

# CTF web 题型解题技巧

## 第一课 思路讲解

CTF web 题型解题技巧 .....	1
工具集: .....	3
常用套路总结 .....	4
直接查看网页源码, 即可找到 flag .....	4
robots.txt .....	4
查看 http 请求/响应 .....	4
不常见类型的请求发送 .....	5
HTTP 头相关的题目 .....	5
修改请求头、伪造 Cookie .....	5
ctf 之流量分析: .....	5
ctf 之日志审计: .....	6
Webshell: .....	6
web 源码泄漏: .....	6
vim 源码泄漏(线上 CTF 常见) .....	6
恢复文件 vim -r index.php, 备份文件泄漏 .....	6
.git 源码泄漏 .....	6
svn 导致文件泄漏 .....	7
Git 源码泄露 .....	7
编码和加解密, 各类编码和加密 .....	7
windows 特性, 短文件名 .....	7
php 弱类型 .....	7
PHP 伪协议 .....	8
绕 waf .....	8
python 爬虫信息处理 .....	8
PHP 代码审计 .....	9
数组返回 NULL 绕过 .....	9
正则表达式相关 .....	9
ereg 正则%00 截断 .....	9
数组绕过 .....	9
单引号绕过 preg_match()正则匹配 .....	9
命令执行漏洞 .....	10
XSS 题目 .....	10
绕过 waf .....	10
长度限制 .....	10
双写 .....	10
等价替代 .....	11
URL 编码绕过 .....	11
Linux 命令使用反斜杠绕过 .....	11

---

URL 二次解码绕过.....	11
数组绕过.....	11
上传绕过.....	11
SQL 注入.....	12
使用 sqlmap.....	12
爆破: .....	12
Python 直接上脚本: .....	12

公众号: LemonSec

---

## 工具集:

基础工具: Burpsuite, python, firefox(hackbar, foxypoxy, user-agent, swither 等)

\*\*\*了解 Burpsuite 的使用方式, 参考《BurpSuite 使用说明》、firefox(hackbar, foxypoxy, user-agent, swither 等)插件的使用给漏洞挖掘带来便利,  
《必不可少的 Firefox 插件》[https://blog.csdn.net/weixin\\_35934768/article/details/80940179](https://blog.csdn.net/weixin_35934768/article/details/80940179)

扫描工具: nmap: Nmap (网络映射器) 是 Gordon Lyon 最初编写的一种安全扫描器, 用于发现计算机网络上的主机和服务, 从而创建网络的“映射”。为了实现其目标, Nmap 将特定数据包发送到目标主机, 然后分析响应。NMAP 强大的网络工具, 用于枚举和测试网络。

网上针对 nmap 的学习教程很多: <https://blog.csdn.net/m1585761297/article/details/80015726>

Nessus: Nessus 号称是世界上最流行的漏洞扫描程序, 全世界有超过 75000 个组织在使用它。该工具提供完整的电脑漏洞扫描服务, 并随时更新其漏洞数据库。Nessus 不同于传统的漏洞扫描软件, Nessus 可同时在本机或远端上遥控, 进行系统的漏洞分析扫描。Nessus 也是渗透测试重要工具之一。

网上针对 Nessus 的学习教程很多: [https://blog.csdn.net/weixin\\_41260116/article/details/88787917](https://blog.csdn.net/weixin_41260116/article/details/88787917)

Openvas: 《openvas 使用入门》<https://www.pianshen.com/article/6269172240/>

\*\*\*了解 nmap 等扫描工具的使用。

sql 注入工具: sqlmap 等, 可以参考我《sqlmap 用户手册》

\*\*\*注入在 CTF WEB 中比较常见, 通过暴库找到 flag

xss 平台: xssplatform: XSS Platform 是一个非常经典的 XSS 渗透测试管理系统《Web 安全之 XSS Platform 搭建及使用实践》<https://www.cnblogs.com/ichunqiu/p/10102531.html>

beef: BeEF-XSS 是一款非常强大的 web 框架攻击平台, 集成了许多 payload, 可以实现许多功能!

《BeEF-XSS 详细使用教程》<https://blog.csdn.net/smling/article/details/106067842>

\*\*\*利用 xss 弹 cookie 的方式弹出 flag

文件上传工具: cknife: Cknife 中国菜刀的使用, 这百度教程很多;

暴力破解工具: burp 暴力破解模块,

md5Crack: MD5Crack 是一款老牌的 md5 解密软件, md5 密码破解软件。MD5Crack 的破解速度及快! 并且支持批量破解, 保存进度和特有的插件等功能。

Hydra: 《爆破工具 Hydra 简单使用》: <https://www.jianshu.com/p/4da49f179cee>

---

## 常用套路总结

### 直接查看网页源码，即可找到 flag

考察基本的查看网页源代码、HTTP 请求、修改页面元素等。  
这些题很简单，比较难的比赛应该不会单独出，就算有因该也是 Web 的签到题。  
实际做题的时候基本都是和其他更复杂的知识结合起来出现。  
姿势：恶补基础知识就行

\*\*\*按 F12 就都看到了，flag 一般都在注释里，有时候注释里也会有一条 hint 或者是对解题有用的信息。

例题：《005-CTF web 题型总结-第五课 CTF WEB 实战练习(一)》bugku-ctf 第一题：web2、bugku-ctf 第二题：计算器、

### robots.txt

例题

《006-CTF web 题型总结-第六课 CTF WEB 实战练习(二)》bugku-ctf 第一题：细心、

例题：

《007-CTF web 题型总结-第七课 CTF WEB 实战练习(三)》bugku-ctf 第二题：phpcmsV9

### 查看 http 请求/响应

使用 burp 查看 http 头部信息，修改或添加 http 请求头（referer--来源伪造，x-forwarded-for--ip 伪造，user-agent--用户浏览器，cookie--维持登陆状态，用户身份识别）

可以用 hackbar，有的也可以写脚本

Get 和 post

例题：《005-CTF web 题型总结-第五课 CTF WEB 实战练习(一)》bugku-ctf 第三题：web 基础\$\_GET、bugku-ctf 第四题：web 基础\$\_POST

响应：

例题：《005-CTF web 题型总结-第五课 CTF WEB 实战练习(一)》bugku-ctf 第八题：你必须让他停下、bugku-ctf 第十一题：头等舱

---

## 不常见类型的请求发送

以前做过一道题考 OPTIONS 请求，可惜题目找不到了，而且那道题也不算很基础。不过如果要发送这类请求，写一个脚本应该就能解决了

## HTTP 头相关的题目

主要是查看和修改 HTTP 头。

姿势：不同的类型有不同的利用方法，基本都离不开抓包，有些简单的也可以利用浏览器 F12 的网络标签解决。但是最根本的应对策略，是

熟悉一些常见请求头的格式、作用等，这样考题目的时候就很容易知道要怎么做了。

查看相应头

有时候响应头里会有 hint 或者题目关键信息，也有时候会直接把 flag 放在响应头里给，但是直接查看响应头拿 flag 的题目不多，因为太简单了。

知识查看的话，可以不用抓包，用 F12 的“网络”标签就可以解决了。

## 修改请求头、伪造 Cookie

常见的有 set-cookie、XFF 和 Referer，总之考法很灵活，做法比较固定，知道一些常见的请求头再根据题目随机应变就没问题了。

有些题目还需要伪造 cookie，根据题目要求做就行了。

可以用 Burp 抓包，也可以直接在浏览器的 F12“网络”标签里改。

域名解析，伪造 host

例题：《005-CTF web 题型总结-第五课 CTF WEB 实战练习(一)》bugku-ctf 第七

题：域名解析、bugku-ctf 第十三题：管理员系统、

## ctf 之流量分析：

流量分析中 wireshark 的使用在下面介绍

例题

《006-CTF web 题型总结-第六课 CTF WEB 实战练习(二)》bugku-ctf 第五题：flag

被盗、bugku-ctf 第六题：这么多数据包、

例题：

《007-CTF web 题型总结-第七课 CTF WEB 实战练习(三)》bugku-ctf 第一题：特殊的后门

---

## ctf 之日志审计:

例题

《006-CTF web 题型总结-第六课 CTF WEB 实战练习(二)》bugku-ctf 第四题: 日志审计、

## Webshell:

通过扫描后台路径, 发现他人留下的 webshell;

例题: 例题: 《005-CTF web 题型总结-第五课 CTF WEB 实战练习(一)》bugku-ctf 第十二题: 网站被黑

## web 源码泄漏:

vim 源码泄漏(线上 CTF 常见)

如果发现页面上有提示 vi 或 vim, 说明存在 swp 文件泄漏, 地址: /.index.php.swp 或 index.php~

恢复文件 vim -r index.php, 备份文件泄漏

地址: index.php.bak, www.zip, htdocs.zip, 可以是 zip, rar, tar.gz, 7z 等

例题:

《005-CTF web 题型总结-第五课 CTF WEB 实战练习(一)》bugku-ctf 第五题: 备份是个好习惯

例题:

《007-CTF web 题型总结-第七课 CTF WEB 实战练习(三)》bugku-ctf 第三题: bugku

导航

## .git 源码泄漏

地址: <http://www.xxx.com/.git/config>, 工具: GitHack, dvcs-ripper

---

## svn 导致文件泄漏

地址: <http://www.xxx.com/.svn/entries>, 工具: dvcs-ripper, seay-svn

## Git 源码泄露

flag 一般在源码的某个文件里, 但也有和其他知识结合、需要进一步利用的情况, 比如 XCTF 社区的 mfw 这道题。

姿势: GitHack 一把梭

## 编码和解密, 各类编码和加密

可以使用在线工具解密, 解码

例题: 《005-CTF web 题型总结-第五课 CTF WEB 实战练习(一)》bugku-ctf 第六题: web3、bugku-ctf 第一题: web4 (看看源代码吧)、

《006-CTF web 题型总结-第六课 CTF WEB 实战练习(二)》bugku-ctf 第三题: 速度要快、bugku-ctf 第四题: cookies 欺骗、bugku-ctf 第五题: never give up

## windows 特性, 短文件名

利用 ~ 字符猜解暴露短文件/文件夹名, 如 backup-81231sadasdasasfa.sql 的长文件, 其短文件是 backup~1.sql, iis 解析漏洞, 绕过文件上传检测

## php 弱类型

php 弱类型第二课会涉及。

例题:

《005-CTF web 题型总结-第五课 CTF WEB 实战练习(一)》bugku-ctf 第五题: 矛盾

---

## PHP 伪协议

在 CTF 中经常出现，也经常跟文件包含，文件上传，命令执行等漏洞结合在一起。

php 伪协议在第二课会涉及；

例题：

例题：

《005-CTF web 题型总结-第五课 CTF WEB 实战练习(一)》bugku-ctf 第二题：flag  
在 index 里

例题：

《006-CTF web 题型总结-第六课 CTF WEB 实战练习(二)》bugku-ctf 第七题：web8  
(txt? ? ? ? )

## 绕 waf

大小写混合，使用编码，使用注释，使用空字节

## python 爬虫信息处理

这类题目一般都是给一个页面，页面中有算式或者是一些数字，要求在很短的时间内求出结果并提交，如果结果正确就可以返回 flag。

因为所给时间一般都很短而且计算比较复杂，所以只能写脚本。这种题目的脚本一般都需要用到 requests 库 BeautifulSoup 库（或者 re 库（正则表达式）），个人感觉使用 BeautifulSoup 简单一些。

姿势：requests 库和 BeautifulSoup 库熟练掌握后，再多做几道题或者写几个爬虫

的项目，一般这类题目就没什么问题了。主要还是对 BeautifulSoup 的熟练掌握，

另外还需要一点点 web 前端（html）的知识。



---

## PHP 代码审计

代码审计覆盖面特别广，分类也很多，而且几乎什么样的比赛都会有，算是比较重要的题目类型之一吧。

姿势：具体问题具体分析，归根结底还是要熟练掌握 PHP 这门语言，了解一些常见的会造成漏洞的函数及利用方法等。

### 数组返回 NULL 绕过

PHP 绝大多数函数无法处理数组，向 md5 函数传入数组类型的参数会使 md5()函数返回 NULL（转换后为 False），进而绕过某些限制。如果上面的代码变成：

```
if(md5($a) === md5($b)) {           //两个等号变成三个
    echo $flag;
}
```

那么利用弱类型 hash 比较缺陷将无法绕过，这时可以使用数组绕过。传入?a[]=1&b[]=2 就可以成功绕过判断。这样的方法也可以用来绕过 sha1()等 hash 加密函数相关的判断，也可以绕过正则判断，可以根据具体情况来灵活运用。

例题：《005-CTF web 题型总结-第五课 CTF WEB 实战练习(一)》bugku-ctf 第九题：  
变量 1

### 正则表达式相关

#### ereg 正则%00 截断

ereg 函数存在 NULL 截断漏洞，使用 NULL 可以截断过滤，所以可以使用%00 截断正则匹配。  
Bugku ereg 正则%00 截断：<http://123.206.87.240:9009/5.php>

#### 数组绕过

正则表达式相关的函数也可以使用数组绕过过滤，绕过方法详见数组返回 NULL 绕过。  
上面那道题也可以用数组绕过。

#### 单引号绕过 preg\_match()正则匹配

在每一个字符前加上单引号可以绕过 preg\_match 的匹配，原理暂时不明。

---

## 命令执行漏洞

### assert()函数引起的命令执行

assert 函数的参数为字符串时，会将字符串当做 PHP 命令来执行。例如：assert('phpinfo()')相当于执行<?php phpinfo() ?>

## XSS 题目

这类题目会涉及到三种 XSS 类型，具体类型要根据题目来判断。一般都是向后台发送一个带有 XSSPayload 的文本，在返回的 Cookie 中含有 flag，解法是在 XSS Payload。这类题目一般都会带有过滤和各种限制，需要了解一些常用的绕过方法。姿势：XSS 归根结底还是 JavaScript，JavaScript 的威力有多大，XSS 的威力就有多大。要知道一些常用的 XSSPayload，还要把三类 XSS 的原理弄明白。做题时需要用到 XSS 平台，网上有公用的，也可以自己在 VPS 上搭一个。

JavisOJ babyxss: <http://web.jarvisoj.com:32800/>

## 绕过 waf

其实绝大多数比较难的题目多多少少都会对输入有过滤，毕竟在现实的网络中肯定是对输入进行限制的，但是这里还是把过滤单独列出来了。姿势：多掌握一些不同的绕过方法。

## 长度限制

有些题目会要求输入较长的文本，但对文本的长度进行了限制。对于这种题目，既可以用 BurpSuite 抓包改包绕过，也可以直接在 F12 里改页面源代码。

Bugku 计算器（修改页面源代码）：<http://123.206.87.240:8002/yanzhengma/>

DVWA 存储型 XSS 的标题栏会对长度进行限制，使用 BurpSuite 抓包绕过。

## 双写

双写可以绕过对输入内容过滤的单个判断，在 XSS、SQL 注入和 PHP 代码审计的题目中比较常见。双写顾名思义就是将被过滤的关键字符写两遍，比如，如果要添加 XSSPayload，又需要插入<script>标签，就可以构造如下的 Payload：<scr<script>ipt>来绕过对<script>标签的单个过滤限制。这样的方法不仅对 XSS 有用，也可以用于代码审计和 SQL 注入。

HGAME2019 有一道 XSS 题目就是过滤了<script>，可以用双写绕过。

---

## 等价替代

就是不用被过滤的字符，而使用没有被过滤却会产生相同效果的字符。比如，如果 SQL 注入题目中过滤了空格，可以用 `/**/` 绕过对空格的限制；XSS 题目如果过滤了 `<script>` 标签，可以使用其他类型的 payload；如果需要使用 `cat` 命令却被过滤，可以使用 `tac`、`more`、`less` 命令来替代等。

## URL 编码绕过

如果过滤了某个必须要用的字符串，输入的内容是以 GET 方式获取的（也就是直接在地址栏中输入），可以采用 url 编码绕过的方式。比如，过滤了 `cat`，可以使用 `c%61t` 来绕过。

## Linux 命令使用反斜杠绕过

在 Linux 下，命令中加入反斜杠与原命令完全等价。例如，`cat` 与 `ca\t` 两条命令等价，效果完全相同。可以利用这个特性来进行一些绕过操作（当然，这个仅限于命令执行漏洞）。

## URL 二次解码绕过

这个类型本来应该放在代码审计里面，但是既然是一种绕过过滤的姿势，就写在这里了。如果源码中出现了 `urldecode()` 函数，可以利用 url 二次解码来绕过。以下是一些常用的 HTML URL 编码：

## 数组绕过

详见 PHP 代码审计的“数组返回 NULL”绕过。数组绕过的应用很广，很多题目都可以用数组绕过。

## 上传绕过

例题：

《006-CTF web 题型总结-第六课 CTF WEB 实战练习(二)》bugku-ctf 第二题：求 getshell

---

## SQL 注入

SQL 注入是一种灵活而复杂的攻击方式，归根结底还是考察对 SQL 语言的了解和根据输入不同数据网页的反应对后台语句的判断，当然也有 sqlmap 这样的自动化工具可以使用。姿势：如果不用 sqlmap 或者用不了，就一定要把 SQL 语言弄明白，sqlmap 这样的自动化工具也可以使用。

### 使用 sqlmap

sqlmap 的应用范围还不大明确，我都是如果 sqlmap 没法注入就手工注入。

sqlmap 教程：<https://www.jianshu.com/p/4509bdf5e3d0>

#### 例题

《006-CTF web 题型总结-第六课 CTF WEB 实战练习(二)》bugku-ctf 第一题：成绩单、bugku-ctf 第三题：多次

### 爆破：

这个可以直接看例题

#### 例题：

《005-CTF web 题型总结-第五课 CTF WEB 实战练习(一)》bugku-ctf 第三题：输入密码查看 flag

### Python 直接上脚本：

#### 例题

《006-CTF web 题型总结-第六课 CTF WEB 实战练习(二)》bugku-ctf 第二题：秋名山老司机