php upload ctf,强网杯CTF防御赛ez_upload Writeup

转载

一只御姐姐 于 2021-03-17 02:02:38 发布 210 收藏
文章标签： php upload ctf

这是强网杯拟态防御线下赛遇到的web题目，本来是不打算分享Writeup的，但是由于问的人很多，于是这里分享给大家。

ez_upload这题算是非常经典的堆叠black trick的题目，算是比较典型的ctf式题目(虽然现在大家都很抵制这样的题目)，这里主要是分享Writeup以及我们队在完成题目时的思考流程。

ez_upload 思考流程

最开始我先描述一下题目逻辑。

1、login.php，登陆页面，只获取了username，没有任何限制，username会在转义后进入session。

2、index.php，页面输出了username，ip(可以被xff覆盖)，以及上传文件列表(不完整，只有10位)。

3、upload.php，上传文件，要求必须上传php，但是又过滤很多，没办法绕过限制。

在拿到题目后，我们可以得到以下信息：

1、登陆无任何限制，只输入用户名，但单引号、双引号、反斜杠会被转义。aaa' => aaa\'

1)hint提到数据库中username的长度为25

2、登陆后，index.php获取ip，这个ip可以被xff覆盖，而且是每次都会获取。

1)xff受到waf限制，形似。

$ip = get_ip_from_xff();

echo $ip;

waf($ip);

但这里只拦截包括单引号、反斜杠

3、上传文件，要求必须上传php，但会被waf拦截。

1)代码形似：

waf($_FILES);

所以和ip那里触发不一致

2)看上去对php的验证在前，在最早的测试中，只有在触发waf的情况下才能被认为是php(猜测)

\n.....

这里的判断看上去完全一致，像是个悖论

3、上述中所有提到的变量，在输出前都是从session里面取得，但提示中数据库存在。

那么猜测有两个数据库操作点。

1、index.php查看文件列表，select filename from uploads where user = '\$user' and ip = "\$ip"? # ip是否参与未知

2、upload.php上传文件，insert into uploads values (id, '\$user', '\$ip', '\$filename')...

###########猜测分割线###########

在最早分析完题目后，由于我们没办法绕过上传，所以重新思考了所有的条件。于是有了下面的猜测：

猜测这里存在二次注入，通过user25位阶段对\的转义，然后转义单引号，这样与下一个单引号闭合，于是完成insert注入。

猜测为注入题目...

文件如果被上传，那么一定可以被index.php看到，那么我们需要假设这个文件一定可以被上传。

但我们传不了，那么有两种假设，有我们忽略的条件或者black trick。

而忽略的条件只有waf(尤其是ip上的

假设ip上的waf是用来测试上传文件的文件名

而insert语句为insert into uploads values (id, '\$user', '\$filename')...

我们可以通过测试ip的waf，知道filename的waf。

但在这种假设下，文件一定可以被上传

从上面的条件思考upload.php的核心代码大致如下

if(!empty($_FILES['upfiles']['tmp_name']))

{

if(is_array($_FILES['upfile'])){

die();

```
}
if(checkIsPhp($_FILES['upfile'])){
die('bu shi php')
}
if(waf($FILES['upfiles'])){
mysql_query('insert')
}else{
die('waf')
}
}
```

重新思考流程后，我们可以想到，这里的文件一定可以被上传(即使不能直接上传php)

在第二天的比赛中，经过测试，我们发现后台的判断非常奇怪，在假设文件可以被上传的情况下，后台大概是判断是不是一个纯粹的php文件，在不考虑强行脑洞的情况下，我们需要寻找一个非 在完成题目之后，我拿到了题目的源码，重新回顾源码后发现一些有趣的东西。

upload.php

```
session_start();
include_once 'lib/clean.php';
include_once 'lib/database.php';
if (isset($_FILES['upfile'])) {
$file = $_FILES['upfile'];
if ($file ['error'] > 0) {
switch ($file ['error']) {
case 1 :
$mes = 'The uploaded file exceeds the value of the upload_max_filesize option in the PHP configuration file';
break;
case 2 :
$mes = 'Exceeded the size of the form MAX_FILE_SIZE limit';
break;
case 3 :
$mes = 'File section was uploaded';
break;
case 4 :
```

```
$mes = 'No upload file selected';
break;
case 6 :
$mes = 'No temporary directory found';
break;
case 7 :
case 8 :
$mes = 'System error';
break;
}
die($mes);
}
$content = file_get_contents($file['tmp_name']);
checkMIME($file);
if (checkContent($content) && checkExts($file['name'])) {
upload($file);
} else {
die('attack detected');
}
} else {
die('file not found');
}
function upload($file)
{
$savepath = dirname(__file__) . '/uploads/';
$filename = explode('.', $file['name']);
$newname = rand_name() . "." . trim(end($filename));
$finalname = $savepath . $newname;
if (move_uploaded_file($file['tmp_name'], $finalname)) {
$db = new Database();
//,1,(select substring(filename,10,10) from(select filename from picture limit 0,1)x))#
```

```
if ($db->insert($_SESSION['username'], getip(), $newname)) {
header('location: index.php');
exit();
}
}
}
function rand_name($l = 64)
{
$str = null;
$Pool = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz_";
$max = strlen($Pool) - 1;
for ($i = 0; $i
$str .= $Pool[rand(0, $max)];
}
return $str;
}
function checkExts($filename)
{
$AllowedExt = array('php', 'php3', 'php4', 'php5', 'pht', 'phtml', 'inc');
$filename = explode('.', $filename);
if (in_array(strtolower($filename[count($filename) - 1]), $AllowedExt)) {
return false;
}
return true;
}
function checkMIME($file)
{
// text/php text/x-php
$php_ext = array("text/php", "text/x-php");
$type = mime_content_type($file['tmp_name']);
if(!in_array(strtolower($type), $php_ext)){
```

```
die("i need php file");

}

}

function checkContent($content)

{

if (stripos($content, '') === 0) {

return false;

}

return true;

}
```

upload中我们一直认为是悖论的过滤，是通过mime_content_type来判断的，这也是为什么我们可以用#!/usr/bin/php绕过的原因，蛮有意思的一个点