

# 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:** April 2022

**Duration:** 45 mins

## 1. Contents

<b>Strengthen Operational Security with IBM WebSphere Automation</b> .....	1
2. Introduction to WebSphere Automation .....	3
3. <b>Business Context</b> .....	4
4. <b>Accessing and starting the environment</b> .....	6
Complete the setup .....	7
5. <b>Accessing the WebSphere Automation UI</b> .....	9
6. <b>Getting the WSA configuration parameters</b> .....	12
7. <b>Configuring Liberty server</b> .....	15
8. <b>Configuring traditional WebSphere (tWAS) v8.5.5</b> .....	18
9. <b>Get insight from WebSphere Automation</b> .....	20
10. <b>Update the Liberty server configuration</b> .....	23
11. <b>Update tWAS server to fix a vulnerability</b> .....	26
12. <b>Remove the Liberty fix to re-introduce a vulnerability</b> .....	28
13. <b>Summary</b> .....	30
14. <b>Appendix</b> .....	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](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.

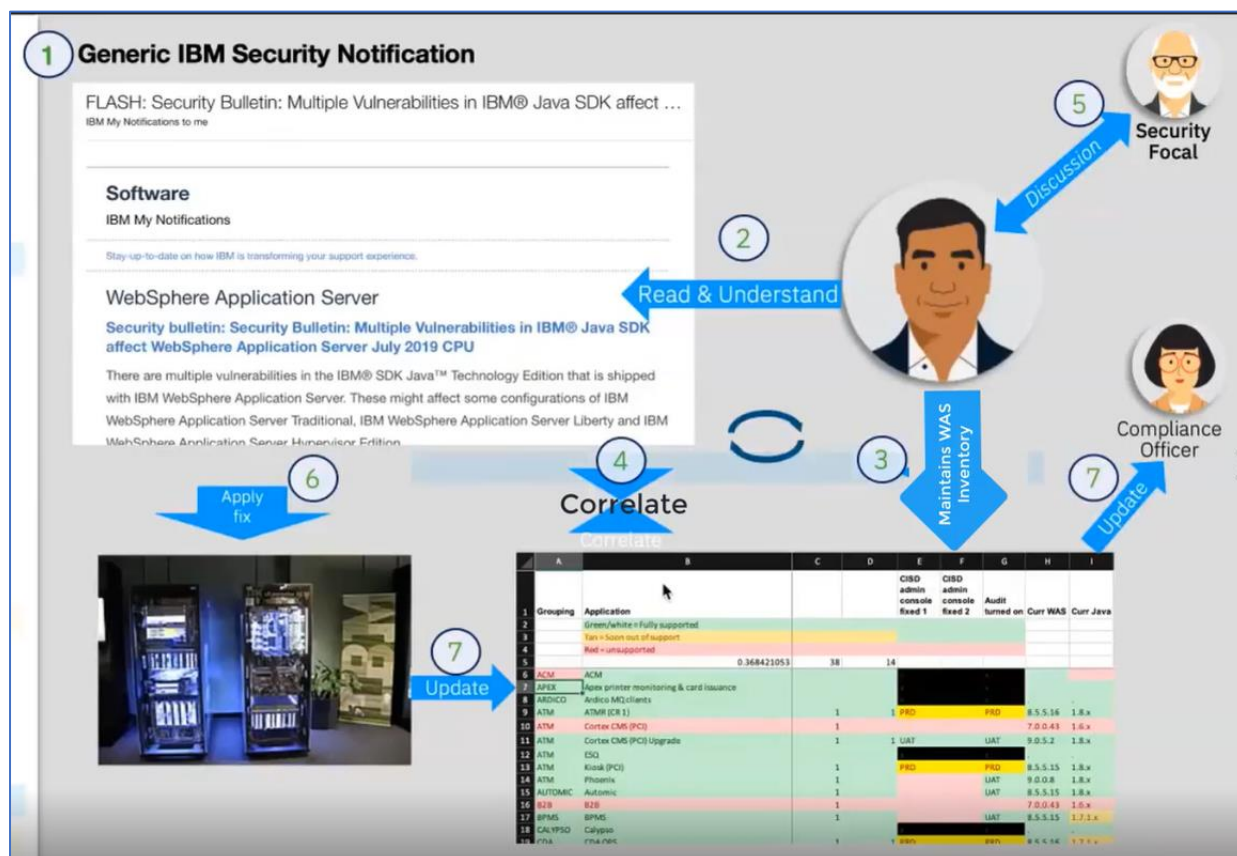
### 3. Business Context

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)	<a href="#">twasServer1</a>	<a href="#">CVE-2021-26296(+26 more)</a>	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18
None (0.0)	<a href="#">oomServer1</a>	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12
None (0.0)	<a href="#">libertyServer1</a>	—	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](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
```

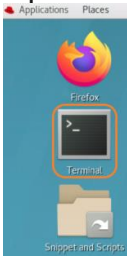
## Complete the setup

### 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.)

```
/usr/IBM/scripts/Backup/setWSA_IP.sh <IP-ADDRESS>
```

Enter passw0rd, when prompted.

```
[ibmdemo@RHEL7WAS1 IBM]$ /usr/IBM/scripts/lab_setWSA_IP.sh 192.168.1.100
Try to set WebSphere Automation target address: 192.168.1.100
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4 ibmdemo-db2 ibmdemo-was ibmdemo-wasxy
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.100 bastion.ocp46.tec.uk.ibm.com bastion
192.168.1.100 cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com cp-console.apps.ocp46.tec.uk.ibm.com api.apps.ocp46.te
c.uk.ibm.com oauth-openshift.apps.ocp46.tec.uk.ibm.com console-openshift-console.apps.ocp46.tec.uk.ibm.com
192.168.1.100 ta.apps.ocp46.tec.uk.ibm.com m2m-ui-wshe.apps.ocp46.tec.uk.ibm.com ibm-licensing-service-instance-ibm-com
mon-services.apps.ocp46.tec.uk.ibm.com
192.168.1.110 RHEL7WAS1.tec.uk.ibm.com RHEL7WAS1
192.168.1.111 RHEL7WAS2.tec.uk.ibm.com RHEL7WAS2
192.168.1.109 instanabackend.tec.uk.ibm.com instanabackend
[ibmdemo@RHEL7WAS1 IBM]$
```

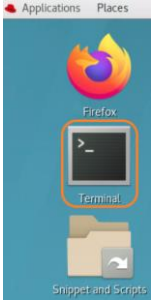
## Set the hostname for the Student VM

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.

```
/usr/IBM/scripts/Backup/setWSA_student.sh $myUserID
```

Enter passw0rd, when prompted.

```
[ibmdemo@RHEL7Guac ~]$ /usr/IBM/scripts/setWSA_student.sh 00
Try to set WebSphere Automation client hostname: 00
Set WebSphere Automation client hostname: ibmdemo-was00
***** Show updated /etc/hosts ***
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4 ibmdemo-db2 ibmdemo-was ibmdemo-was00
***** Show updated hostname ***
ibmdemo-was00
```



## 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.

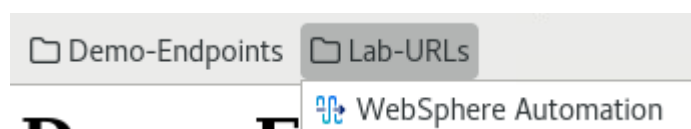
For this lab, WebSphere Automation is pre-installed on an OCP cluster. You do not have your individual WebSphere Automation installation. Let's access your environment.

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

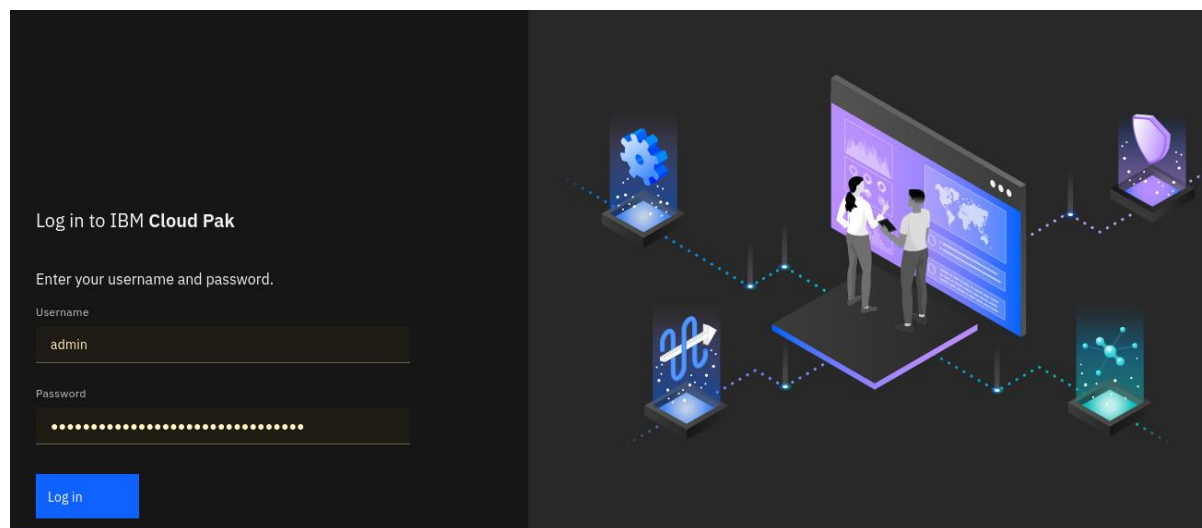


Enter the following URL (there is a WebSphere Automation link on bookmark toolbar):

<https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com>



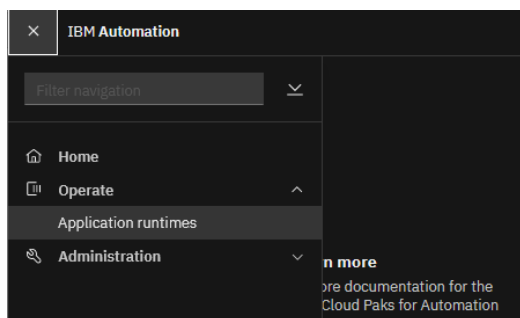
2. On the login panel, click on IBM provided credentials, then use the prefilled login credentials and click on **Log in**.  
(user: admin, password: JnarVX84CKz3bAWWqrtjXHF4N3M3UwiW)



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**.

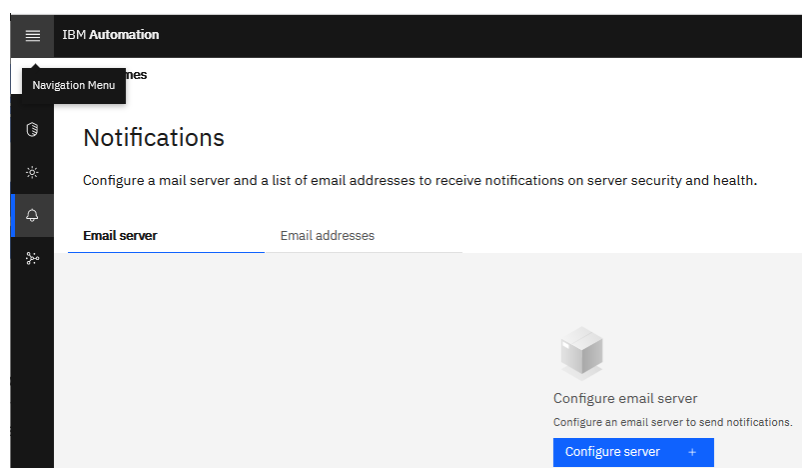
Application Runtimes represent the Traditional WebSphere and WebSphere Liberty servers that have been registered with IBM Automation.



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 console. 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 has several sections: 'SMTP server' with a text input containing 'smtp.ibm.com' and a 'SMTP port' input containing '587'; 'Sender email address' with a text input containing 'no-reply@notifications.ibm.com'; 'SMTP server credentials' with a 'Username' input containing 'Enter username' and a 'Password' input containing 'Enter password' with a toggle icon; and 'Certificate' with a text area containing '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

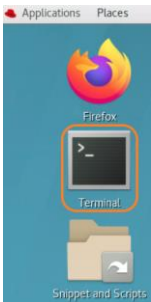
Add each of your WebSphere Application Server and WebSphere Liberty servers to WebSphere Automation by registering them with the **usage metering** service.

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@RHEL7WAS1 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@RHEL7WAS1 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@RHEL7WAS1 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@RHEL7WAS1 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@RHEL7WAS1 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
cu9xYl1564jKmHhkfAMhbxD0CzRP0A8xntR7fLIn1X14Gxx2K6KDLkH8J3GF0DGAyKKRZBCg0f+7QE2btWNJhekHnFfdHZTCuN+hrRj11g2BwTnTptyXEa+fg2AjXHPNi
InnCSKo9LZht0B1ZYC18ccdnbrN1C+AytN1Znuf3MyYf0raGfmfEFuKcgvEb0gSLULFCv0vLZikTFjWD8C14Gw+NTPhxJ4oxNbUfc2aTdcz4leif7H/frYaHvW+102WaL0
ZDB0E4fEaQYutVoy+5N9utxjGsPaDuWI+/POv0nwULQ4z/3XAFQ1a1+B9dkbJyDdJJO9PHKRAICBMQ9LJTCYFiAerUf+be1+3Td5cjCA4ekV0pmc8rwd/cayPZD54cPD0
ZNXD0cURKVI4Mv6QLYftXa+Pt+6w395wXRQkCF3tJ4iYELwzb0XSEWjE5X21a5bdsHug7txGmB3eg+Bv+kIMM208FNyrvAdWiXgiXSKnhX7cgBbWpUnaP8L3chL1jNLV6jg
YfIt0FM2nKd2VexYOWICIGznWDH2jkindYN6pskeesfBR5fxpEe7ipdYnpa5xtrV65ekTY6BgCZ3SPUwfe2pqFJ45B0VdxNNSYkzjX3r0908H4F/A3hGwVgos6qd59w0
rhf32KECa0Z0YUnapBf0GfD/GCICKM3aRZkgIwbfTPUvBR2EGRzBz4Zjh+NEWndiMIuVksZLzbQ+OZLk9HqSDmXKIM10vQdLu7Hxsj1NxxjKRYjHmBxks+v0tBn08vQR
5eN9CEj9w0wP6E9FQPAcTA07AJXGIA6IC0QFHTA+3XFj36bs/CMkGDPdyQ0wR1t1L6FKWRVeqbRzZ/dP0k13v/yNYnS1qWvTLV80YeQukQkzPtkHeI3hegnfk0Kd+pL
kJZMKH01kdL5jMpqpcu+VDNJ4pmvPQfLo10N7qKKG9nfsNFni/2MqJkzQB4r02YwVuDs/1DXT8aGcQ+B7RwaCcZCxsIvLKMgQWKAU2z5YAhoc9IilWxZPUFroPucFuhTew
eyMVxILayJMFHGS5Y3ru4pG0Q++IKjjgT29KC8/ZDPEB64qZQDyxYXQE5d3yU+Hj6hqxskBS8V2jVqhTHU7r2kLIugpSXIx+kGuJsdnpuaF5Y0khL3Jpules7TSxm9Qjig
c/+FokMsPKA055LMgwMBX2NZW7YVCuor9BJ/VQ+n5eDxPUwjpSRycjDuQ9nREJ4laUvRE0p6245N/V5Vkv+z0cwsTL9LVEHU0YCAT+StKJN9/8JBY/bDHwBMMIb+br9B3
N/TInY3Bww57LXpeACYNh4XKbUQWR7uEKCC+xSmaMXjJq+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@RHEL7WAS1 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-----
MIIDlDCCAnyGAwIBAgIRAL5GfEd0Y4I652Z/vp6vgvLmDQYJKoZIhvcNAQELBQAw
JzEELCMGA1UEAxMCsUJNIEF1dG9tYXRpb24gRm91bmRhdGlvbiBDQTAEfW0yMjAx
MjExMTUyMzNaFw0yMjAxMjExMTUyMzNaMA4xDDAKBgNVBAMTA2N2ZDCCASiW0YJKo
ZIhvcNAQEBBQADggEPADCCAQoCggEBALi+rc9ev1xfEbQKXsRkc/vhrJeitv2
w3+Nj/fUQ33CtD1UjDQIKMfc7uJvXE13GyIYbbe/wAHgSA8hN2et7Q4HzGH0Dx
4L6C84DhEe0R++z2n8MUAMeSyjgVJaIcVBPc9PZoxL9e2cvBuiLFtxCgDq0BKMO
9KDoNu3N2fIVGcgjMn6fg5EWklrG049HI+TD0sGBf0qdaU83j09Dv1MrJl8k4y
xX60PMBRmKZfw2R+vZvDTTNoq5oKXIbGhamIiUyFTMmOeaLq9GGNTsDaKCjP3j6
zzX3k52CQH8ld0AVBkmITD1SAeRGZ8bmipQV1tCwVQGS9D5/rOylHgsCAwEAa0B
0zCB0A0BgNVHQ8BAf8EBAMCBAwEwYDVR0LBAwwCgYIKwYBBQUHAwEwDAYDVR0T
AQH/BAIwADAfBgNVHSMGDAwGBS15Ua2rq+VAMBjLx4/scnAGkx/DB6BgNVHREE
czBxggNjcG5CGGNwZC53ZWJzcGhlcmlUeXV0b21hdGlvb0Icy3BkLndLynNwaGvy
ZS1hdXRvbwF0aw9uLnN2Y4IyY3BkLXdlYnNwaGvyZS1hdXRvbwF0aw9uLmFwcHMU
b2NwNDYudG9jLnVrLmli5j2b0wDQYJKoZIhvcNAQELBQADggEBAQgAsaA87L41G
b2THLTVYqXvKf9f/GIUnBgvIVagH/B9T41Px/grc0mOyYQ0LiR2Bk9GqBp1SddiZ
srI9fPPx000RX1lwfkqtmDK1CdQMIF0yJLfhZNYaZlccW0u0hqPdfCyZgsdQfK
qq5SMVc+0BhfXnu1tweV2hxqIgM2clATNYEuhkZclrMInQKGP47Nh9Qr6c1lVTvq
ECMPYxYzSbNbwM7VtmHG1MnouAH54p57S6JFuTad9FtYQe3Ge4/qB0FgA2afA
QBjaTviZ09NzTFvKj5JmLB0/ePEChAI60eoqSKAe8KMLFWHi3r1IaoHQr5te1J
Ik9gg+wm1go=
-----END CERTIFICATE-----
```

- 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.pl2 -storetype PKCS12 -storepass meterPwd -v -trustcacerts -noprompt
[ibmdemo@RHEL7WAS1 WSA]$ keytool -import -trustcacerts -file WSA_metering_certificate_file.pem -keystore WSA_metering_Key.pl2 -storetype PKCS12 -storepass meterPwd -v -trustcacerts -noprompt
Certificate was added to keystore
[Storing WSA_metering_Key.pl2]
```

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

```
ls -lrt WSA_metering*
```

As you can see the following files have been generated:

```
[ibmdemo@RHEL7WAS1 WSA]$ ls -lrt WSA_metering*
-rw-rw-r--. 1 ibmdemo ibmdemo 84 Feb 2 12:01 WSA_metering_URL.txt
-rw-rw-r--. 1 ibmdemo ibmdemo 1368 Feb 2 12:06 WSA_metering_api-key.txt
-rw-rw-r--. 1 ibmdemo ibmdemo 1302 Feb 2 12:08 WSA_metering_certificate_file.pem
-rw-rw-r--. 1 ibmdemo ibmdemo 1210 Feb 2 12:11 WSA_metering_Key.pl2
```

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@RHEL7WAS1 Liberty]$ mkdir $myWorkingDir/Liberty
[ibmdemo@RHEL7WAS1 Liberty]$ cd $myWorkingDir/Liberty
[ibmdemo@RHEL7WAS1 Liberty]$ java -jar /var/IBM/software/WAS/wlp-base-all-21.0.0.12.jar -acceptLicense /$myWorkingDir/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/userXX/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@RHEL7WAS1 Liberty]$ $myWorkingDir/Liberty/wlp/bin/server create libertyServer$myUserID

Server libertyServerXX 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@RHEL7WAS1 Liberty]$ mkdir -p $myWorkingDir/Liberty/wlp/usr/servers/libertyServer$myUserID/configDropins/defaults
[ibmdemo@RHEL7WAS1 Liberty]$ cp /var/IBM/software/WAS/WSA_server.xml $myWorkingDir/Liberty/wlp/usr/servers/libertyServer$myUserID/configDropins/defaults
```

4. Next you have to 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@RHEL7WAS1 Liberty]$ echo "WSA_metering_URL=$(cat $myWorkingDir/WSA/WSA_metering_URL.txt)" > $myWorkingDir/WSA/bootstrap.properties
[ibmdemo@RHEL7WAS1 Liberty]$ echo "WSA_metering_keystore=$myWorkingDir/WSA/WSA_metering_Key.p12" >> $myWorkingDir/WSA/bootstrap.properties
[ibmdemo@RHEL7WAS1 Liberty]$ echo "WSA_metering_api-key=$(cat $myWorkingDir/WSA/WSA_metering_api-key.txt)" >> $myWorkingDir/WSA/bootstrap.properties
[ibmdemo@RHEL7WAS1 Liberty]$ cat $myWorkingDir/WSA/bootstrap.properties
WSA_metering_URL=https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi
WSA_metering_keystore=/var/IBM/userXX/WSA/WSA_metering_Key.p12
WSA_metering_api-key=cu9xY11564jKmHhkfMhbxD0CzSRP0A8xntR7fL1n1X146xx2K6KDlKHJ3GF0DGYKKRZBCg8f+7QE2btWNJhekhnfDhZTCuN+hrRjJ1lg2BwTnTPtyXEa+Fg2AjXHPN1In
nCSK9LZht0B1ZYC18ccdnbrN1C+AytN1Znuf3MyYf0aGfmfEfUkCgvEb0gS1ULFCv0vLZikTFjW08C14Gw+NTPhxJ4oxNbUfc2aTdcz4leifE7H/frYahVw+102WaL0ZDBQe4fEaQYutVoy+5N9utxjG
sPaDuWI+/P0v0nmULQ4z/3XAF0ial+B9dkkBJydbJJORpHKRatCBM091JTYCFFIAerUf+bel+3Td5cJA4ekV0pmc8rwd/cayPZD54cPD0zNXD0cURkVI4Mv6QLYfTXa+Pt+6w395wXQKcF3tJ4iyELwz
b0XS5EWjE5X21a5bdsHug7TxGmB3eg+BV+kIMP200FNyrVadW1XgiXSknH7cgBbwPunaP813chl1jNlV6jgYFitoFM2nKd2VExYOWICCIGznWDH22jkindYN6pskeesfBR5fxpEe7ipdYnpa5xtrv65eKt
Y6BgC23PUwfe2pQfJ45B0VdxNNSYKzjcX3r0908H4F/A3hGwvgos6qd59w0rhf32IKEa0Z0YUynapBf0GfD/GC1KM3aRZkgIwBfTPUvBR2EGRzBz4zJh+NEWndiMImUvKszK1zbQ+0ZLk9Hg5DmXKIM1
0vQdLu7Hxsj1NxxjKRYjHmbXks+v0tBn08vqr5eN9CEj9wn0wP6E9FQPAcTAo7A7jXG1A61C0QFHTA+3XFj36bs/CMekQ6PdY0QWRit1L6FKWRVeqbRzZ/dP0ki3v/yNynS1qWvTLV8oYeQukQkWzPtkhe
I3hegnfk0kd+plKJZMqH01KD15jMqpcu+VDN14pmvP0fLo10N7qKG9nfsNFnI/2MqJkzQB4r02YwVuDs/IDXT8aGcQ+87RwaCcZCcksIvLKMGQWkaU2z5YAhoC9IilWxZPUfR0PuCuFhTeweyMvXILay
JMfhG55Y3rU4pG00++IKjgT29KC8/ZDPEB64qZ00yxYQE5d3yU+Hj6hqsksB58V2jVqhTHHU7r2kLIugpSIXx+kgUjSdpnuaF5Y0khL3jpuLes775xw90jigc/+FoKMsPKA055LMgWgNBX2NZW7yVCuo
r9BJ/VQ+n5e0xPwjp5RycjDuQ9nREJ41aUvRE0p6245N/V5VK+z0cWtL9RLVEHU0YCAT+sTKjN9/8JBV/bdHwBMM1b+br9B3N/TInY3Bws7LXpeAcYNh4XKbUWQR7uEkCC+xSmaMXjJq+tpMTHS7F
Q==
```

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@RHEL7WAS1 Liberty]$ echo "bootstrap.include=$myWorkingDir/WSA/bootstrap.properties" >> $myWorkingDir/Liberty/wlp/usr/servers/libertyServer$myUser
ID/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@RHEL7WAS1 Liberty]$ $myWorkingDir/Liberty/wlp/bin/server start libertyServer$myUserID
```

```
Starting server libertyServerXX.
Server libertyServerXX started with process ID 26910.
```

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
```

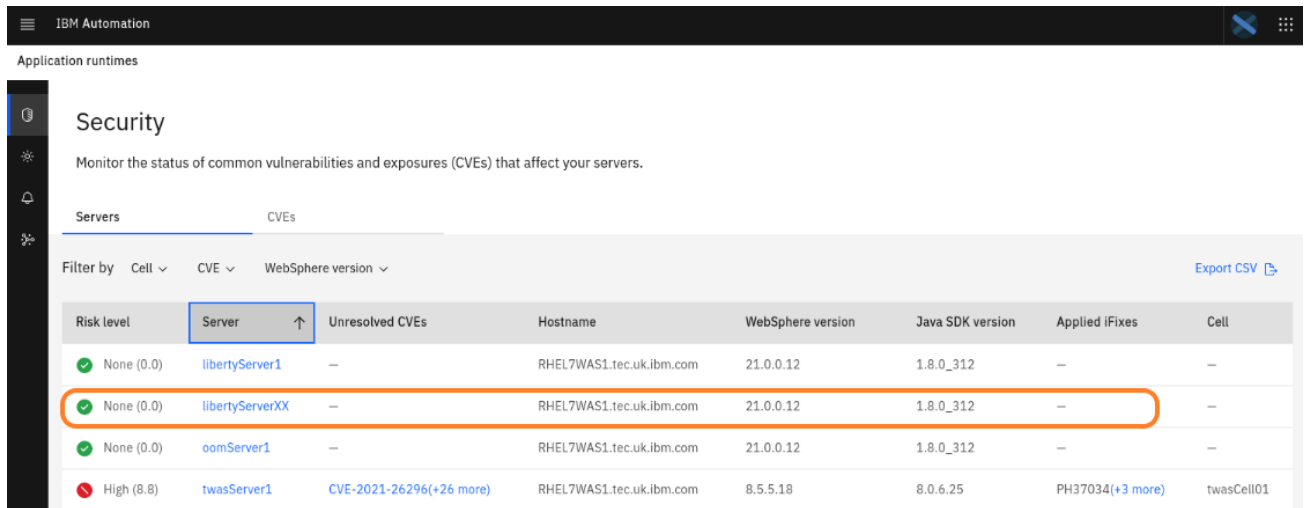
```
[02/02/22 12:53:13:362 GMT] 0000002b com.ibm.ws.usage.metering.common.RegisterTask I CWMKR0400I: The server was registered with the IBM Clo
ud Private Metering service on the specified URL https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi.
```

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 IBM Automation Security console. The 'Security' section is active, displaying a table of servers. The table has columns for Risk level, Server, Unresolved CVEs, Hostname, WebSphere version, Java SDK version, Applied iFixes, and Cell. The 'Server' column is highlighted with a blue box, and the second row is highlighted with an orange box.

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
None (0.0)	libertyServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—
None (0.0)	libertyServerXX	—	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	—	—
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

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@RHEL7WAS1 Liberty]$ $WAS_HOME/bin/manageprofiles.sh -create \
> -profileName WASrsvmyUserID \
> -serverName twasServermyUserID \
> -templatePath $WAS_HOME/profileTemplates/default \
> -enableAdminSecurity false
INSTCONFSUCCESS: Success: Profile WASrsvXX now exists. Please consult /usr/IBM/WAS855ND/profiles/WASrsvXX/logs/AboutThisProfile.txt for more information a
bout this profile.
```

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

```
cat /usr/IBM/WAS855ND/profiles/WASrsv1/logs/AboutThisProfile.txt | grep SOAP
```

```
[ibmdemo@RHEL7WAS1 Liberty]$ cat /usr/IBM/WAS855ND/profiles/WASrsvmyUserID/logs/AboutThisProfile.txt
Application server environment to create: Application server
Location: /usr/IBM/WAS855ND/profiles/WASrsvXX
Disk space required: 200 MB
Profile name: WASrsvXX
Make this profile the default: False
Node name: RHEL7WAS1Node01
Host name: RHEL7WAS1.tec.uk.ibm.com
Enable administrative security (recommended): False
Administrative console port: 9061
Administrative console secure port: 9044
HTTP transport port: 9081
HTTPS transport port: 9444
Bootstrap port: 2810
SOAP connector port: 8881
Run application server as a service: False
Create a Web server definition: False
Performance tuning setting: Standard
```

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@RHEL7WAS1 Liberty]$ /usr/IBM/WAS855ND/profiles/WASrsvmyUserID/bin/startServer.sh twasServermyUserID
ADMU0116I: Tool information is being logged in file
          /usr/IBM/WAS855ND/profiles/WASrsvXX/logs/twasServerXX/startServer.log
ADMU0128I: Starting tool with the WASrsvXX profile
ADMU3100I: Reading configuration for server: twasServerXX
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server twasServerXX open for e-business; process id is 29315
```

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.

```
[ibmdemo@RHEL7WAS1 Liberty]$ cp /var/IBM/software/WAS/configuretWasUsageMetering.py /usr/IBM/WAS855ND/profiles/WSASrv$myUserID/bin
```

Switch to the WAS bin directory and run the wsadmin script

[illegible]

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

**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
High (8.8)	<a href="#">twasServerXX</a>	<a href="#">CVE-2021-26296(+26 more)</a>	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	<a href="#">PH37034(+3 more)</a>	RHEL7WAS1Node01Cell
High (8.8)	<a href="#">twasServer1</a>	<a href="#">CVE-2021-26296(+26 more)</a>	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	<a href="#">PH37034(+3 more)</a>	twasCell01
None (0.0)	<a href="#">oomServer1</a>	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—
None (0.0)	<a href="#">libertyServerXX</a>	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—
None (0.0)	<a href="#">libertyServer1</a>	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—

Look for the registration message in the log file

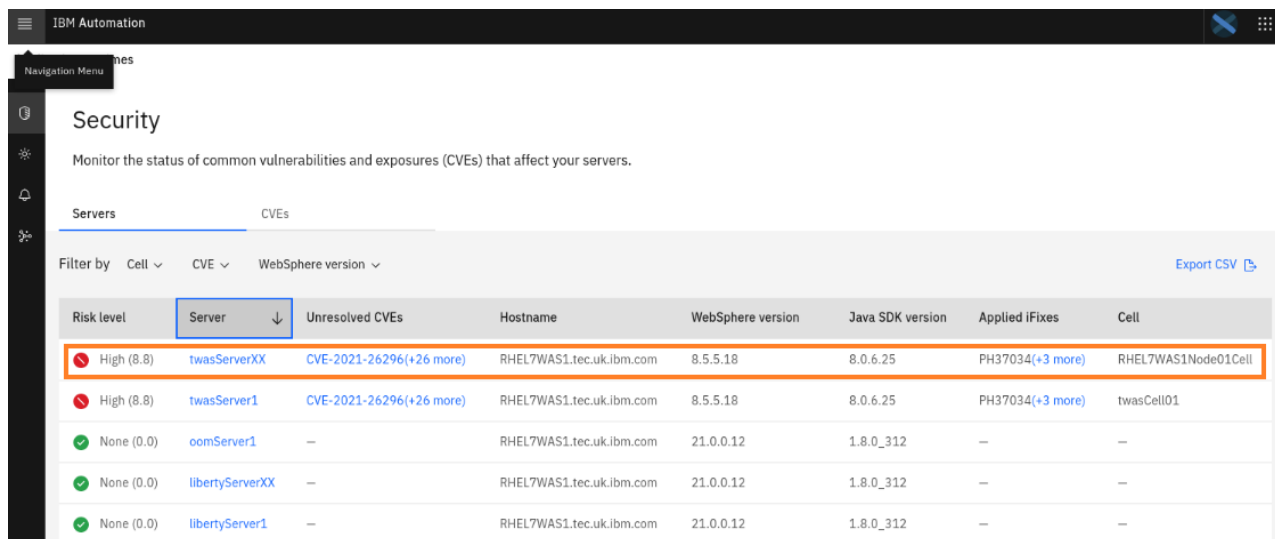
```
[02/02/22 13:21:34:394 GMT] 00000069 RegisterTask I CWWKR0400I: The server was registered with the IBM Cloud Private Metering service on the specified URL https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi.
```

If there is an issue, then restart the server.

```
/usr/IBM/WAS855ND/profiles/WSASrv01/bin/stopServer.sh twasServer01
/usr/IBM/WAS855ND/profiles/WSASrv01/bin/startServer.sh twasServer01
```

## 9. Get insight from WebSphere Automation

Switch to the browser tab for WebSphere Automation and you can see that the WebSphere Traditional instance server1 has several unresolved CVEs.



The screenshot shows the IBM Automation Security interface. A navigation menu on the left highlights 'Security'. The main content area is titled 'Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.' Below this, there are tabs for 'Servers' and 'CVEs'. A filter bar allows filtering by 'Cell', 'CVE', and 'WebSphere version'. An 'Export CSV' link is present. The main table lists servers with columns for Risk level, Server, Unresolved CVEs, Hostname, WebSphere version, Java SDK version, Applied iFixes, and Cell. The first row, for 'twasServerXX', is highlighted with an orange border.

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
High (8.8)	twasServerXX	<a href="#">CVE-2021-26296(+26 more)</a>	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)	RHEL7WAS1Node01Cell
High (8.8)	twasServer1	<a href="#">CVE-2021-26296(+26 more)</a>	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)	libertyServerXX	—	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	—	—

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

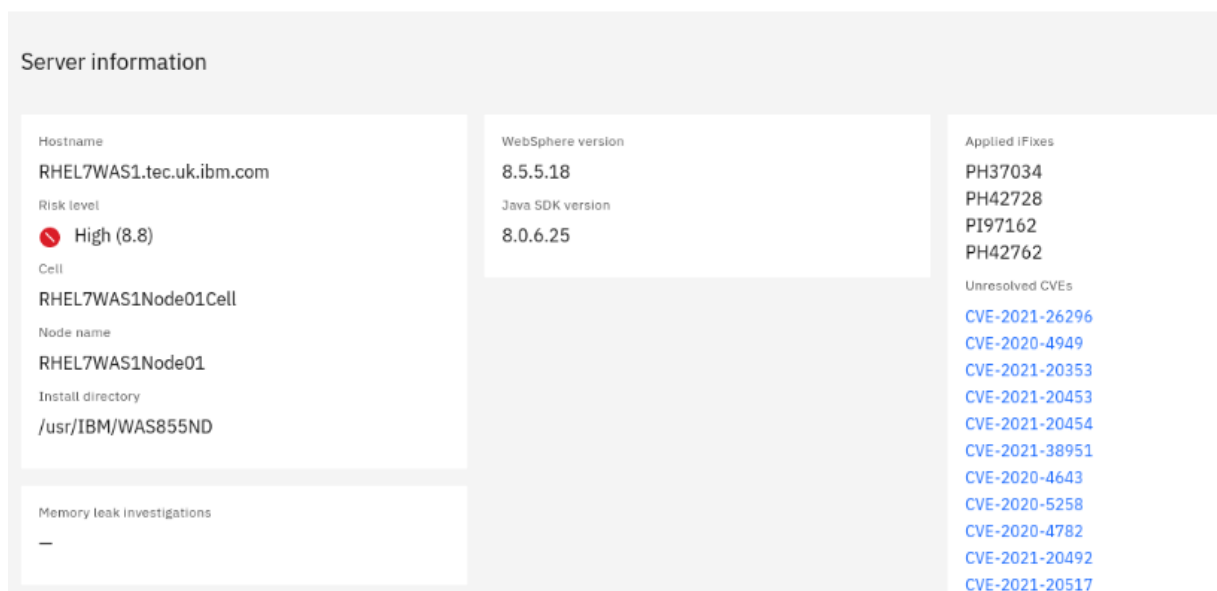


This is a cropped view of the table from the previous screenshot, with the row for 'twasServerXX' highlighted with an orange border.

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
High (8.8)	twasServerXX	<a href="#">CVE-2021-26296(+26 more)</a>	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)	RHEL7WAS1Node01Cell

This will show all the vulnerabilities that have been found for the server.

twasServerXX



The screenshot shows the 'Server information' page for 'twasServerXX'. It is divided into three main sections: Hostname, WebSphere version, and Applied iFixes. The Hostname section includes Risk level, Cell, Node name, and Install directory. The WebSphere version section includes Java SDK version. The Applied iFixes section lists several iFixes and a list of Unresolved CVEs.

Server information		
Hostname	WebSphere version	Applied iFixes
RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	PH37034
Risk level	Java SDK version	PH42728
High (8.8)	8.0.6.25	PI97162
Cell		PH42762
RHEL7WAS1Node01Cell		Unresolved CVEs
Node name		<a href="#">CVE-2021-26296</a>
RHEL7WAS1Node01		<a href="#">CVE-2020-4949</a>
Install directory		<a href="#">CVE-2021-20353</a>
/usr/IBM/WAS855ND		<a href="#">CVE-2021-20453</a>
		<a href="#">CVE-2021-20454</a>
		<a href="#">CVE-2021-38951</a>
		<a href="#">CVE-2020-4643</a>
		<a href="#">CVE-2020-5258</a>
		<a href="#">CVE-2020-4782</a>
		<a href="#">CVE-2021-20492</a>
		<a href="#">CVE-2021-20517</a>


The list will help you to consolidate maintenance efforts.

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

## Server information

<b>Hostname</b> <b>RHEL7WAS1.tec.uk.ibm.com</b>	<b>WebSphere version</b> <b>8.5.5.18</b>	<b>Applied iFixes</b> <b>PH37034</b> <b>PH42728</b> <b>PI97162</b> <b>PH42762</b>
<b>Risk level</b>  <b>High (8.8)</b>	<b>Java SDK version</b> <b>8.0.6.25</b>	<b>Unresolved CVEs</b> <b>CVE-2021-26296</b> <b>CVE-2020-4949</b> <b>CVE-2021-20353</b> <b>CVE-2021-20453</b> <b>CVE-2021-20454</b> <b>CVE-2021-38951</b>
<b>Cell</b> <b>RHEL7WAS1Node01Cell</b>		
<b>Node name</b> <b>RHEL7WAS1Node01</b>		
<b>Install directory</b> <b>/usr/IBM/WAS855ND</b>		

A new tab will open, and you will be routed to 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.



[Support](#)
[Downloads](#)
[Documentation](#)
[Forums](#)
[Cases](#)
[Monitoring](#)
[Manage support account](#)

## 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

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 **Back**.

IBM Automation

Application runtimes

[Back](#) /

twasServerXX

You are back on the main 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 ▼ [Export CSV](#)

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
High (8.8)	<a href="#">twasServerXX</a>	<a href="#">CVE-2021-26296(+26 more)</a>	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	<a href="#">PH37034(+3 more)</a>	RHEL7WAS1Node01Cell
High (8.8)	<a href="#">twasServer1</a>	<a href="#">CVE-2021-26296(+26 more)</a>	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	<a href="#">PH37034(+3 more)</a>	twasCell01
None (0.0)	<a href="#">libertyServerXX</a>	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—
None (0.0)	<a href="#">oomServer1</a>	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—
None (0.0)	<a href="#">libertyServer1</a>	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—

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	Cell
High (8.8)	twasServerXX	CVE-2021-26296(+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)	RHEL7WAS1Node01Cell
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)	libertyServerXX	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—

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 $myWorkingDir/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="280XX"
    httpsPort="210XX" />

  <!-- 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
```

```
[02/02/22 14:17:41:313 GMT] 00000037 com.ibm.ws.kernel.feature.internal.FeatureManager A CWKKF0012I: The server installed the following feature
s: [jaxb-2.2, jaxws-2.2].
[02/02/22 14:17:41:314 GMT] 00000037 com.ibm.ws.kernel.feature.internal.FeatureManager A CWKKF0008I: Feature update completed in 3.003 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.

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
High (8.8)	twasServerXX	CVE-2021-26296(+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)	RHEL7WAS1Node01Cell
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
Medium (4.8)	libertyServerXX	CVE-2022-22310	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—

Click on the CVE to open the related CVE alert.

IBM Automation

</

IBM Support

Search support or find a product

## 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 and a small script has been created. The script mainly stops the server, applies the iFix and starts the server. Content:

```
echo "Apply Fix"
/var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01
echo "Apply iFix to resolve CVE"
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
```

To apply the Liberty iFix, go to a command shell and execute the script

```
/usr/IBM/labs/WAS/wlp_applyFix.sh
```

```
[ibmdemo@RHEL7WAS1 bin]$ /usr/IBM/scripts/lab_wlp_applyFix.sh
Apply Fix

Stopping server libertyServerXX.
Server libertyServerXX stopped.
Apply iFix to resolve CVE
Successfully extracted all product files.

Starting server libertyServerXX.
Server libertyServerXX started with process ID 4664.
```

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.

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
High (8.8)	twasServerXX	CVE-2021-26296(+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)	RHEL7WAS1Node01Cell
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)	libertyServerXX	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	PH42074	—

## 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.

The screenshot shows the IBM Automation Security interface. The main table lists servers and their unresolved CVEs. A modal window titled 'Unresolved CVEs' is open, showing details for CVE-2021-26296 (High 8.8) in Apache MyFaces. The modal also lists other CVEs like CVE-2020-4949.

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version
High (8.8)	twasServerXX	CVE-2021-26296 (+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18
High (8.8)	twasServer1	CVE-2021-26296 (+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18

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

The security bulletin opens and you can see that the vulnerability is in Apache MyFaces. If you scroll down and click on the related fix, you will see that the recommended fix is PH36923.

The left page is the 'Security Bulletin: Vulnerability in Apache MyFaces affects WebSphere Application Server (CVE-2021-26296)'. It includes a summary and vulnerability details. The right page is the 'PH34711: Vulnerability in Apache MyFaces affects WebSphere Application Server (CVE-2021-26296 CVSS 8.8)' page. It includes a download section and a note that the fix has been superseded by a later interim fix (PH36923).

The fixes have already been downloaded and a script has been created.  
Content of the script:

```
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"
echo "Stop Server"
$WAS855ND_PROFILE/bin/stopServer.sh twasServer01
echo "Apply Fix $fixID"
$IMCL_HOME/imcl install $fixID -repositories /var/IBM/software/WAS/$fixRepo -
installationDirectory $WAS855ND_HOME -log /var/IBM/temp/$fixID.log
echo "Start Server"
$WAS855ND_PROFILE/bin/startServer.sh twasServer01
```

The script basically stops the tWAS instance, uses IBM Installation Manager to apply a fix for WAS and then starts the tWAS instance again.

```
/usr/IBM/labs/WSA/was_applyFixes.sh
```

tWAS fix, go to a command shell and execute the script

To apply the

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

```
[ibmdemo@RHEL7WAS1 bin]$ /usr/IBM/scripts/lab_was_applyFixes.sh
Stop Server
ADMU0116I: Tool information is being logged in file
           /usr/IBM/WAS855ND/profiles/WSASrvXX/logs/twasServerXX/stopServer.log
ADMU0128I: Starting tool with the WSASrvXX profile
ADMU3100I: Reading configuration for server: twasServerXX
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server twasServerXX stop completed.

Apply Fix 8.5.5.5-WS-WAS-IFPH36923
Installed 8.5.5.5-WS-WAS-IFPH36923_8.5.5005.20210520_1002 to the /usr/IBM/WAS855ND directory.
Start Server
ADMU0116I: Tool information is being logged in file
           /usr/IBM/WAS855ND/profiles/WSASrvXX/logs/twasServerXX/startServer.log
ADMU0128I: Starting tool with the WSASrvXX profile
ADMU3100I: Reading configuration for server: twasServerXX
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server twasServerXX open for e-business; process id is 8270
```

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	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
High	(8.2)	twasServerXX	CVE-2020-4949(+25 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+5 more)	RHEL7WAS1Node01Cell

## 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 script to uninstall an iFix from Liberty and see how the security vulnerability gets back into WebSphere Automation.

Content of the script:

```
/var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01
echo "Remove iFix"
rm /var/IBM/temp/Liberty/wlp/lib/com.ibm.ws.jaxws.common_1.0.59.c1211220211208-1644.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
/var/IBM/temp/Liberty/wlp/bin/server start libertyServer01
```

The script basically stops the Liberty instance, uninstalls the fixes and then starts the Liberty instance again.

To remove the Liberty fix, go to a command shell and execute the script

```
/usr/IBM/labs/WSA/wlp_removeFix.sh
```

```
[ibmdemo@RHEL7WAS1 bin]$ /usr/IBM/scripts/lab_wlp_removeFix.sh
Stopping server libertyServerXX.
Server libertyServerXX stopped.
Remove iFix

Starting server libertyServerXX.
Server libertyServerXX started with process ID 8986.
```

To remove the tWAS fix, go to a command shell and execute the script

```
/usr/IBM/labs/WSA/was_removeFixes.sh
```

```
[ibmdemo@RHEL7WAS1 bin]$ /usr/IBM/scripts/lab_was_removeFixes.sh
Stop Server
ADMU0116I: Tool information is being logged in file
           /usr/IBM/WAS855ND/profiles/WSASrvXX/logs/twasServerXX/stopServer.log
ADMU0128I: Starting tool with the WSASrvXX profile
ADMU3100I: Reading configuration for server: twasServerXX
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server twasServerXX stop completed.

Remove Fix 8.5.5.5-WS-WAS-IFPH36923
Uninstalled 8.5.5.5-WS-WAS-IFPH36923_8.5.5005.20210520_1002 from the /usr/IBM/WAS855ND directory.
Start Server
ADMU0116I: Tool information is being logged in file
           /usr/IBM/WAS855ND/profiles/WSASrvXX/logs/twasServerXX/startServer.log
ADMU0128I: Starting tool with the WSASrvXX profile
ADMU3100I: Reading configuration for server: twasServerXX
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server twasServerXX open for e-business; process id is 10231
```

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

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
High (8.8)	twasServerXX	CVE-2021-26296(+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)	RHEL7WAS1Node01Cell
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
Medium (4.8)	libertyServerXX	CVE-2022-22310	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	—	—
None (0.0)	libertyServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—

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](https://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
/usr/IBM/labs/WSA/Backup/setWSA_IP.sh <IP-ADDRESS>

# Set your UserID
export myUserID=<student number>

# Set the hostname for the Student VM
/usr/IBM/scripts/Backup/setWSA_student.sh $myUserID

# Accessing the WSA UI
https://cp-console.apps.ocp46.tec.uk.ibm.com/oidc/login.jsp
IBM provide credentials: admin/passw0rd
WSA credentials: admin/JnarVX84CKz3bAWWqrtjXHF4N3M3UwiW

# Getting configuration Parameters

# /usr/IBM/labs/WSA/retrieveWSADetails.sh
echo "***** Retrieve WSA Details *****"
mkdir /var/IBM/temp/WSA
cd /var/IBM/temp/WSA

oc login -s api.apps.ocp46.tec.uk.ibm.com:6443 -u admin -p passw0rd --insecure-
skip-tls-verify=true
oc project websphere-automation
# Metering API
echo "***** Retrieve WSA metering URL *****"
oc get route cpd -o jsonpath=https://{.spec.host}/websphereauto/meteringapi >
WSA_metering_URL.txt && cat WSA_metering_URL.txt
# API Key:
echo "***** 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 && echo
WSA_metering_api-key.txt && echo
#Usage Metering Certificate
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 a Keystore for metering
echo "***** Create WSA truststore *****"
keytool -import -trustcacerts -file WSA_metering_certificate_file.pem -keystore
WSA_metering_Key.pl2 -storetype PKCS12 -storepass meterPwd -v -trustcacerts -
noprompt

# List all generated assets
echo "***** List Files with WSA Details *****"
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

# Restart tWAS
/usr/IBM/WAS855ND/profiles/WSASrv01/bin/stopServer.sh twasServer01
/usr/IBM/WAS855ND/profiles/WSASrv01/bin/startServer.sh twasServer01
```



```

# 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
# Via script:
/usr/IBM/labs/WSA/wlp_applyFix.sh
# Manually
echo "Apply Fix"
/var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01
echo "Apply iFix to resolve CVE"
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
# Via script:
/usr/IBM/labs/WSA/was_applyFixes.sh
# Manually
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"
echo "Stop Server"
$WAS855ND_PROFILE/bin/stopServer.sh twasServer01
echo "Apply Fix $fixID"
$IMCL_HOME/imcl install $fixID -repositories /var/IBM/software/WAS/$fixRepo -
installationDirectory $WAS855ND_HOME -log /var/IBM/temp/$fixID.log
echo "Start Server"
$WAS855ND_PROFILE/bin/startServer.sh twasServer01

# Remove fix from Liberty
# Via script:
/usr/IBM/labs/WSA/wlp_removeFix.sh
# Manually
# Stop Server
/var/IBM/temp/Liberty/wlp/bin/server stop libertyServer01
echo "Remove iFix"
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
# Via script:
/usr/IBM/labs/WSA/was_removeFixes.sh

# 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

```