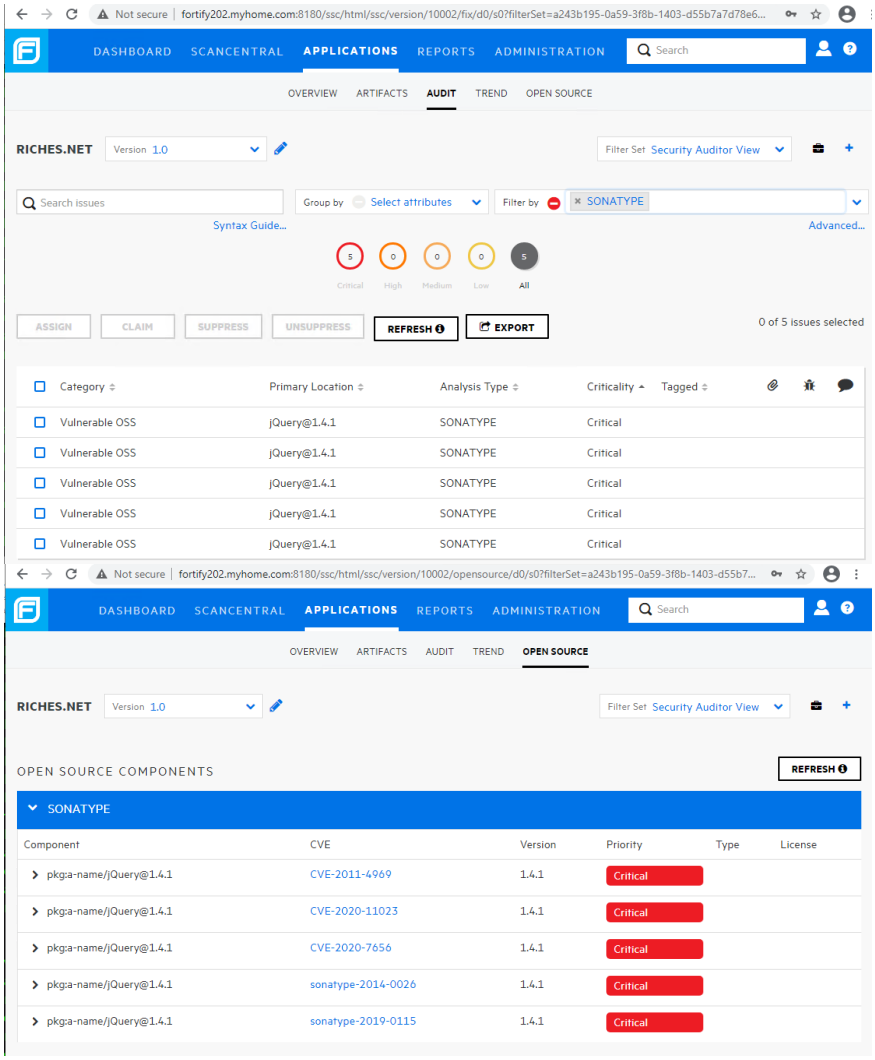
Run the below command after updating the version id.

```
sourceandlibscanner -auto -bt msbuild -bf RichesDotnet.sln -scan -
sonatype -iqurl http://172.17.5.224:8070 -nexusauth
ssciquser:ssciquser@123 -iqappid Riches.net.1.0 -stage build -
upload -ssc http://fortify202.myhome.com:8180/ssc -ssctoken
e5f8bfdf-c3f4-4e55-b155-7df4f951b7ce -versionid 10002
```



Verify in SSC –





Log Files

Log files are located in the following directories:

Windows:

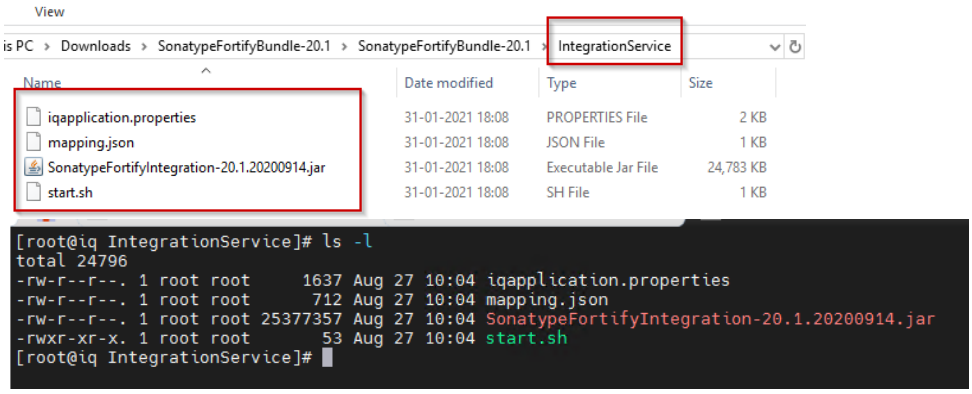
C:\Users\<username>\AppData\Local\Fortify\sca<version>\log
C:\Users\<username>\AppData\Local\Fortify\sourceandlibscanner-<version>\log

Linux and macOS:

<userhome>/.fortify/sca<version>/log
<userhome>/.fortify/sourceandlibscanner-<version>/log

Deploy Integration Service on SonaType IQ Server

Upload the folder "IntegrationService" to IQ Server's / folder.



Open iqapplication.properties file in vi and modify it based on your environment –

```
# Default port to listen on, set as needed
server.port=8182

# URL and creds for IQ Server
iqserver.url=http://172.17.5.225:8070/
iqserver.username=admin
iqserver.password=Passw0rd

# URL and creds (CIToken token) to SSC server: see https://www.microfocus.com/documentation/fortify-software-
security-center/2010/SSC_Help_20.1.0/index.htm#SSC_UG/Gen_Auth_Tokens.htm
sscserver.url=http://172.17.5.227:8080/ssc/
sscserver.token=ZTVmOGJmZGYtYzNmNC00ZTU1LWIxNTUtN2RmNGY5NTFiN2Nl

# work directory where JSON files are stored
loadfile.location=.work/
mapping.file=mapping.json

# Update the mapping.file with project values dynamically passed as request parameters
update.mapping.file=true

# Define which report type to view (raw, vulnerabilities, policy = default)
iq.report.type=vulnerabilities

# directory/file where log files are stored
logfile.location=.work/Servicelog.log
logLevel=info

# cron expression; it consists of 7 fields
# <second> <minute> <hour> <day-of-month> <month> <day-of-week> <year>
# <year> field is optional. Rest all are required
# Some examples are as follows:
# Running every 12 hours starting at 6 - 0 0/720 6 * * ?
# Running every 12 hours starting at midnight - 0 0/720 0 * * ?
# Running every 6 hrs starting at 6 AM - 0 0/360 6 * * ?
# For more details please visit - https://www.baeldung.com/cron-expressions
# Currently scheduled to run at 6 AM and then every 6 hours.
scheduling.job.cron=0 0/360 6 * * ?

# Set it to true if wanted to close the process after next scheduled run or
# leave set to true if you want to use your own scheduler
# also http://localhost/killProcess will stop the process
KillProcess=false
```

Save the file.
Open mapping.json file in vi and edit as per environment.

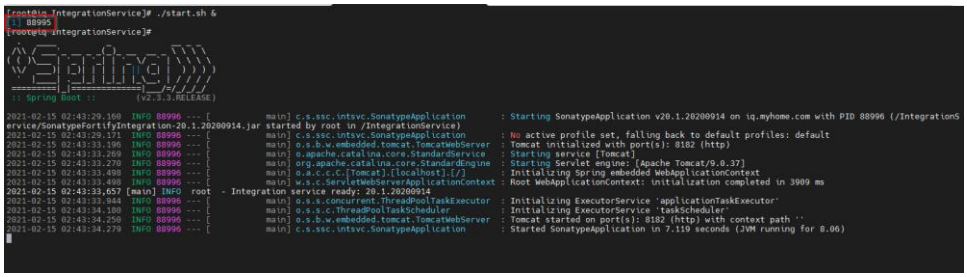
```
[
    {
      "sonatypeProject": "petclinic",
      "sonatypeProjectStage": "release",
      "fortifyApplication": "PetClinic",
      "fortifyApplicationVersion": "1.0"
    },
    {
      "sonatypeProject": "petclinic",
      "sonatypeProjectStage": "build",
      "fortifyApplication": "PetClinic",
```

```
    "fortifyApplicationVersion": "2.0"
  },
  {
    "sonatypeProject": "struts-showcase",
    "sonatypeProjectStage": "stage-release",
    "fortifyApplication": "Struts Showcase",
    "fortifyApplicationVersion": "1.0"
  },
  {
    "sonatypeProject": "webgoat8",
    "sonatypeProjectStage": "release",
    "fortifyApplication": "Webgoat8",
    "fortifyApplicationVersion": "1.0"
  },
  {
    "sonatypeProject": "WebGoat5.0",
    "sonatypeProjectStage": "build",
    "fortifyApplication": "MF_Internal.WebGoat_via_SonaType",
    "fortifyApplicationVersion": "5.0"
  }
]
```

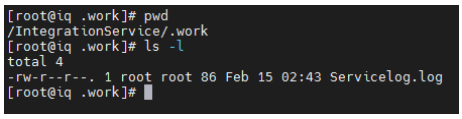
Save and Close the file.

Start the service using –

```
# chmod +x start.sh
# ./start.sh &
```



It will create a folder named work. Which will contain the log file.



< !! **End of the Document** !! >