Simple Network

This project is a **docktical** toy example.

The two main files are the **docker-compse.yml** file and the **docktical.yml** file.

For this example the project will be composed of two containers. One container will be the **attacker** container and the other the **webserver** container. They will be connected on a network named **lan**.

```
name: example
services:
  webserver:
    build: ./webserver
    networks:
      lan:
        ipv4_address: 10.5.0.3
  attacker:
    build: ./attacker
    networks:
      lan:
        ipv4_address: 10.5.0.2

networks:
  lan:
    ipam:
      config:
        - subnet: 10.5.0.0/16
          gateway: 10.5.0.1
```

Example usage

To start the practical work run:

```
docktical start
```

It will open two terminals named “Attacker” and “Server Logs”.

If use the command **ifconfig**, we see that our container is well connected on the network with the specified IP address.

```
# Attacker
user@f6662b141f54:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.5.0.2  netmask 255.255.0.0  broadcast 10.5.255.255
        ether 02:42:0a:05:00:02  txqueuelen 0  (Ethernet)
        RX packets 134  bytes 23197 (23.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

To check if the web server is running we can firstly try to ping it by using the **`ping`** command:

```
# Attacker
user@f6662b141f54:~$ ping 10.5.0.3
PING 10.5.0.3 (10.5.0.3) 56(84) bytes of data.
64 bytes from 10.5.0.3: icmp_seq=1 ttl=64 time=0.210 ms
64 bytes from 10.5.0.3: icmp_seq=2 ttl=64 time=0.119 ms
^C
--- 10.5.0.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1030ms
rtt min/avg/max/mdev = 0.119/0.164/0.210/0.045 ms
```

We can then send an HTTP request with **`curl`**:

```
# Attacker
user@f6662b141f54:~$ curl 10.5.0.3
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/
strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Directory listing for /</title>
</head>
<body>
<h1>Directory listing for /</h1>
<hr>
<ul>
<li><a href=".bash_logout">.bash_logout</a></li>
<li><a href=".bashrc">.bashrc</a></li>
<li><a href=".profile">.profile</a></li>
<li><a href="init.sh">init.sh</a></li>
</ul>
<hr>
</body>
</html>
```

The user can see that the server logs in recording the connection on the “Server Logs” terminal.

```
# Server Logs
10.5.0.2 - - [19/Oct/2023 11:41:53] "GET / HTTP/1.1" 200 -
10.5.0.2 - - [19/Oct/2023 12:45:47] "GET / HTTP/1.1" 200 -
```

If one of the terminal is closed during the practical work it is possible to reopen it by using the command:

```
docktical open "terminal name"
```

Once the practical work is done it is possible to stop it by using:

```
docktical stop
```