

Strengthen Operational Security with IBM WebSphere Automation



Owner: Lars Besselmann, IBM

Current document version: 3.0

Used software: IBM WebSphere Automation 1.3

Last updated: May 2022

Duration: 45 mins

1. Contents

Strengthen Operational Security with IBM WebSphere Automation	1
1. Contents.....	2
2. Introduction to WebSphere Automation.....	3
3. Business Context.....	4
4. Accessing and starting the environment	6
Complete the setup	7
5. Accessing the WebSphere Automation UI.....	9
6. Getting the WSA configuration parameters.....	12
7. Configuring Liberty server.....	14
8. Configuring traditional WebSphere (tWAS) v8.5.5	17
9. Get insight from WebSphere Automation.....	19
10. Update the Liberty server configuration	23
11. Update tWAS server to fix a vulnerability.....	26
12. Remove the Liberty fix to re-introduce a vulnerability.....	28
13. Summary.....	30
14. Appendix.....	31
Lab_WSAcommands.txt.....	31



Note: To ease the copy and paste, the commands used in the lab have been documented in the file
https://larsbesselmannibm.github.io/labs/WSA/lab_WSAcommands.txt

2. Introduction to WebSphere Automation

[IBM WebSphere Automation](#) is focused on delivering value into existing WebSphere Application Server (WAS) environments, helping administrators reduce the cost, effort, and risk of addressing common vulnerabilities, automating tasks, and remediating capacity incidents.

It removes manual toil so that your team can spend more time innovating while minimizing the cost of extending the life and maximizing the ROI of your WebSphere investments.

In conversations with customers, the same three concerns come up repeatedly. Organizations need to keep their IT estate secure and compliant, resilient to disruption and running optimally while reducing costs and maximizing ROI.

WebSphere Automation helps organizations gain visibility, operational efficiencies, and cost savings quickly by extending the life of WebSphere investments and giving teams time back to focus on unlocking new value and fixing the imbalance of pure maintenance versus innovation tasks.

- WebSphere operators and administrators save time and embrace DevSecOps by implementing patches more efficiently on virtual and container environments to keep operations compliant and secure.
- Enhance remediation capabilities with insights and recommendations to improve the speed and depth of understanding of outages and anomalies as they occur.
- Augment the operational experience with access to simplified and consolidated information that enables teams to act.

With WebSphere Automation, security, business efficiency and resiliency become standard. IBM can meet you wherever you are in your optimization and automation journeys to help you quickly deliver value and increase ROI, all while laying a solid automation foundation to support future growth.

IBM WebSphere Automation is available as a stand-alone offering or as an addition to IBM Cloud Pak® for Watson AIOps. As part of IBM Automation platform, IBM WebSphere Automation includes containerized components and common software services on top of a common automation layer, to manage WebSphere's incidents, hybrid applications, and cost with complete observability, governance, and compliance.

Deploy virtually anywhere through containers supported by Red Hat® OpenShift® software, on IBM Cloud®, on essentially any existing infrastructure on-premises, or through private and public clouds. Use only the capabilities you need with a fully modular approach that's designed to be easy to consume.

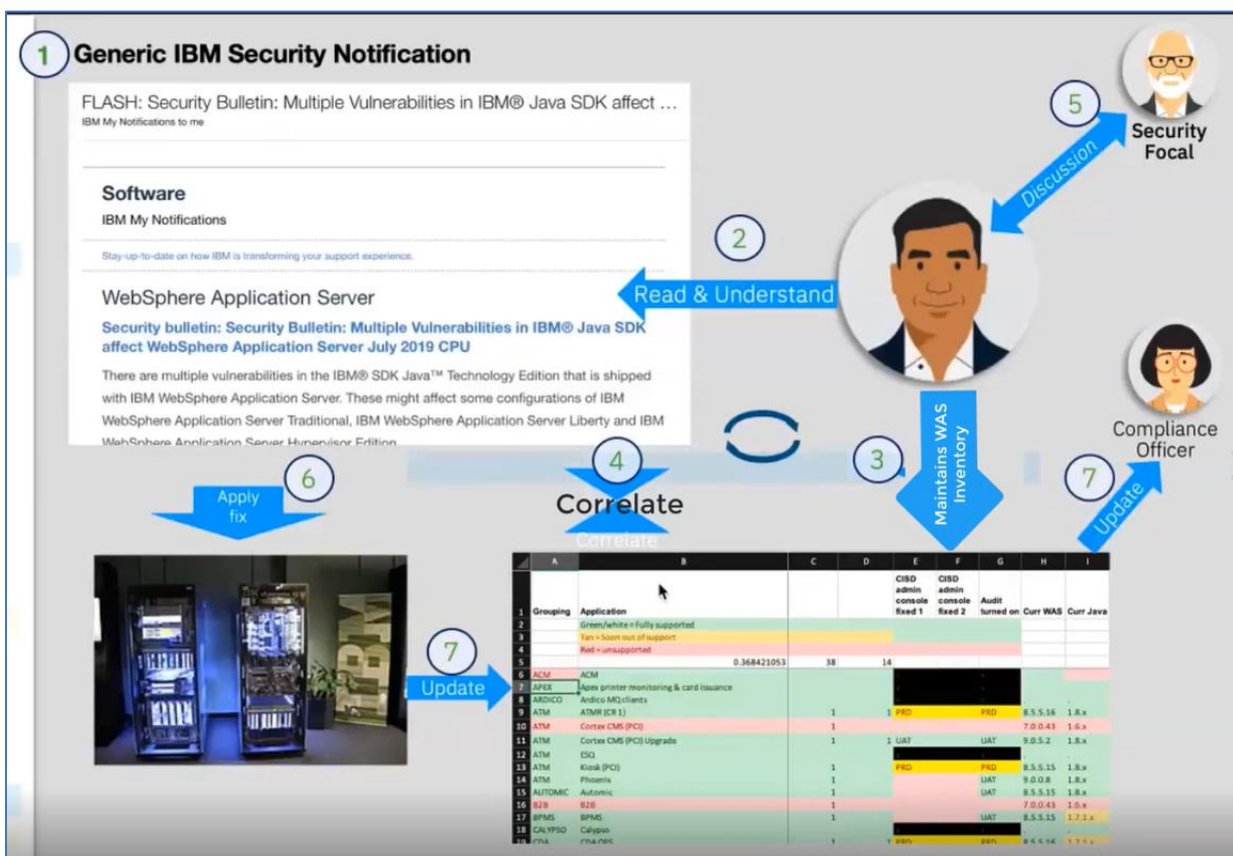
3. Business Context

You are a WebSphere Administrator, part of a WebSphere Operations Team responsible for maintaining security compliance of the WebSphere estate in the enterprise. A typical “as-is” process for maintaining security compliance for WebSphere environments is depicted below.

Today (as-is):

1. IBM sends generic “FLASH” to indicate a new WAS security bulletin.
2. You subscribe and receive IBM Security Bulletins to be aware about vulnerabilities, its potential impact, severity, and recommended solutions.
3. Generally, WAS inventory is maintained in spreadsheets.
4. Based on that, you check if this CVE applies to your managed inventory (Spreadsheet)
5. You determine if an APAR / Fix Pack upgrade should be applied to existing environment
6. You deploy the fix to the impacted environments
7. You update the WAS inventory (Spreadsheet) and provide up-to-date reports to audit and compliance teams

As is, your inventory is a spreadsheet, containing all information about your servers, such as the versions of the installed servers, which operating system they're installed on, and iFixes which have been applied, etc.



Currently, this is a very manual, time-consuming process, and you'd like to automate this process to direct valuable time and resource elsewhere. This is where **IBM WebSphere Automation will help!**

You would like to have:

- **Management dashboard:** Consolidated dashboard increases awareness and response time to common vulnerabilities and exposures (CVEs).

- **Automated vulnerability tracking:** Let WebSphere Automation track new security bulletins across your existing traditional WebSphere and Liberty environments, on virtual machines or containers.
- **Contextual notifications:** Receive security bulletin notifications only when new vulnerabilities affect the environment you manage, reducing noise and interruptions to the WebSphere operations team.
- **Shared, live visibility to key stakeholders:** WebSphere operators and security compliance teams can see the real-time security posture of the WebSphere estate, accelerating action and minimizing the risk of miscommunication.

In this lab, you use the IBM WebSphere Automation to secure operations to reduce risk and meet compliance.

At the end of this lab, you will be able to connect teams with the most relevant information through a single dashboard. This enables you to discover, analyze and remediate common vulnerabilities and exposures across instances. Furthermore, this information can be exported to a CSV file to be shared amongst the broader team.

IBM Automation

Application runtimes

Security

Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.

Servers

CVEs

Filter by

Cell

CVE

WebSphere version

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version
High (8.8)	twasServer1	CVE-2021-26296(+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18
None (0.0)	oomServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12
None (0.0)	libertyServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12

4. Accessing and starting the environment

The environment consists of two instances:

- A workstation which is a RHEL VM dedicated to one user. It is called **Student VM** in the lab instructions and contains WebSphere Application Server Traditional and Liberty installations.
- A WebSphere Automation instance which is a shared environment. It is called **WSA environment** in the lab instructions.

You get access to the WSA environment via the Student VM.

1. Access the Student VM

- a. Use the connection details that have been provided to you.
- b. If you are connected via VNC, use the URL that has been provided to you, for example:
<https://iccve.uk.ibm.com/cloudhur2>.

2. Login to the Student VM.

- a. If you are connected via VNC, you should be automatically logged in as ibmdemo.

Otherwise log in as user “ibmdemo” and enter “passw0rd” as the password:

Password: `passw0rd` (lowercase with a zero instead of the o)

For your convenience, there are several scripts that ease the administration.



Note: To ease the copy and paste, the commands used in the lab have been stored into the file `lab_WSAcommands.txt`.

The file is accessible via browser at

https://larsbesselmannibm.github.io/labs/WSA/lab_WSAcommands.txt

If you want to copy it to your local system, use the following command to copy it to your desktop:

```
curl https://larsbesselmannibm.github.io/labs/WSA/lab_WSAcommands.txt >
~/Desktop/lab_WSAcommands.txt
```

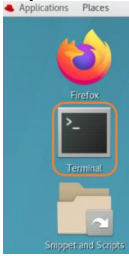
Complete the setup

Set the address for the WSA target

As this is a shared environment, the WebSphere Automation instance is rebuilt from time to time with a new IP address. Therefore you might have to adjust the hostname file.

If you are provided a new IP address and are asked to change the address, these are the steps to do so.

1. Open a terminal window by clicking its icon from the Desktop toolbar.



2. Run the following command with the IP address provided by the instructor.
(This is only needed if a remote WSA instance is used and has been updated.)

```
sudo sed -i 's/10.99.99.23/10.139.195.190/g' /etc/hosts
cat /etc/hosts
```

Enter passw0rd, when prompted.

```
[ibmdemo@RHEL7Guac ~]$ sudo sed -i 's/10.99.99.23/10.139.195.190/g' /etc/hosts
[ibmdemo@RHEL7Guac ~]$ cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4 ibmdemo-db2 ibmdemo-was ibmdemo-wasxy
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.139.195.190 cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com cp-console.apps.ocp46.tec.uk.ibm.com api.apps.ocp46.
tec.uk.ibm.com oauth-openshift.apps.ocp46.tec.uk.ibm.com
10.99.99.36 instanabackend instanabackend.tec.uk.ibm.com
10.99.99.37 rhel7was1
```

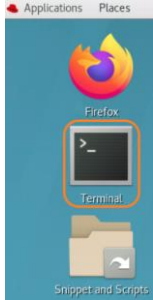
Set the hostname for the Student VM

As all Student VMs are based on the same image, all VMs use the same hostname and internal IPs. WebSphere Automation does the registration based on IP address and hostname, so all VMs would result into the same WSA registration. To avoid that, there are two options:

- Use user-specific names for the traditional WAS and Liberty server instances
- Adjust the hostname to make it user-specific

In the lab, we will use the second approach and adjust the hostname.

1. Open a terminal window by clicking its icon from the Desktop toolbar.



2. Set your UserID

Run the following command with the student number provided by the instructor.

```
# Set your UserID
export myUserID=<student number>
```

3. Run the following command with the student number provided by the instructor.

```
export newHost="ibmdemo-was"$myUserID
echo "192.168.1.100 $newHost" | sudo tee -a /etc/hosts
sudo hostname $newHost
```

Enter passw0rd, when prompted.

```
[ibmdemo@RHEL7Guac ~]$ # Set your UserID
[ibmdemo@RHEL7Guac ~]$ export myUserID=01
[ibmdemo@RHEL7Guac ~]$ # Set the hostname for the Student VM
[ibmdemo@RHEL7Guac ~]$ export newHost="ibmdemo-was"$myUserID
[ibmdemo@RHEL7Guac ~]$ echo "192.168.1.100 $newHost" | sudo tee -a /etc/hosts
192.168.1.100 ibmdemo-was01
[ibmdemo@RHEL7Guac ~]$ sudo hostname $newHost
```


5. Accessing the WebSphere Automation UI

A WebSphere administrator sets up WebSphere Automation by registering and configuring WebSphere Application Servers and WebSphere Liberty servers for vulnerability tracking and by configuring email notifications.

WebSphere administrators can also view the results of vulnerability assessment in WebSphere Automation to plan their response for the WebSphere Application Server and WebSphere Liberty servers that they manage.

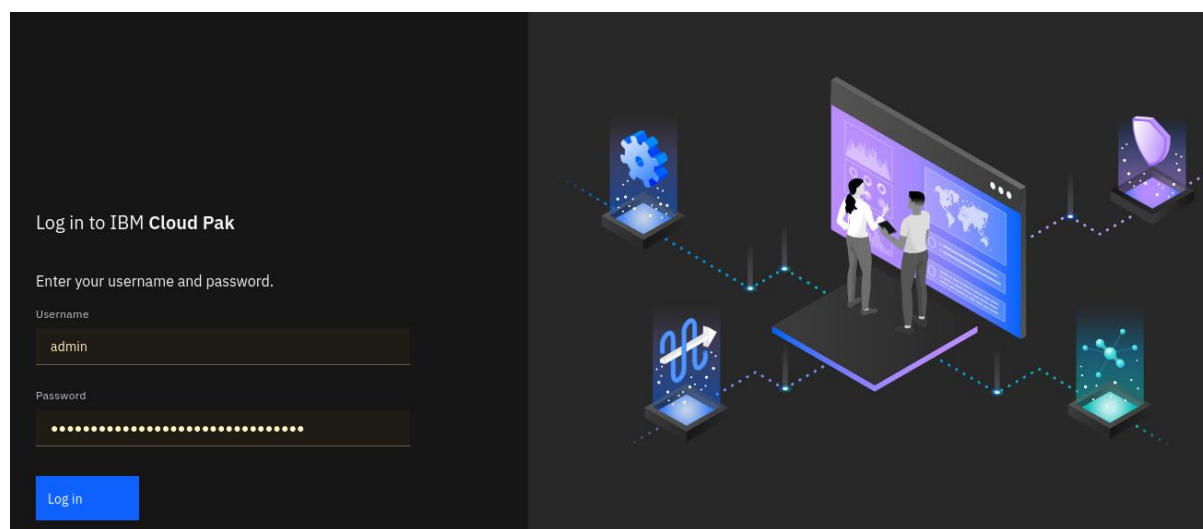
For this lab, WebSphere Automation is pre-installed on an OCP cluster. You do not have your individual WebSphere Automation installation. Let's access your environment.

1. On the *Student VM* and open a browser (ignore any comments to recover a session).



Enter the following URL: <https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com>
(don't use the WebSphere Automation link on bookmark toolbar)

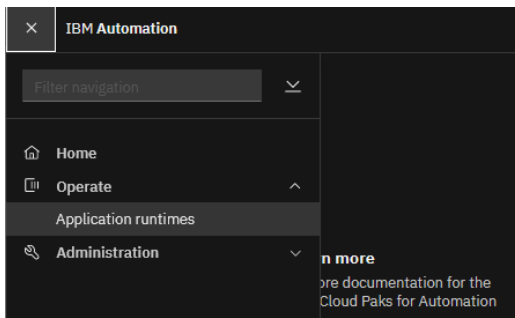
2. On the login panel, click on IBM provided credentials, then use the prefilled login credentials and click on **Log in**.
(user: admin, password: JnarVX84CKz3bAWWqrtjXHF4N3M3UwiW)



Note: If necessary, accept all the warnings and certificates. Depending on your browser, you might have to scroll down to permit access.

3. You should automatically be routed to the **Application Runtimes** page.
If not, open the **Menu**, click **Operate**, and then click **Application runtimes**.

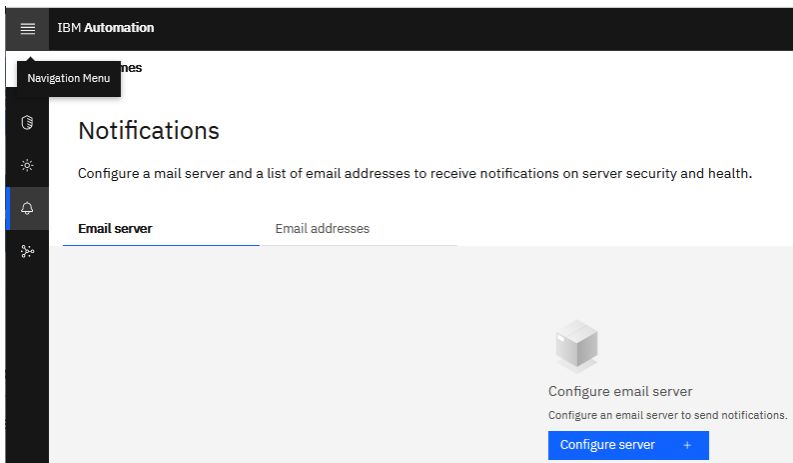
Application Runtimes represent the Traditional WebSphere and WebSphere Liberty servers that have been registered with IBM Automation.



4. The Application runtimes looks like the screenshot below. As you can see, there are three servers already registered. For all registered servers, you can see the version unresolved CVEs as well as applied fixes. The sorting is initially based on Risk Level.

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
High (8.8)	twasServer1	CVE-2021-26296(+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)	twasCell01
None (0.0)	oomServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—
None (0.0)	libertyServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—

5. Before you start to register servers to the Dashboard, you can configure an email to received notifications about CVEs. We do not use the notification in the lab, as we have a shared environment. But below you can find the steps how to configure it.
 - a. Open the **Notification** menu.



- b. On the panel, you could enter
 - i. Your corporate email server
 - ii. The email address of the security administrators to be notified

Please do not enter any details here as this is a shared environment.

The screenshot shows the 'Configure email server' page in the IBM Automation application. The page has a dark sidebar on the left with icons for home, settings, notifications, and a gear icon. The main content area is titled 'Configure email server' with a subtitle 'Configure an email server to send notifications.' Below this is a form titled 'Email server configuration'. The form contains several input fields: 'SMTP server' with the value 'smtp.ibm.com', 'SMTP port' with the value '587', 'Sender email address' with the value 'no-reply@notifications.ibm.com', 'SMTP server credentials' section with 'Username' (value 'Enter username') and 'Password' (value 'Enter password' and a toggle icon), and a 'Certificate' section with the instruction 'Paste your certificate in PEM format'.

IBM Automation

Application runtimes

Configure email server

Configure an email server to send notifications.

Email server configuration

SMTP server

smtp.ibm.com

SMTP port

587

Sender email address

no-reply@notifications.ibm.com

SMTP server credentials

Username

Enter username

Password

Enter password

Certificate

Paste your certificate in PEM format

6. Getting the WSA configuration parameters

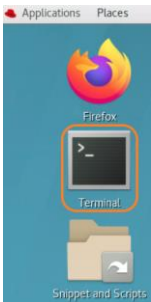
Add each of your WebSphere Application Server and WebSphere Liberty servers to WebSphere Automation by registering them with the **usage metering** service.

To register your application servers with the usage metering service in WebSphere Automation, you must configure the usage metering feature in each application server. To configure the usage metering feature in each of your application servers, you must obtain the following usage metering details:

- **URL:** The URL of the usage metering service in WebSphere Automation. This service registers WebSphere Application Server and Liberty servers with WebSphere Automation so that you can track security vulnerabilities.
- **API Key:** The token used to authenticate the WebSphere Application Server and Liberty servers during the registration process.
- **Usage metering certificate:** The certificate that contains the public key. This key allows an application server that is registering with WebSphere Automation to do an SSL handshake with the metering service.

Usually, you would get them directly from the WebSphere Automation administrator as you would not have an OpenShift CLI on your WebSphere machines. But in the lab environment, we have the client installed and access to the cluster. Let's get these configuration parameters.

1. Return to the desktop and open a new **terminal** window.



2. The following steps retrieve the WebSphere Automation configuration and store the results under \$myWorkingDir/WSA.

- a. Create the directory to store the WSA assets

```
mkdir /var/IBM/temp/WSA
cd /var/IBM/temp/WSA
```

- b. Log into OpenShift and switch to the project **websphere-automation**:

```
oc login -s api.apps.ocp46.tec.uk.ibm.com:6443 -u admin -p passw0rd --insecure-skip-tls-verify=true
oc project websphere-automation

[ibmdemo@RHEL7Guac WSA]$ oc login -s api.apps.ocp46.tec.uk.ibm.com:6443 -u admin -p passw0rd --insecure-skip-tls-verify=true
Login successful.

You have access to 70 projects, the list has been suppressed. You can list all projects with ' projects'

Using project "websphere-automation".
[ibmdemo@RHEL7Guac WSA]$ oc project websphere-automation
Already on project "websphere-automation" on server "https://api.apps.ocp46.tec.uk.ibm.com:6443".
```

- c. Use the **oc** command to get the URL of the usage metering service in WebSphere Automation and save it to a file WSA_metering_URL.txt.

```
oc get route cpd -o jsonpath=https://{.spec.host}/websphereauto/meteringapi >
WSA_metering_URL.txt && cat WSA_metering_URL.txt
```

```
[ibmdemo@RHEL7Guac WSA]$ # Retrieve WSA metering URL
[ibmdemo@RHEL7Guac WSA]$ oc get route cpd -o jsonpath=https://{.spec.host}/websphereauto/meteringapi > WSA_metering_URL.txt && cat WSA_metering_URL.txt
https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi [ibmdemo@RHEL7Guac WSA]$
```

- d. Get the api-key that will be used to authenticate the WebSphere Application Server and Liberty servers during the registration process. Save it to a file named `WSA_metering_api-key.txt`.

```
oc get secret wsa-secure-metering-apis-encrypted-tokens -o jsonpath='{.data.wsa-secure-metering-apis-sa}' | base64 -d > WSA_metering_api-key.txt && cat WSA_metering_api-key.txt
&& echo

[ibmdemo@RHEL7Guac WSA]$ # Retrieve WSA API Key
[ibmdemo@RHEL7Guac WSA]$ oc get secret wsa-secure-metering-apis-encrypted-tokens -o jsonpath='{.data.wsa-secure-metering-apis-sa}' | base64 -d > WSA_metering_api-key.txt && cat WSA_metering_api-key.txt && echo
cu9xY11564jKmHhkfaMHbxD0CzSRPOA8xntR7fL1n1Xi4Gxx2K6KD1kH8J3GF0DGAYKKRZBCg0f+7QE2btWNJhekHnFfdHZTCuN+hrRJj1lg2BwTnTptyXEa+Fg2AjXH
PNIInnCSKo9LZht0BLZYC18ccdnbrN1C+AytnLZnuf3MyYf0raGfmfEFuKcgvEb0gSLULFCv0vLZikTFjWD8C14Gw+NTPhxJ4oxNbUfc2aTdcz4leifE7H/frYaHwv+1
02WaLoZDBQe4EaQYutVoy+5N9utxjGsPaDuWI+/P0v0nwULQ4z/3XAFQia1+B9dkkBJyDdJJ0RpHKRatCBMQ9LJTCYfFiAerUf+be1+3Td5cjcA4ekV0pmc8rwd/cay
PZD54cPD0zNXD0cURkVI4Mv6QLYfTxa+Pt+6w395wXQKcF3Tj4iyELwzB0XSEWJE5X21a5bdsHug7txGmB3eg+Bv+kIMM200FNYrVadWiXgiXSKnhX7cgBbWpUnaP8l
3chl1jNlV6jgyFItoFM2nKd2VEXY0WICCIgznWDH22jkindYN6pskeesfBR5fxpEe7ipdYnpa5xtrV6SeKTY6BgCZ3SPUwfe2pqfJ45B0VdxNNSYKzjcX3r0908H4F/A
3hGwvgo6qd59w0rhf32IKECa0Z0YUynapBf0GfD/GCICKM3aRZkgIwBfTPUvBR2EGRzBz4zJh+NEWNdIMImUvKszKLzbQ+OZLk9HqSDmXKIM10vQdLu7Hxsj1NxxjKRY
jHmBxks+v0tBn08vQR5eN9CEj9wn0wP6E9FQPAcTAo7AjXGIA6IC0QFHTA+3XFj36bs/CMmkDGpDYQwRit1L6FKWRVqlqbRzZ/dPoki3v/yNynS1qWvTLV8oYeQukQk
WzPtKHeI3hegnfk0kd+pLkZMKH01kDl5jMpqpccu+VDNJ4pmvPQfLo10N7qKKG9nfnFNI/2MqkzQB4r02YwVuD5/iDXt8aGcQ+B7RwaCcZCxsIvLKMgQWKAu2z5YA
hoC9iIlWxZPUFR0pCufuHeweyMVxILayJmFHG55Y3rU4p600++IKjJgT29KC8/ZDEPB64qZQDyxYXQE5d3yU+Hj6hqxskB58V2jVqhTHHU7r2kLIugpSXiX+kGuJsdp
nuaf5Y0khL3Jpules7TSxm9Qjicg/+FokMsPkA055LMgWgNBX2NZW7yVCuor9BJ/VQ+n5eDxPUwjpSRycjDuQ9nREJ4LaUVRrE0p06245N/V5VK+zcw5TL9RlVEHU0Yc
AT+sTkjN9/8DhWBMlB+br9B3N/TInY3Bw5w7LXpAcYnh4XKbUWQR7UEKCc+s5maMXjJq+tpMTHSf7FQ==
```

- e. Finally, get the Server certificate that is used for SSL handshake between the servers and IBM Automation, and save it to a file named `WSA_metering_certificate_file.pem`.

```
oc get secret external-tls-secret -o jsonpath='{.data.cert\.crt}' | base64 -d > WSA_metering_certificate_file.pem && cat WSA_metering_certificate_file.pem

[ibmdemo@RHEL7Guac WSA]$ oc get secret external-tls-secret -o jsonpath='{.data.cert\.crt}' | base64 -d > WSA_metering_certificate_file.pem && cat WSA_metering_certificate_file.pem
-----BEGIN CERTIFICATE-----
MIIDlDCCAnygAwIBAgIRAKrnatKpPR9K3XRo0Zo+0wDQYJKoZIhvcNAQELBQAw
JzE1MCcmAg1UEA/McS1UJNEF1dG9tYXRpb24gRm91bmRhZGlvb1B0QTAeFw0yMj
-----
```

- f. As we retrieved all required information from WSA, log out of OpenShift

```
oc logout
```

- g. To ease the reuse, we store the certificate in a separate keystore that can be reused for any outbound connectivity to WebSphere Automation. We use the keytool that is part of the JDK to create the keystore.

```
keytool -import -trustcacerts -file WSA_metering_certificate_file.pem -keystore WSA_metering_Key.p12 -storetype PKCS12 -storepass meterPwd -v -trustcacerts -noprompt

[ibmdemo@RHEL7Guac WSA]$ # Create WSA truststore
[ibmdemo@RHEL7Guac WSA]$ keytool -import -trustcacerts -file WSA_metering_certificate_file.pem -keystore WSA_metering_Key.p12 -storetype PKCS12 -storepass meterPwd -v -trustcacerts -noprompt
Certificate was added to keystore
[Storing WSA_metering_Key.p12]
```

- h. Now let's list all generated assets:

```
ls -lrt WSA_metering*
```

As you can see the following files have been generated:

```
[ibmdemo@RHEL7Guac WSA]$ # List all generated assets
[ibmdemo@RHEL7Guac WSA]$ ls -lrt WSA_metering*
-rw-rw-r--. 1 ibmdemo ibmdemo 84 Apr 25 16:56 WSA_metering_URL.txt
-rw-rw-r--. 1 ibmdemo ibmdemo 1368 Apr 25 16:59 WSA_metering_api-key.txt
-rw-rw-r--. 1 ibmdemo ibmdemo 1302 Apr 25 17:00 WSA_metering_certificate_file.pem
-rw-rw-r--. 1 ibmdemo ibmdemo 1210 Apr 25 17:01 WSA_metering_Key.p12
```

Great! Now you have all the configuration parameters necessary to register the application servers with the usage metering service in WebSphere Automation.

In the next section, you register your first server in WebSphere Automation.

7. Configuring Liberty server

In this section, you configure a Liberty Server instance to register to WebSphere Automation. The Liberty binaries have been installed to /usr/IBM/Liberty/wlp.

Since Liberty servers are easily created, you will first create a new Liberty server and start it.

1. Install a new Liberty server, using the command below:

```
mkdir /var/IBM/temp/Liberty
cd /var/IBM/temp/Liberty
java -jar /var/IBM/software/WAS/wlp-base-all-21.0.0.12.jar -acceptLicense
//var/IBM/temp/Liberty

[ibmdemo@RHEL7Guac WSA]$ #Install Liberty
[ibmdemo@RHEL7Guac WSA]$ mkdir /var/IBM/temp/Liberty
[ibmdemo@RHEL7Guac WSA]$ cd /var/IBM/temp/Liberty
[ibmdemo@RHEL7Guac Liberty]$ java -jar /var/IBM/software/WAS/wlp-base-all-21.0.0.12.jar -acceptLicense //var/IBM/temp/Liberty
Before you can use, extract, or install IBM WebSphere Application
Server, you must accept the terms of IBM International Program License
Agreement and additional license information. Please read the following
license agreements carefully.

The --acceptLicense argument was found. This indicates that you have
accepted the terms of the license agreement.

Extracting files to /var/IBM/temp/Liberty/wlp
Successfully extracted all product files.
```

2. Create a new Liberty server instance, using the command below:

```
/var/IBM/temp/Liberty/wlp/bin/server create libertyServer01

[ibmdemo@RHEL7Guac Liberty]$ # Create a Liberty instance
[ibmdemo@RHEL7Guac Liberty]$ /var/IBM/temp/Liberty/wlp/bin/server create libertyServer01

Server libertyServer01 created.
```

3. Configure the usage metering in the new server. This is configured in the Liberty **server.xml** file. To allow reuse, we configure a separate server.xml with parameters. The server.xml looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
  <!-- Enable features -->
  <featureManager>
    <feature>usageMetering-1.0</feature>
    <feature>transportSecurity-1.0</feature>
  </featureManager>

  <keyStore id="WSA_metering_keyStore"
    password="meterPwd"
    location="${WSA_metering_keystore}"
    type="PKCS12" />

  <ssl id="WSA_metering_SSL" keyStoreRef="defaultKeyStore"
    trustStoreRef="WSA_metering_keyStore" sslProtocol="TLSv1.2" />
  <usageMetering
    url="${WSA_metering_URL}"
    sslRef="WSA_metering_SSL"
    apiKey="${WSA_metering_api-key}" />
</server>
```

- a. Take a look at the server.xml file above

- i. The usageMetering feature has been enabled and defined
- ii. SSL has been configured to use the truststore containing the WSA certificate.
- iii. The WSA details have been specified via variables WSA_metering_keystore, WSA_metering_URL and WSA_metering_api-key, which will be defined later.

- b. The above shown server.xml file has already been created. You could copy the content into the existing server.xml file (which has been created via server create), you could also use an include statement in the existing server.xml file. A third option, that you will use here, is the concept of config dropins, where you just copy the configuration into the appropriate directory and it will be picked up automatically. Here, you will use the configDropins/defaults directory.

```
mkdir -p /var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/configDropins/defaults
cp /var/IBM/software/WAS/WSA_server.xml
/var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/configDropins/defaults
```

```
[ibmdemo@RHEL7Guac Liberty]$ # Use the configDropins/defaults directory
[ibmdemo@RHEL7Guac Liberty]$ mkdir -p /var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/configDropins/defaults
[ibmdemo@RHEL7Guac Liberty]$ cp /var/IBM/software/WAS/WSA_server.xml /var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/configDropins/defaults
```

4. Next you must define the variables WSA_Metering_Keystore, WSA_Metering_URL and WSA_Metering_api-key. This can be done in the bootstrap.properties file. Instead of doing copy and paste, use the commands below.

```
echo "WSA_metering_URL=$(cat $myWorkingDir/WSA/WSA_metering_URL.txt)" >
$myWorkingDir/WSA/bootstrap.properties
echo "WSA_metering_keystore=$myWorkingDir/WSA/WSA_metering_Key.p12" >>
$myWorkingDir/WSA/bootstrap.properties
echo "WSA_metering_api-key=$(cat $myWorkingDir/WSA/WSA_metering_api-key.txt)" >>
$myWorkingDir/WSA/bootstrap.properties
cat $myWorkingDir/WSA/bootstrap.properties
```

```
[ibmdemo@RHEL7Guac Liberty]$ cp /var/IBM/software/WAS/WSA_server.xml /var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/configDropins/defaults
[ibmdemo@RHEL7Guac Liberty]$ echo "WSA_metering_URL=$(cat /var/IBM/temp/WSA/WSA_metering_URL.txt)" > /var/IBM/temp/WSA/bootstrap.properties
[ibmdemo@RHEL7Guac Liberty]$ echo "WSA_metering_keystore=/var/IBM/temp/WSA/WSA_metering_Key.p12" >> /var/IBM/temp/WSA/bootstrap.properties
[ibmdemo@RHEL7Guac Liberty]$ echo "WSA_metering_api-key=$(cat /var/IBM/temp/WSA/WSA_metering_api-key.txt)" >> /var/IBM/temp/WSA/bootstrap.properties
[ibmdemo@RHEL7Guac Liberty]$ cat /var/IBM/temp/WSA/bootstrap.properties
WSA_metering_URL=https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi
WSA_metering_keystore=/var/IBM/temp/WSA/WSA_metering_Key.p12
WSA_metering_api-key=cu9xY11564jKmHhkfaMHbxD0CzsRPOA8xntR7fLln1Xi4Gxx2K6KdLkH8J3GfODGAYKKRZBCg0f+7QE2btWNJhekHnFfdHZTCuN+hrRJ11g2BwTnTPtyXEa+Fg2AjXHPNiInnCSko9LZht0BlZyC18ccdnbrN1C+AytlnLznuf3MyYf0raGfmfEFuKcgvEb0gSLULFCv0vLZikTfJWD8C14Gw+NTPhxJ4oxNbUfc2aTdcz4LeifE7H/frYahVw+102WaLOZDBQE4fEaQYutVoy+5N9utxjGsPaDuWI+/POv0nwULQ4z/3XAFQial+B9dkkBjyDdJJ0RPHKRATCBMQ9LJTCYfFiAerUf+be1+3Tdd5cjcA4ekV0pmc8rwd/cayPZDS4cPD0zNXD0cURkVI4Mv6QLYfTXa+Pt+6w395wXRQkcF3tJ4iyELwzb0XSEWjE5X21a5bdsHug7TxGmB3eg+Bv+kIMM200FNYrVadWiX
```

5. Finally you have to add the bootstrap.properties file to the Liberty configuration. This could be done by merging it with the Liberty bootstrap file via copy and paste, another approach is to use an include to add it to the bootstrap file.

```
echo "bootstrap.include=/var/IBM/temp/WSA/bootstrap.properties" >>
/var/IBM/temp/Liberty/wlp/usr/servers/libertyServer$myUserID/bootstrap.properties
```

```
[ibmdemo@RHEL7Guac Liberty]$ echo "bootstrap.include=/var/IBM/temp/WSA/bootstrap.properties" >> /var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/bootstrap.properties
```

6. Now everything is configured, so let's start the Liberty server and it should register to the WebSphere Automation instance automatically.

```
/var/IBM/temp/Liberty/wlp/bin/server start libertyServer01
```

```
[ibmdemo@RHEL7Guac Liberty]$ /var/IBM/temp/Liberty/wlp/bin/server start libertyServer01
```

```
Starting server libertyServer01.
Server libertyServer01 started with process ID 18407.
```

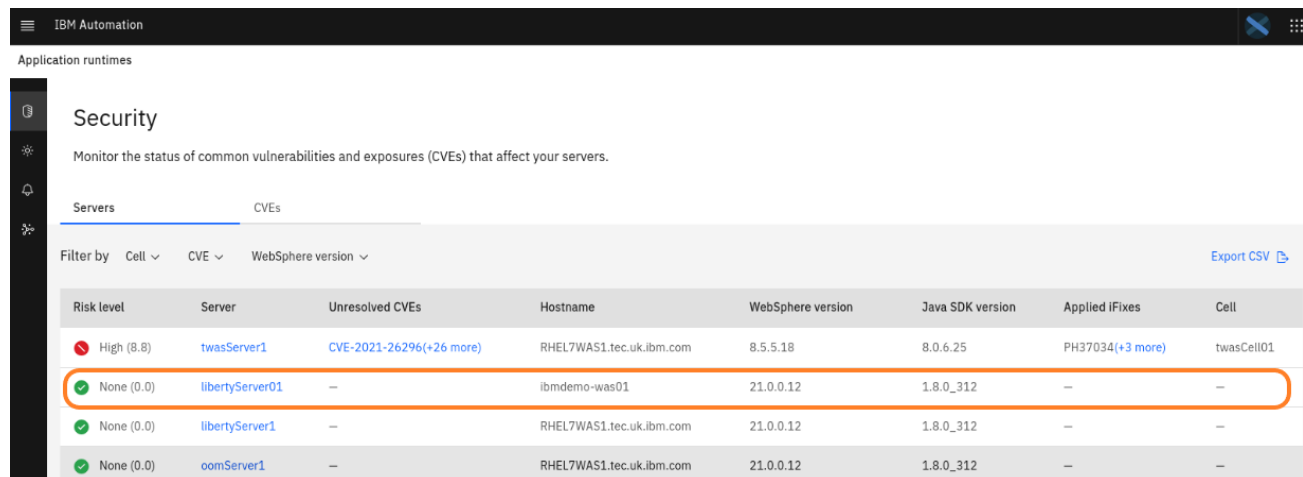
7. View the Liberty server messages.log file with cat and find the message indicating that the server was registered to the metering service.

```
cat $myWorkingDir/Liberty/wlp/usr/servers/libertyServer01/logs/messages.log
```

```
[25/04/22 17:10:43:445 BST] 00000031 com.ibm.ws.usage.metering.common.RegisterTask I CWWKR0400I: The server was registered with the IBM Cloud Private Metering service on the specified URL https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi.
```

8. In the WebSphere Automation UI, navigate to the **Servers** tab, to view the list of registered servers. The new Liberty server should be registered.

- a. Confirm that the Liberty server is registered in the WebSphere Automation Application runtimes page. If the Liberty server was successfully registered, it is displayed in the Application Runtimes in IBM automation UI.
- i. The hostname of the server is the same for all attendees, the Server name is different, so you might have to scroll to find your server.
 - ii. You can click on **Server** to sort by hostname



The screenshot shows the 'Security' section of the IBM Automation UI. It displays a table of servers with columns for Risk level, Server, Unresolved CVEs, Hostname, WebSphere version, Java SDK version, Applied IFixes, and Cell. The row for 'libertyServer01' is highlighted with an orange box.

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied IFixes	Cell
High (8.8)	twasServer1	CVE-2021-26296(+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)	twasCell01
None (0.0)	libertyServer01	—	ibmdemo-was01	21.0.0.12	1.8.0_312	—	—
None (0.0)	libertyServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—
None (0.0)	oomServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—

You have now registered your first Liberty server and can see if there are any vulnerabilities. As the server does not use a lot of features so far, there should be a Risk Level of 0, but this could change if new vulnerabilities are identified. It will also change during the lab.

8. Configuring traditional WebSphere (tWAS) v8.5.5

In this section, you configure a traditional WebSphere Application Server to your WebSphere Automation dashboard. With traditional WebSphere, you use the wsadmin script to configure the usage metering service.

```
export WAS_HOME=/usr/IBM/WAS855ND
$WAS_HOME/bin/manageprofiles.sh -create \
  -profileName WASrsv01 \
  -serverName twasServer01 \
  -templatePath $WAS_HOME/profileTemplates/default \
  -enableAdminSecurity false
```

This might take some minutes, but finally you should see a message like this:

```
[ibmdemo@RHEL7Guac ~]$ export WAS_HOME=/usr/IBM/WAS855ND
[ibmdemo@RHEL7Guac ~]$ $WAS_HOME/bin/manageprofiles.sh -create \
> -profileName WASrsv01 \
> -serverName twasServer01 \
> -templatePath $WAS_HOME/profileTemplates/default \
> -enableAdminSecurity false
INSTCONFSUCCESS: Success: Profile WASrsv01 now exists. Please consult /usr/IBM/WAS855ND/profiles/WASrsv01/logs/About
ThisProfile.txt for more information about this profile.
```

Find out the SOAP port – this is the port we use to configure tWAS via script.

```
cat /usr/IBM/WAS855ND/profiles/WASrsv01/logs/AboutThisProfile.txt | grep SOAP
```

```
[ibmdemo@RHEL7Guac ~]$ cat /usr/IBM/WAS855ND/profiles/WASrsv01/logs/AboutThisProfile.txt | grep SOAP
SOAP connector port: 8881
```

Now let's start the newly created server instance:

```
/usr/IBM/WAS855ND/profiles/WASrsv01/bin/startServer.sh twasServer01
```

After a minute or so, you should see a message that it has been started.

```
[ibmdemo@RHEL7Guac ~]$ /usr/IBM/WAS855ND/profiles/WASrsv01/bin/startServer.sh twasServer01
ADMU0116I: Tool information is being logged in file
          /usr/IBM/WAS855ND/profiles/WASrsv01/logs/twasServer01/startServer.log
ADMU0128I: Starting tool with the WASrsv01 profile
ADMU3100I: Reading configuration for server: twasServer01
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server twasServer01 open for e-business; process id is 6922
```

The approach to configure WAS Traditional is a bit different than the one for Liberty:

- As with Liberty, you first have to retrieve the metering URL as well as the API key. We will re-use the content of the two files that we created for Liberty.
- The WSA certificate will be retrieved from the WSA instance directly.
- To configure WAS, IBM provides a ready to use wsadmin script, you can find details here: <https://www.ibm.com/docs/en/ws-automation?topic=vulnerabilities-adding-websphere-application-server-traditional-server>
The content of the script has been copied into the file configuretWasUsageMetering.py.

Copy the file into the WAS bin directory of the server.

```
cp /var/IBM/software/WAS/configuretWasUsageMetering.py
  /usr/IBM/WAS855ND/profiles/WASrsv01/bin
```

```
[ibmdemo@RHEL7Guac ~]$ cp /var/IBM/software/WAS/configuretWasUsageMetering.py /usr/IBM/WAS855ND/profiles/WASrsv01/bin
```

Switch to the WAS bin directory and run the wsadmin script

```
cd /usr/IBM/WAS855ND/profiles/WSASrv01/bin
./wsadmin.sh -lang jython -connType SOAP -port 8881 -f configuretWasUsageMetering.py
url=$(cat $myWorkingDir/WAS/WSA_metering_URL.txt) apiKey=$(cat
$myWorkingDir/WAS/WSA_metering_api-key.txt) trustStorePassword=meterPwd

[ibmdemo@RHEL7Guac ~]$ cd /usr/IBM/WAS855ND/profiles/WSASrv01/bin
[ibmdemo@RHEL7Guac bin]$ ./wsadmin.sh -lang jython -connType SOAP -port 8881 -f configuretWasUsageMetering.py url=$(cat
/var/IBM/temp/WAS/WSA_metering_URL.txt) apiKey=$(cat /var/IBM/temp/WAS/WSA_metering_api-key.txt) trustStorePassword=me
terPwd
WASX7209I: Connected to process "twasServer01" on node ibmdemo-was01Node01 using SOAP connector; The type of process i
s: UnManagedProcess
WASX7303I: The following options are passed to the scripting environment and are available as arguments that are stored
in the argv variable: "[url=https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi, apiK
ey=cu9xY11564jKmHhkfaMhbx0CzSRP0A8xntR7fL1n1Xi4Gxx2K6KDLkH8J3GF0DGAYKKRZBCg0f+7QE2btWNJhekHnFfdHZTCuN+hrRJj1lg2BwTnTPT
yXea+FG2AjXHPNiInnCSKo9LZht0BLZyC18ccdnbrN1C+AytN1Znuf3MyYf0raGfmfEFuKcgvEb0gSLULFCv0vLZikTFjWD8C14Gw+NTPhxJ4oxNbUfc2aT
dcz4leife7H/frYaHvW+102WaLOZDBQE4fEaQYutVoy+5N9utxjGsPaDuWI+/P0v0nwULQ4z/3XAFQia1+B9dkkBjyDdJJORPHKRATCBM09LJTCYffIAerU
f+be1+3Td5cjcA4ekV0pmc8rwd/cayPZD54cPD0zNXD0cURkVI4Mv6QLYfTxa+Pt+6w395wXQRkcF3tJ4iyELwzB0XSEWjE5X21a5bdsHug7txGmB3eg+Bv
+KIM200FNYrVadWiXgiXSKnhX7cgBbWpUnaP8l3chL1jNLV6jgYFItoFM2nKd2VExYOWICCgznWDH22jkindYN6pskeesfBR5fxpEe7ipdYnpa5xtrV6S
eKTY6BgCZ3SPUwfe2pqfJ45B0vdxNNSYkzjcX3r0908H4F/A3hGwVgos6qd59w0rhf32IKECa0Z0YUynapBf0GfD/GCIKM3aRZkgIwbfTPUvBR2EGRzBz4z
Jh+NEWNdIMImUvKszKlzbQ+OZLk9Hq5DmXKIM10vQdLu7Hxsj1NxxjKRYjHmBxks+v0tBn08vqR5eN9CEj9wn0wP6E9FQPACtao7AjXGIA6IC0QFHTA+3XF
j36bs/CMmkDGPdYQ0wRit1L6FKWRVclqBRzZ/dPOki3v/yNYnS1qWvTLV8oYeQukQkKwZPtKHeI3hegnfk0Kd+pLkJZMKH01kDl5jMpqpcu+VDNJ4pmvPQfL
o10N7qKKG9nfsNFnI/2MqJkzQB4r02YwVuDs/dXT8aGcQ+B7RwaCcZcxsIvLKMgQWKAU2z5YAhoC9IilWxZPUFR0PuCFuhTeweyMVxILayJMFH55Y3rU
4pG00++IKjgt29KC8/ZDPEB64qZQDyYXQe5d3yU+Hj6hqxsKBS8V2jVqhTHHU7r2kLIugpSXIx+kGuJsdnpuaF5Y0khL3Jpules7T5xm9Qjigc/+FokMs
PKA055LMgWgNBX2NZW7yVCuor9BJ/VQ+n5eDxPUwjpSRycjDuQ9nREJ4laUVrE0p06245N/V5VK+z0cwsTL9RLVEHU0YcAT+STKJN9/8JBY/bDhWBWMIb+b
r9B3N/TInY3BwWS7LXpeAcYNh4XKBuWQR7uEkCC+xSmaMXjJq+tpMTHSf7FQ==, trustStorePassword=meterPwd]"
Input arguments:
url: https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi
apiKey: cu9xY11564jKmHhkfaMhbx0CzSRP0A8xntR7fL1n1Xi4Gxx2K6KDLkH8J3GF0DGAYKKRZBCg0f+7QE2btWNJhekHnFfdHZTCuN+hrRJj1lg2
BwTnTPTyXea+FG2AjXHPNiInnCSKo9LZht0BLZyC18ccdnbrN1C+AytN1Znuf3MyYf0raGfmfEFuKcgvEb0gSLULFCv0vLZikTFjWD8C14Gw+NTPhxJ4oxN
bUfc2aTdcz4leife7H/frYaHvW+102WaLOZDBQE4fEaQYutVoy+5N9utxjGsPaDuWI+/P0v0nwULQ4z/3XAFQia1+B9dkkBjyDdJJORPHKRATCBM09LJTCY
ffIAerUf+be1+3Td5cjcA4ekV0pmc8rwd/cayPZD54cPD0zNXD0cURkVI4Mv6QLYfTxa+Pt+6w395wXQRkcF3tJ4iyELwzB0XSEWjE5X21a5bdsHug7txGm
B3eg+Bv+kIM200FNYrVadWiXgiXSKnhX7cgBbWpUnaP8l3chL1jNLV6jgYFItoFM2nKd2VExYOWICCgznWDH22jkindYN6pskeesfBR5fxpEe7ipdYnpa
5xtrV6SeKTY6BgCZ3SPUwfe2pqfJ45B0vdxNNSYkzjcX3r0908H4F/A3hGwVgos6qd59w0rhf32IKECa0Z0YUynapBf0GfD/GCIKM3aRZkgIwbfTPUvBR2E
GRzBz4zJh+NEWNdIMImUvKszKlzbQ+OZLk9Hq5DmXKIM10vQdLu7Hxsj1NxxjKRYjHmBxks+v0tBn08vqR5eN9CEj9wn0wP6E9FQPACtao7AjXGIA6IC0QF
HTA+3XFj36bs/CMmkDGPdYQ0wRit1L6FKWRVclqBRzZ/dPOki3v/yNYnS1qWvTLV8oYeQukQkKwZPtKHeI3hegnfk0Kd+pLkJZMKH01kDl5jMpqpcu+VDNJ4
pmvPQfLo10N7qKKG9nfsNFnI/2MqJkzQB4r02YwVuDs/dXT8aGcQ+B7RwaCcZcxsIvLKMgQWKAU2z5YAhoC9IilWxZPUFR0PuCFuhTeweyMVxILayJMFH
G55Y3rU4pG00++IKjgt29KC8/ZDPEB64qZQDyYXQe5d3yU+Hj6hqxsKBS8V2jVqhTHHU7r2kLIugpSXIx+kGuJsdnpuaF5Y0khL3Jpules7T5xm9Qjigc
/+FokMsPKA055LMgWgNBX2NZW7yVCuor9BJ/VQ+n5eDxPUwjpSRycjDuQ9nREJ4laUVrE0p06245N/V5VK+z0cwsTL9RLVEHU0YcAT+STKJN9/8JBY/bDhW
BWMIB+br9B3N/TInY3BwWS7LXpeAcYNh4XKBuWQR7uEkCC+xSmaMXjJq+tpMTHSf7FQ
trustStorePassword: *****
Creating keystore meteringTrustStore ...
Keystore was created: meteringTrustStore(cells/ibmdemo-was01Node01Cell|security.xml#KeyStore_1650907694817)
Retrieving signer from port ...
Signer was retrieved from host: cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com, port: 443 and store to keystore: me
teringTrustStore
Creating was-usage-metering.properties file with all specified properties ...
Copying keystore meteringTrustStore.p12 and was-usage-metering.properties to all servers ...
keystoreFile meteringTrustStore.p12 was created on all servers.
was-usage-metering.properties was created on all servers.
No sync on WebSphere Base Server!
```

Switch to the browser and you should see that the server has been registered.

IBM Automation

Application runtimes

Security

Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.

Servers CVEs

Filter by Cell CVE WebSphere version

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes
High (8.8)	twasServer01	CVE-2021-26296(+26 more)	ibmdemo-was01	8.5.5.18	8.0.6.25	PH37034(+3 more)
High (8.8)	twasServer1	CVE-2021-26296(+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)
None (0.0)	libertyServer01	—	ibmdemo-was01	21.0.0.12	1.8.0_312	—
None (0.0)	libertyServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—
None (0.0)	oomServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—

Optional: Look for the registration message in the log file

```
cat /usr/IBM/WAS855ND/profiles/WSASrv01/logs/twasServer01/SystemOut.log

[25/04/22 18:28:23:736 BST] 00000048 RegisterTask I CWWKR0400I: The server was registered with the IBM Cloud Private Mete
ring service on the specified URL https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi.
```

9. Get insight from WebSphere Automation

Switch to the browser tab for WebSphere Automation and you can see that the WebSphere Traditional instance twasServer01 has several unresolved CVEs. Look for your server based on the hostname.

IBM Automation

Application runtimes

Security

Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.

Servers CVEs

Filter by Cell CVE WebSphere version

Export CSV

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
None (0.0)	libertyServer01	—	ibmdemo-was01	21.0.0.12	1.8.0_312	—	—
High (8.8)	twasServer01	CVE-2021-26296(+26 more)	ibmdemo-was01	8.5.5.18	8.0.6.25	PH37034(+3 more)	ibmdemo-was01Node01Cell

Click on **twasServer01** to get more details on the issues related to the server.

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
None (0.0)	libertyServer01	—	ibmdemo-was01	21.0.0.12	1.8.0_312	—	—
High (8.8)	twasServer01	CVE-2021-26296(+26 more)	ibmdemo-was01	8.5.5.18	8.0.6.25	PH37034(+3 more)	ibmdemo-was01Node01Cell

This will show all the vulnerabilities that have been found for the server and how long the exposure exists. As we just created the server, it displays 1 day for exposure.

Application runtimes

Back /

twasServer01

Information Vulnerabilities Health investigations

CVEs Fix history

A list of CVEs that affect this server.

Filter by CVE

Risk level	CVE	Status	Days exposed	Detection date
High (8.8)	CVE-2021-26296	Unresolved	1	25/04/2022 18:28:42
High (8.2)	CVE-2021-20353	Unresolved	1	25/04/2022 18:28:37
High (8.2)	CVE-2021-20453	Unresolved	1	25/04/2022 18:28:38
High (8.2)	CVE-2020-4949	Unresolved	1	25/04/2022 18:28:32
High (8.2)	CVE-2021-20454	Unresolved	1	25/04/2022 18:28:44

The list will help you to consolidate maintenance efforts.

Click on the **Unresolved CVE** CVE-2021-26296 to get details about the CVE.

CVE information

Apache MyFaces is vulnerable to cross-site request forgery, caused by improper validation of user-supplied input. By persuading an authenticated user to visit a malicious Web site, a remote attacker could send a malformed HTTP request to perform unauthorized actions. An attacker could exploit this vulnerability to perform cross-site scripting attacks, Web cache poisoning, and other malicious activities.

[View](#)

Risk level: High (8.8) Days exposed: 92 Detected date: 24/01/2022 15:36:09

Affected servers

A list of servers that are affected by this CVE.

Filter by: Vulnerability status: All WebSphere version: ▾

Servers	WebSphere version	Java SDK version	Hostname	Cell	Vulnerability status	Days exposed	Detection date
twasServer01	8.5.5.18	8.0.6.25	ibmdemo-was01	ibmdemo-was01Node01Cell	Unresolved	1	25/04/2022
twasServer1	8.5.5.18	8.0.6.25	RHEL7WAS1.tec.uk.ibm.com	twasCell01	Unresolved	92	24/01/2022

In the lower area, you can see that one of the servers is since quite some time vulnerable. In the upper area, you can see a short description about the CVE and can click on “View” to open the related security bulletin. This will open the support page which provide the details about the CVE, the CVSS Base score (which makes up the risk level) and how to resolve it. Close the tab or switch back to the WebSphere Automation tab.

IBM | Support | Downloads ▾ | Documentation ▾ | Forums | Cases ▾ | Monitoring ▾ | Manage support account ▾

Search support or find a product

Security Bulletin: Vulnerability in Apache MyFaces affects WebSphere Application Server (CVE-2021-26296)

Security Bulletin

Summary

There is a vulnerability in the Apache MyFaces library used by WebSphere Application Server. This has been addressed.

Vulnerability Details

CVEID: [CVE-2021-26296](#)

DESCRIPTION: Apache MyFaces is vulnerable to cross-site request forgery, caused by improper validation of user-supplied input. By persuading an authenticated user to visit a malicious Web site, a remote attacker could send a malformed HTTP request to perform unauthorized actions. An attacker could exploit this vulnerability to perform cross-site scripting attacks, Web cache poisoning, and other malicious activities.

CVSS Base score: 8.8

CVSS Temporal Score: See: <https://exchange.xforce.ibmcloud.com/vulnerabilities/197017> for the current score.

CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H)

Affected Products and Versions

Affected Product(s)	Version(s)
WebSphere Application Server Liberty	17.0.0.3 - 21.0.0.3
WebSphere Application Server	9.0
WebSphere Application Server	8.5
WebSphere Application Server	8.0

[www.ibm.com](#)

Scroll down to the Section about Remediation/Fixes and click on the fix for WAS 8.5.5

Remediation/Fixes

The recommended solution is to apply the interim fix, Fix Pack or PTF containing the APAR for each named product as soon as practical.

For WebSphere Application Server Liberty 17.0.0.3 - 21.0.0.3 using the jsf-2.0, jsf-2.2 or jsf-2.3 feature:

- Upgrade to minimal fix pack levels as required by interim fix and then apply Interim Fix [PH34711](#)
- OR--
- Apply Fix Pack 21.0.0.4 or later (targeted availability 2Q2021).

For WebSphere Application Server traditional and WebSphere Application Server Hypervisor Edition:

For V9.0.0.0 through 9.0.5.7:


- Upgrade to minimal fix pack levels as required by interim fix and then apply Interim Fix [PH34711](#)
- OR--
- Apply Fix Pack 9.0.5.8 or later (targeted availability 2Q2021).

For V8.5.0.0 through 8.5.5.19:

- Upgrade to minimal fix pack levels as required by interim fix and then apply Interim Fix [PH34711](#)
- OR--
- Apply Fix Pack 8.5.5.20 or later (targeted availability 3Q2021).

The support page tells you that the fix has been superseded with the fix PH36923. This is the fix that we will apply later.

IBM Support

 Search support or find a product


PH34711: Vulnerability in Apache MyFaces affects WebSphere Application Server (CVE-2021-26296 CVSS 8.8)

Download

Abstract

Vulnerability in Apache MyFaces affects WebSphere Application Server (CVE-2021-26296 CVSS 8.8)

Download Description


 **THIS FIX HAS BEEN SUPERSEDED BY A LATER INTERIM FIX**


This fix has been superseded by the fix for APAR [PH36923](#). Download and install the fix for [PH36923](#) to resolve PH34711.

Switch back to the WebSphere Automation tab and click on the Security link.

IBM Automation

Application runtimes

 Back /

 CVE-2021-26296

You are back on the Security page.

IBM Automation

Application runtimes



Security

Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.

Servers

CVEs

Filter by Cell CVE WebSphere version

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
 High (8.8)	twasServer01	CVE-2021-26296(+26 more)	ibmdemo-was01	8.5.5.18	8.0.6.25	PH37034(+3 more)	ibmdemo-was01Node01Cell
 High (8.8)	twasServer1	CVE-2021-26296(+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)	twasCell01

Export CSV

As you have seen, you can easily see which servers are in risk and if work can be consolidated. WebSphere Automation also helps you to answer questions like “are we impacted by CVE xxx?”.

You can also export the findings as a csv file to import it into existing tools and processes.

Export CSV

10. Update the Liberty server configuration

As you have seen, the Liberty instance right now does not have any issue.

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes
None (0.0)	libertyServer01	—	ibmdemo-was01	21.0.0.12	1.8.0_312	—
High (8.8)	twasServer01	CVE-2021-26296(+26 more)	ibmdemo-was01	8.5.5.18	8.0.6.25	PH37034(+3 more)

In this section, you will simulate the situation of an application deployment, where an updated application introduces new Liberty features.

Open the Liberty server configuration to add jaxws-2.0 as new features.

```
gedit /var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/server.xml
```

Add the line `<feature>jaxws-2.2</feature>` as shown in the screenshot below and save the file. Then close the editor.

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

  <!-- Enable features -->
  <featureManager>
    <feature>jsp-2.3</feature>
    <feature>jaxws-2.2</feature>
  </featureManager>

  <!-- To access this server from a remote client add a host attribute to the following element, e.g. host="*" -->
  <httpEndpoint id="defaultHttpEndpoint"
    httpPort="298XX"
    httpsPort="218XX" />

  <!-- Automatically expand WAR files and EAR files -->
  <applicationManager autoExpand="true"/>

</server>
```

As Liberty is configured for dynamic updates, the configuration change is applied on the fly. You can see this in the server log messages.log

```
cat $myWorkingDir/Liberty/wlp/usr/servers/libertyServer01/logs/messages.log
```

```
[25/04/22 19:08:25:049 BST] 0000002b com.ibm.ws.config.xml.internal.ConfigRefresher
in 2.134 seconds.
[25/04/22 19:08:25:062 BST] 0000002a com.ibm.ws.kernel.feature.internal.FeatureManager
-2.2, jaxws-2.2].
[25/04/22 19:08:25:063 BST] 0000002a com.ibm.ws.kernel.feature.internal.FeatureManager
A CWWKGF0017I: The server configuration was successfully updated
A CWWKGF0012I: The server installed the following features: [jaxb
A CWWKGF0008I: Feature update completed in 2.128 seconds.
```

The question is: Is the environment now still secure? Wasn't there a CVE alert around JAXWS some weeks ago?

Let's switch to WebSphere Automation and look for your instance. You should see, that WebSphere Automation identified an unresolved vulnerability for your liberty server instance.

Application runtimes

Security

Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.

Servers CVEs

Filter by Cell CVE WebSphere version Export CSV

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
Medium (4.8)	libertyServer01	CVE-2022-22310	ibmdemo-was01	21.0.0.12	1.8.0_312	—	—
High (8.8)	twasServer01	CVE-2021-26296(+26 more)	ibmdemo-was01	8.5.5.18	8.0.6.25	PH37034(+3 more)	ibmdemo-was01Node01Cell

Click on the CVE to get more details.

CVE-2022-22310

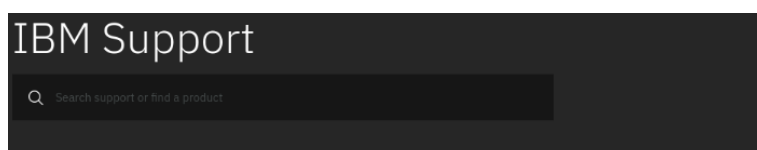
CVE information

IBM WebSphere Application Server Liberty could provide weaker than expected security. A remote attacker could exploit this weakness to obtain sensitive information and gain unauthorized access to JAX-WS applications.

[View](#)

Risk level: Medium (4.8) Days exposed: 1 Detected date: 20/04/2022 11:07:17

Click on “View” to open the related security bulletin.



Security Bulletin: IBM WebSphere Application Server Liberty is vulnerable to an Information Disclosure (CVE-2022-22310)

Security Bulletin

Summary

IBM WebSphere Application Server Liberty is vulnerable to an Information Disclosure. This has been addressed.

Vulnerability Details

CVEID: [CVE-2022-22310](#)

DESCRIPTION: IBM WebSphere Application Server Liberty could provide weaker than expected security. A remote attacker could exploit this weakness to obtain sensitive information and gain unauthorized access to JAX-WS applications.

CVSS Base score: 4.8

CVSS Temporal Score: See: <https://exchange.xforce.ibmcloud.com/vulnerabilities/217224> for the current score.

CVSS Vector: (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N)

Affected Products and Versions

Affected Product(s)	Version(s)
WebSphere Application Server Liberty	21.0.0.10 - 21.0.0.12

If you scroll down, you can see how to resolve this.

Remediation/Fixes

The recommended solution is to apply the interim fix or Fix Pack containing APAR for each named product as soon as practical.

For WebSphere Application Server Liberty 21.0.0.10 - 21.0.0.12 using the jaxws-2.2 feature:

· Upgrade to minimal fix pack levels as required by interim fix and then apply Interim Fix [PH42074](#)

--OR--

· Apply Liberty Fix Pack 22.0.0.1 or later (targeted availability 1Q2022).

Additional interim fixes may be available and linked off the interim fix download page.

So, let's apply the related fix.

The related fix has already been downloaded. To install the fix, we will stop the Liberty instance, apply the fix and start the Liberty instance again.


```
# Apply fix to Liberty
/var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01
java -jar /var/IBM/software/WAS/210012-extended-archive-ifph42074.jar --installLocation
/var/IBM/temp/Liberty/wlp --suppressInfo
/var/IBM/temp/Liberty/wlp/bin/server start libertyServer01
```

```
[ibmdemo@RHEL7Guac bin]$ # Apply fix to Liberty
[ibmdemo@RHEL7Guac bin]$ /var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01
Stopping server libertyServer01.
Server libertyServer01 stopped.
[ibmdemo@RHEL7Guac bin]$ java -jar /var/IBM/software/WAS/210012-extended-archive-ifph42074.jar --installLocation /var/IBM/temp/Liberty/wlp --suppressInfo
Successfully extracted all product files.
[ibmdemo@RHEL7Guac bin]$ /var/IBM/temp/Liberty/wlp/bin/server start libertyServer01
Starting server libertyServer01.
Server libertyServer01 started with process ID 8899.
```

Switch to the WebSphere Automation panel and you should see that the server is back to risk level 0 and, in addition you should see the applied iFix.

Application runtimes

Security

Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.

Servers CVEs

Filter by Cell ▾ CVE ▾ WebSphere version ▾

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes
None (0.0)	libertyServer01	—	ibmdemo-was01	21.0.0.12	1.8.0_312	PH42074

Click on the server and you can see that the server was 1 day exposed.

libertyServer01

Information Vulnerabilities Health investigations

CVEs Fix history

A list of CVEs that affect this server.

Filter by CVE ▾

Risk level	CVE	Status	Days exposed ⓘ	Detection date ⓘ
Medium (4.8)	CVE-2022-22310	Resolved	1	25/04/2022 19:08:30

Click on Fix history to get more details:

Information Vulnerabilities Health investigations

CVEs Fix history

A history of fixes on this server.

Filter by Fix ▾ Action: All ▾ Resolved CVEs ▾ Last 30 days ▾

Fix	Action	Resolved CVEs	Date
PH42074	Installed	CVE-2022-22310	25/04/2022 19:19:57

11. Update tWAS server to fix a vulnerability

In this section, you will apply an iFix to the traditional WebSphere server to remove some vulnerability.

As you can see there are several unresolved vulnerabilities in tWASServerXX. Click on the (+26 more) in the section of unresolved CVE, then click on the highest.

Application runtimes

Security

Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.

Servers CVEs

Filter by Cell CVE WebSphere version

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes
High (8.8)	twasServer01	CVE-2021-26296(+26 more)	ibmdemo-was01	8.5.5.18	8.0.6.25	PH37034(+3 more)

Unresolved CVEs

Server twasServer1

CVEs

Risk level	Unresolved CVEs
High (8.8)	CVE-2021-26296
High (8.2)	CVE-2020-4949
High (8.2)	CVE-2021-20353

You get more details about the CVE and can open the related security bulletin which tells you more details about the vulnerability around Apache MyFaces. If you scroll down and click on the related fix, you will see that the recommended fix is PH36923.

IBM Support

Search support or find a product

Security Bulletin: Vulnerability in Apache MyFaces affects WebSphere Application Server (CVE-2021-26296)

Security Bulletin

Summary

There is a vulnerability in the Apache MyFaces library used by WebSphere Application Server. This has been addressed.

Vulnerability Details

CVEID: [CVE-2021-26296](#)

DESCRIPTION: Apache MyFaces is vulnerable to cross-site request forgery, caused by improper validation of user-supplied input. By persuading an authenticated user to visit a malicious Web site, a remote attacker could send a malformed HTTP request to perform unauthorized actions. An attacker could exploit this vulnerability to perform cross-site scripting attacks, Web cache poisoning, and other malicious activities.

CVSS Base score: 8.8

CVSS Temporal Score: See: <https://exchange.xforce.ibmcloud.com/vulnerabilities/197017> for the current score.

CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H)

IBM Support

Search support or find a product

PH34711: Vulnerability in Apache MyFaces affects WebSphere Application Server (CVE-2021-26296 CVSS 8.8)

Download

Abstract

Vulnerability in Apache MyFaces affects WebSphere Application Server (CVE-2021-26296 CVSS 8.8)

Download Description

THIS FIX HAS BEEN SUPERSEDED BY A LATER INTERIM FIX

This fix has been superseded by the fix for APAR [PH36923](#). Download and install the fix for [PH36923](#) to resolve PH34711.

The fixes have already been downloaded, so you just have to execute the following commands which will stop the WAS instance, deploy the fix and then start the WAS instance again.

```
# Apply fix to tWAS
export fixID="8.5.5.5-WS-WAS-IFPH36923"
export fixRepo="8.5.5.5-ws-was-ifph36923.zip"
export WAS855ND_HOME="/usr/IBM/WAS855ND"
export WAS855ND_PROFILE="$WAS855ND_HOME/profiles/WSASrv01"
export IMCL_HOME="/usr/IBM/IM/eclipse/tools"
# Stop Server
$WAS855ND_PROFILE/bin/stopServer.sh twasServer01
# Apply fix
$IMCL_HOME/imcl install $fixID -repositories /var/IBM/software/WAS/$fixRepo -
installationDirectory $WAS855ND_HOME -log /var/IBM/temp/$fixID.log
# Start Server
$WAS855ND_PROFILE/bin/startServer.sh twasServer01
```

This might take some minutes but finally you should see something like

```
[ibmdemo@ibmdemo-was01 bin]$ # Apply fix to twas
[ibmdemo@ibmdemo-was01 bin]$ export fixID="8.5.5.5-WAS-IFPH36923"
[ibmdemo@ibmdemo-was01 bin]$ export fixRepo="8.5.5.5-ws-was-ifph36923.zip"
[ibmdemo@ibmdemo-was01 bin]$ export WAS855ND_HOME="/usr/IBM/WAS855ND"
[ibmdemo@ibmdemo-was01 bin]$ export WAS855ND_PROFILE="$WAS855ND_HOME/profiles/WSASrv01"
[ibmdemo@ibmdemo-was01 bin]$ export IMCL_HOME="/usr/IBM/IM/eclipse/tools"
[ibmdemo@ibmdemo-was01 bin]$ $WAS855ND_PROFILE/bin/stopServer.sh twasServer01
ADMU0116I: Tool information is being logged in file
/usr/IBM/WAS855ND/profiles/WSASrv01/logs/twasServer01/stopServer.log
ADMU0128I: Starting tool with the WSASrv01 profile
ADMU3100I: Reading configuration for server: twasServer01
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server twasServer01 stop completed.

[ibmdemo@ibmdemo-was01 bin]$ $IMCL_HOME/imcl install $fixID -repositories /var/IBM/software/WAS/$fixRepo -installationDirectory $WAS855ND_HOME -log /var/IBM/temp/$fixID.log
Installed 8.5.5.5-WAS-IFPH36923_8.5.5005.20210520_1002 to the /usr/IBM/WAS855ND directory.

[ibmdemo@ibmdemo-was01 bin]$ $WAS855ND_PROFILE/bin/startServer.sh twasServer01
ADMU0116I: Tool information is being logged in file
/usr/IBM/WAS855ND/profiles/WSASrv01/logs/twasServer01/startServer.log
ADMU0128I: Starting tool with the WSASrv01 profile
ADMU3100I: Reading configuration for server: twasServer01
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server twasServer01 open for e-business; process id is 10316
```

Switch to the browser tab with the WebSphere Automation Runtime panel and you should see that the fixes have been applied and that the risk level dropped as expected from 8.8 (Critical) to 8.2 (High).

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied IFixes
None (0.0)	libertyServer01	—	ibmdemo-was01	21.0.0.12	1.8.0_312	PH42074
High (8.2)	twasServer01	CVE-2020-4949(+25 more)	ibmdemo-was01	8.5.5.18	8.0.6.25	PH37034(+5 more)

12. Remove the Liberty fix to re-introduce a vulnerability

The same as you can apply fixes to resolve a vulnerability, the same an uninstall can re-introduce an issue.

If there is time, you can use the following commands to uninstall an iFix from Liberty and see how the security vulnerability gets back into WebSphere Automation.

```
# Stop Server
/var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01
rm /var/IBM/temp/Liberty/wlp/lib/com.ibm.ws.jaxws.common_1.0.59.cl211220211208-1644.jar
rm /var/IBM/temp/Liberty/wlp/lib/fixes/210012-extended-archive-IFPH42074_21.0.0012.20220115_0043.xml
rm /var/IBM/temp/Liberty/wlp/lib/fixes/210012-extended-archive-IFPH42074_21.0.0012.20220115_0043.lpmf
# Start Server
/var/IBM/temp/Liberty/wlp/bin/server start libertyServer01
```

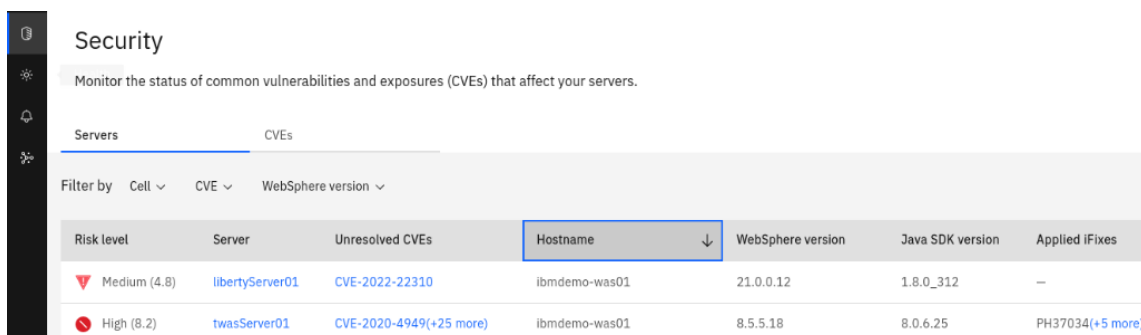
The commands basically stop the Liberty instance, uninstalls the fixes and then starts the Liberty instance again.

```
[ibmdemo@RHEL7Guac bin]$ # Remove fix from Liberty
[ibmdemo@RHEL7Guac bin]$ # Stop Server
[ibmdemo@RHEL7Guac bin]$ /var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01

Stopping server libertyServer01.
Server libertyServer01 stopped.
[ibmdemo@RHEL7Guac bin]$ rm /var/IBM/temp/Liberty/wlp/lib/com.ibm.ws.jaxws.common_1.0.59.cl211220211208-1644.jar
[ibmdemo@RHEL7Guac bin]$ rm /var/IBM/temp/Liberty/wlp/lib/fixes/210012-extended-archive-IFPH42074_21.0.0012.20220115_0043.xml
[ibmdemo@RHEL7Guac bin]$ rm /var/IBM/temp/Liberty/wlp/lib/fixes/210012-extended-archive-IFPH42074_21.0.0012.20220115_0043.lpmf
[ibmdemo@RHEL7Guac bin]$ # Start Server
[ibmdemo@RHEL7Guac bin]$ /var/IBM/temp/Liberty/wlp/bin/server start libertyServer01

Starting server libertyServer01.
Server libertyServer01 started with process ID 10635.
```

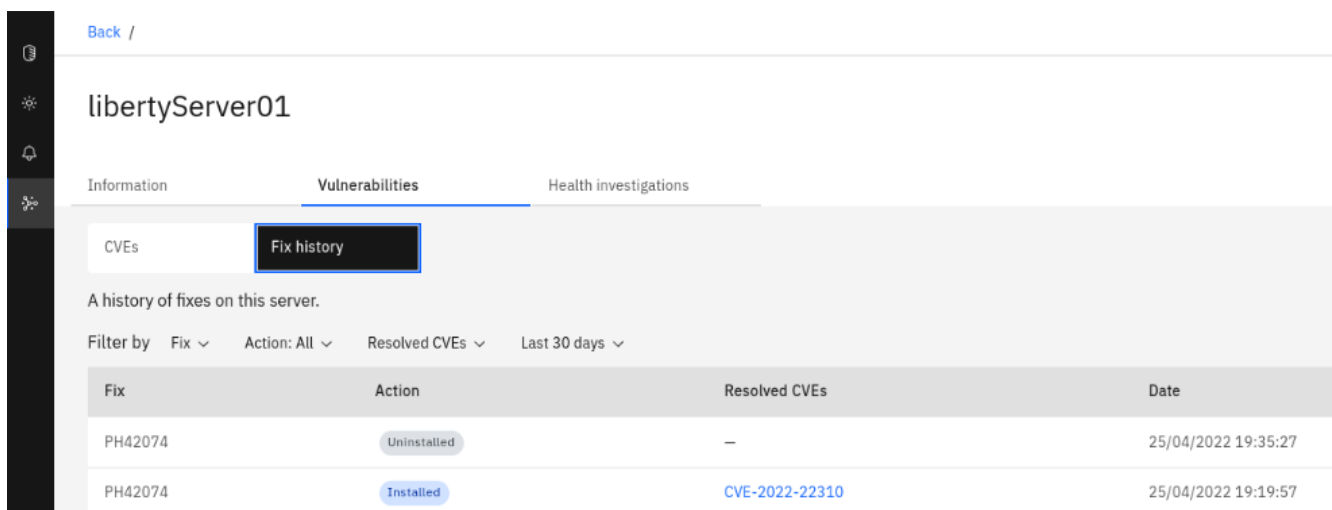
Finally, your WebSphere Automation panel should look as the initial one after registering the servers.



The screenshot shows the 'Security' tab in the WebSphere Automation console. It displays a table of vulnerabilities for two servers. The table has columns for Risk level, Server, Unresolved CVEs, Hostname, WebSphere version, Java SDK version, and Applied iFixes. The 'Hostname' column is highlighted with a blue box.

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes
Medium (4.8)	libertyServer01	CVE-2022-22310	ibmdemo-was01	21.0.0.12	1.8.0_312	—
High (8.2)	twasServer01	CVE-2020-4949(+25 more)	ibmdemo-was01	8.5.5.18	8.0.6.25	PH37034(+5 more)

Click on the server and then on Fix history to get the details:



The screenshot shows the 'libertyServer01' page in the WebSphere Automation console. The 'Vulnerabilities' tab is selected, and the 'Fix history' sub-tab is highlighted with a blue box. Below the sub-tabs, there is a message 'A history of fixes on this server.' and a filter section. The main table shows the history of fixes, with columns for Fix, Action, Resolved CVEs, and Date.

Fix	Action	Resolved CVEs	Date
PH42074	Uninstalled	—	25/04/2022 19:35:27
PH42074	Installed	CVE-2022-22310	25/04/2022 19:19:57

To clean up the environment, stop the servers

```
/var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01  
/usr/IBM/WAS855ND/profiles/WSASrv01/bin/stopServer.sh twasServer01
```

13. Summary

Congratulations! You have completed the WebSphere Automation lab.

With automated tooling and insights, IBM WebSphere Automation enables teams to modernize and secure IT estates, adapt and respond to incidents efficiently, and optimize WebSphere operations. WebSphere system operators and administrators can reduce the cost, effort, and risk of addressing vulnerabilities, automate critical activities, and preserve uptime with early detection, notification, and remediation of incidents.

IBM WebSphere Automation helps teams remove manual toil to work less on maintenance tasks and more on strategic activities, while unlocking new value, extending the life, and increasing ROI of WebSphere investments.

IBM WebSphere Automation is part of IBM Automation, a set of shared automation services that help you get insight into how your processes run, visualize hotspots and bottlenecks, and use financial impact information to prioritize which issues to address first.

To learn more about IBM WebSphere Automation, visit ibm.com/cloud/websphere-automation.

14. Appendix

Lab_WSAcommands.txt

```
The WSA commands are available at
https://larsbesselmannibm.github.io/labs/WSA/lab_WSAcommands.txt
If you want to copy it to your local system, use
curl https://larsbesselmannibm.github.io/labs/WSA/lab_WSAcommands.txt >
~/Desktop/lab_WSAcommands.txt

# Set the WSA endpoint to the IP address provided by the lab provider (new address
is 10.139.195.190)
sudo sed -i 's/10.99.99.23/10.139.195.190/g' /etc/hosts
cat /etc/hosts

# Set your UserID
export myUserID=<student number>

# Set the hostname for the Student VM
export newHost="ibmdemo-was"$myUserID
echo "192.168.1.100 $newHost" | sudo tee -a /etc/hosts
sudo hostname $newHost

# Accessing the WSA UI
https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com

IBM provide credentials: admin/passw0rd
WSA credentials: admin/JnarVX84CKz3bAWWqrtjXHF4N3M3UwiW

# Getting configuration Parameters

# Retrieve WSA Details
mkdir /var/IBM/temp/WSA
cd /var/IBM/temp/WSA

# Log into OpenShift and switch to the project websphere-automation
oc login -s api.apps.ocp46.tec.uk.ibm.com:6443 -u admin -p passw0rd --insecure-
skip-tls-verify=true
oc project websphere-automation

# Retrieve WSA metering URL
oc get route cpd -o jsonpath=https://{.spec.host}/websphereauto/meteringapi >
WSA_metering_URL.txt && cat WSA_metering_URL.txt

# Retrieve WSA API Key
oc get secret wsa-secure-metering-apis-encrypted-tokens -o jsonpath='{.data.wsa-
secure-metering-apis-sa}' | base64 -d > WSA_metering_api-key.txt && cat
WSA_metering_api-key.txt && echo

# Retrieve WSA Metering Certificate
oc get secret external-tls-secret -o jsonpath='{.data.cert\.crt}' | base64 -d >
WSA_metering_certificate_file.pem && cat WSA_metering_certificate_file.pem

# Log out
oc logout

# Create WSA truststore
keytool -import -trustcacerts -file WSA_metering_certificate_file.pem -keystore
WSA_metering_Key.p12 -storetype PKCS12 -storepass meterPwd -v -trustcacerts -
noprompt

# List all generated assets
ls -lrt WSA_metering*
```

```

# Install Liberty
mkdir /var/IBM/temp/Liberty
cd /var/IBM/temp/Liberty
java -jar /var/IBM/software/WAS/wlp-base-all-21.0.0.12.jar -acceptLicense
//var/IBM/temp/Liberty

# Create a Liberty instance
/var/IBM/temp/Liberty/wlp/bin/server create libertyServer01

# Use the configDropins/defaults directory
mkdir -p
/var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/configDropins/defaults
cp /var/IBM/software/WAS/WSA_server.xml
/var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/configDropins/defaults

echo "WSA_metering_URL=$(cat /var/IBM/temp/WSA/WSA_metering_URL.txt)" >
/var/IBM/temp/WSA/bootstrap.properties
echo "WSA_metering_keystore=/var/IBM/temp/WSA/WSA_metering_Key.p12" >>
/var/IBM/temp/WSA/bootstrap.properties
echo "WSA_metering_api-key=$(cat /var/IBM/temp/WSA/WSA_metering_api-key.txt)" >>
/var/IBM/temp/WSA/bootstrap.properties
cat /var/IBM/temp/WSA/bootstrap.properties

echo "bootstrap.include=/var/IBM/temp/WSA/bootstrap.properties" >>
/var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/bootstrap.properties

/var/IBM/temp/Liberty/wlp/bin/server start libertyServer01

cat /var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/logs/messages.log

**** tWAS ***
# Create profile for standalone WAS
export WAS_HOME=/usr/IBM/WAS855ND
$WAS_HOME/bin/manageprofiles.sh -create \
    -profileName WSASrv01 \
    -serverName twasServer01 \
    -templatePath $WAS_HOME/profileTemplates/default \
    -enableAdminSecurity false

# Get ports
cat /usr/IBM/WAS855ND/profiles/WSASrv01/logs/AboutThisProfile.txt | grep SOAP

# Start tWAS
/usr/IBM/WAS855ND/profiles/WSASrv01/bin/startServer.sh twasServer01

# Configure tWAS for WSA
cp /var/IBM/software/WAS/configuretWasUsageMetering.py
/usr/IBM/WAS855ND/profiles/WSASrv01/bin
cd /usr/IBM/WAS855ND/profiles/WSASrv01/bin
./wsadmin.sh -lang jython -connType SOAP -port 8881 -f
configuretWasUsageMetering.py url=$(cat /var/IBM/temp/WSA/WSA_metering_URL.txt)
apiKey=$(cat /var/IBM/temp/WSA/WSA_metering_api-key.txt)
trustStorePassword=meterPwd

# Optional: Look for the registration message in the log file
cat /usr/IBM/WAS855ND/profiles/WSASrv01/logs/twasServer01/SystemOut.log

```



```

# Configure a new feature for an updated application
gedit /var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/server.xml
    <feature>jaxws-2.2</feature>

cat /var/IBM/temp/Liberty/wlp/usr/servers/libertyServer01/logs/messages.log

# Apply fix to Liberty
/var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01
java -jar /var/IBM/software/WAS/210012-extended-archive-ifph42074.jar --
installLocation /var/IBM/temp/Liberty/wlp --suppressInfo
/var/IBM/temp/Liberty/wlp/bin/server start libertyServer01

# Apply fix to tWAS
export fixID="8.5.5.5-WS-WAS-IFPH36923"
export fixRepo="8.5.5.5-ws-was-ifph36923.zip"
export WAS855ND_HOME="/usr/IBM/WAS855ND"
export WAS855ND_PROFILE="$WAS855ND_HOME/profiles/WSASrv01"
export IMCL_HOME="/usr/IBM/IM/eclipse/tools"
# Stop Server
$WAS855ND_PROFILE/bin/stopServer.sh twasServer01
# Apply fix
$IMCL_HOME/imcl install $fixID -repositories /var/IBM/software/WAS/$fixRepo -
installationDirectory $WAS855ND_HOME -log /var/IBM/temp/$fixID.log
# Start Server
$WAS855ND_PROFILE/bin/startServer.sh twasServer01

# Remove fix from Liberty
# Stop Server
/var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01
rm /var/IBM/temp/Liberty/wlp/lib/com.ibm.ws.jaxws.common_1.0.59.cl211220211208-
1644.jar
rm /var/IBM/temp/Liberty/wlp/lib/fixes/210012-extended-archive-
IFPH42074_21.0.0012.20220115_0043.xml
rm /var/IBM/temp/Liberty/wlp/lib/fixes/210012-extended-archive-
IFPH42074_21.0.0012.20220115_0043.lpmf
# Start Server
/var/IBM/temp/Liberty/wlp/bin/server start libertyServer01

# Remove fix from tWAS
export fixID="8.5.5.5-WS-WAS-IFPH36923"
export fixRepo="8.5.5.5-ws-was-ifph36923.zip"
export WAS855ND_HOME="/usr/IBM/WAS855ND"
export WAS855ND_PROFILE="$WAS855ND_HOME/profiles/WSASrv01"
export IMCL_HOME="/usr/IBM/IM/eclipse/tools"
# Stop Server
$WAS855ND_PROFILE/bin/stopServer.sh twasServer01
# Uninstall Fix
$IMCL_HOME/imcl uninstall $fixID -installationDirectory $WAS855ND_HOME -log
/var/IBM/temp/$fixID.log
# Start Server
$WAS855ND_PROFILE/bin/startServer.sh twasServer01

# Cleanup:
# Stop servers
/var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01
/usr/IBM/WAS855ND/profiles/WSASrv01/bin/stopServer.sh twasServer01

# Remove profiles
cd ~
rm -rf /var/IBM/temp
/usr/IBM/WAS855ND/bin/manageprofiles.sh -delete -profileName WSASrv01
rm -rf /usr/IBM/WAS855ND/profiles/WSASrv01

# Cleanup WSA console:
# Change /etc/hosts
sudo cp /etc/hosts.wsa /etc/hosts
cat /etc/hosts

```