[网络安全自学篇] 四.实验吧CTF实战之WEB渗透和隐写术解 密



Azreal、Z 于 2020-07-02 16:49:28 发布 726 论 收藏 4 分类专栏: <u>网络安全自学篇</u>文章标签: <u>安全</u> 原文链接: <u>https://blog.csdn.net/Eastmount/article/details/98529597</u> 版权



网络安全自学篇专栏收录该内容

6 篇文章 1 订阅

订阅专栏

本篇文章分享实验吧CFT实战的题目,涉及WEB渗透和隐写术常见题型,包括"这是什么"、"天网管理系统"、"忘记密码"、 "false"、"天下武功唯快不破"和"隐写术之水果、小苹果"。非常有意思的文章,作为在线笔记,希望对入门的博友们有帮助,大神 请飘过,谢谢各位看官!

一.WEB之这是什么

题目地址:	http://www.shiyanbar.com/ctf/56
解题链接:	http://ctf5.shiyanbar.com/DUTCTF/1.html
题目描述:	

╱ 实验	♪ <u>○</u> ○□□ 社実验更简単	į		决斗场 U	-SaaS		
您的位	置:首页>决斗场>i	训练题库>CTF题	i库>what a fuck!这是什	-么鬼东西?			
	what a fuck!这	是什么鬼东西?	' 分值:10				
	来源: DUTCTF	难度:易	参与人数:17148人	Get Flag : 9678,	人 答题人数:10020人	解题通过率:97%	
	what a fuck!这是(解题链接: http://	十么鬼东西? /ctf5.shiyanbar.	com/DUTCTF/1.html	通过			
						提交 #dim.g. et.dvc.ixin/[2]29 2989	

打开链接如下图所示,确实是什么鬼东西。

Cttf5.shiyanbar.com/DUTCTF/1. × +	-		×
← → C ③ 不安全 ctf5.shiyanbar.com/DUTCTF/1.html	QZ	8	0

 $0[(i_{0}+i_{0})[+0]+([i_{0}]+0](0)](+i_{1}+i_{1}+i_{2})]+(i_{0}+i_{0})[i_{1}+i_{1}+i_{2}+i_{0}]+(i_{0}+i_{0})[i_{1}+i_{1}+i_{2}+i_$ [+(]]+([i])+([i]([1]) + [1](1]) + [1+1] + ([1+1])0) + (!!0 + (!)(!+!+0))(!+0) + (!+0) + (!(0) + (0)(!+1))(+0) + (!(0) + (0)(0)(!+1) + (!(0) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1) + ((0)(!+1)) + (((0)(!+1)))))))))))))))))))))+ ! + (1)(1)] + (i(1 + 1))(i + 1 + i + 0) + (i(1 + 1))(i + 0)(i + 0)(i + 0 + i + 0) + (i(1 + 1))(i + 1 + 0)(i + 1 + 0)(i + 0)(i[i+0+i+0]+([](i]+0]+([](i]+0])+([](i]+0])+([i]+0])+([i]+0])+([i]+0]+([i]+0])+([i]+0)+([i]+0])+([i]+0)+([i]+0])+([i]+0)+([i]+[+!+0]+(+(!+0)+([+1+0]+([0+0)[+0+1]+([0+0)[+0]+[i+0+i+0]+(ii0+0)(+i+0)((i+0)(+i+0)(+i+0)(+i+0)(+i+0)(+i+0)(+i+0)(+i+0)(+i+0)(+i+0)(+i+0)(+i+0[] + ([[0]] + []([1]) + [] + [] + []) + ([[0]] + ([] + []) + ([] + ([] + []) + ([] + []) + ([] + []) + ([] + []) + ([] + ([] + []) + ([] + []) + ([] + ([] + []) + ([] + ([] + []) + ([] + ([] + []) + ([] + ([] + []) + ([] + ([] + []) + ([] + ([] + []) + ([] + ([] + []) + ([] + ([] + []) + ([] + ([] + []) + ([](i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|+(i||+|)|)|+(i|||||+|)|+(i|||+|)|+(i||+|)|+(i|||+|)|+(i|||+|)|+(i|||+|)|+(()(!+0+!+0+!+0)+(!(0+0)(!+0)+(!(0+0)(!+0)+(!(0+0)(!+0)+(!(0+0)(!+1)+(!(0+0)(!+1)+(!(0+0)(!+0+1)+(!(0+0+0)(!+0+1)+(!(0+0)(!+0+1)+(!(0+0)(!+0+1)))))))(!!] + []) + (!!] + (!]) + (!!] + !] + (!] + !!] + (!] + (!]) + (!]) + (!]) + (!]) + (!] + (!]) + (!]) + (!] + (!] + (!]) + (!] + (!]) + (!] + (!]) + (!] + (!]) + (!] + (!]) + (!] + (!]) + (!] + (!] + (!]) + (!] + (!]) + (!] + (!]) + (!] + (!]) + (!] + (!] + (!]) + (!] + (!]) + (!] + (!] + (!]) + (!] + (!]) + (!] + (!] + (!]) + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!] + (!] + (!] + (!] + (!]) + (!] +(+)) + ((-))

题目解析:

1.它们其实是Jother编码,它是一种运用于Javascript语言中利用少量字符构造精简的匿名函数方法对于字符串进行的编码方式, 其中少量字符包括"[","]","{","}","(",")","!","+"。这些字符就能完成对任意字符串的编码,本质上是一种Javascript的编 码,其优点是代码字符就那么几个,比较好记,缺点是编码极其冗长和复杂。 这种编码一般现在只会出现在CTF比赛中,实际开发中用的到就很少了。

2.打开Chrome浏览器,按F12键选择控制台Console,将代码复制过去回车即可得到flag值。

Ctf5.shiyanbar.co	m/DUTCTF/1. ×	+				—]	×
\leftrightarrow \rightarrow C (i)	不安全 ctf5.sh	iyanbar.com/[OUTCTF/1.htm			Q		0	0
(ii[]+[])[i+[]+i+[]+i+] $(ii[]+[])[i+[]+i+[]+i+]$ $(ii[]+[])[i+[]+i+[]+i+]$ $(ii[]+[])[i+[]+i+]]$ $(ii]+[])[i+[]+i+]]+(ii]$ $(ii+[]+i+[]+i+]]+(ii]$ $(ii+[]+i+[]+i+]]+(ii]$ $(ii+[]+i+]]+(ii]+(ii]+]$ $(ii+[]+i+]]+(ii]+(ii]+]$ $(ii+[]+i+]]+(ii]+(ii)]$ $(ii+[]+i+]]+(ii]+(ii)]+(ii]$ $(ii+[]+i+]]+(ii]+(ii)]+(ii]$ $(ii+[]+i+]]+(ii]+(ii)]+(ii)$ $(ii+[]+i+]]+(ii]+(ii)]+(ii)$ $(ii+[]+i+]]+(ii)]+(ii)$ $(ii+[]+i+]]+(ii)$ $(ii+[]+i+])$ $(ii+[]+i+]]+(ii)$ $(ii+[]+i+])$ $(ii+[]+i+]]+(ii)$ $(ii+[]+i+])$ $(ii+[$	+ () (()) (+ ! + () + ()) (+ ! + ()) (+ ! + () + ()	$\begin{array}{c} (])] + (! [] + []) [! + ! + [] + [] \\ + (]]) [+ ! + [] + (!] \\ + (]]) [+ ! + [] + (!] \\ + (] []) [+ ! + [] + (!] \\ + (] []) [+ ! + [] + (!] \\ + (] []) [+ ! + [] + (!] \\ + (]]) [+ ! + [] + (!] \\ + (]]) [+ ! + [] \\ + (]]) [+ ! + [] \\ + (]] \\ + (]]) [+ ! + [] \\ + (]] \\ + (]] \\ + (]] \\ + (]] \\ + (]] \\ + (]] \\ + (] \\ + (]] \\ + (] \\ + (]] \\ + (] \\ + (] \\ + (]] \\ + ($	$\begin{array}{c} [] + ! + []] + (!![] + \\ []] + (!]] + (!![] + \\ []] + (!]] + (!] \\ []] + (!] + []) \\ [] + [] \\ [] + []) \\ [] $	+ [])[+[]] + (!] + (!] + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!]) + (!] + (!] + (!] + (!]) + (!] + (!] + (!] + (!] + (!] + (!]) + (!] +	$\begin{array}{c} + (1)(1 + (1) + ($	+[]+[]+ +(!![]+ +(!![]+ +(!![]+ +(!]]]+(!]])[+!+[] +[]]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!!]]+(!)[]+(!)[]+]]+(!)[]+(!)[]+]+(!)[]	!+[]]. ![]+[] ![]+[!]]+(! +[]+[]]+[]]]+(!! (![]+ [([][(! [(][(! +[]]+ [(]](!]+[])]]+[]]+	+(!! +[]+!])[!+ []+[]) [+ []+[]) []+[]) []+[]] []+[]] []+[]] []+[]] []+! []+[]] []+! []+[]]+	+
[])[+[]]+(!![]+[])[+!+	·[]]+([][[]]+[])[+	[]]+([][(![]+[])[·	+[]]+([![]]+[][[]])[+!+[]+[+	-[]]]+(![]+[)[!+[]	+!+[]]+(ii	•
🔽 📋 🛛 Elements	Console Sour	ces Network	Performance	Memory	Application	>>			×
🕩 🛇 top	• •	Filter	D	efault levels 🔻					\$
[!+[]+!+[]+!+[]]+([]]+[][[]])[+!+[]+ [+!+[]])[+!+[]+[]+ []]+([][]]+[])[+] [+!+[]+[]]+(!]]	(!![]+[])[+!+[]] +[+[]]]+(![]+[])[+[]]]+(!![]+[])[[]]+(!![]+[])[+!]]+(!![]+[])[+!]+[])[!+[]+!+[] [!+[]+!+[]]+(!! +!+[]]]((!![]+ +[]]+([][[]]+[] +(!![]+[])+[]]+[]]+]+[]]+(![]]+(] []+[])[+[]]+(!] [])[+!+[]]+(!![])[+!+[]]+(![]])[+!+[]]+(+[![]+(!![]+[])[!+[])[!+[]]]	[])[+[]]+(!! !![]+[])[!+[[]+[])[!+[]+ []]+[][(![]+ []+!+[]+!+[]	[]+[][(![]+]+!+[]+!+[] !+[]+!+[]]+ [])[+[]]+([]]+(!![]+[])	-[])[+]+(!! -(!![]- ![]]+ [+!+[[]]+([]+[]) [][]])[· [][[]]])[·	[]) +]) +!+	



3.得到该题的密码: lhatejs。

Ctf5.shiyanbar.com/DUTCTF/1. × +	_			×
← → C ③ 不安全 ctf5.shiyanbar.com/DUTCTF/1.html	Q	☆	θ	0
[][(!]+[])[+[]]+([!]] []+[])[+!+[]]]((]]([] []+!+[]]+(!![]+[])[+ []+!+[]]+(!![]+[])[+ [!+!]]+(!!]+!]]+(! []]]+(![]+[])[!+[]+!+	!+[] -(!![[]]] [+!]])[- !+[]	+!+[]+[]) +(![]+ +!+[] +!+[]]]+(!! +[])[!+ -(![]+[] -(![]+[])
[]+[])[+[]]+(!![]+(]](![]+[])[+[]]+(![]]+([![]]+([]])[+!+[]+[]+[])]+(![]+[])[!+[]+!+[]]+(!![]+(!)[])+(!)[]+(!)[])	+[]]+((!![]+	[])[!+	N
[i+[]+i+[]+i+[]]+(ii[]+[])[+i+[]]+(ii[]+[])[+i])+(ii[]+[])[+i])[+i]]+(ii[]+[])[(ii[]+[])[(ii]+[])[+i])[+i])[+i]+(ii[]+[])[i+i])[i+i])[i+i]	[+:+ [+:[]]]	- []] + (i[]	(;;[]+[] +[])[!•) 29890

正确答案: lhatejs 参考链接: https://www.writeup.top/391.html

二.隐写术之水果

隐写术是一门关于信息隐藏的技巧与科学,所谓信息隐藏指的是不让除预期的接收者之外的任何人知晓信息的传递事件或者信息 的内容。隐写术的英文叫做Steganography,来源于特里特米乌斯的一本讲述密码学与隐写术的著作Steganographia,该书书名 源于希腊语,意为"隐秘书写"。在CTF题目中,图片隐写题属于杂项的一部分,题目较为简单。 题目地址: http://www.shiyanbar.com/ctf/1903 解题链接: http://ctf5.shiyanbar.com/stega/pic.png 题目描述:

╱ 实验	ノご 記号 社家	宗验更简单!			决斗场	U-SaaS			
您的位	置:首页>	>决斗场>训练题	5库>CTF题库>z	K果					
	水果	分值:10							
	来源: 北	邮天枢战队	难度:易	参与人数:4845人	Get Flag	:2412人	答题人数:2765人	解题通过率:87%	
	flag就隐藏在这些鲜艳的水果中,仔细找就能找到 key格式:CTF{xxx}								
	解题链接	: http://ctf5.	shiyanbar.com/	/stega/pic.png	通过				



打开网页如下图所示,显示一张水果的图片,flag就隐藏在图片中。作者第一反应是查看源代码,哈哈~原谅我这个小白第一次 学习。



题目解析:

1.将图片另存为本地。

2.从CSDN下载Stegsolve工具,它是用于图像解析的工具,然后导入本地图片,按方向键右键不断切换,直到出现下图的二维码。





如下图所示:



在这里插入图片描述 (图片加载失败) 3.使用手机扫描二维码,得到一串数字,我们根据数值可以分析,这是十进制的ASCI码。

45 46 45 46 32 45 32 46 46 45 46 32 46 45 46 46 32 46 46 46 32 45 46 46 46 32 46 46 45 45 46 45 32 45 46 46 46 32 46 46 32 46 45 46 46 32 4.将数字转换为ASCII, 45对应"-"、46对应"."、32对应空格。

45 46 45 46 32 - . - . 45 32 -46 46 45 46 32 . . - . 46 45 46 46 32 . - . . 46 46 46 32 . . . 45 46 46 46 32 - . . . 46 46 45 45 46 45 32 ..--.-45 46 46 46 32 - . . . 46 46 46 32 46 45 46 46 32

5.它们就是传说中的摩斯密码。根据下面的对照表,其结果为: CTFLSB BSL

			,
a · -	1	r · - ·	1 ·
ä · - · -	1	s · · ·	2 · ·
b - • • •	k	t -	3 · · ·
C - · - ·		u · · -	4 · · · · -
ch	m	ö · ·	5 · · · ·
d - • •	n	v · · · -	6 - • • • •
e ·	0	w ·	7 · · ·
f++-+	ä.	¥ - · · -	8 · ·
g ·	0	× - · · -	9
ň · · · ·	p	7	0

摩尔斯电码(又译为摩斯密码, Morse code)是一种时通时断的信号代码,通过不同的排列顺序来表达不同的英文字母、数字和标点符号。它发明于1837年,发明者有争议,是 美国人塞缪尔·莫尔斯或者艾尔菲德·维尔。 摩尔斯电码是一种早期的数字化通信形式,但是它不同于现代只使用零和一两种状态的二进制代码,它的代码包括五种: 点、划、点和划之间的停顿、每个字符之间短的停顿、每个词之间中等的停顿以及句子之间长的停顿。

正确答案: CTF{lsb_bsl} 参考链接: https://blog.csdn.net/miko2018/article/details/81627130 https://www.cnblogs.com/nul1/p/9594387.html https://blog.csdn.net/u012486730/article/details/82016706

三.隐写术之小苹果

题目原理和上一题一样。 题目地址: http://www.shiyanbar.com/ctf/1928 解题链接: http://ctf5.shiyanbar.com/stega/apple.png 题目描述:





的位置:首页>决斗场>训	练题库>CTF题库	2>小苹果			
小苹果 分值:10)				
来源: hanyuhang	难度:易	参与人数:5922人	Get Flag : 2531人	答题人数:2833人	解题通过率:89%
flag格式: CTF{}					
解题链接: http://c	tf5.shiyanbar.co	m/stega/apple.png	通过		
					提交 dm.g.et Micin int/日25/29

题目打开也是一张图片,中国结。



题目解析:

1.下载图片至本地并打开,得到如下二维码:

StegSolve 1.3 by C	_	\times
File Analyse Help		
Red plane 4		

htt <mark>ps://disevol</mark> n.je/deixin/201999990

2.二维码包含如下数字。

\u7f8a\u7531\u5927\u4e95\u592b\u5927\u4eba\u738b\u4e2d\u5de5

这是unicode编码的方式,让我们在相关网站(搜索"unicode解码即可")中进行解码,得到中文"羊由大井夫大人王中工",这是一种从未见过的加密方式。

Unicode编码 UTF-8编码 URL编码/解码 Unix时间戳 Ascii/Nat	ive编码互知	传 Hex编码/解码			
\u7f8a\u7531\u5927\u4e95\u592b\u5927\u4eba\u738b\u4e2d\u 5	ı5de	羊由大井夫大人王中	I		
ASCII 转	Jnicode	Unicode 转 ASCII	Unicode 转中文	中文、簇、从njcode,	清空结果。 ₈₉₀

3.通过百度了解到该加密为当铺密码,曾在CTF题目中出现过,我们按照编码规则进行解码,得到数字:9158753624。

当铺密码是一种将中文和数字进行转化的密码,算法相当简单:当前汉字有多少笔画出头,就是转化成数字几。 "羊由大井夫大人王中工"对应的数字为"9158753624" 4.再回头分析图片可知,里面包含了一个压缩文件,我们通过修改扩展名为.ZIP并解压,得到了apple.mp3的音频文件。



5.使用mp3隐写术工具MP3Stego的Decode.exe对其进行解码,密码就是我们刚刚得到的那串数字9158753624。解码后得到字 串Q1RGe3hpYW9fcGluZ19ndW99。

6.通过尝试,在base64解码中得到了正确的结果: CTF{xiao_ping_guo}。 正确答案: CTF{xiao_ping_guo}

四.WEB之天网管理系统

题目地址: http://www.shiyanbar.com/ctf/1810 解题链接: http://ctf5.shiyanbar.com/10/web1/index.php 题目描述:

实验吧 让实验更简单	ŗ		决斗场	U-SaaS			
您的位置:首页>决斗场>;	训练题库>CTF题	库>天网管理系统					
天网管理系统	分值:10						
来源: zusheng	难度:易	参与人数:15031人	Get Flag : 5	334人	答题人数:5519人	解题通过率:97%	
天网你敢来挑战嘛 格式:ctf{ }							
解题链接: http://	/ctf5.shiyanbar.o	com/10/web1/ 通道	<u>1</u>				
)
						提交 約mgetWreinter/E228	129290

题目显示如下图所示,需要输入正确的用户名和密码获取flag。

天网管理系统

安全与你同在

账户:admin 密码:admin

就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。

	大家请放心使用我们的产	^左 品。	
用户名:	admin		
密码:	admin		
登入系统			

考点: PHP弱类型 题目解析: 1.查看网页源代码如下所示,注意注释的提示。

<!-- <pre>\$test=\$_GET['username']; \$test=md5(\$test); if(\$test=='0') -->

* NDOUY /
<h1>天网管理系统</h1>
<
k户:admin 密码:admin
就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。
<pre><form action="index.php" enctype="multipart/form-data" method="post"></form></pre>
▼
▼
▼/tr\
<pre></pre>
<pre> // U/ / // U/ //</pre>
► <\LF>
▶
••
\$test=\$_GET['username'];</td
Liftherman and a second and a

2.需要用户名传入一个字符串,并且它经过md5加密后要等于0。

注意,PHP某些情况会把类数值数据(如含有数字的字符串等)转换成数值处理。在使用"=="运算符对两个字符串进行比较时,PHP会把类数值的字符串转换为数值进行比较,如果参数是字符串,则返回字符串中第一个不是数字的字符之前的数字串 所代表的整数值。比如: '3' == '3ascasd'结果为true。

因此只要找到一个字串加密后第一个字符为0即可,这里提供几个: 240610708、aabg7XSs



3.用户名输入"aabg7XSs",此时返回的提示信息如下图所示。 http://ctf5.shiyanbar.com/10/web1/user.php?fame=hjkleffifer



安全与你同在

账户:admin 密码:admin

就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。

大家请放心使用我们的产品。 用户名: adminaabg7XSs 密码: admin

登入系统		
/user.php	?fame=hjkleffifer	>

https://blagsc&dhug.et&weiniet/212529896

4.访问该页面显示内容如下图所示:

函数serialize()是对输入的数据进行序列化转换,把变量和它们的值编码成文本形式。 函数unserialize()是还原已经序列化的对象,对单一的已序列化的变量进行操作,将其转换回反序列化 PHP 的值。

<pre>\$unserialize_str = \$_POST['password'];</pre>
<pre>\$data_unserialize = unserialize(\$unserialize_str);</pre>
<pre>if(\$data_unserialize['user'] == '???' && \$data_unserialize['pass']=='???') {</pre>
<pre>print_r(\$flag);</pre>
}

Ctf5.shiyanl	par.com/10/web1/u × +		—			\times
$\leftarrow \ \ \rightarrow \ \ G$	① 不安全 ctf5.shiyanba	r.com/10/web1/user.php?fame	Q	☆	θ	0

\$unserialize_str = \$_POST['password']; \$data_unserialize = unserialize(\$unserialize_str); if(\$data_unserialize['user'] == '???' && \$data_unserialize['pass']=='???') { print_r(\$flag); } 伟大 的科学家php方言道:成也布尔,败也布尔。回去吧骚年

https://blogsc&dm.g.et/weixet/Ek2529890

这段代码是将Post提交的密码值经过unserialize()函数 反序列化处理,得到一个数组,要求数组里的user和pass都等于值"???",此时输出flag。那么,这个"???"又是什么内容呢?

5.此时"成也布尔,败也布尔"提醒我们。

bool类型的true跟任意字符串可以弱类型相等。因此我们可以构造bool类型的序列化数据,无论比较的值是什么,结果都为true。(a代表array,s代表string,b代表bool,而数字代表个数/长度)

```
<?php
error_reporting(0);
$test='';
$test=array("user"=>1,"pass"=>1);
echo var_dump($test);
echo var_dump(serialize($test));
```

```
$test1='';
$test1=array("user"=>true,"pass"=>true);
echo var_dump($test1);
echo var_dump(serialize($test1));
}>
```

找个在线PHP网站进行测试,输出如下图所示: string(36) "a:2:{s:4:"user";i:1;s:4:"pass";i:1;}"



● 文本方式显示 ● html方式显示

```
array(2) {
  ["user"]=>
  int(1)
  ["pass"]=>
  int(1)
}
string(36) "a:2:{s:4:"user";i:1;s:4:"pass";i:1;}"
array(2) {
  ["user"]=>
  bool(true)
  ["pass"]=>
  bool(true)
}
string(36) "a:2:{s:4:"user";b:1;s:4:"pass";b:1;}"
```

https://blogsc&dm.g.et&veixet/2k2529890

6.构造password值为: a:2:{s:4:"user";b:1;s:4:"pass";b:1;}, 输出最后的flag。

天网管理系统

安全与你同在

账户:admin 密码:admin

就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。

大家请放心使用我们的产品。

用户名: admin

密码: 登入系	a:2:{s:4	:"user";b:1;s	:4:"pass					
ctf{dwo	duwkhduv	w5465}					2529290	
正确结果: 参考链接: https://blog. https://www	ctf{dwduwkhd csdn.net/dong .cnblogs.com/	uw5465} gyanwen6036/ /ssooking/p/58	article/details/776509 77086.html	21				
五.WE 题目地址: 解题链接: 题目描述:	B之忘记智 http://www.sh http://ctf5.shi	咨码 iiyanbar.com/c yanbar.com/10	tf/1808 //upload/step1.php					
実验吧 您的位置:首	让实验更简单! 一页>决斗场>训	练题库>CTF题®	■>忘记密码了	决斗场	U-SaaS			
志证	密码了分	重:20						
来源	: Justatest	难度:中	参与人数:18626人	Get Flag :	5829人	答题人数:6201人	解题通过率:94%	
找回: 格式 解题:	密码 :SimCTF{ } 链接: http://c	tf5.shiyanbar.c	om/10/upload/ 通	过				

https://biogsc#dimg.etwo把放在/El25329290

题目显示如下图所示,需要输入正确的邮箱找回密码。

logic	× +	- 🗆 X
$\leftarrow \rightarrow G$	① 不安全 ctf5.shiyanbar.com/10/upload/step1.php	९ 🖈 🔒 ዕ



考点: vim备份文件泄露

题目解析:

1.首先我们随便输入一个密码,如"123456"看返回结果。



返回如下图所示,注意"step2.php"页面。

你邮箱收到的重置密码链接为 ./step2.php?email=youmail@mail.com&check=??????

确定 https://bilbcsc4/Limpedwiter/compensation

2.查看源代码,发现提醒用户名为admin,输入邮箱为"admin@simplexue.com"。

html
<html></html>
▼ <head></head>
<meta charset="utf-8"/>
<meta content="IE=edge,chrome=1" http-equiv="X-UA-Compatible"/>
<meta content="webkit" name="renderer"/>
<meta content="admin@simplexue.com" name="admin"/>
<meta content="Vim" name="editor"/>
<title>logic</title>
▶ <style type="text/css"></style>
▶ <body></body>

🗋 logic			×	+		-			×
$\leftarrow \rightarrow$	C	① 不安全	ctf5.shiy	/anbar.com	/10/upload/step1.php?emailAddress=admin%40simplexue.com	Q, ·		9	0
					ctf5.shiyanbar.com 显示 邮件发到管理员邮箱了,你看不到的 确定				
					找回密码step1				
					input regist email:				
					提交				
					https://	iögscédiage	usive in inter	<u>E4259</u> 8	9890

输入该邮箱发现Scripts提醒变成了"邮箱已送到管理员邮箱了,你看不到",真是逗~

3.这里有个细节,Step2.php页面跳转了一下,然后又跳转回step1,说明step2里面有猫腻!页面跳转这么快,那我们该怎么去 看这个页面呢?这时候要用到一个名叫Burp Suite的神器,抓包拦截。 Step2.php显示立刻跳转:

logic	× +	- 🗆 X
$\leftarrow \ \ \rightarrow \ \ \mathbf{G}$	③ 不安全 ctf5.shiyanbar.com/10/upload/step2.php?email=test@test.com	९ 🖈 🔒 🕚
check error!		
	找回密码step2	

	email:	
	test@test.com	
	token:	
	提交	
正在等待 ctf5.shiyanbar.com 的响应		https://blogscscillag.estade.ixid/ <u>5428828890</u>

方法一:在Target查看目录树发现有个"submit.php"文件。

Burp Suite Free Edition v1 7 26 - Temporary Proje	ct										-	×
Burp Intruder Repeater Window Help												
Target Proxy Spider Scanner Intruder Repeater	Sequencer Decoder Co	omparer	Extender Project options	User options	Alerts							
Site map Scope												
Filter: Hiding not found items; hiding CSS, image and general	binary content; hiding 4xx respo	onses; hidin	g empty folders									?
▶ 🗍 http://ctf5.shiyanbar.com	Host	Method	URL				Params	Status 🔺	Length	MIME type	Title	
	http://ctf5.shiyanbar.com	GET	/10/upload/step1.php					200	1088	HTML	logic	
	http://ctf5.shiyanbar.com	GET	/10/upload/step1.php?ema	ilAddress=admin ⁴	%40simple	exue.com		200	1162	HTML	logic	
	http://ctf5.shiyanbar.com	GET	/10/upload/step1.php?ema	ilAddress=youma	ail%40mail	.com		200	1203	HTML	logic	
	http://ctf5.shiyanbar.com	GET	/10/upload/step2.php?ema	il=test@test.com				200	1227	HTML	logic	
	http://ctf5.shiyanbar.com	GET	/10/upload/step2.php					302	1272	HTML	logic	
	http://ctf5.shiyanbar.com	GET	/10/upload/step2.php?ema	il=youmail@mail	il.com&che	eok=???????				HTML		
	http://ctf5.shiyanbar.com	GET	/10/upload/submit.php									
												7.
	Request Response											
	Raw Params Headers	s Hex										
	<pre>GTT /10/upload/step2.ph Host: ct5.shiyanbar.co Accept: */* Accept-Language: en Connection: close</pre>	p?email≕y m	oumail@mail.com✓=?	?????? HTTP/1	1.1							
	? < + >	Type a	search term							https://blbgsc//bl	ngetidein <mark>et/</mark>	1 matches

方法二:使用Repeater,查看响应Response。

Burp Intruder Repeater Window Help Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts Intercept HTTP history WebSockets history Options Request to http://ctf5.shiyanbar.com:80 [106.2.25.10] Forward Drop Intercept is on Action Raw Params Headers Hex Send to Spider Do an active scan Send to Intruder Ctrl+1 More: ctrl>1.0 (windows IIT 10.0; Wow Accept: text/html, applicat ion/xhtml+xml, applicat ion (blocker Request in browser Send to Decoder Request in browser
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts Intercept HTTP history WebSockets history Options Options Intercept
Intercept HTTP history WebSookets history Options Image: Request to http://ctf5.shiyanbar.com:80 [106.2.25.10] Comment this item Image: Request to http://ctf5.shiyanbar.com:80 [106.2.25.10] Forward Drop Intercept is on Action Comment this item Image: Request to http://ctf5.shiyanbar.com Raw Params Headers Hex Send to Spider Do an active scan GET /10/upload/step1.php?email Address=admin*9 Send to Intruder Ctrl+1 Most: ctf5.shiyanbar.com Send to Repeater Ctrl+R Vser-Agent: Mosilation/shtml+xml, appli Send to Sequencer Accept-Language: sh'cTI, shiyanbar.com/10/upload/ Send to Comparer Send to Decoder Send to Decoder Request in browser
Request to http://ctf5.shiyanbar.com:80 [106.2.25.10] Forward Drop Intercept is on Action Raw Params Headers Hex Send to Spider Do an active scan Do an active scan Send to Intruder Ctrl+I Send to Repeater Accept: textion/xhtml+xml, applic Accept: send to Comparer Send to Comparer Send to Comparer Send to Decoder Send to Decoder Request in browser For the other series
Engagement tools [Pro version only] Change request method Change body encoding Copy URL Copy as curl command Copy to file

	Don't intercept requests Do intercept	*	
	Convert selection	►	
	URL-encode as you type		
Type a search term	Cut	Ctrl+X	https://blogsckdim.pet/weinev_9_metches

将GET方法的网址修改为step2.php,然后响应表单提交为"submit.php"。



4.赶紧查看该网页,结果提醒"you are not an admin",有权限访问该页面,但不是管理员不透露信息。有意思~

Ctf5.shiyan	oar.com/10/upload/ ×	+
$\leftarrow \ \rightarrow \ {\tt G}$	① 不安全 ctf5.shi	/anbar.com/10/upload/submit.php

you are not an admin

https://blagsc&dm.g.et/weinin/El25292990

5.再回到最初step1.php的源代码,这里有个非常重要的提示信息——编辑器采用的是VIM。

html
<html></html>
<head></head>
<meta charset="utf-8"/>
<pre><meta content="IE=edge,chrome=1" http-equiv="X-UA-Compatible"/></pre>
<meta content="webkit" name="renderer"/>
<meta content="admin@simplexue.com" name="admin"/>
<pre><meta content="Vim" name="editor"/></pre>
<script>alert("你邮箱收到的重置密码链接为 ./step2.php?email=youmail@mail.com&check=?????")</script>
<title>logic</title>
<style type="text/css">body,html{ position: relative; height 10</style>
▼ <body></body>
<pre><form action="./step1.php" method="GET"></form></pre>
<h1>找回密码step1</h1>
input pogist smail.

https://blogsc&dm.g.et/wieiniet/El2529890

```
VIM备份文件(参考Sp4rkW大神) 默认情况下使用VIM编程,在修改文件后系统会自动生成一个带~
的备份文件,某些情况下可以对其下载进行查看。例如,index.php普遍意义上的首页,它的备份文件则为index.php~。
VIM中的swp即swap文件,在编辑文件时产生,它是隐藏文件,如果原文件名是submit,则它的临时文件".submit.swp"。如果文件正常退出,则此文件自动删除。
```

这个题目叫备份文件泄露,我们知道这个VIM编辑器可以存放临时文件,而临时文件会存放信息,咱们可以尝试一下访问临时文件,格式如下:

ctf5.shiyanbar.com/10/upload/.submit.php.swp

PS: 因为vim备份文件是隐藏文件,所以需要加上一个".submit.php.swp"。

Ctf5.shiyanbar.com/10/upload/ × +	- 🗆 X	
← → C ① 不安全 ctf5.shiyanbar.com/10/upload/submit.php.swp	९ ☆ 8 0	
404		
Submit.php.swp	全部显示 ×	< 90

6.尝试打开.submit.php.swp文件。

🔚 submit. php. swp 🔀 1这一行是省略的代码..... 2 3 4 /* 5 如果登录邮箱地址不是管理员则 die() 6 数据库结构 7 8 -- 表的结构 `user` 9 10 ---11 CREATE TABLE IF NOT EXISTS `user` (12 `id` int(11) NOT NULL AUTO_INCREMENT, 13 1.4 'username' wanahaw (255) MOT NITE

```
username varchar(200) NOT NULL,
14
15
       `email` varchar(255) NOT NULL,
      `token` int(255) NOT NULL DEFAULT '0',
16
17
     PRIMARY KEY (`id`)
18
    ) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;
19
20
21
     -- 转存表中的数据 `user`
22
23
     INSERT INTO `user` (`id`, `username`, `email`, `token`) VALUES
24
    (1, '****不可见***', '***不可见***', 0);
25
26
     */
27
28
29
     ......这一行是省略的代码......
30
31
    if(!empty($token)&&!empty($emailAddress)){
32
        if(strlen($token)!=10) die('fail');
33
        if($token!='0') die('fail');
34
        $sql = "SELECT count(*) as num from `user` where token='$token' AND email='$emailAddress'";
35
        $r = mysql_query($sql) or die('db error');
36
        $r = mysql_fetch_assoc($r);
37
        $r = $r['num'];
38
        if($r>0){
39
            echo $flag;
        }else{
40
           echo "失败了呀";
41
42
        3
43
    }
44
45
```

重点是后面的if判断语句,这个条件必须要满足token的长度必须等于10,并且token的值为0,咱们可以构造十个0试试。

7.最终构造的结果如下:

http://ctf5.shiyanbar.com/10/upload/submit.phpemailAddress=admin@simplexue.com&token=000000000

🗋 ctf5.shiyanb	bar.com/10/upload/ × +	
\leftrightarrow \rightarrow G	① 不安全 ctf5.shiyanbar.com/10/upload/submit.php?emailAddress=admin@simplexue.com&tol	(en=0000000000
flag is SimCTF{	{huachuan_TdsWX}	
正确答案: flag 参考链接:	g is SimCTF{huachuan_TdsWX}	

https://www.cnblogs.com/ECJTUACM-873284962/p/7860788.html https://blog.csdn.net/wy_97/article/details/76559354

六.WEB之false

题目地址: http://www.shiyanbar.com/ctf/1787 解题链接: http://ctf5.shiyanbar.com/web/false.php 题目描述:

文章 实验吧 让实验更简单!	决斗场 U-Sa	aaS	
您的位置:首页>决斗场>训练题库>CTF题库>FALSE			
FALSE 分值:10			
来源: iFurySt 难度:易 参与人数:10141人	Get Flag:5543人	答题人数:5599人	解题通过率:99%
PHP代码审计 hint:sha1函数你有认真了解过吗?听说也有人用md5a 格式:CTF{}	碰撞o(′ □ ╰)o		
解题链接: http://ctf5.shiyanbar.com/web/false.php	通过		
			<mark>提交</mark> gsc&dim.g et&deixid(日25 129290

题目显示如下图所示。

PHP Code Audit × +	- 🗆 X
← → C ① 不安全 ctf5.shiyanbar.com/web/false.php	९ 🖈 🔒 🖸
Login first!	
Login	
View the source code	
http:	s://didgsc&dim.get&deixin/2028288890

考点: PHP代码审计 (PHP Code Audit) 题目解析: 1.首先随便输入内容,点击"Login"按钮。

PHP Code Audit × +	- 🗆 X
← → C ▲ 不安全 ctf5.shiyanbar.com/web/false.php?name=1&password=2	॰ ९ 🖈 🔒 🗘
Invalid password.	
1 •	
Login	
View the source code	

2.点击"View the source code"获取源代码如下所示。

```
<?php
if (isset($_GET['name']) and isset($_GET['password'])) {
    if ($_GET['name'] == $_GET['password'])
        echo '<p>Your password can not be your name!';
    else if (sha1($_GET['name']) === sha1($_GET['password']))
        die('Flag: '.$flag);
    else
        echo 'Invalid password.';
}
else{
    echo 'Login first!';
}
```

它的含义是GET获取name和password,然后进行判断。

(1) if (\$_GET['name'] == \$_GET['password']),用户名和密码相等,提示如下。

```
ctf5.shiyanbar.com/web/false.php?name=1&password=1
```

Your password can not be your name!

Login //blogscs/dm.p.at/weixist/El2529290

(2) else if (sha1(\$_GET['name']) === sha1(\$_GET['password'])),用户名名和密码的sha1加密散列值相等,执行die函数。

(3) 以上都不是返回"Invalid password"。

(4) 未输入用户名和密码,提示"Login first"。

3.函数说明:

•die()函数:停止程序运行,输出内容

- sha1()函数:计算字符串 "Hello" 的 SHA-1 散列。默认的传入参数类型是字符串型
- isset()函数: 检测变量是否已设置并且非 NULL。

• 若变量不存在则返回 FALSE,若变量存在且其值为NULL,也返回 FALSE,若变量存在且值不为NULL,则返回 TURE。同时 检查多个变量时,每个单项都符合上一条要求时才返回 TRUE,否则结果为 FALSE。

参考官网: https://www.php.net/manual/zh/function.isset.php

```
<?php
$a = array ('test' => 1, 'hello' => NULL, 'pie' => array('a' => 'apple'));
var dump(isset($a['test']));
                                   // TRUE
var_dump(isset($a['foo']));
                                     // FALSE
var_dump(isset($a['hello']));
                                     // FALSE
// 键 'hello' 的值等于 NULL,所以被认为是未置值的。
// 如果想检测 NULL 键值,可以试试下边的方法。
var_dump(array_key_exists('hello', $a)); // TRUE
// Checking deeper array values
var_dump(isset($a['pie']['a']));
                                     // TRUE
var_dump(isset($a['pie']['b']));
                                     // FALSE
var_dump(isset($a['cake']['a']['b'])); // FALSE
?>
```

4.这里需要执行"if (sha1(\$_GET['name']) === sha1(\$_GET['password']))"语句。

重点: sha1()函数默认的传入参数类型是字符串型,也可以传入其他类型,使其返回值为false,如数组类型。再加上题目标题false,可以想到构造FALSE===FALSE拿到flag。

==: 比较运算符号 不会检查条件式的表达式的类型 ===: 恒等计算符,同时检查表达式的值与类型。

构造网址:

5.name和password为数组,并且值不相等,提交即可获得flag。



Flag: CTF{t3st_th3_Sha1}

https://blogsc/scha.g.et/weinint/Ek252929290

正确结果: Flag: CTF{t3st_th3_Sha1}

七.WEB之天下武功唯快不破

题目地址: http://www.shiyanbar.com/ctf/1854 解题链接: http://ctf5.shiyanbar.com/web/10/10.php 题目描述:

シシシション	2 P 让实验更简单	Ê !		决斗场	U-SaaS	i		
您的位置	閏:首页>决斗场>	训练题库>CTF题	ī库>FALSE					
	FALSE 分值:	10						
	来源: iFurySt	难度:易	参与人数:10141人	Get Flag : 55	43人	答题人数:5599人	解题通过率:99%	
	PHP代码审计 hint:sha1函数你 格式:CTF{}	有认真了解过吗	?听说也有人用md5碰撞	≨o(╯ □ ╰)o				
i	解题链接: http://	/ctf5.shiyanbar.	.com/web/false.php	通过				
)
							提交 gsc&dm.getXdeix(n//212)	929890

题目显示如下图所示,提醒"You must do it as fast as you can!"。

Ctf5.shiyanb	ar.com/web/10/10 × +			
$\leftarrow \ \ \rightarrow \ \ \mathbf{G}$	→ C ③ 不安全 ctf5.shiyanbar.com/web/10/10.php			

There is no martial art is indefectible, while the fastest speed is the only way for long success. >>>>>>----You must do it as fast as you can!----<<<<<

https://blagsc&dhug.et&weixiet/Ek2529890

考点: Python脚本 题目解析: 1.尝试SQL注入都无反应,接着查看源代码,发现一个提示信息:

<!-- please post what you find with parameter:key -->

2.根据题目内容,试图将网页链接速度放慢,这里可以采Burp Suite抓包,Proxy的intercept载入网页,并将抓到的信息发到 repeater中Go一下,会发现一个FLAG值。另一种方法,Chrome浏览器审查网络状态。

 	C ctf5 shivanbar.com/web/10/10 × +						-	-		×
There is no martial art is indefectible, while the fastest speed is the only way for long success. >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	← → C ③ 不安全 ctf5.shiyanbar.com	/web/10/10.php					Q	☆	Θ	0
Image: Console Sources Network Performance Memory Application Security Audits : × Image: Console Sources Network Performance Memory Application Security Audits : × Image: Console Sources Network Performance Memory Application Security Audits : × Image: Console Sources Network Performance Memory Application Security Audits : × Image: Console Sources Network Performance Memory Application Security Audits : × Image: Console Sources Network Performance Memory Application Security Audits : × Image: Console Sources Network Performance Memory Application Doc WS Manifest Other : × Image: Console Sources Network Performance Source Network Performance Source Network Performance Source Accept: text/html, application/xmt];q=0.9, image/webp, image/appn;*/*;q=	There is no martial art is indefectible, while the fastest speed is the only way for long success.									
R Elements Console Sources Network Performance Memory Application Security Audits I X Image: Source in the security View: Image: Source in the security Group by frame Preserve log Disable cache Offline Online Image: Source in the security Image:										
Image: Server in ginx/1.10.2 View: Image: Server in ginx/1.10.2 Image: Server in ginx/1.10.2 Content-Type: text/html in application/xhtml+xml, application/xml;q=0.9, image/appg,*/*;q=	🕞 🖬 🛛 Elements Console Sources Network	Performance	Memory Applic	ation Security	Audits				:	×
Filter Hide data URLs All XHR JS CSS Ing Media For Doc WS Manifest Other 10 ms 20 ms 30 ms 40 ms 50 ms 60 ms 70 ms 80 ms 90 ms 100 ms 110 Name X Headers Preview Response Cookies Timing 10.php X Headers Preview Response Cookies Timing 10.php X Headers Preview Response Cookies Timing 10.php X Headers Preview Response Cookies Timing Server: ninx1.10.2 Transfer-Encoding: chunked X-Powered-By: PHP/5.5.38 Fequest Headers view source Accept: text/html, application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q= Fequest Headers view source	🔎 🛇 🛛 🎟 🍸 🔍 View: 📰 🛬 🗔 Group by frame 📄 Preserve log 🗔 Disable cache 📄 Offline Online 🔹 🔻									
10 ms 20 ms 30 ms 40 ms 50 ms 60 ms 70 ms 80 ms 90 ms 100 ms 110 Name X Headers Preview Response Cookies Timing I 10,php X Headers Preview Response Cookies Timing I 10,php X Headers Preview Response Cookies Timing I 10,php X Headers Preview Response Cookies Timing I I I Date: Tue, 06 Aug 2019 10:48:54 GMT FLAG: UDBTVF9USEITXIQwX0NINE5HRV9GTDRHOnZhbmRmQXp1Zg== Server: Image: Content-Type: Server:	Filter 🗌 Hide data URLs All X	HR JS CSS Img M	ledia Font Doc V	VS Manifest Ot	her					
Name × Headers Preview Response Cookies Timing I 10.php Content-Type: text/html Date: Tue, 06 Aug 2019 10:48:54 GMT FLAG: UDBTVF9USEITX1QwX0NINE5HRV9GTDRH0nZhbmRmQXp1Zg== Server: nginx/1.10.2 Transfer-Encoding: chunked X-Powered-By: PHP/5.5.38 View source Accept: text/html,application/xml;q=0.9,image/webp,image/apng,*/*;q=	10 ms 20 ms 30 ms	40 ms 50	ms 60 ms	70 ms	80 ms	90 ms		100 ms		110
Name × Headers Preview Response Cookies Timing 10.php Content-Type: text/html Date: Tue, 06 Aug 2019 10:48:54 GMT FLAG: UDBTVF9USE1TX1QwX0NINE5HRV9GTDRHOnZhbmRmQXp1Zg== Server: nginx/1.10.2 Transfer-Encoding: thuked X-Powered-By: PHP/5.5.38 Request Headers view source Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q= Tuest of text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=										
Content-Type: text/html Date: Tue, 06 Aug 2019 10:48:54 GMT FLAG: UDBTVF9USEITX1QwX0NINE5HRV9GTDRH0nZhbmRmQXp1Zg== Server: nginx/1.10.2 Transfer-Encoding: chunked X-Powered-By: PHP/5.5.38 Request Headers view source Accept: text/html,application/xml;q=0.9,image/webp,image/apng,*/*;q=	Name	× Headers P	review Response	Cookies Tir	ning					
Date: Tue, 06 Aug 2019 10:48:54 GMT FLAG: UDBTVF9USEITX1QwX0NINE5HRV9GTDRHOnZhbmRmQXp1Zg== Server: nginx/1.10.2 Transfer-Encoding: chunked X-Powered-By: PHP/5.5.38 ▼ Request Headers view source Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=	10.php Content-Type: text/html						•			
FLAG: UDBTVF9USElTX1QwX0NINE5HRV9GTDRH0nZhbmRmQXp1Zg== Server: nginx/1.10.2 Transfer-Encoding: chunked X-Powered-By: PHP/5.5.38 ▼ Request Headers view source Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=		Date: Tue, 06 Aug 2019 10:48:54 GMT								
Server: nginx/1.10.2 Transfer-Encoding: chunked X-Powered-By: PHP/5.5.38 Request Headers view source Accept: text/html,application/xml;q=0.9,image/webp,image/apng,*/*;q=		FLAG: UDBTVF	FLAG: UDBTVF9USElTX1QwX0NINE5HRV9GTDRHOnZhbmRmQXp1Zg==							
Iransfer-Encoding: chunked X-Powered-By: PHP/5.5.38 Request Headers view source Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=		Server: nginx/1.10.2								
Request Headers view source Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=		X-Powered-By: PHD/5 5 38								
<pre></pre>									-	
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/appg,*/*;q=	• Request Headers view source									
Trequests 1.4b4 B transferred 1 Einish: YU ms 1.10 M(onte a g	= 529255									

3.在响应头中发现了FLAG,看起来像是一个Base64编码,尝试在线解码。

在线加密解密(采用Crypto-JS实现) ▲ Feedback						
加密/解密 散列/哈希 BASE64	图片/BASE64转换					
明文: POST_THIS_T0_CH4NGE_FL4G:vandfA	zuf BA	BASE64: UDBTVF9USEITX1QwX0NINE5HRV9GTDRHOnZhbmRmQXp1Zg== ASE64编码				

但是该值每次生成的值是随机的。

FLAG: UDBTVF9USEITX1QwX0NINE5HRV9GTDRHOnZhbmRmQXp1Zg==

解码: POST_THIS_T0_CH4NGE_FL4G:vandfAzuf

FLAG: UDBTVF9USEITX1QwX0NINE5HRV9GTDRHOktsSVBLWmVkOQ==

解码: P0ST_THIS_T0_CH4NGE_FL4G:KIIPKZed9

4.回想之前的注释(please post what you find with parameter:key)以及解密后的FLAG值,需要快速提交POST,故采用Python 脚本实现。哈哈,又回到熟悉的语言。

```
#-*- coding: utf8 -*
import requests
import base64
url = 'http://ctf5.shiyanbar.com/web/10/10.php'
s = requests.session()
response = s.get(url)
#获取FLAG值
#FLAG: UDBTVF9USELTX1QwX0NINE5HRV9GTDRHOnZhbmRmQXp1Zg==
head = response.headers
flag = base64.b64decode(head['FLAG']).split(':')[1]
print(flag)
#设置POST请求
pdata = {'key': flag}
result = s.post(url=url, data=pdata)
print(result.text) #响应
```

5.运行得到如下结果。

>>> MDt0Wr8ys CTF{Y0U_4R3_1NCR3D1BL3_F4ST!} >>>

正确答案: CTF{Y0U_4R3_1NCR3D1BL3_F4ST!}

参考链接: https://www.jianshu.com/p/11b5c1bd62d0 https://blog.csdn.net/miko2018/article/details/83314088 https://blog.csdn.net/dongyanwen6036/article/details/77358693