绕过黑名单检测实现文件上传

讲师：空白

学院介绍

学院宗旨：专注网安人才实战技能培养

学院官网：https://edu.heetian.com/

合天网安实验室：https://www.hetianlab.com/

主打课程：

《web安全》：OWASP TOP 10漏洞原理及测试

《渗透测试》：渗透测试流程及工具的使用

《安全开发》：用python写一个综合的扫描器

《CTF-PWN》：CTF中的PWN相关

目录
CONTENTS

/01 文件上传基本概述

1.1 文件上传简述

顾名思义，文件上传就是将客户端的文件上传到服务器的过程称为文件上传。比如QQ空间发表说说上传的图片、招聘网上传简历、合天网安实验室修改头像、将文件上传到网盘等，这些都是文件上传。

```
<html>
<head></head>
<body></body>
<form enctype="multipart/form-data" action="02.php" method="POST">
    Send this file: <input name="userfile" type="file" />
    <input type="submit" value="Send File" />
</form>
</html>
```

1.2 文件上传漏洞简述

上传文件的时候，如果服务器端后端语言未对上传的文件进行严格的验证和过滤，就容易造成上传任意文件的情况。常见场景是web服务器允许用户上传图片或者普通文本文件保存，而用户绕过上传机制上传恶意代码并执行从而控制服务器。

1.3 文件上传漏洞的危害

攻击者通过上传恶意文件传递给解释器去执行，然后就可以在服务器上执行恶意代码，进行数据库执行、服务器文件管理、命令执行等恶意操作。从而控制整个网站，甚至是服务器。

1.4 文件上传漏洞的必备条件

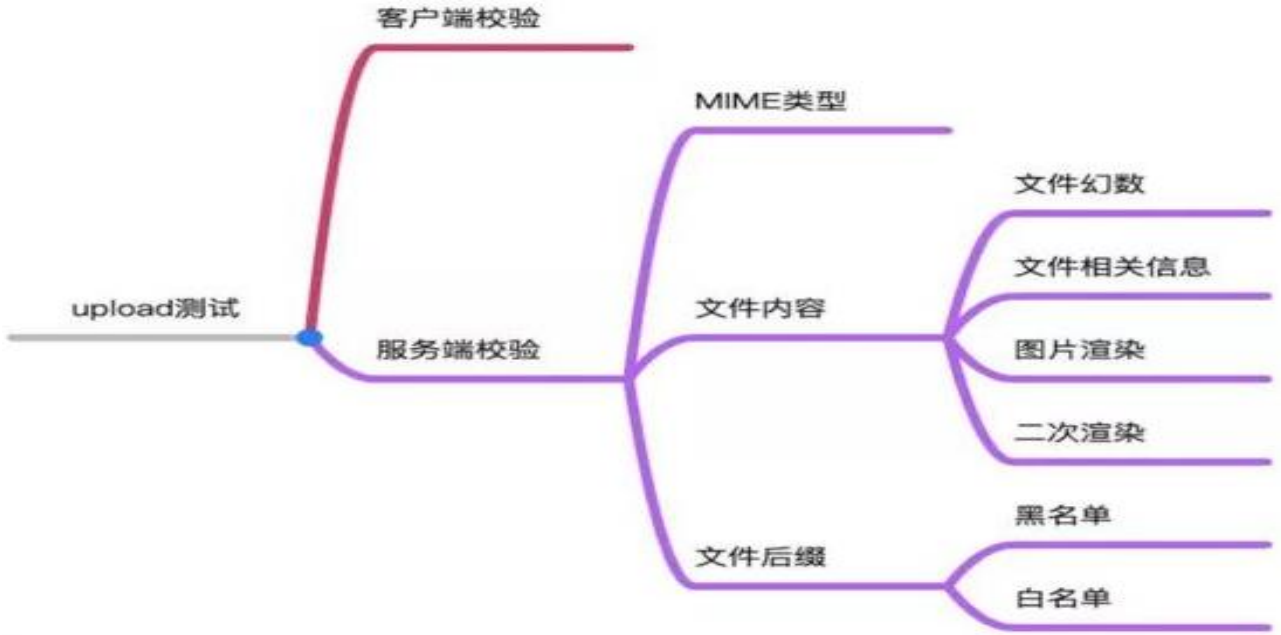
- 文件上传功能能正常使用
- 上传文件路径可知
- 上传文件可以被访问
- 上传文件可以被解析

1.5 为什么要进行文件上传

危害最大化

1.6 检测上传文件的方式

- 客户端JavaScript检测（检测文件扩展名）
- 服务端MIME类型检测（检测content-type内容）
- 服务端文件扩展名检测 (检测跟文件extension相关的内容)
- 服务端文件内容检测（检测内容是否合法是否含有恶意代码）等。



/02 客户端及MIME类型检测

2.1 客户端浏览器检测

当用户在客户端选择文件点击上传的时候，客户端还没有向服务器发送任何消息，就对本地文件进行检测来判断是否是可以上传的类型，这种方式称为前台脚本检测扩展名。

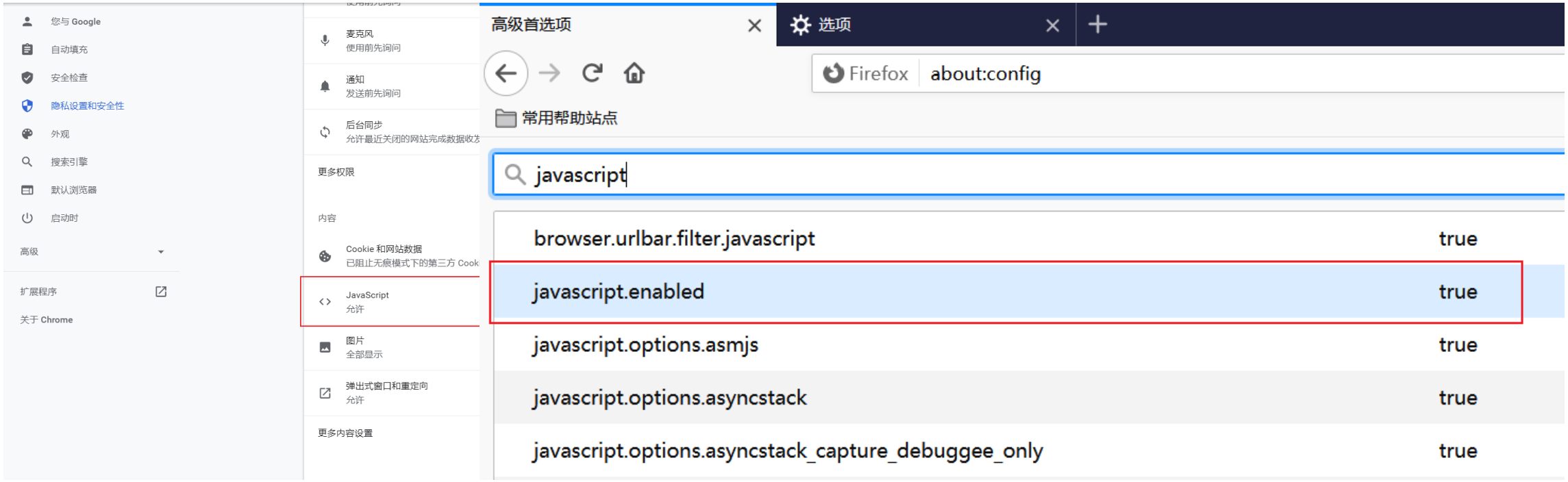
```
<script type="text/javascript">
    function checkFile() {
        var file = document.getElementsByName('upfile')[0].value;
        if (file == null || file == "") {
            alert("你还没有选择任何文件，不能上传!");
            return false;
        }
        //定义允许上传的文件类型
        var allow_ext = ".jpg|.jpeg|.png|.gif|.bmp|";

        //提取上传文件的类型
        var ext_name = file.substring(file.lastIndexOf("."));
        //alert(ext_name);
        //alert(ext_name + "|");

        //判断上传文件类型是否允许上传
        if (allow_ext.indexOf(ext_name + "|") == -1) {
            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;
            alert(errMsg);
            return false;
        }
    }
</script>
```

2.1.1 绕过客户端检测实现上传

在本地浏览器客户端禁用JS：可使用火狐浏览器的Noscript插件、浏览器禁用JS等方式实现。



2.2 服务器端检测MIME类型

当浏览器在上传文件到服务器的时候，服务器对所上传文件的Content-Type类型进行检测，如果是允许的，则可以正常上传，否则上传失败。

```
6   if (isset($_POST['submit'])) {
7       if (file_exists(UPLOAD_PATH)) {
8           if (($_FILES['upload_file']['type'] == 'image/jpeg') || ($_FILES['upload_file']['type']
    == 'image/png') || ($_FILES['upload_file']['type'] == 'image/gif')) {
9               if (move_uploaded_file($_FILES['upload_file']['tmp_name'], UPLOAD_PATH . '/' .
    $_FILES['upload_file']['name'])) {
10                  $img_path = UPLOAD_PATH . $_FILES['upload_file']['name'];
11                  $is_upload = true;
12
13              }
14          } else {
15              $msg = '文件类型不正确，请重新上传！';
16          }
17      } else {
18          $msg = UPLOAD_PATH.'文件夹不存在,请手工创建！';
19      }
20  }
```

2.2.1 MIME理解

MIME (Multipurpose Internet Mail Extensions) 是描述消息内容类型的因特网标准。用来表示文档、文件或字节流的性质和格式。在http数据包中在Content-Type字段显示。

- 超文本标记语言.html文件：text/html
- 普通文本.txt文件：text/plain
- PDF文档.pdf：application/pdf
- Microsoft Word文件.word：application/msword
- PNG图像.png：image/png
- GIF图像.gif：image/gif
- MPEG文件.mpg、.mpeg：video/mpeg
- AVI文件.avi：video/x-msvideo

```
POST /Login.action HTTP/1.1
Host: www.hetianlab.com
Connection: close
Content-Length: 320
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/89.0.4389.82 Safari/537.36 Edg/89.0.774.48
Content-Type: application/x-www-form-urlencoded
Origin: https://www.hetianlab.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://www.hetianlab.com/loginLab.do
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-GB;q=0.8,en;q=0.7,en-US;q=0.6
Cookie:
UM_distinctid=177f21a33d7632-06a473fa080ffd-7a667166-144000-177f21a33d8b83;
CNZZDATA1279677270=897547635-1614674536-%7C1614674536; register=; platform=os;
JSESSIONID=488AF286D523389CE9C264A0981EF5D1.jvm3; noticeFlag=77c6447e;
```

2.2.2 绕过MIME检测实现文件上传

利用Burp Suite截取并修改数据包中的Content-Type字段的值为正常值从而进行绕过。

/03 黑名单检测及绕过方法

3.1 黑名单概念

一般情况下，代码文件里会有一个数组或者列表，该数组或者列表里会包含一些非法的字符或者字符串，当数据包中含有符合该列表的字符串时，即认定该数据包是非法的。

```
...
$deny_ext = array('.asp','.aspx','.php','.jsp');
...
if(!in_array($file_ext, $deny_ext)) {
    ...
} else {
    $msg = '不允许上传.asp,.aspx,.php,.jsp后缀文件！ ';
}
```

3.2 如何确认黑白名单

因为黑名单是不允许我们的数据包含有符合黑名单列表的字符串，所以我们只需要**随意构造**一个不在它列表中的数据包即可。

3.3 利用后缀大小写绕过

在Windows中，大小写是不敏感的。

例如：“index.html”和“index.htmL”访问的结果是一样的。

3.4 利用空格绕过

在Windows中，文件保存的时候如果文件后缀名末尾有空格会自动去掉。

例如：“phpinfo.php ” Windows会自动去掉末尾的空格变成"phpinfo.php"。

3.5 利用点号（·）绕过

在Windows中，文件保存的时候会自动去掉文件后缀后的点号。

例如：“index.html.”在保存的时候就变成了“index.html”。

3.6 利用（::$DATA）绕过

在Windows中如果文件名+“::$DATA”会把::$DATA之后的数据当成文件流处理，不会检测后缀名，且保持::$DATA之前的文件名。

例如:“phpinfo.php::$DATA”Windows会自动去掉末尾的::$DATA变成“phpinfo.php”。

3.7 利用双写后缀绕过

有些代码中，会将数据包中符合黑名单列表的字符串替换为空。

比如：“index.php”变为“index”。

3.8 利用.htaccess文件绕过

.htaccess文件(或者"分布式配置文件"），全称是Hypertext Access(超文本入口)。提供了针对目录改变配置的方法， 即，在一个特定的文档目录中放置一个包含一个或多个指令的文件， 以作用于此目录及其所有子目录。作为用户，所能使用的命令受到限制。

比如新建一个.htaccess文件：

<FilesMatch "as.png">

setHandler application/x-httpd-php

</FilesMatch>

通过一个.htaccess 文件调用 php 的解析器去解析一个文件名中只要包含"as.png"这个字符串的任意文件，所以无论文件名是什么样子，只要包含"as.png"这个字符串，都可以被以 php 的方式来解析，一个自定的.htaccess文件就可以以各种各样的方式去绕过很多上传验证机制。

感谢您的聆听

学习工具、资料及课程回放

扫码免费领取