盘企 – LCMS 代码审计 (vv2021.0521152900+v2021.0528154955)

奇安信攻防社区 – 某小型 cms 代码审计

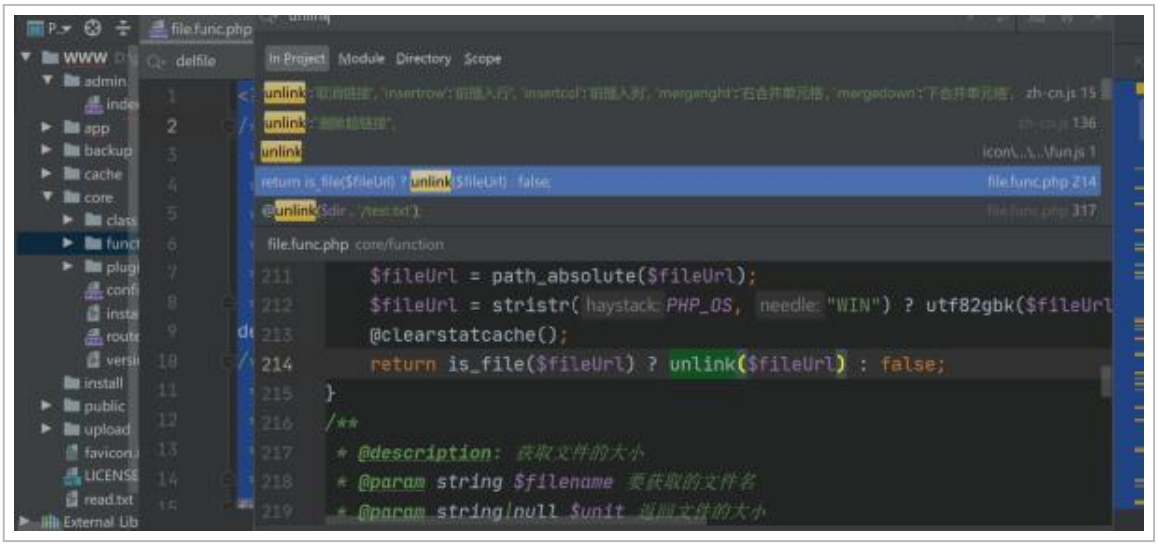
审计两个版本 v2021.0521152900 和 v2021.0528154955，7 月 2 号爆出的 v2021.0521152900 存在任意文件删除和任意文件上传，审计出来以后又审计了一下下一版本，图片都是当时审计时做的笔记...

审计两个版本 v2021.0521152900 和 v2021.0528154955，7 月 2 号爆出的 v2021.0521152900 存在任意文件删除和任意文件上传，审计出来以后又审计了一下下一版本，图片都是当时审计时做的笔记的图片

暑假七月份想训练一下代码审计，于是去 cnvd 上看近几天有哪些 cms 有编号，然后找到了这个小 cms

任意文件删除漏洞产生原因

由于已经知道是什么漏洞了，所以直接去搜索一下相应的函数即可，于是搜索了一下 unl ink() 函数，翻着看了一下定位到了这个文件



(https://shs3.b.qianxin.com/attack_forum/2021/08/attach-af6a3b9d1c14900386de4c4ec9abb31e366d91ef.jpg)

af0a3b5d1e14900880de4c4ec5abb51e5000d51ef.jpg)

代码如下

```
function delfile($fileUrl)
```

看到

```
{
```

跟进 path_absolute 函数

```
$fileUrl = path_absolute($fileUrl);
```

可以看到过滤规则，但是开发人员忽略了 cms 是搭建在 windwos 系统上那么就可以利用..\ 来进行跨目录

知道这一点后找哪里应用了 delfile()

在后台中有个备份功能，当备份了以后可以执行删除操作

(https://shs3.b.qianxin.com/attack_forum/2021/08/attach-5f66898c40c35b0889fb9f500a21847b362f2408.png)

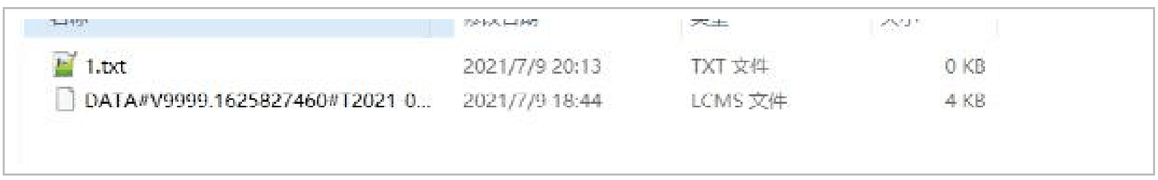
删除操作在 datab ase.class.php 中

```
$fileUrl = stristr(PHP_OS, &quot;WIN&quot;) ? utf82gbk($fileUrl) : $fileUrl;
```

可以看到这里应用了 delfile()

于是 bp 抓包，修改一下数据包

当时在备份文件目录下创建了一个 txt 文件



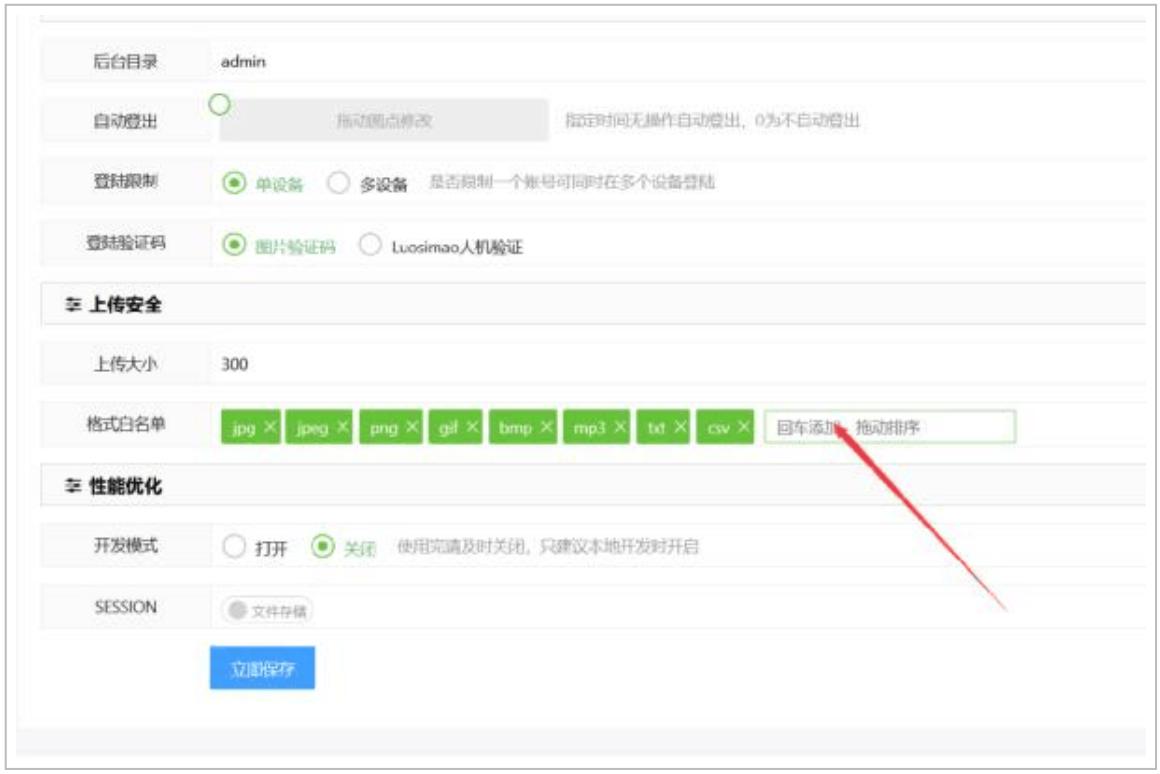
(https://shs3.b.qianxin.com/attack_forum/2021/08/attach-

8786e243c243e16f2b5c4e18908cb81c5958b8e3.png)

在数据包中将 data 名修改为 1.txt 发送即可，但是当时没有将 bp 数据包截图，只有下一版本的任意文件删除漏洞有截图

任意文件上传

漏洞产生的原因在于没有将后添加进白名单的文件名进行检测过滤



(https://shs3.b.qianxin.com/attack_forum/2021/08/attach-42a226cc8a3cc346e465b67674dcbc6698c901b5.png)

然后直接到这里添加附件

(https://shs3.b.qianxin.com/attack_forum/2021/08/attach-65084fae948a1d878e780676cc3c0867271e21ca.png)
直接上传即可 getshell

漏洞产生原因:
上传控制文件位于：upload.class.php
upload 先截取文件的后缀名, 截取以后再对后缀名和白名单文件名列表进行比较

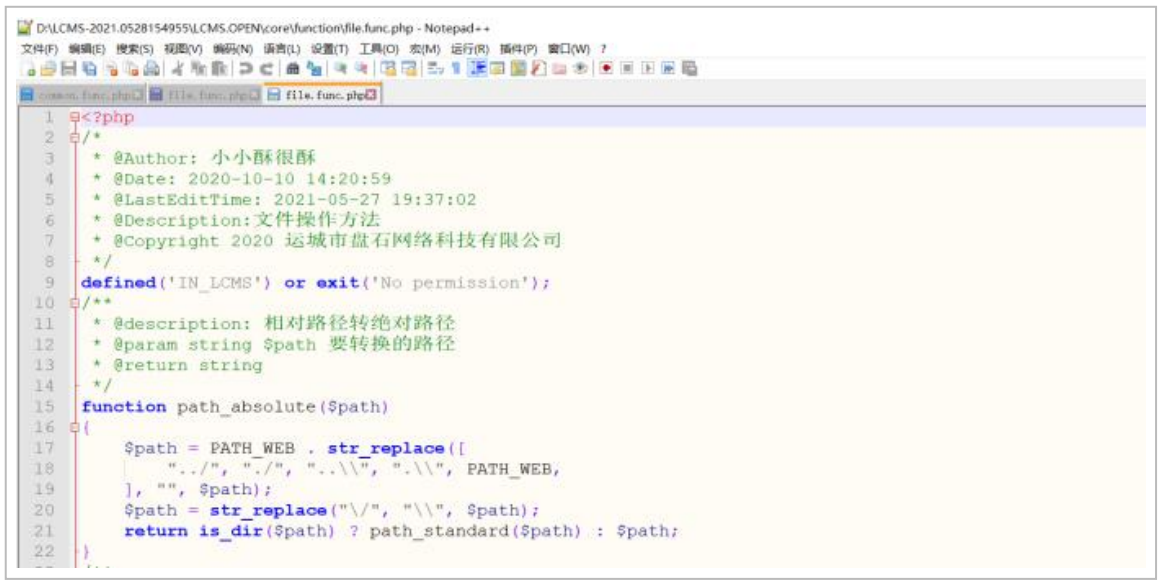
```
@clearstatcache();
```

当添加进去文件白名单后, upload.class.php 过滤规则中就会有你写的白名单文件名

```
81      public static function mime($mime = "")
82      {
83          $allmime = [
84              "image/jpeg"                    => "jpeg",
85              "image/png"                     => "png",
86              "image/bmp"                     => "bmp",
87              "image/gif"                     => "gif",
88              "image/webp"                    => "webp",
89              "image/vnd.wap.wbmp"            => "wbmp",
90              "image/x-up-wpng"               => "wpng",
91              "image/x-icon"                  => "ico",
92              "image/svg+xml"                 => "svg",
93              "image/tiff"                    => "tiff",
94              "audio/mpeg"                    => "mp3",
95              "audio/ogg"                     => "ogg",
96              "audio/x-wav"                   => "wav",
97              "audio/x-ms-wma"                => "wma",
98              "audio/x-ms-wmv"                => "wmv",
99              "video/mp4"                     => "mp4",
100             "video/mpeg"                    => "mpeg",
101             "video/quicktime"               => "mov",
102             "flv-application/octet-stream" => "flv",
103             "application/json"              => "json",
104             "application/octet-stream"      => "rar"
```

(https://shs3.b.qianxin.com/attack_forum/2021/08/attach-9e936acd353e00293eba8babc07e27d166064bc8.png)

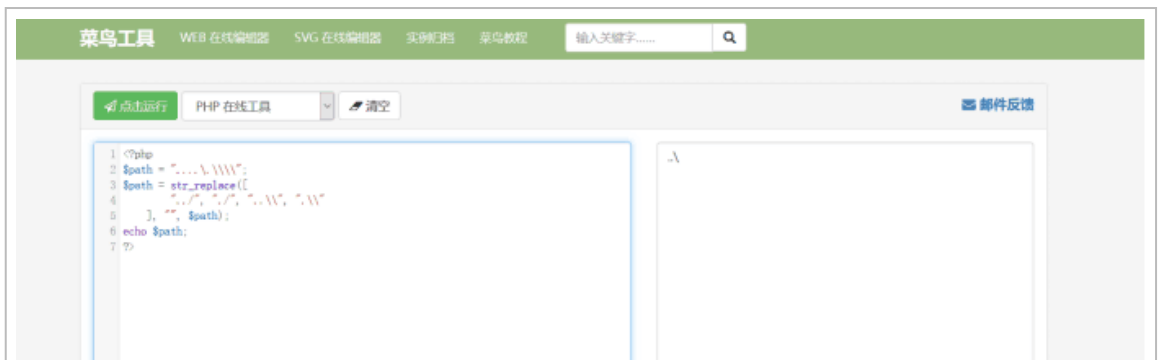
下一版本任意文件删除

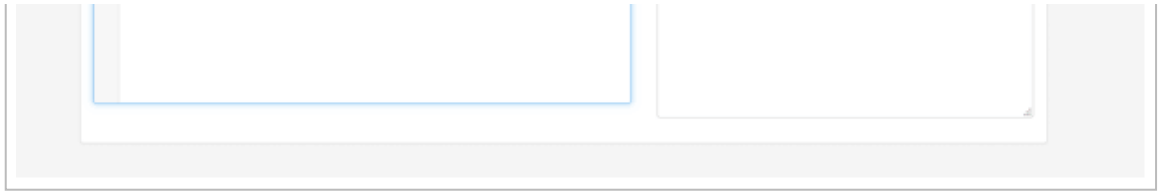


(https://shs3.b.qianxin.com/attack_forum/2021/08/attach-069f87327741a10c3c0daf090e872939fde1bc16.jpg)

可以看到由于上一版本的过滤原因，这一版本添加了过滤规则，在过滤规则中添加了 ..\ 和.\

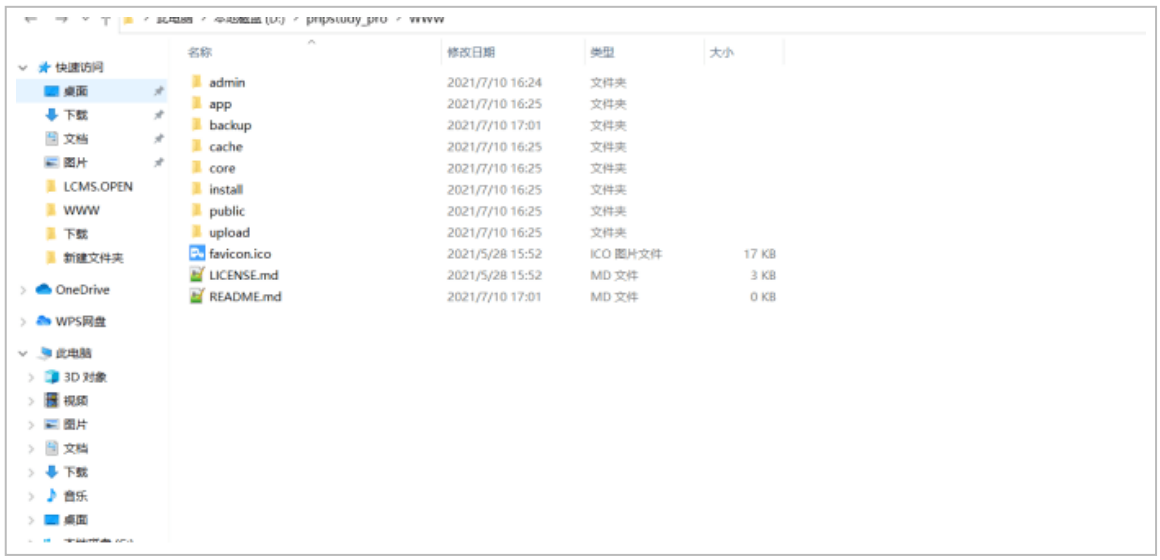
但是加了之后真的安全了吗？表面上看起来雀食如此，但根据过滤规则我在在线 php 工具上写出了代码，然后进行调试



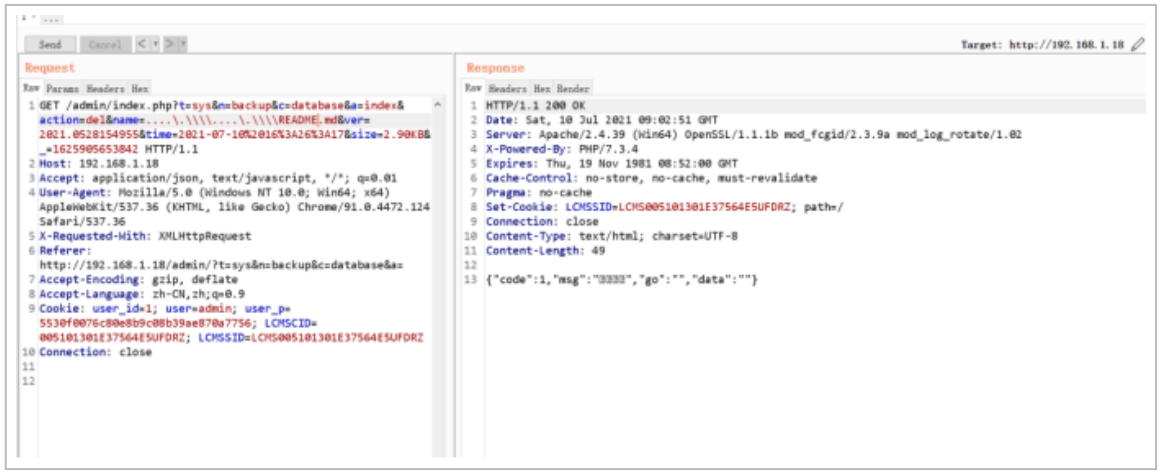


(https://shs3.b.qianxin.com/attack_forum/2021/08/attach-4458f73d0940f591e04628bf444cc1c84c5e4bbd.png)
成功了，可以看到最后我的 payload 最后输出成为了..\，这就代表着在 windows

系统下依然存在任意文件删除漏洞
打开 bp 再次抓包，这一次当时选择了根目录下的 README.md 文件

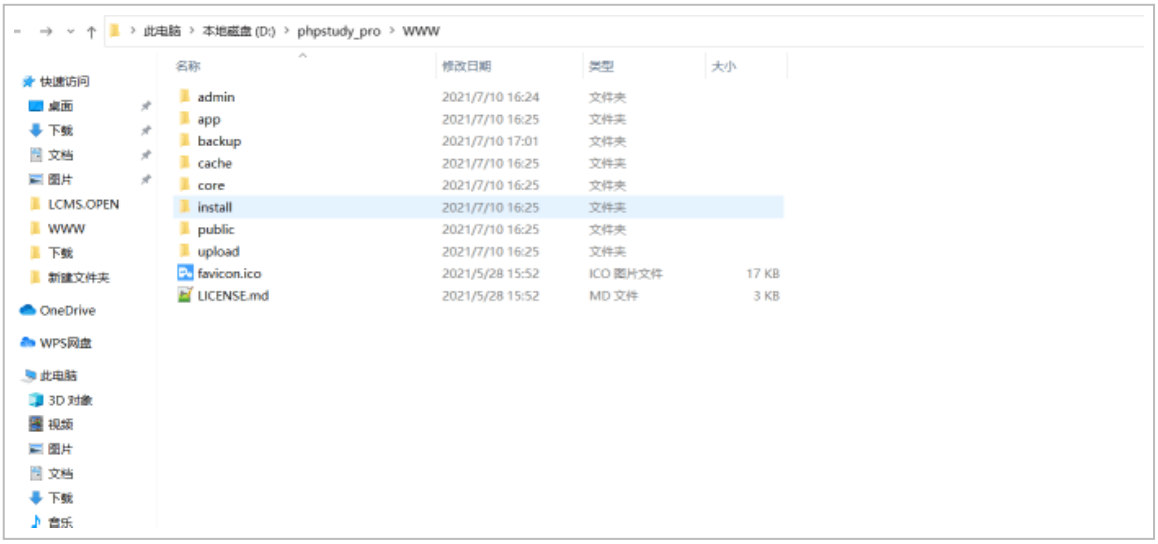


(https://shs3.b.qianxin.com/attack_forum/2021/08/attach-07053a7dff66c644064a523a86e5e800806c5c67.png)
然后像之前一样修改数据包，根据目录放上 payload



(https://shs3.b.qianxin.com/attack_forum/2021/08/attach-27e6cdca860b21902d7e2a5f2e66a1d50523e9db.png)

发送之后就可以看见



(https://shs3.b.qianxin.com/attack_forum/2021/08/attach-e07c1773c936c08bcc51802b67494e5b340d855a.png)

直接删除掉了.

措辞轻浮，内容浅显，操作生疏。不足之处欢迎大师傅们指点和纠正，感激不尽。