Micro Focus
Fortify WebInspect and OAST on Docker

Software Version: 22.1.0
Windows® and Linux® operating systems

User Guide

Document Release Date: June 2022
Software Release Date: June 2022

Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

https://www.microfocus.com

Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Except as specifically indicated otherwise, a valid license from Micro Focus is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

This document was produced on May 31, 2022. To check for recent updates or to verify that you are using the most recent edition of a document, go to:

https://www.microfocus.com/support/documentation

Contents

Preface

Contacting Micro Focus Fortify Customer Support

Visit the Support website to:

- Manage licenses and entitlements
- Create and manage technical assistance requests
- Browse documentation and knowledge articles
- Download software
- Explore the Community

https://www.microfocus.com/support

For More Information

For more information about Fortify software products:

https://www.microfocus.com/cyberres/application-security

About the Documentation Set

The Fortify Software documentation set contains installation, user, and deployment guides for all Fortify Software products and components. In addition, you will find technical notes and release notes that describe new features, known issues, and last-minute updates. You can access the latest versions of these documents from the following Micro Focus Product Documentation website:

https://www.microfocus.com/support/documentation

To be notified of documentation updates between releases, subscribe to Fortify Product Announcements on the Micro Focus Community:

https://community.microfocus.com/cyberres/fortify/w/fortify-product-announcements

Fortify Product Feature Videos

You can find videos that highlight Fortify products and features on the Fortify Unplugged YouTube channel:

https://www.youtube.com/c/FortifyUnplugged

Change Log

The following table lists changes made to this document. Revisions to this document are published between software releases only if the changes made affect product functionality.

Software Release / Document Version	Changes
22.1.0	Updated: • PowerShell script content with `--hostname` and `ScannerDescription`. See "Running the Container in ScanCentral DAST Mode" on page 22. • Corrected `-v` option sample and description for running the OAST container. See "Running the OAST Container" on page 31.
21.2.0 and 22.1.0 / January 28, 2022	Added: • Content for Fortify OAST. See "Fortify WebInspect and OAST on Docker" on page 9 and "Using the OAST Docker Image" on page 26.
21.2.0	Updated: • Docker image version number.
21.1.0 / August 18, 2021	Updated: • Docker image name with current version number. See "Using the WebInspect Docker Image" on page 14 and "Pulling an Image for API and CLI Modes" on page 15. Removed: • References to the `latest` image tag.
21.1.0	Added: • Description of the ScanCentral DAST Utility Service mode. See "Understanding the Operation Modes" on page 14. Updated: • Information about requesting access to the Fortify Docker repository. See "Pulling an Image for API and CLI Modes" on page 15.

Software Release / Document Version	Changes
	- Script name for pulling sensor image for use with Fortify ScanCentral DAST, environment parameter for DAST API root URL, and DAST artifact ZIP file name. See "Running the Container in ScanCentral DAST Mode" on page 22.
20.2.0	Added: - Information for running the Docker image in Fortify ScanCentral DAST mode. See "Running the Container in ScanCentral DAST Mode" on page 22. Updated: - Links to Docker-related websites. See "Setting Up Docker" on page 10. - Description of image naming convention to remove details about version tags. See "Using the WebInspect Docker Image" on page 14. - Instructions for configuring an environment file and running the container to indicate they apply only to CLI and API modes. See "Configuring the Environment File for CLI and API Modes" on page 16 and "Running the Container in CLI and API Modes" on page 19.

Chapter 1: Fortify WebInspect and OAST on Docker

Fortify engineers have created a Fortify WebInspect image that is available for download on the Docker container platform. The image includes the full version of Fortify WebInspect 22.1.0 software, but is intended to be used in automated processes as a headless scanner configured by way of the command line interface (CLI) or the application programming interface (API). It can also be run as a Fortify ScanCentral DAST sensor and used in conjunction with Fortify Software Security Center.

Fortify OAST

Fortify engineers have created an out-of-band application security testing (OAST) server image that is available for download on the Docker container platform. The image allows you to configure local DNS service and is intended for use in networks that lack an Internet connection.

Tip: By default, Fortify WebInspect build 21.2.0.117 (or later) or Fortify WebInspect on Docker 21.2.0.118 (or later) use the Micro Focus public OAST server. For networks that have Internet access, configuring a local OAST infrastructure is not necessary.

Initially, Fortify OAST 22.1.0 detects the Log4Shell vulnerability that allows attackers to run malicious code on the affected server. The detection of other vulnerabilities related to out-of-band attacks is planned for future releases.

Important! You must use the Fortify OAST image with Fortify WebInspect build 21.2.0.117 (or later) or Fortify WebInspect on Docker 21.2.0.118 (or later). Only these versions of Fortify WebInspect support Fortify OAST.

What is Docker?

Docker is a platform that facilitates creating, deploying, and running applications. Developers can package their application and all dependencies, including the platform and all its dependencies, into one logical package called a container or image. You can download a Docker image and run the application contained therein on a virtual machine (VM).

Benefits of Docker

Using a Docker image makes configuring the various prerequisite dependencies unnecessary, and can reduce the time it takes to deploy an instance of the application.

Docker is command-line driven, so it is easy to integrate into build processes, making Docker perfect for automation. As part of an automated build process, you can download a Fortify WebInspect image from the Docker repository, conduct a scan, and then remove the image from your VM.

For more information about Docker, visit https://www.docker.com.

Supported Versions

Fortify WebInspect on Docker runs on Docker Enterprise Edition version 18.09 or later.

Fortify OAST on Docker runs on the Docker Engine on Ubuntu for Linux. For more information, see https://docs.docker.com/engine/install/ubuntu/.

Audience

This document is intended for users who are familiar with Fortify WebInspect, in particular its CLI and API, and the License and Infrastructure Manager (LIM). Users should also have experience installing, configuring, and using Docker.

Requesting Access to Fortify Docker Repository

Access to the Fortify Docker repository requires credentials and is granted through your Docker ID. To access the Fortify Docker repository, email your Docker ID to fortifydocker@microfocus.com.

Setting Up Docker

Before you can run Docker containers, you must set up Docker according to the process described in the following table.

Stage	Description
1.	Download and install Docker for Windows.
2.	Configure your machine for Docker containers.
3.	Register and start the Docker service.

For information about Docker Engine Enterprise, see https://docs.mirantis.com/docker-enterprise/v3.0/dockeree-products/docker-ee/windows.html.

For additional Docker documentation, see https://docs.docker.com/.

Related Documents

This topic describes documents that provide information about Micro Focus Fortify software products.

> **Note:** You can find the Micro Focus Fortify Product Documentation at https://www.microfocus.com/support/documentation. Most guides are available in both PDF and HTML formats. Product help is available within the Fortify LIM product and the Fortify WebInspect products.

All Products

The following documents provide general information for all products. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website.

Document / File Name	Description
About Micro Focus Fortify Product Software Documentation About_Fortify_Docs_<*version*>.pdf	This paper provides information about how to access Micro Focus Fortify product documentation. **Note:** This document is included only with the product download.
Micro Focus Fortify License and Infrastructure Manager Installation and Usage Guide LIM_Guide_<*version*>.pdf	This document describes how to install, configure, and use the Fortify License and Infrastructure Manager (LIM), which is available for installation on a local Windows server and as a container image on the Docker platform.
Micro Focus Fortify Software System Requirements Fortify_Sys_Reqs_<*version*>.pdf	This document provides the details about the environments and products supported for this version of Fortify Software.
Micro Focus Fortify Software Release Notes FortifySW_RN_<*version*>.pdf	This document provides an overview of the changes made to Fortify Software for this release and important information not included elsewhere in the product documentation.
What's New in Micro Focus Fortify Software <version> Fortify_Whats_New_<*version*>.pdf	This document describes the new features in Fortify Software products.

Micro Focus Fortify ScanCentral DAST

The following document provides information about Fortify ScanCentral DAST. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at https://www.microfocus.com/documentation/fortify-ScanCentral-DAST.

Document / File Name	**Description**
Micro Focus Fortify ScanCentral DAST Configuration and Usage Guide SC_DAST_Guide_<*version*>.pdf	This document provides information about how to configure and use Fortify ScanCentral DAST to conduct dynamic scans of Web applications.

Micro Focus Fortify WebInspect

The following documents provide information about Fortify WebInspect. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at https://www.microfocus.com/documentation/fortify-webinspect.

Document / File Name	**Description**
Micro Focus Fortify WebInspect Installation Guide WI_Install_<*version*>.pdf	This document provides an overview of Fortify WebInspect and instructions for installing Fortify WebInspect and activating the product license.
Micro Focus Fortify WebInspect User Guide WI_Guide_<*version*>.pdf	This document describes how to configure and use Fortify WebInspect to scan and analyze Web applications and Web services. **Note:** This document is a PDF version of the Fortify WebInspect help. This PDF file is provided so you can easily print multiple topics from the help information or read the help in PDF format. Because this content was originally created to be viewed as help in a web browser, some topics may not be formatted properly. Additionally, some interactive topics and linked content may not be present in this PDF version.
Micro Focus Fortify WebInspect and OAST on Docker User Guide	This document describes how to download, configure, and use Fortify WebInspect and Fortify OAST that are

Document / File Name	**Description**
WI_Docker_Guide_<*version*>.pdf	available as container images on the Docker platform. The Fortify WebInspect image is intended to be used in automated processes as a headless sensor configured by way of the command line interface (CLI) or the application programming interface (API). It can also be run as a Fortify ScanCentral DAST sensor and used in conjunction with Fortify Software Security Center. Fortify OAST is an out-of-band application security testing (OAST) server that provides DNS service for the detection of OAST vulnerabilities.
Micro Focus Fortify WebInspect Tools Guide WI_Tools_Guide_<*version*>.pdf	This document describes how to use the Fortify WebInspect diagnostic and penetration testing tools and configuration utilities packaged with Fortify WebInspect and Fortify WebInspect Enterprise.
Micro Focus Fortify WebInspect Agent Installation Guide WI_Agent_Install_<*version*>.pdf	This document describes how to install the Fortify WebInspect Agent for applications running under a supported Java Runtime Environment (JRE) on a supported application server or service and applications running under a supported .NET Framework on a supported version of IIS.
Micro Focus Fortify WebInspect Agent Rulepack Kit Guide WI_Agent_Rulepack_Guide_<*version*>.pdf	This document describes the detection capabilities of Fortify WebInspect Agent Rulepack Kit. Fortify WebInspect Agent Rulepack Kit runs atop the Fortify WebInspect Agent, allowing it to monitor your code for software security vulnerabilities as it runs. Fortify WebInspect Agent Rulepack Kit provides the runtime technology to help connect your dynamic results to your static ones.

Chapter 2: Using the WebInspect Docker Image

The following paragraphs describe the Windows versions, database version, and naming convention of the Fortify WebInspect image on Docker.

Image Naming Convention

The Fortify Docker repository uses the following naming convention for the Fortify WebInspect image:

```
fortifydocker/webinspect:<version>
```

The latest image version that is available as of this writing is:

```
fortifydocker/webinspect:22.1
```

For more information about the version that is available, refer to the Readme file in the fortifydocker/webinspect repository.

Windows Version Available

This release of Fortify WebInspect 22.1 image is available in Windows version 1809.

> **Important!** Before you can run the Fortify WebInspect image, you must install Microsoft update KB4561608 on the host machine. For more information, see https://support.microsoft.com/en-us/topic/june-9-2020-kb4561608-os-build-17763-1282-437af506-e3ef-a8a1-09e7-26cc94e509c7.

Database Version

The Fortify WebInspect 22.1 image includes the SQL Server 2017 Express edition database.

Understanding the Operation Modes

The Fortify WebInspect image can run in one of four operation modes in a container as described in the following table.

Mode	Description
1	**WebInspect CLI mode.** Use this mode to conduct scans using options available in the command-line interface. For an entire list of CLI options, see the "Command Line Execution" topic in the *Micro Focus Fortify WebInspect User Guide*.
2	**WebInspect API mode.** Use this mode to conduct scans using the endpoints available in the Fortify WebInspect REST API. After the Docker container starts, you can navigate to the following URL to browse the Swagger documentation from your local machine: http://<*hostname*>:8083/webinspect/swagger/docs/v1 If you map ports from the container to the host machine as shown in the Docker run command, you can access it using localhost as <*hostname*>. Otherwise, use the IP address of the Docker host machine.
3	**ScanCentral DAST mode.** Use this mode to conduct scans from the ScanCentral DAST user interface in Fortify Software Security Center. For more information about Fortify ScanCentral DAST, see *Micro Focus Fortify ScanCentral DAST Configuration and Usage Guide*. **Note:** This description is provided for informational purposes only. You do not configure an environment file for this mode. For more information, see "Running the Container in ScanCentral DAST Mode" on page 22.
4	**ScanCentral DAST Utility Service mode.** Use this mode to run the Fortify WebInspect image as a ScanCentral DAST Utility Service container. **Note:** This description is provided for informational purposes only. You do not configure an environment file for this mode. You pull this image and start the container using either the Docker compose file or one of the PowerShell scripts that the Fortify ScanCentral DAST Configuration Tool generates. For more information, see *Micro Focus Fortify ScanCentral DAST Configuration and Usage Guide*.

Pulling an Image for API and CLI Modes

After starting the Docker service and requesting access to the private Fortify WebInspect repository on Docker Hub, you can pull an image of Fortify WebInspect from the Fortify Docker repository as described in this topic.

Important! Before you can run the Fortify WebInspect image, you must install Microsoft update KB4561608 on the host machine. For more information, see https://support.microsoft.com/en-us/topic/june-9-2020-kb4561608-os-build-17763-1282-437af506-e3ef-a8a1-09e7-26cc94e509c7.

Note: Instructions for pulling an image apply only when running the container in API and CLI modes. To pull an image when running the container in ScanCentral DAST Mode, see "Running the Container in ScanCentral DAST Mode" on page 22.

To pull the current version of the Fortify WebInspect image:

- In PowerShell, enter the following command:

  ```
  docker pull fortifydocker/webinspect:22.1
  ```

Configuring the Environment File for CLI and API Modes

After you download a Fortify WebInspect image from the Docker repository, you must configure an environment (`.env`) file that defines how the image will operate. For more information, see https://docs.docker.com/compose/env-file.

In the environment file, configure the operation mode, licensing (if required), and options as described in the following sections.

Configuring the Operation Mode (Required)

You must specify a mode for the image. For more information about the modes, see "Understanding the Operation Modes" on page 14.

In the environment file, specify the operation mode as follows:

```
# WebInspect Container Mode

mode=<number>
```

The following example sets the image to run in WebInspect CLI mode:

```
# WebInspect Container Mode

mode=1
```

Configuring Licensing (Required for CLI and API Modes)

You must configure licensing for the image when running in CLI and API modes. Currently, licensing must be handled by a License and Infrastructure Manager (LIM). In the environment file, type the following information for your LIM installation to configure licensing for this instance of Fortify WebInspect:

```
# Licensing

limURL=<LIM_URL>

limPool=<LIM_pool>

limPswd=<LIM_password>
```

For more information about using the LIM, see the *Micro Focus Fortify License and Infrastructure Manager Installation and Usage Guide*.

Configuring CLI Mode Options

You must configure CLI options to use WebInspect CLI mode. You can configure any of the available CLI options as scan arguments in the environment file. For the complete list of CLI options, see the "Command Line Execution" topic in the *Micro Focus Fortify WebInspect User Guide*.

In the environment file, type the following to configure the CLI options to use in the scan. Substitute *<options>* with your specific options:

```
# WebInspect CLI scan options

scanArgs=<options>
```

The following example performs a crawl-only scan of zero.webappsecurity.com and exports the results to the `zero.scan` file:

```
# WebInspect CLI scan options

scanArgs=-u http://zero.webappsecurity.com -c -es zero.scan
```

Sample CLI Environment File

The following is a sample environment file for WebInspect CLI mode to run a full audit:

```
#!-- WebInspect Docker Mode. --!
#!-- Sample configuration for CLI mode. --!

# 1 = CLI mode
mode=1
```

```
# Licensing
limURL=http://xxx.xx.xxx.xxx/LIM.Service/
limPool=xxxxxx
limPswd=******

# WebInspect options - for use in scan mode
# Full audit
scanArgs=-u http://zero.webappsecurity.com -es c:\host\zero.scan

# Full audit with macro
#scanArgs=-u http://zero.webappsecurity.com -xd -es c:\host\zero.scan -
macro c:\host\zero_macro.webmacro

# Crawl only
#scanArgs=-u http://zero.webappsecurity.com -es c:\host\zero.scan -c

# Full audit with settings file and reporting
#scanArgs=-u http://zero.webappsecurity.com -s c:\host\Settings.xml -r
Vulnerability -y Standard -f c:\host\Report -gp -es c:\host\zero.scan
```

The full audit with macro, crawl only, and full audit with settings file and reporting examples are commented out in this sample file.

Configuring API Mode Options

You must configure API options to use WebInspect API mode. To conduct a scan that uses the Fortify WebInspect API, you must provide the host, port, and authentication type parameters for the API server as described in the following table.

Parameter	Description
`RCServerHost`	Specifies the hostname that the WebInspect API Server should listen on. Use `+` for all.
`RCServerPort`	Specifies the WebInspect API Server port to listen on.
`RCServerAuthType`	Specifies the WebInspect API Server authentication type. The value can be one of the following: • `None` • `Basic` • `NTLM` • `ClientCert`

In the environment file, provide the details for your Fortify WebInspect REST API using the following parameters:

```
# WebInspect API

RCServerHost=<hostname>

RCServerPort=<port_number>

RCServerAuthType=<auth_type>
```

Sample API Environment File

The following is a sample environment file for WebInspect API mode:

```
#!-- WebInspect Docker Mode. --!
#!-- Example configuration for API mode. --!

# 2 = WebInspect API mode
mode=2

# Licensing
limURL=http://xxx.xx.xxx.xxx/LIM.Service/
limPool=xxxxxx
limPswd=*****

# WebInspect API settings
RCServerHost=+
RCServerPort=8083

# RCServerAuthType: None, Basic, NTLM, ClientCert
RCServerAuthType=None
```

What's next?

After you have configured and saved your environment file, you can run the image in a container. Go to "Running the Container in CLI and API Modes" below.

Running the Container in CLI and API Modes

This topic provides a sample Docker run command for the WebInspect CLI and API modes. The Docker run command uses CLI options that define the container’s resources at runtime. To understand how the Docker CLI options used in the samples determine how the container is run, see "Understanding the Docker CLI Options" on the next page.

> **Note:** If proxy settings are required, see "Using Proxy Settings" on the next page.

Sample Docker Run Command for CLI Mode

The following example uses Docker CLI options to run the container in CLI mode:

```
docker run -d --rm -v c:/scans:c:/host --env-file ScanMode.env --memory=16g
--cpus=4 --name webinspect fortifydocker/webinspect:22.1
```

For more information about image filenames and version numbers, see "Pulling an Image for API and CLI Modes" on page 15.

Sample Docker Run Command for API Mode

The following example uses Docker CLI options to run the container in API mode:

```
docker run -d --rm -p 8083:8083 --env-file APIMode.env --memory=16g --
cpus=4 --name webinspect_api fortifydocker/webinspect:22.1
```

For more information about image filenames and version numbers, see "Pulling an Image for API and CLI Modes" on page 15.

Understanding the Docker CLI Options

The following table describes the Docker CLI options used in "Sample Docker Run Command for CLI Mode" above and "Sample Docker Run Command for API Mode" above.

Option	Description
`-d`	Runs the container in the background and displays the container ID.
`--cpus`	Specifies the number of CPUs to allocate to the container. Fortify recommends 2 CPUs.
`--env-file`	Identifies the `.env` file to use. For more information, see "Configuring the Environment File for CLI and API Modes" on page 16.
`--memory`	Specifies the amount of memory to allocate to the container. Fortify recommends 16 GB.
`-p`	Maps a port inside the container to a port on the host system. **Important!** This option is required when using WebInspect API mode.

Option	Description
`--rm`	Automatically removes the container when it exits.
`-v`	Maps the volume (or folder) from the container to a folder on the host system. Separate multiple folder names with a colon.

Tip: For more information and a complete list of Docker run options, see https://docs.docker.com/engine/reference/commandline/run.

Using Proxy Settings

You cannot pass proxy settings directly to the WebInspect image through command line arguments or in the .env file. However, you can use the following process to use proxy settings for a scan.

Stage	Description
1.	Create a custom WebInspect settings file that includes the proxy settings.
2.	Save the file on the Docker host machine.

3.	Use the following options:
	- The `-s` WebInspect CLI option as a scan argument (`scanArgs`) in the .env file to pass the settings file, as shown in the following example:
	`scanArgs=-u http://zero.webappsecurity.com/`
	`   -s c:\host\CustomSettings.xml -es c:\host\zero.scan -xd`
	- The `-v` Docker CLI option in the Docker run command to map the folder with the settings to a folder in the container, as shown in the following example:
	`docker run -v c:/widocker:c:/host --env-file config.env`
	`   fortifydocker/webinspect`

Running the Container in ScanCentral DAST Mode

The ScanCentral DAST Configuration Tool creates and downloads the following PowerShell scripts that you can use to pull and start a new sensor container:

- `pull-and-start-sensor-container.ps1` - This PowerShell script pulls the Fortify WebInspect image from Docker Hub, and then starts the container.
- `pull-sensor-image.ps1` - This PowerShell script pulls the Fortify WebInspect image from Docker Hub, but does not start the container.
- `start-sensor-container.ps1` - This PowerShell script starts the Fortify WebInspect container, but does not pull the image.

You can find these files in the `DAST-start.zip` file along with the other ScanCentral DAST launch artifacts. For more information about the ScanCentral DAST Configuration Tool, see the *Micro Focus Fortify ScanCentral DAST Configuration and Usage Guide*.

Sample Script

The script you use should be similar to the following example:

```
docker run -d --restart always --name scancentral-dast-sensor --hostname
   "<MyHostName>"
   -e "mode=3" -e "RCServerHost=+" -e "RCServerPort=8089"
   -e "RCServerUseHTTPS=false" -e "RCServerAuthType=none"
   -e "DASTApiRootUrl=http://<IP_Address>:<Port>/api"
   -e "AllowNonTrustedServerCertificate=true"
   -e "ServiceToken=QgitxRErVP5Eh7hr2Bnuig=="
   -e "ScannerDescription=<MyDASTSensorName>"
   -e "ScannerPoolId=0" --memory=8g --cpus=2 fortifydocker/webinspect:22.1
```

About the ServiceToken

The `ServiceToken` is the encrypted "Sensor Service Token" that is set by the administrator using the ScanCentral DAST Configuration tool. This value should be protected.

> **Caution!** A change to the `ServiceToken` value by the configuration tool requires all sensor containers to be updated with the new value.

About the ScannerDescription

The `ScannerDescription` allows you to provide a description for the sensor service. This value is displayed in the Description column of the Sensors list table on the ScanCentral DAST tab in Fortify

Software Security Center. The description is shown in the sample script as an environment variable, but you can also provide it in the `appsettings.json` file.

About the ScannerPoolId

The `ScannerPoolId` is the sensor pool ID number in Fortify Software Security Center. If you need to hand-edit the `ScannerPoolId` in your script, you can find the sensor pool ID number in Fortify Software Security Center on the **SCANCENTRAL** > **DAST** > **Sensor Pools** page. Select the sensor pool in the list and view the Pool ID in the detail panel.

> **Tip:** Setting `ScannerPoolId` to `0` automatically allocates the sensor to the Default pool.

Using PowerShell Scripts

These PowerShell scripts offer the following options:

- Use one script to pull the Fortify WebInspect image and then start the container.
- Use two scripts: one to pull the image, and then another to start the container.

You use the script or scripts on the host where you want to run the Fortify WebInspect container.

Using One Script

Use the following process to use a single PowerShell script to pull the image and start the container.

Stage	Description
1.	Copy the `pull-and-start-sensor-container.ps1` file to the host where you want to run the Fortify WebInspect container.
2.	On this same host, start Windows PowerShell ISE as Administrator. For more information about using PowerShell, refer to your Windows PowerShell documentation.
3.	To avoid errors regarding non-digitally signed scripts, run the contents of the `pull-and-start-sensor-container.ps1` script: 1. Copy the contents from the `pull-and-start-sensor-container.ps1` script. 2. Paste the contents in the PowerShell ISE script pane. 3. Click the **Run Selection** icon. **Note:** Alternatively, you can set the execution policy to allow all scripts, and then run the script as follows: `& "<drive>:<path_to_script>\pull-and-start-sensor-container.ps1"`

Stage	Description
	For more information about setting the execution policy, refer to your Windows PowerShell documentation. The Fortify WebInspect image is pulled and the container is started.

Using Two Scripts

Use the following process to use separate pull and start PowerShell scripts.

Stage	Description
1.	Copy the following files to the host where you want to run the Fortify WebInspect container: • `pull-sensor-image.ps1` • `start-sensor-container.ps1`
2.	On this same host, start Windows PowerShell ISE as Administrator. For more information about using PowerShell, refer to your Windows PowerShell documentation.
3.	Pull the image. To avoid errors regarding non-digitally signed scripts, run the contents of the `pull-sensor-image.ps1` script: 1. Copy the contents from the `pull-sensor-image.ps1` script. 2. Paste the contents in the PowerShell ISE script pane. 3. Click the **Run Selection** icon. **Note:** Alternatively, you can set the execution policy to allow all scripts, and then run the script as follows: `& "`*`<drive>`*`:`*`<path_to_script>`*`\pull-sensor-image.ps1"` For more information about setting the execution policy, refer to your Windows PowerShell documentation. The Fortify WebInspect image is pulled.
4.	Start the container. To avoid errors regarding non-digitally signed scripts, run the contents of the `start-sensor-container.ps1` script:

<table>
<tr><th>Stage</th><th>Description</th></tr>
<tr><td></td><td>
<ol>
<li>Copy the contents from the <code>start-sensor-container.ps1</code> script.</li>
<li>Paste the contents in the PowerShell ISE script pane.</li>
<li>Click the <b>Run Selection</b> icon.</li>
</ol>
<b>Note:</b> Alternatively, if you set the execution policy to allow all scripts as described in Stage 3, you can run the script as follows:
<code>& "<i><drive></i>:<i><path_to_script></i>\start-sensor-container.ps1"</code>
The Fortify WebInspect container is started.
</td></tr>
</table>

Chapter 3: Using the OAST Docker Image

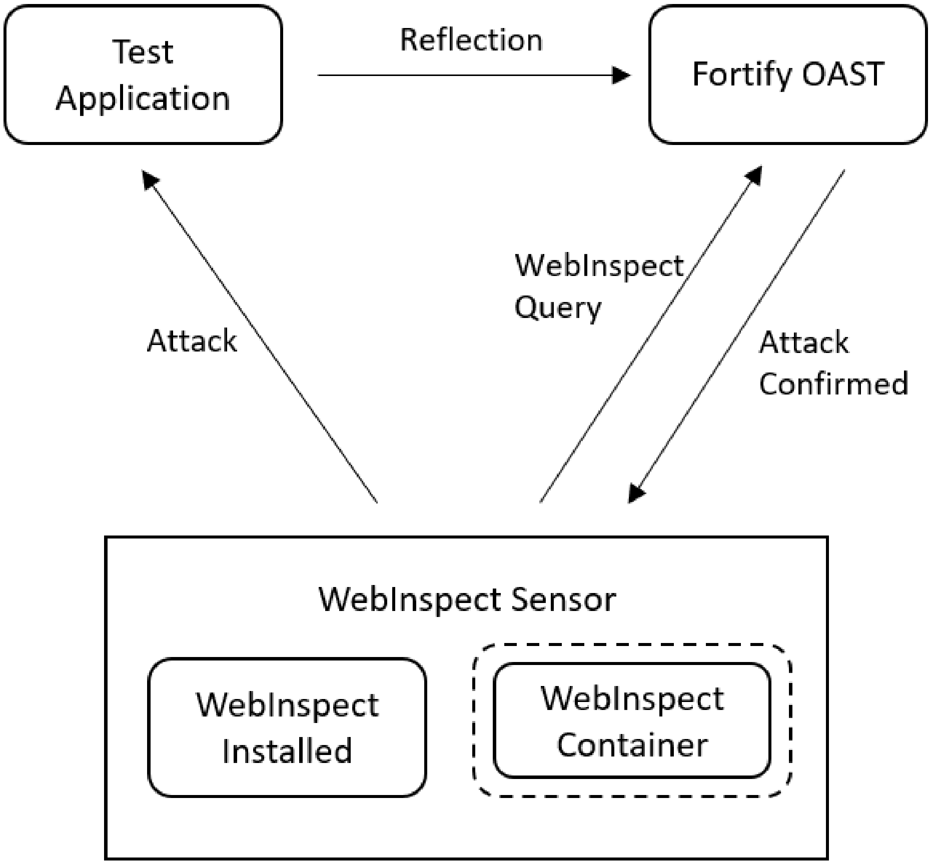
The following paragraphs describe how the Fortify OAST on Docker image works and how to configure and run it in a container.

How Fortify OAST Works

OAST vulnerabilities do not reflect back to Fortify WebInspect, making them difficult to detect with traditional DAST scanning. The Fortify OAST server provides DNS service for the detection of out-of-band attack vulnerabilities. You configure and use the server with a desktop version of Fortify WebInspect or with WebInspect on Docker.

With the Log4Shell vulnerability, if WebInspect is able to detect the vulnerability, then the application server under test will send a DNS lookup to the Fortify OAST server. Fortify WebInspect will then query the Fortify OAST server to determine whether it received the DNS lookup. If the Fortify OAST server received it, then the application server is susceptible to the vulnerability.

The following diagram illustrates how Fortify WebInspect works with Fortify OAST during a scan to detect the Log4Shell vulnerability.



Understanding the Configuration Process

The following table describes the process of configuring and using Fortify OAST in conjunction with a Fortify WebInspect scan.

Stage	Description
1.	Prepare a Linux VM machine with Ubuntu 20.04, 18.04, 16.04 LTS x64. This machine will be the host for the Fortify OAST image.
2.	Install Docker Engine on Ubuntu for Linux on the Linux VM machine. For more information, see https://docs.docker.com/engine/install/ubuntu/.
3.	Pull the Fortify OAST Docker image. See "Pulling the Fortify OAST Image" on the next page.
4.	Configure settings on the Ubuntu Linux Docker host machine to disable the embedded DNS server and use a manual DNS configuration. See " Configuring the Ubuntu Linux Docker Host Machine" on the next page.
5.	Run the Fortify OAST container. See "Running the OAST Container" on page 31.
6.	Do one of the following: • Configure Fortify WebInspect to use the Fortify OAST server. See "Configuring Fortify WebInspect for OAST" on page 32. • Pass environment variables in the Docker run command to start your Fortify WebInspect container and use the Fortify OAST server. See "Running Fortify WebInspect on Docker with Fortify OAST" on page 34.
7.	Do one of the following: • Configure the target web application to use the Fortify OAST server. See "Configuring the Target Application for OAST " on page 35. • Run the target application container with the Fortify OAST server. See "Running the Target Application in Docker with OAST" on page 37.

About the OAST Image

The Fortify OAST image runs on a Linux VM Machine. It provides DNS service for the detection of OAST vulnerabilities, and it is intended for use in networks that lack an Internet connection.

Image Naming Convention

The Fortify Docker repository uses the following naming convention for the Fortify OAST image:

```
fortifydocker/fortify-oast:<version.linux_os_version>
```

The latest image version that is available as of this writing is:

```
fortifydocker/fortify-oast:22.1.alpine.3.14.3
```

Note: The image version includes the Alpine Linux operating system build number that is used in the image.

For more information about the version that is available, refer to the Readme file in the fortifydocker/fortify-oast repository.

Pulling the Fortify OAST Image

After installing the Docker Engine on Ubuntu for Linux, starting the Docker service, and requesting access to the private FortifyOAST repository on Docker Hub, you can pull an image of Fortify OAST from the Fortify Docker repository as described in this topic.

To pull the current version of the Fortify OAST image:

- At the terminal prompt on the Ubuntu Linux Docker host machine, enter the following command:

  ```
  docker login
  docker pull fortifydocker/fortify-oast:22.1.alpine.3.14.3
  docker logout
  ```

Configuring the Ubuntu Linux Docker Host Machine

You must determine if the ports needed by the Fortify OAST server are currently in use. If they are, you must edit the default settings in the `resolved.conf` file on the Ubuntu Linux Docker host machine to disable the embedded DNS server and use a manual DNS configuration. Afterward, you must reboot the host machine.

Checking Port Usage

The Fortify OAST server requires the following ports:

- 443/TCP
- 53/TCP
- 53/UDP

By default, the Ubuntu OS allocates its local DNS resolver to ports 53/TCP and 53/UDP. However, port 443 is not allocated.

To check whether these required ports are used on the Ubuntu Linux Docker host machine:

- At the terminal prompt on the host machine, enter the following command:

```
netstat -antu
```

If port 443 is in use, either find the server that is using it and free the port or use an Ubuntu OS with default settings. If ports 53/TCP and 53/UDP are in use, you can free them as described in "Editing the Configuration File to Free Ports" below.

Editing the Configuration File to Free Ports

Ubuntu 20.04 may allocate 53/TCP and 53/UDP ports by default for the `systemd-resolved` system service that provides network name resolution on the local DNS server. You can reconfigure them in the `resolved.conf` file.

To edit the configuration file:

1. At the terminal prompt on the Ubuntu Linux Docker host machine, enter the following command:

   ```
   sudo nano /etc/systemd/resolved.conf
   ```

   The following example shows the `resolved.conf` file contents.

   ```
   [Resolve]
   #DNS=
   #FallbackDNS=
   #Domains=
   #LLMNR=no
   #MulticastDNS=no
   #DNSSEC=no
   #Cache=yes
   #DNSStubListener=yes
   ```

2. Remove the number sign (#) from the front of the line for `DNS`.

3. After `DNS=`, enter the IP address for your working primary local network DNS server.
4. Remove the number sign (#) from the front of the line for `DNSStubListener`.
5. Change the `DNSStubListener` setting to `no`.

   The updated `resolved.conf` file should resemble the following example.

   ```
   [Resolve]
   DNS=<ip_address>
   #FallbackDNS=
   #Domains=
   #LLMNR=no
   #MulticastDNS=no
   #DNSSEC=no
   #Cache=yes
   DNSStubListener=no
   ```

6. Save your changes.

Creating a Symbolic Link

At this point, the Ubuntu embedded DNS server is disabled. You must configure Ubuntu Linux to use manual DNS configuration to resolve hostnames. For manual DNS configuration, Linux reads settings from `/etc/resolv.conf`. Therefore, you must create a symbolic link to this file from the `systemd` service that provides network name resolution to local applications.

To create the symbolic link:

- At the terminal prompt on the Ubuntu Linux Docker host machine, enter the following command:

  ```
  sudo ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf
  ```

The following table describes the options used in the command.

Option	Description
`-s`	Creates a symbolic link instead of a hard link.
`-f`	Removes existing files from the destination directory.

Rebooting the Ubuntu Linux Docker Host Machine

After configuring the changes, you must reboot the Ubuntu Linux Docker host machine. Refer to your Ubuntu documentation for details.

Running the OAST Container

After your DNS configurations are complete and the host machine has been rebooted, you can run the OAST container.

To run the container:

- At the terminal prompt on the Ubuntu Linux Docker host machine, enter the following commands:

```
mkdir -p "<host_path>/certs"
docker run -d \
    --name <string> \
    --restart unless-stopped \
    -p 443:443 \
    -p 0.0.0.0:53:53/tcp \
    -p 0.0.0.0:53:53/udp \
    -e "WIH_DOMAIN=<domain_name>" \
    -e "WIH_IPv4_PUBLICIP=<ip_address>" \
    -v "<host_path>/certs:/etc/wihorizon/certs" \
    --log-opt max-size=20m \
    --log-opt max-file=5 \
    fortifydocker/fortify-oast:22.1.alpine.3.14.3
```

Tip: The backslash (\) indicates the end of line for the Linux OS.

Understanding the Run Command Options

The following table describes the options used in the run command.

Option	Description
`-d`	Runs the container in the background and prints the container ID.
`--name`	Specifies the name of your Fortify OAST container. Any string is valid. Examples in this table use `wihorizon`.
`--restart unless-stopped`	Restarts the container unless the container is manually stopped.
`-p 443:443`	Publishes the container's main TCP ingress port to the host.

Option	Description
`-p 0.0.0.0:53:53/udp`	Publishes the container's UDP DNS server port to the host.
`-p 0.0.0.0:53:53/tcp`	Publishes the container's TCP DNS server port to the host.
`-e "WIH_DOMAIN=<domain_name>"`	Configures the local domain name. For example: `-e "WIH_DOMAIN=local-fortify-oast.net"`
`-e "WIH_IPv4_PUBLICIP=<ip_address>"`	Configures the local IP address for the Ubuntu Linux Docker host machine that is exposed to the Fortify WebInspect sensor.
`-v "<host_path>/certs:/etc/wihorizon/certs"`	Adds a volume for a Fortify OAST auto-generated certificates directory. This directory safeguards the certificates in case the Fortify OAST container needs to be removed or upgraded. For example: `-v "$HOME/.wihorizon/certs:/etc/wihorizon/certs" \`
`--log-opt max-size=20m`	Limits the Docker log file size to the specified number of megabytes. This setting prevents log files from consuming too much disc space.
`--log-opt max-file=5`	Limits the number of Docker log files to the specified number. When the number is reached, Docker removes the oldest log file and starts a new one.

Configuring Fortify WebInspect for OAST

You can use the Fortify OAST server with a classic Fortify WebInspect installation or with the Fortify WebInspect on Docker image. This topic describes the required configuration changes to support the Fortify OAST server with a classic installation. For information using Fortify OAST with the Fortify WebInspect container, see "Running Fortify WebInspect on Docker with Fortify OAST" on page 34.

Configuring Access to the Fortify OAST Server

You must configure either your network or your sensor to provide access to the Fortify OAST server.

To configure access, do one of the following:

- Add the domain name that you configured for the `WIH_DOMAIN` option to your local DNS server.
- Edit the host file on the Fortify WebInspect machine to point to the Docker host IP address that you configured for `WIH_PUBLICIP` option.

For more information, see "Running the OAST Container" on page 31.

Verifying Access to the Fortify OAST Server

Verify that the Fortify OAST server works on the Fortify WebInspect machine.

To verify access:

- In PowerShell on the sensor machine, enter the following command:

  ```
  nslookup 00000000-0000-0000-0000-000000000000.<WIH_DOMAIN> <WIH_DOMAIN>
  ```

  Using the example from "Running the OAST Container" on page 31, the command would be as follows:

  ```
  nslookup 00000000-0000-0000-0000-000000000000.local-fortify-oast.net
  local-fortify-oast.net
  ```

  If the Fortify OAST server works, you should see `127.0.0.1` as the resolved address, as shown in the following example:

  ```
  Server:         local-fortify-oast.net
  Address:        <WIH_IPv4_PUBLICIP>#53

  Name:   00000000-0000-0000-0000-000000000000.local-fortify-oast.net
  Address: 127.0.0.1
  ```

Verify the Fortify OAST Docker Logs

Verify that the Fortify OAST server is logging its connection to the sensor in the Docker container log file.

To verify log files:

- At the terminal prompt on the Ubuntu Linux Docker host machine, enter the following commands:

  ```
  docker logs <fortify_oast_container_name>
  ```

  Using the example from "Running the OAST Container" on page 31, the command would be as follows:

  ```
  docker logs wihorizon
  ```

You should see output similar to the following:

```
0/00, 00:00:00  00 | [DNS] Detected correlation GUID 00000000-0000-0000-
0000-000000000000 p0
```

Configure Fortify WebInspect to Use the Local Domain

You must configure the sensor to use the local domain name that you configured for the `WIH_DOMAIN` option.

To use the local domain:

1. Close all Fortify WebInspect instances.
2. On the sensor machine, open the command line prompt and navigate to the Fortify WebInspect installation directory.

   > **Tip:** By default, the installation directory is `C:\Program Files\Fortify\Fortify WebInspect\`.

3. At the command prompt, enter the following command:

   ```
   WIConfig.exe -WIOASTServerAddress "<WIH_DOMAIN>"
   ```

   Using the example from "Running the OAST Container" on page 31, the command would be as follows:

   ```
   WIConfig.exe -WIOASTServerAddress "local-fortify-oast.net"
   ```

Running Fortify WebInspect on Docker with Fortify OAST

You can pass environment variables in the Docker run command to start your Fortify WebInspect to use the Fortify OAST server that you configured.

To start the sensor with Fortify OAST:

- In PowerShell on the sensor machine, enter the following command:

```
docker run -d `
       --name <container_name> `
       -p 8089:8089 `
       -e 'mode=<number>' `
       -e 'limURL=http://<ip_address>/LIM.Service/' `
       -e 'limPool=<string>' `
```

```
        -e 'limPswd=<string>' `
        -e 'RCServerHost=+' `
        -e 'RCServerPort=<port_number>' `
        -e 'RCServerAuthType=None' `
        -e 'WIOASTServerAddress=<WIH_DOMAIN>' `
fortifydocker/webinspect:22.1
docker exec webinspect cmd /c "echo <WIH_IPv4_ADDRESS> <WIH_DOMAIN> >>
C:\Windows\System32\drivers\etc\hosts"
```

Understanding the Run Command Options

The following table describes the options used in the run command.

Option	**Description**
`-d`	Runs the container in the background and prints the container ID.
`--name`	Specifies the name of your Fortify WebInspect container. Any string is valid. Examples in this table use `webinspect`.
`-p 8089:8089`	Publishes the Fortify WebInspect REST API port to the host.
`mode`	Specifies the operation mode for the container. For more information, see "Understanding the Operation Modes" on page 14.
`limURL`, `limPool`, `limPswd`	Configures licensing. For more information, see "Configuring Licensing (Required for CLI and API Modes) " on page 17.
`RCServerHost`, `RCServerPort`, `RCServerAuthType`	Configures access to the Fortify WebInspect API server. For information about the API server options, see "Configuring API Mode Options" on page 18.
`WIOASTServerAddress`	Configures Fortify WebInspect to use the local Fortify OAST server. Using the example from "Running the OAST Container" on page 31, the command would be as follows: ``-e 'WIOASTServerAddress=local-fortify-oast.net' ` ``

Configuring the Target Application for OAST

You must configure the target web application to use the Fortify OAST server for DNS lookup requests from Fortify WebInspect or run the target application container with Fortify OAST. This topic describes the required changes to the target application. For information about running the target application container, see "Running the Target Application in Docker with OAST" on the next page.

Adding the Local Domain Server

Add the local domain name that you configured for the `WIH_DOMAIN` option to the web application network as the primary DNS server or add it as a primary DNS server for the machine that hosts the target web application. In a Linux OS, for example, you can edit the `/etc/resolv.conf` file by adding the Fortify OAST server as the primary DNS server and using the real network DNS server as secondary:

```
nameserver <WIH_IPv4_ADDRESS>
nameserver <dns_server_ip_address>
```

Verifying Application Access to the Fortify OAST Server

Verify that the Fortify OAST server works for the target web application machine.

To verify access:

- At the terminal prompt on the web application machine, enter the following command:

  ```
  nslookup 00000000-0000-0000-0000-000000000000.<WIH_DOMAIN>
  ```

  Using the example from "Running the OAST Container" on page 31, the command would be as follows:

  ```
  nslookup 00000000-0000-0000-0000-000000000000.local-fortify-oast.net
  ```

  If the Fortify OAST server works, you should see `127.0.0.1` as the resolved address, as shown in the following example:

  ```
  Server:         local-fortify-oast.net
  Address:        <WIH_IPv4_PUBLICIP>#53

  Name:   00000000-0000-0000-0000-000000000000.local-fortify-oast.net
  Address: 127.0.0.1
  ```

Verify the Fortify OAST Docker Logs

Verify that the Fortify OAST server is logging its connection to the target web application in the Docker container log file.

To verify log files:

- At the terminal prompt on the Ubuntu Linux Docker host machine, enter the following command:

  ```
  docker logs <fortify_oast_container_name>
  ```

  Using the example from "Running the OAST Container" on page 31, the command would be as follows:

  ```
  docker logs wihorizon
  ```

  You should see output similar to the following:

  ```
  0/00, 00:00:00  00 | [DNS] Detected correlation GUID 00000000-0000-0000-
  0000-000000000000 p0
  ```

Running the Target Application in Docker with OAST

If the target web application resides in a Docker image, you can run the target application container with Fortify OAST.

To start the application with Fortify OAST:

- At the terminal prompt on the web application container, enter the following commands:

  ```
       docker run -d \
       --name <container_name> \
       --restart unless-stopped \
       -p <port>:<port> \
       --dns <WIH_IPv4_ADDRESS> \
       --dns <ip_address> \
       --dns <ip_address> \
       --log-opt max-size=20m \
       --log-opt max-file=5 \
  <docker_repo>/<application_name>:latest
  ```

Understanding the Run Command Options

The following table describes the options used in the run command.

Option	Description
`-d`	Runs the container in the background and prints the container ID.

Option	Description
`--name`	Specifies the name of your test application container.
`--restart unless-stopped`	Restarts the container unless the container is manually stopped.
`-p <port>:<port>`	Publishes the container's port to the host.
`--dns`	Indicates the various DNS servers to use. The first entry adds the Fortify OAST server as the primary DNS server. Each of the following `--dns` entries are real network DNS servers that respond to all regular nameserver queries so that the container environment can perform all required nslookups.
`--log-opt max-size=20m`	Limits the Docker log file size to the specified number of megabytes. This setting prevents log files from consuming too much disc space.
`--log-opt max-file=5`	Limits the number of Docker log files to the specified number. When the number is reached, Docker removes the oldest log file and starts a new one.

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email.

> **Note:** If you are experiencing a technical issue with our product, do not email the documentation team. Instead, contact Micro Focus Fortify Customer Support at https://www.microfocus.com/support so they can assist you.

If an email client is configured on this computer, click the link above to contact the documentation team and an email window opens with the following information in the subject line:

Feedback on User Guide (Fortify WebInspect and OAST on Docker 22.1.0)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to fortifydocteam@microfocus.com.

We appreciate your feedback!