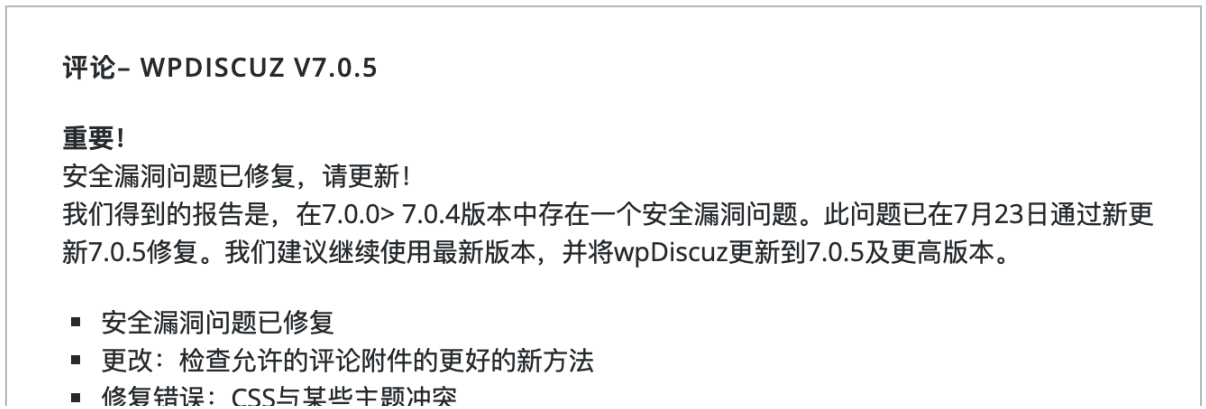
wordpress 评论插件 wpDiscuz 任意文件上传漏洞分析

> 先知社区，先知安全技术社区

前言

在 t00ls 上看到一个老哥 wordpress 站点被搞了，下载了 access 日志分析了一下，发现攻击路径是先访问了一个页面，然后访问 /wp-admin/admin-ajax.php?action=wmuUploadFiles 后直接就 shell 落地了，确定是这个模块有问题了；后来说是 wpdiscuz 评论插件的漏洞，看了下官网 V7.0.5 修复了一次安全漏洞，于是下载 V7.0.3 版本学习一下（7.0.4 没找到）。



- 已修复的错误：已修复了其他一些小错误

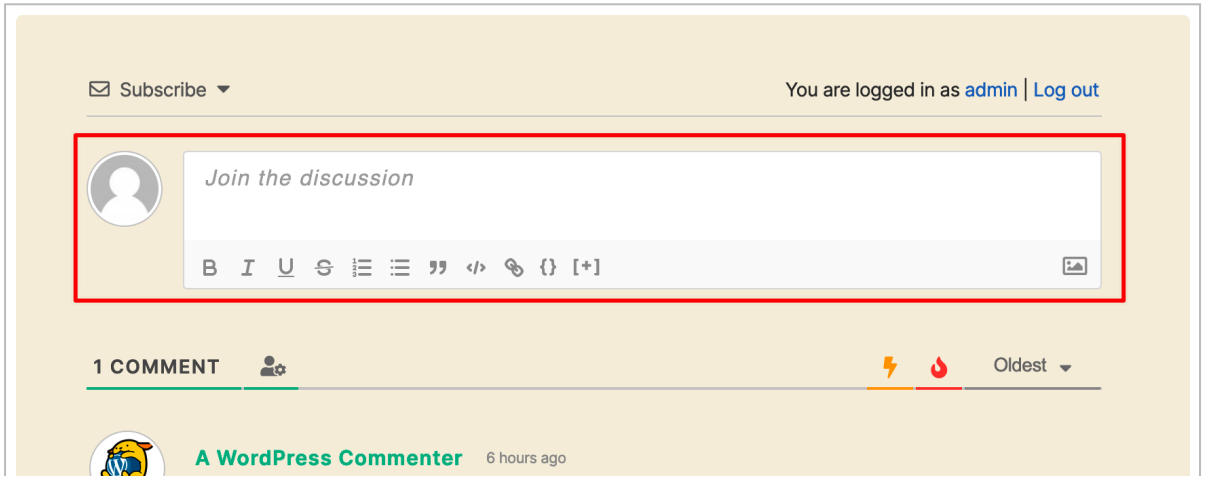
(https://xzfile.aliyuncs.com/media/upload/picture/20200811112118-b8b4ef36-db81-1.png)

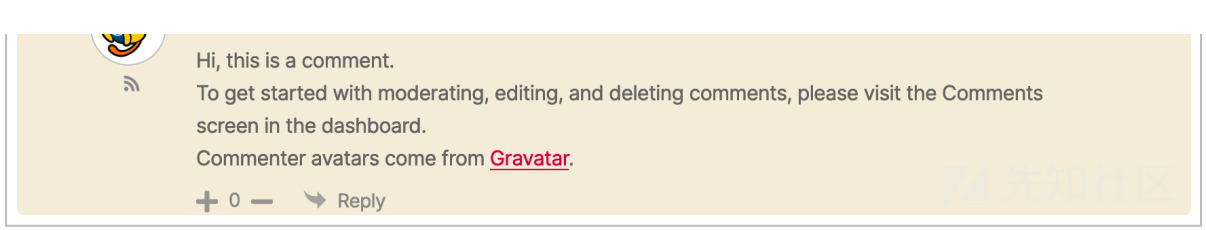
环境

- php 5.6.40
- mysql 5.7.26
- Phpstorm
- Wordpress 5.4.1 + wpdiscuz V 7.0.3

分析

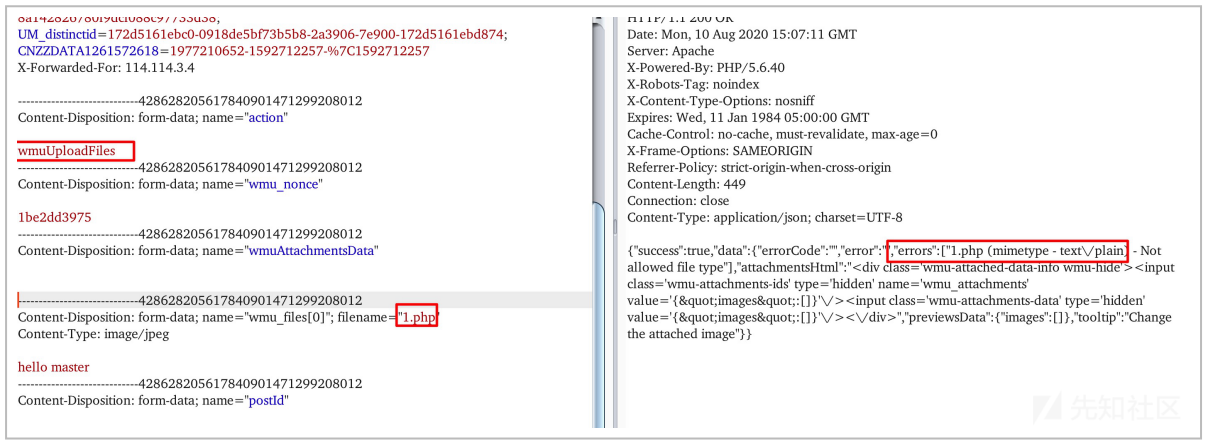
1. 环境搭建后，手动安装 wpdiscuz 插件后，看到文章下增加评论模块





(https://xzfile.aliyuncs.com/media/upload/picture/20200811112222-de9c392a-db81-1.png)

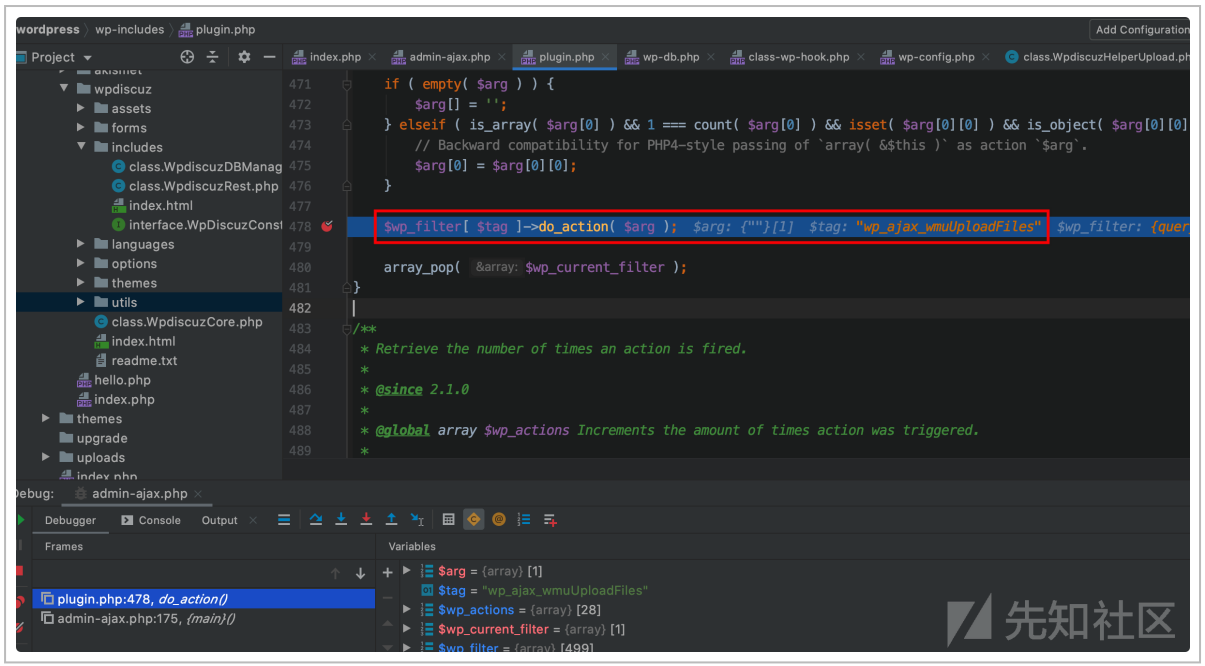
2.phpstorm 导入 web 目录，点击图片按钮，上传一个 php 文件测试一下，上传路径是 http://127.0.0.1:8888/wordpress/wp-admin/admin-ajax.php，默认是上传不了的。(http://127.0.0.1:8888/wordpress/wp-admin/admin-ajax.php%EF%BC%8C%E9%BB%98%E8%AE%A4%E6%98%AF%E4%B8%8A%E4%BC%A0%E4%B8%8D%E4%BA%86%E7%9A%84%E3%80%82)



(https://xzfile.aliyuncs.com/media/upload/picture/20200811112245-ecc0e1b8-

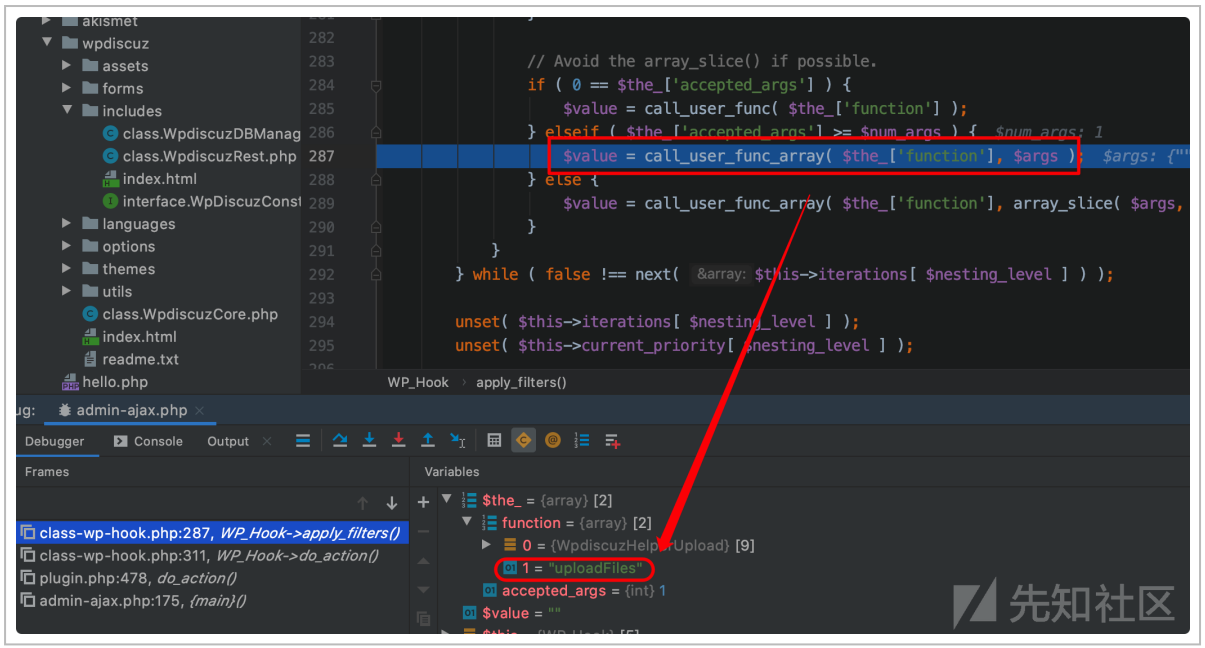
db81-1.png)

3. 从入口点分析，如图是 wp_filter 的 action 过滤



(https://xzfile.aliyuncs.com/media/upload/picture/20200811112258-f41cee5c-db81-1.png)

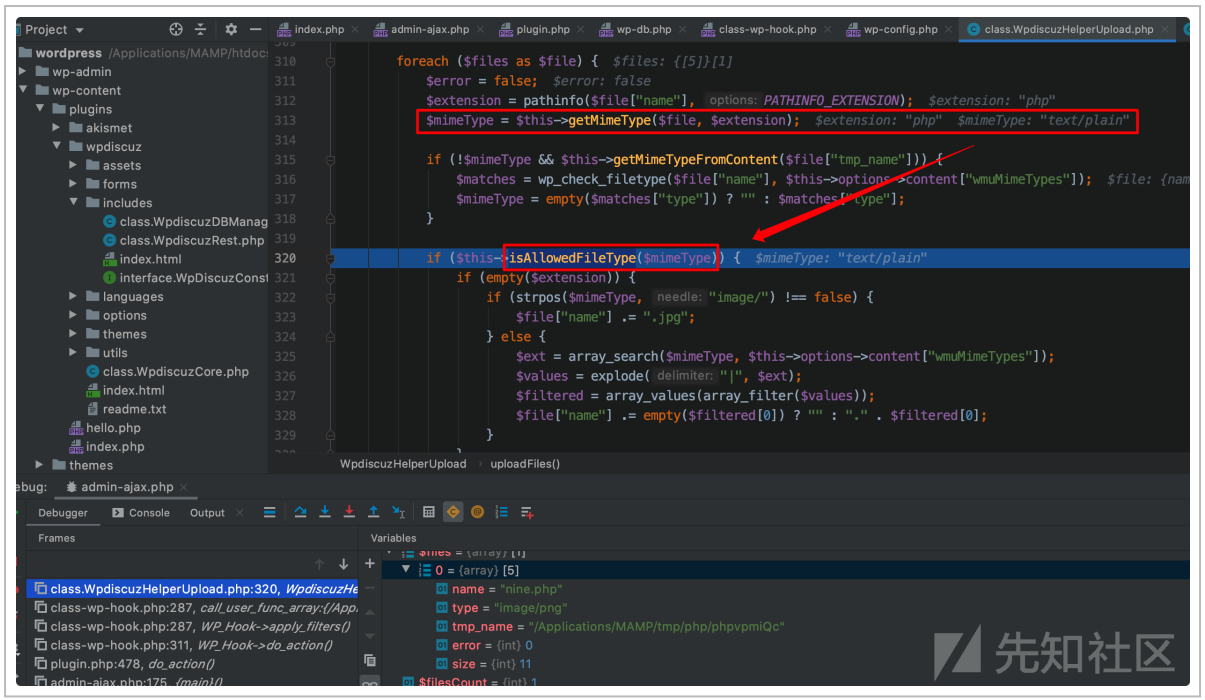
4. 跟进去，可以看到上传的功能点，再进去



(https://xzfile.aliyuncs.com/media/upload/picture/20200811112311-fc084d82-db81-1.png)

5. 可以看到如图位置，使用 getMimeType 方法根据文件内容获取文件类型，并不是通过文

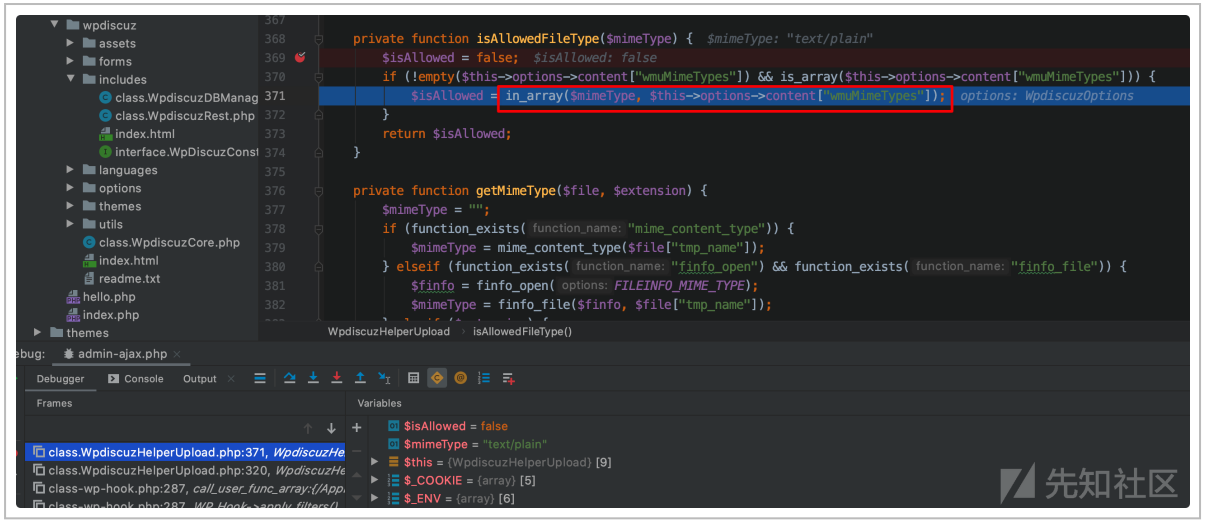
件后缀名判断，进一步根据 $mineType 判断是否是允许的上传类型。



(https://xzfile.aliyuncs.com/media/upload/picture/20200811112322-02e61b20-db82-1.png)

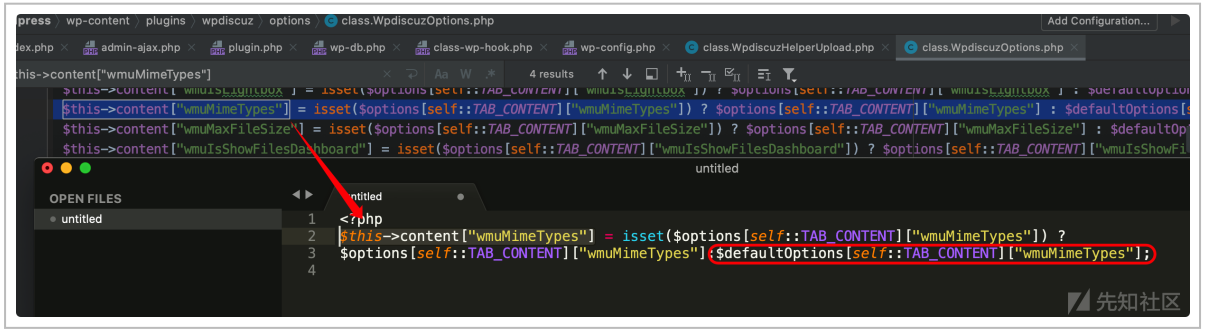
6. 跟入查看 isAllowedFileType 方法，在判断 $mineType 是否在 $this->options->

6. 跟入查看 isAllowedFileType 方法，在判断 $mimeType 是否在 $this -> options -> content["wmuMimeTypes"] 中存在。



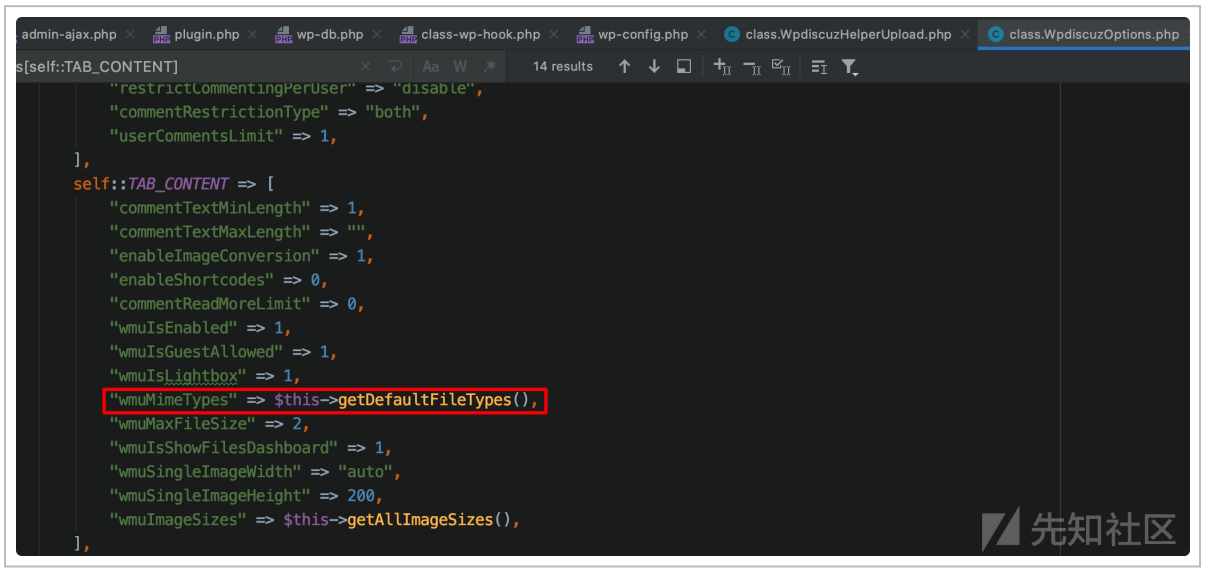
(https://xzfile.aliyuncs.com/media/upload/picture/20200811112338-0c0d25d6-db82-1.png)

7. 如图，进入 $options 中，可以 content["wmuMimeTypes"] 使用三目运算判断，搜索上下文得知，结果就是 $defaultOptions[self::TAB_CONTENT]["wmuMimeTypes"]



(https://xzfile.aliyuncs.com/media/upload/picture/20200811112351-13df129c-db82-1.png)

8. 进入 $defaultOptions 中可以得到最终 $this -> options -> content["wmuMimeTypes"] 的值是几种常见的图片类型。



(https://xzfile.aliyuncs.com/media/upload/picture/20200811112401-1a2f0b48-db82-1.png)

```
public function getDefaultFileTypes() {
    $types = ["jpg|jpeg|jpe" => "image/jpeg", "gif" => "image/gif", "png" => "image/png", "bmp" => "image/bmp", "tiff|tif" => "image/tiff", "ico" => "image/x-icon"];
    $types = apply_filters("wpdiscuz_mu_file_types", $types);
    return $types;
}
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200811112413-2145b152-db82-1.png)

9. 很明显此时文件类型已经通过 getMimeType() 方法修改为 text/plain 了，但是回到进入 isAllowedFileType 的代码，发现程序只在此处对上传文件进行了判断后，直接保存了文件。

```
if ($this->isAllowedFileType($mimeType)) {
    if (empty($extension)) {  $extension: "png"
        if (strpos($mimeType, needle: "image/") !== false) {
            $file["name"] .= ".jpg";
        } else {
            $ext = array_search($mimeType, $this->options->content["wmuMimeTypes"]);
            $values = explode( delimiter: "|", $ext);
            $filtered = array_values(array_filter($values));
            $file["name"] .= empty($filtered[0]) ? "" : "." . $filtered[0];
        }
    }
    $file["type"] = $mimeType;
} else {
    $error = true;
    $response["errors"][] = $file["name"] . " " . (current_user_can( capability: "manage_options") ? "(mimetype – " . $mimeType . ") " : "") . "– " . $this->opt
}

do_action( tag: "wpdiscuz_mu_preupload", $file);

if (!$error) {  $error: false
    $attachmentData = $this->uploadSingleFile($file);  $attachmentData: {id => 12, url => "http://127.0.0.1:8888/wordpress/wp-content/uploads/2020/08/nine-1
    if ($attachmentData) {
        if (strpos($file["type"], needle: "image/") !== false) {
            $attachmentIds[self::KEY_IMAGES][] = $attachmentData["id"];
            $attachmentsData[self::KEY_IMAGES][] = $attachmentData;
        } else {
            $attachmentIds = apply_filters("wpdiscuz_mu_add_attachment_ids", $attachmentIds, $attachmentData, $file);
            $attachmentsData = apply_filters("wpdiscuz_mu_add_attachments_data", $attachmentsData, $attachmentData, $file);  $attachmentData: {id => 12, url
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200811112424-27b61c2-

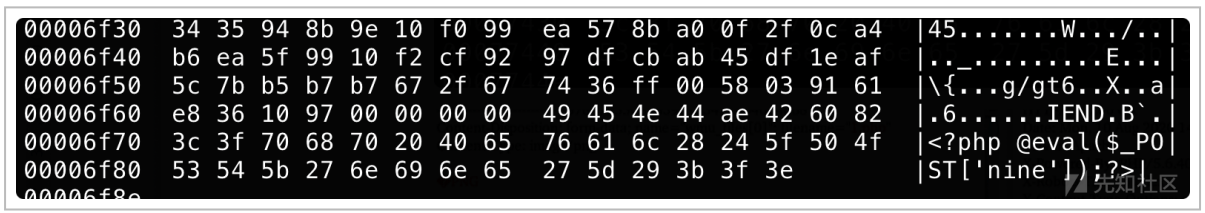
(https://xzfile.aliyuncs.com/media/upload/picture/20200811112424-27b61c2a-db82-1.png)

利用

如此，程序只是根据文件内容判断文件类型，并未对文件后缀进行效验，构造一个图片马，或者手动在 webshell 前面加上图片头信息即可绕过。

1. 把后门文件追加到图片后

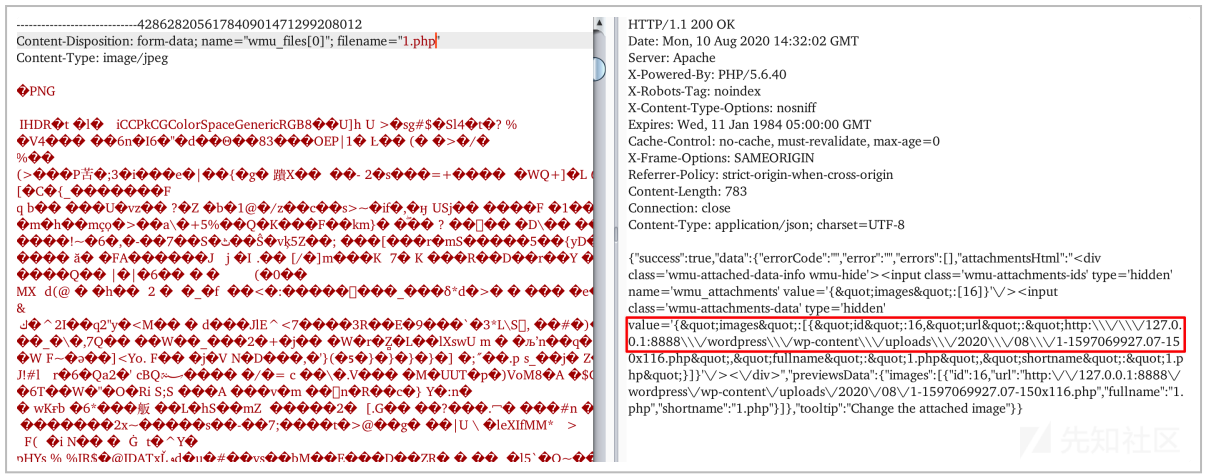
(https://xzfile.aliyuncs.com/media/upload/picture/20200811112435-2e0f59c4-db82-1.png)



(https://xzfile.aliyuncs.com/media/upload/picture/20200811112445-3418eeb6-db82-1.png)

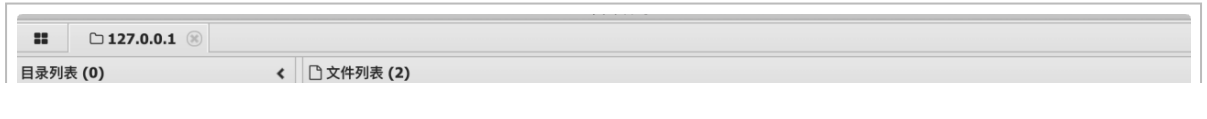
db82-1.png)

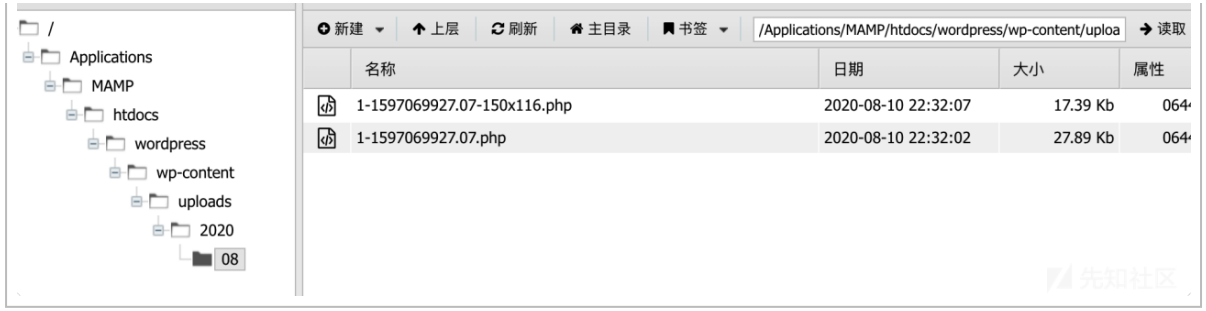
2. 上传并修改后缀名为 php，可以看到返回路径



(https://xzfile.aliyuncs.com/media/upload/picture/20200811112456-3a87ccfe-db82-1.png)

3. 连接 webshell

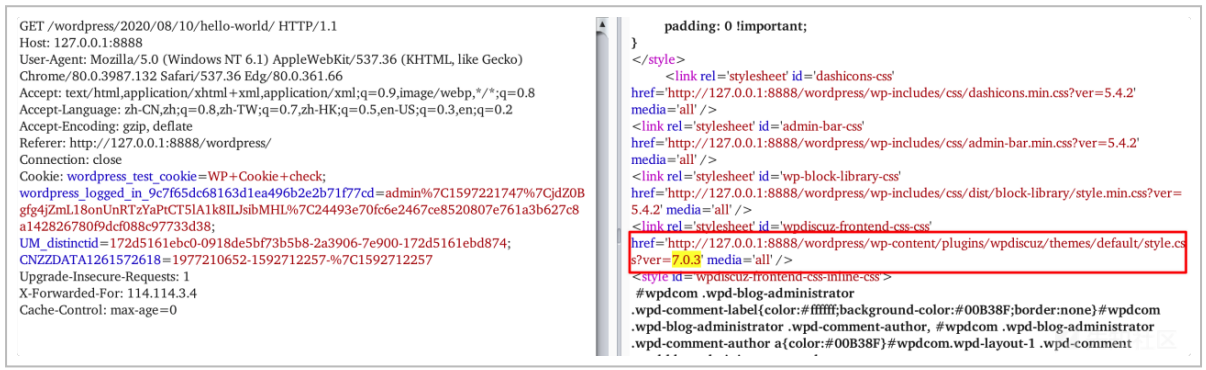




(https://xzfile.aliyuncs.com/media/upload/picture/20200811112507-415254dc-db82-1.png)

检测

在装上 wpdiscuz 插件后，每个文章中都会带有如下标签信息，且带有版本号，可利用此特征编写脚本或者御风插件。



(https://xzfile.aliyuncs.com/media/upload/picture/20200811112517-47448fae-db82-1.png)

简单的检测脚本，exp 功能删除了

```
import requests
import re
import sys

class wpdiscuz():
    def __init__(self):
        self.s = requests.session()
        self.s.headrs = {
            "User-Agent":
            "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/80.0.3987.132 Safari/537.36 Edg/80.0.361.66"
        }
        self.nonce = ""
        self.state = False

    def check(self, url):
        res = self.s.get(url=url)

        pat1 = "wpdiscuz/themes/default/style\.css\?ver=(.*?)'"
        reSearch1 = re.search(pat1, res.text)
        if reSearch1 == None:
            print("%s 评论插件不存在任意文件漏洞" % url)
            return
```

```
            mess = reSearch1.group(0)
            version = reSearch1.group(1)
            # 判断版本
            vers = version.split(".")
            if (len(vers) == 3):
                if int(vers[0]) == 7:
                    if int(vers[2]) <= 4:
                        print(url + " 存在任意文件上传漏洞 wpdiscuz版本为 %s" % version)
                        self.state = True

            if self.state == True:
                # nonce
                pat2 = '"wmuSecurity":"(.*?)"'
                reSearch2 = re.search(pat2, res.text)
                nonce = reSearch2.group(1)
                self.nonce = nonce
            else:
                print("%s 评论插件不存在任意文件漏洞" % url)

        def exp(self, url, project, filepath):
            pass

    if __name__ == "__main__":
        wpdiscuz = wpdiscuz()
        url = sys.argv[1]
        print("检测漏洞结果:")
        wpdiscuz.check(url)
```