Εργασία Εξαμήνου Ασφάλεια Δικτύων και Επικοινωνιών

Γουργολίτσα Αικατερίνη Α.Μ.171193

Docker

Αρχικά θα πρέπει να κάνουμε μία προεργασία και να εγκαταστήσουμε τον docker και κάποια εργαλεία που θα χρησιμοποιήσουμε στην συνέχεια για την υλοποήση της εργασίας.

Εγκατάσταση Docker

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs)
stable"

sudo apt update

sudo apt install -y docker-ce
 echo   "or"
sudo apt install docker*

sudo systemctl status docker
sudo usermod -aG docker username     #οπου username το όνομα

του χρήστη που θα δουλεύει με docker kate
```

Docker compose

```
 sudo curl -L
"https://github.com/docker/compose/releases/download/1.24.1/do
cker-compose-$(uname -s)-$(uname -m)" -o
/usr/local/bin/docker-compose
 sudo chmod +x /usr/local/bin/docker-compose

 sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-
compose
```

Swarmlab-sec

```
cd /tmp
mkdir test
cd test
git clone https://git.swarmlab.io:3000/swarmlab/swarmlab-
sec.git
cd swarmlab-sec
mkdir project
cd project
../install/usr/share/swarmlab.io/sec/swarmlab-sec create
../install/usr/share/swarmlab.io/sec/swarmlab-sec up size=3
σηκώνουμε 3 clusters
```

Εγκατάσταση εργαλείων

```
sudo apt update
sudo apt install hping3
sudo apt install medusa
sudo apt-get install fail2ban
sudo apt install apache2
sudo apt install openvpn
```

1. Dos/DDos Attacks

1.1. Βρίσκουμε την ip

Αρχικά συνδεόμαστε στον Master και βρίσκουμε την ip

```
../install/usr/share/swarmlab.io/sec/swarmlab-sec login
ifconfig
```

Εδώ θα πάρουμε την ip 172.19.0.2

```
kate@kate-VirtualBox:/tmp/test/swarmlab-sec/project$
kate@kate-VirtualBox:/tmp/test/swarmlab-sec/project$ ../install/usr/share/swarml
ab.io/sec/swarmlab-sec login
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

docker@8d98c9101dc3:/project$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.19.0.2  netmask 255.255.0.0  broadcast 172.19.255.255
        ether 02:42:ac:13:00:02  txqueuelen 0  (Ethernet)
        RX packets 68  bytes 8549 (8.5 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 3208  bytes 202104 (202.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3208  bytes 202104 (202.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker@8d98c9101dc3:/project$
```

1.2. Εύρεση live hosts

Βρίσκουμε τους live hosts ,τις ips διευθύνσεις των workers και ανοιχτές πόρτες.

```
nmap -sP 172.19.0.*
nmap -p 172.19.0.3
```

```
docker@8d98c9101dc3:/project$ nmap -sP 172.19.0.*

Starting Nmap 7.60 ( https://nmap.org ) at 2021-01-09 09:46 UTC
Nmap scan report for 172.19.0.1
Host is up (0.00091s latency).
Nmap scan report for 8d98c9101dc3 (172.19.0.2)
Host is up (0.00067s latency).
Nmap scan report for project_worker_1.project_net (172.19.0.3)
Host is up (0.00060s latency).
Nmap scan report for project_worker_2.project_net (172.19.0.4)
Host is up (0.00045s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 16.09 seconds
docker@8d98c9101dc3:/project$
```

```
docker@8d98c9101dc3:/project$ nmap -p- 172.19.0.3

Starting Nmap 7.60 ( https://nmap.org ) at 2021-01-09 09:53 UTC
Nmap scan report for project_worker_1.project_net (172.19.0.3)
Host is up (0.00033s latency).
Not shown: 65534 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 3.91 seconds
docker@8d98c9101dc3:/project$
```

1.3. Επίθεση DOS

Για να ξεκινήσουμε την επίθεση πρέπει πρώτα να έχουμε εγκαταστήσει το hping3(όπως αναφέρθηκε παραπάνω) .Στη συνέχεια συνδεόμαστε στον worker_1 ως εξής ssh docker@172.19.0.3 password:docker.

Επειτα τρέχουμε την παρακάτω εντολή για να επιτεθούμε στον worker_2 με ip 172.19.0.4.

```
sudo hping3 -V -c 400 -d 120 -S -p 22 --flood --rand-source
172.19.0.4
```

όπου,
-c:αριθμός πακέτων (400)
-d:μέγεθος πακέτων (120)
-S:τύπος των πακέτων(SYN)
-p:αριθμός port (22)
-flood:κατακλυσμός από πακέτα
-rand-source:εμφάνιση τυχαίων source ips

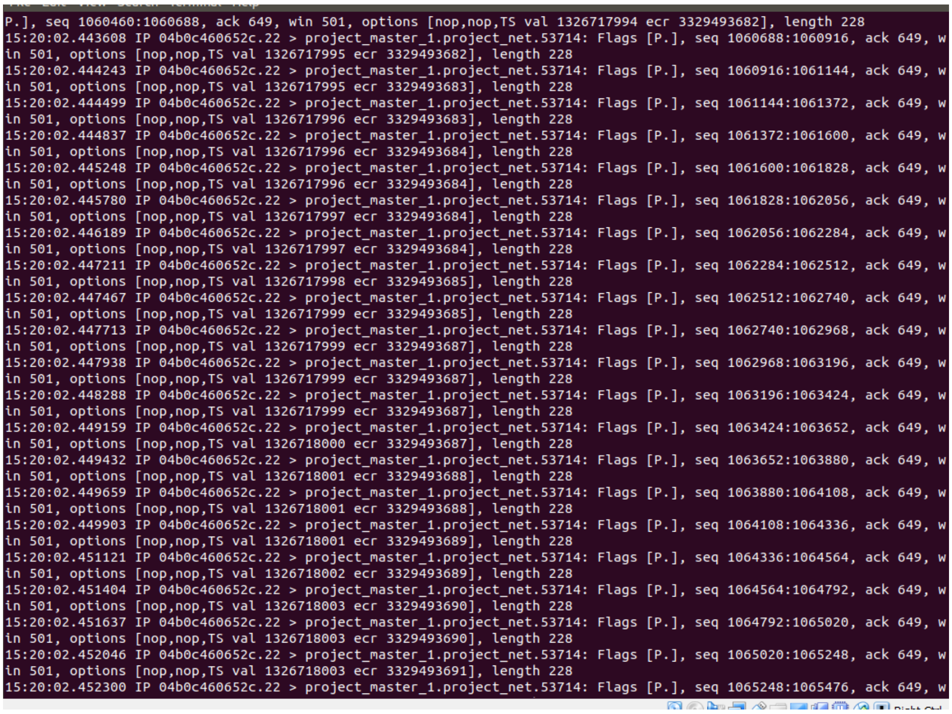
1.4. Παρακολούθηση επίθεσης

Για να παρακολουθήσουμε την επίθεση συνδεόμαστε στον worker_2 (ssh docker@172.19.0.4,password:docker) και τρέχουμε την παρακάτω εντολή

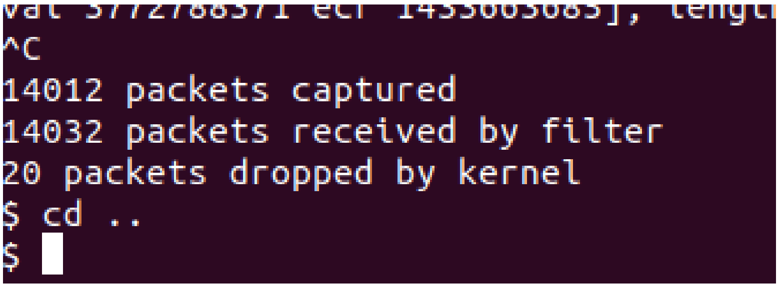
```
sudo tcpdump scr  172.19.0.3
ή tcpdump port 22 && 'tcp[tcpflags]== tcp-syn'
```

tcpflags ==tcp-syn : για την καταμέτρηση των πακέτα SYN που φτάνουν .

Το tcpdump μας δίνει μια περιγραφή για τα πακέτα που εντόπισε, όπως για παράδειγμα πληροφορίες για το μέγεθος και το περιεχόμενο των πακέτων. Τέλος, μόλις ολοκληρωθεί η διαδικασία εμφανίζεται το πλήθος των πακέτων που καταγράφηκαν, το πλήθος των πακέτων που εντοπίστηκαν με βάση το φίλτρο και το πλήθος των πακέτων που απορρίφθηκαν.Τα αποτελέσματα φαίνονται στις εικόνες που ακολουθούν.

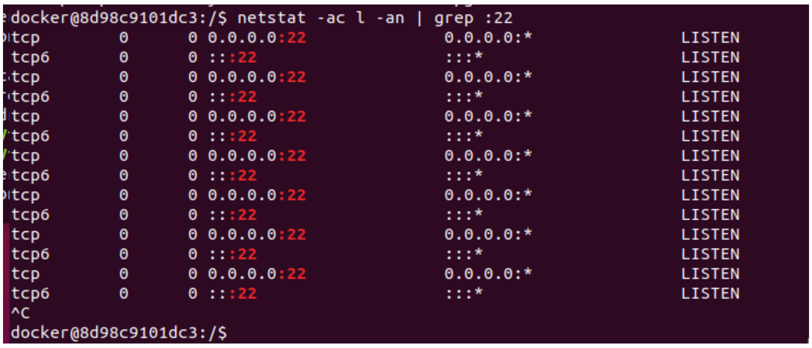


```
P.], seq 1060460:1060688, ack 649, win 501, options [nop,nop,TS val 1326717994 ecr 3329493682], length 228
15:20:02.443608 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1060688:1060916, ack 649, w
in 501, options [nop,nop,TS val 1326717995 ecr 3329493682], length 228
15:20:02.444243 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1060916:1061144, ack 649, w
in 501, options [nop,nop,TS val 1326717995 ecr 3329493683], length 228
15:20:02.444499 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1061144:1061372, ack 649, w
in 501, options [nop,nop,TS val 1326717996 ecr 3329493683], length 228
15:20:02.444837 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1061372:1061600, ack 649, w
in 501, options [nop,nop,TS val 1326717996 ecr 3329493684], length 228
15:20:02.445248 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1061600:1061828, ack 649, w
in 501, options [nop,nop,TS val 1326717996 ecr 3329493684], length 228
15:20:02.445780 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1061828:1062056, ack 649, w
in 501, options [nop,nop,TS val 1326717997 ecr 3329493684], length 228
15:20:02.446189 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1062056:1062284, ack 649, w
in 501, options [nop,nop,TS val 1326717997 ecr 3329493684], length 228
15:20:02.447211 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1062284:1062512, ack 649, w
in 501, options [nop,nop,TS val 1326717998 ecr 3329493685], length 228
15:20:02.447467 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1062512:1062740, ack 649, w
in 501, options [nop,nop,TS val 1326717999 ecr 3329493685], length 228
15:20:02.447713 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1062740:1062968, ack 649, w
in 501, options [nop,nop,TS val 1326717999 ecr 3329493687], length 228
15:20:02.447938 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1062968:1063196, ack 649, w
in 501, options [nop,nop,TS val 1326717999 ecr 3329493687], length 228
15:20:02.448288 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1063196:1063424, ack 649, w
in 501, options [nop,nop,TS val 1326717999 ecr 3329493687], length 228
15:20:02.449159 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1063424:1063652, ack 649, w
in 501, options [nop,nop,TS val 1326718000 ecr 3329493687], length 228
15:20:02.449432 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1063652:1063880, ack 649, w
in 501, options [nop,nop,TS val 1326718001 ecr 3329493688], length 228
15:20:02.449659 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1063880:1064108, ack 649, w
in 501, options [nop,nop,TS val 1326718001 ecr 3329493688], length 228
15:20:02.449903 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1064108:1064336, ack 649, w
in 501, options [nop,nop,TS val 1326718001 ecr 3329493689], length 228
15:20:02.451121 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1064336:1064564, ack 649, w
in 501, options [nop,nop,TS val 1326718002 ecr 3329493689], length 228
15:20:02.451404 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1064564:1064792, ack 649, w
in 501, options [nop,nop,TS val 1326718003 ecr 3329493690], length 228
15:20:02.451637 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1064792:1065020, ack 649, w
in 501, options [nop,nop,TS val 1326718003 ecr 3329493690], length 228
15:20:02.452046 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1065020:1065248, ack 649, w
in 501, options [nop,nop,TS val 1326718003 ecr 3329493691], length 228
15:20:02.452300 IP 04b0c460652c.22 > project_master_1.project_net.53714: Flags [P.], seq 1065248:1065476, ack 649, w
```



```
^C
14012 packets captured
14032 packets received by filter
20 packets dropped by kernel
$ cd ..
$
```

Το netstat εξετάζει τις συνδέσεις TCP .Εδώ η κίνηση στην πόρτα 22



```
docker@8d98c9101dc3:/$ netstat -ac l -an | grep :22
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
^C
docker@8d98c9101dc3:/$
```

Το netstat όπως βλέπουμε δεν κατάφερα να λειτουργήσει σωστά.

1.5. Αντιμετώπιση επίθεσης

Για την προστασία έναντι των επιθέσεων αυτών θα τροποποιήσουμε το iptables ,όπως παρουσιάζεται και στην συνέχεια.

```
sudo iptables -I INPUT -s 172.19.0.3 -j DROP
sudo iptables -L
sudo iptables -nvL
```

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  172.19.0.3           anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination
DOCKER-USER  all  --  anywhere             anywhere
DOCKER-ISOLATION-STAGE-1  all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ES
TABLISHED
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ES
```

```
Chain INPUT (policy ACCEPT 440 packets, 264K bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 DROP       all  --  *      *       172.19.0.3           0.0.0.0/0


Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

8732K  407M DOCKER-USER  all  --  *      *       0.0.0.0/0            0.0.0.0/0

8732K  407M DOCKER-ISOLATION-STAGE-1  all  --  *      *       0.0.0.0/0
   0.0.0.0/0
75542   25M ACCEPT     all  --  *      br-96dbdf922b22  0.0.0.0/0            0.0
.0.0/0            ctstate RELATED,ESTABLISHED
66558 3977K DOCKER     all  --  *      br-96dbdf922b22  0.0.0.0/0            0.0
.0.0/0
8589K  378M ACCEPT     all  --  br-96dbdf922b22 !br-96dbdf922b22  0.0.0.0/0
       0.0.0.0/0
66558 3977K ACCEPT     all  --  br-96dbdf922b22 br-96dbdf922b22  0.0.0.0/0
      0.0.0.0/0
```

Ωστόσο απο την παραπάνω διαδικασία δεν κατάφερα να πάρω τα επιθυμητά αποτελέσματα.

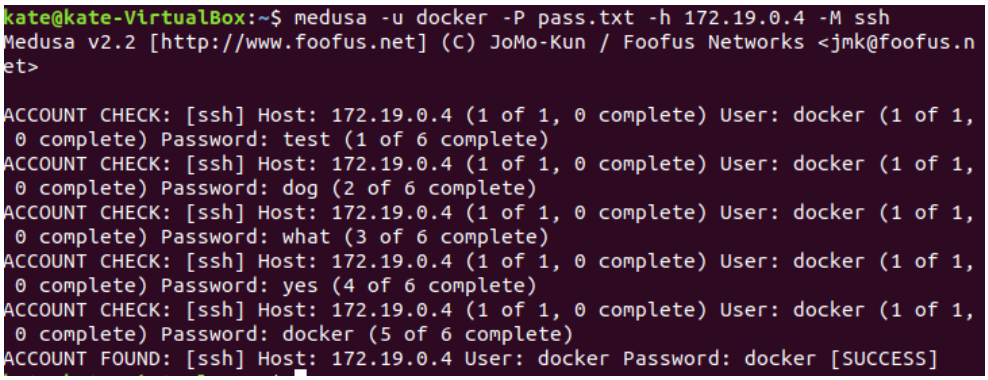
2. SSH Brute Force Attacks

Το SSH είναι ένα ασφαλές πρωτόκολλο το οποίο επιτρέπει τη μεταφορά δεδομένων και την επικοινωνία μεταξύ δύο μελών του ίδιου δικτύου.

2.1. Medusa

Για την επίτευξη μιας SSH Brute Forceεπίθεσης χρησιμοποιήσαμε το εργαλείο medusa και ένα pass.txt αρχείο με διάφορους κωδικούς. Θα επιτεθουμε και πάλι στον worker_2 με Ip 172.19.0.4

```
medusa -u docker -P pass.txt -h  172.19.0.4 -M ssh
```



Η επίθεση ήταν επιτυχής και βρήκαμε τον κωδικό του worker_2.

2.2. Αντιμετώπιση της επίθεσης

Για να αντιμετωπίσουμε την επίθεση θα χρησιμοποιήσουμε το εργαλείο Fail2Ban.Το Fail2Ban είναι ένα πλαίσιο λογισμικού πρόληψης εισβολών που προστατεύει τους διακομιστές των υπολογιστών από βίαιες επιθέσεις.

Εμείς στην εργασία τροποποιούμε το αρχείο jail.local ,όπως παρουσιάζεται στην συνέχεια,αφού πρώτα αντιγράψουμε το jail.conf στο αρχείο jail.local με την παρακάτω εντολή ,καθώς ενδέχεται να ενημερωθεί και το ίδιο το αρχείο και να χαθούν οι ρυθμίσεις που έχει εισάγει ο διαχειριστής.

```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

```
[sshd]
enabled = true
port = ssh
action = iptables-multiport
logpath = /var/log/auth.log
```

```
maxretry = 3
```

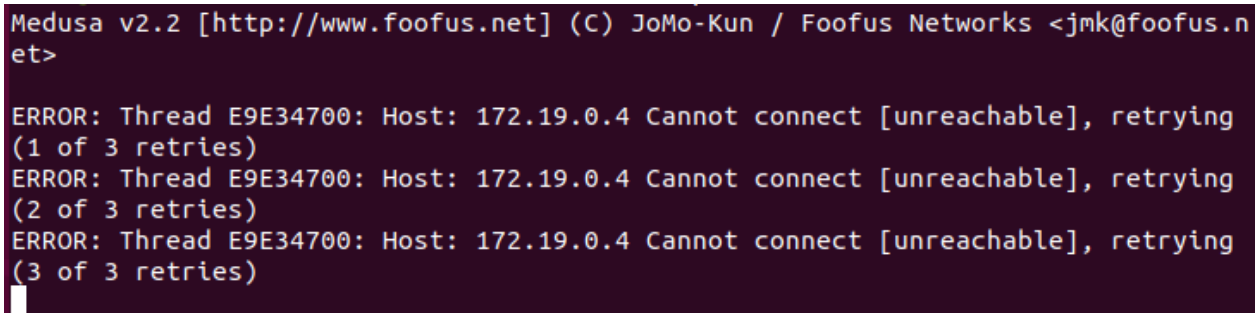
enables=true: για την ενεργοποίηση του μηχανισμού

port=ssh Η πόρτα της ssh (22)
action = iptables-multiport iptables για την απαγόρευση εισόδου
logpath = /var/log/auth.log Αρχείο καταγραφής
maxretry = 5 Μέγιστος αριθμός προσπαθειών

Τρέχουμε παλι την παρακάτω εντολή και παρατηρούμε ότι τώρα δεν μπορεί να πραγματοποιηθεί η επίθεση .

```
medusa -u docker -P pass.txt -h  172.19.0.4 -M ssh
```

fail2ban



Επιπλεόν δεν ήταν επιτυχης η προσπάθεια για συνδέσεις μέσω key.

3. Local/Remote SSH Forwarding

Το SSH Tunneling, είναι η δυνατότητα χρήσης του ssh για τη δημιουργία αμφίδρομης κρυπτογραφημένης σύνδεσης δικτύου μεταξύ μηχανών μέσω των οποίων μπορούν να ανταλλάσσονται δεδομένα, συνήθως TCP / IP.

Παρέχει επίσης έναν τρόπο για να ασφαλίσετε την κυκλοφορία δεδομένων οποιασδήποτε δεδομένης εφαρμογής μέσω προώθησης θύρας, ουσιαστικά διοχέτευση οποιασδήποτε θύρας TCP / IP μέσω SSH. Αυτό σημαίνει ότι η κυκλοφορία δεδομένων της εφαρμογής κατευθύνεται να ρέει μέσα σε μια κρυπτογραφημένη σύνδεση SSH, έτσι ώστε να μην μπορεί να υποβλεφθεί ή να υποκλαπεί κατά τη μεταφορά.

Αρχικά εγκαθιστούμε τον apache και τρέχουμε την υπηρεσία και βλέπουμε και την ip του μηχανήματός μας.

```
sudo apachectl start
sudo netstat -antlupe
```

```
docker@644617e2efba:/project$ sudo apachectl start
 [sudo] password for docker:
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.19.0.2. Set the 'Serv
 erName' directive globally to suppress this message
docker@644617e2efba:/project$ sudo netstat -antlupe
 Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       User       Inode      PID/Program name

tcp        0      0 127.0.0.11:43263        0.0.0.0:*               LISTEN      0          50813      -

tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      0          112487     11591/apache2

tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      0          51038      26/sshd

tcp6       0      0 :::22                   :::*                    LISTEN      0          51040      26/sshd

 udp        0      0 127.0.0.11:38650        0.0.0.0:*                           0          50812      -

 docker@644617e2efba:/project$
```

3.1. Local port forwarding

Στο local port forwarding , ο πελάτης επιθυμεί να αποκτήσει πρόσβαση σε κάποια υπηρεσία μιας απομακρυσμένης μηχανής.

Για την υλοποίηση του Local port forwarding τρεχουμε την παρακάτω εντολή

```
ssh -nNT -L 8000:localhost:80 kate@127.0.0.11
```

```
docker@644617e2efba:/project$ ssh -nNT -L 8000:localhost:80 kate@127.0.0.11
The authenticity of host '127.0.0.11 (127.0.0.11)' can't be established.
ECDSA key fingerprint is SHA256:q9CWGiHX+7JCNR3RaEkKJDtEKfKhSgJElnKUzfCyckA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '127.0.0.11' (ECDSA) to the list of known hosts.
kate@127.0.0.11's password:
```

Η παραπάνω εντολή δημιουργεί μια σήραγγα ssh μεταξύ του υπολογιστή και του διακομιστή και προωθεί όλη την κίνηση από το localhost: 80 στο localhost: 8000 .

3.2. Remote port forwarding

Στο remote port forwarding σκοπός είναι η αποκτήση πρόσβασης σε υπηρεσίες ή πόρους του απομακρυσμένου εξυπηρετητή, οι οποίες δεν είναι διαθέσιμες .Γι ‘αυτό το λόγο ,προωθείται η κίνηση από μία πύλη της τοπικής μηχανής σε μια πύλη της απομακρυσμένης μηχανής.

Για την υλοποίηση του remote port forwarding τρεχουμε την παρακάτω εντολή

```
ssh -nNT -R 4000:localhost:3000 user@127.0.0.11
```

```
docker@644617e2efba:/project$ ssh -nNT -R 4000:localhost:3000 kate@127.0.0.11
kate@127.0.0.11's password:
Permission denied, please try again.
kate@127.0.0.11's password:
Permission denied, please try again.
kate@127.0.0.11's password:
```

Η παραπάνω εντολή δημιουργεί μια σήραγγα ssh μεταξύ του υπολογιστή σας και του διακομιστή και προωθεί όλη την κίνηση από το localhost: 3000 (στον υπολογιστή σας) στο localhost: 4000 (στο πλαίσιο του διακομιστή).

βλέπουμε την σελίδα του apache αν τρέξουμε curl localhost:4000.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitio
nal.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2016-11-16
    See: https://launchpad.net/bugs/1288690
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }

  body, html {
    padding: 3px 3px 3px 3px;

    background-color: #D8DBE2;

    font-family: Verdana, sans-serif;
    font-size: 11pt;
    text-align: center;
```

4. VPN

Ένα εικονικό ιδιωτικό δίκτυο (VPN) επεκτείνει ένα ιδιωτικό δίκτυο σε ένα δημόσιο δίκτυο και επιτρέπει στους χρήστες να στέλνουν και να λαμβάνουν δεδομένα σε κοινόχρηστα ή δημόσια δίκτυα σαν να ήταν οι υπολογιστικές τους συσκευές συνδεδεμένες απευθείας στο ιδιωτικό δίκτυο.

4.1. Δημιουργία VPN

Στον φάκελο swarmlab-sec δημιουργούμε έναν νέο φάκελο τον vpn.Εκεί φτάχνουμε δύο νέα αρχεία
το create-vpn.sh και το create-user.sh. ,όπως φαίνεται και στη συνέχεια.

create-vpn.sh

```
#!/bin/bash
IP=127.0.0.1                                                #
Server IP
P=1194                                                      #
Server Port
OVPN_SERVER='10.80.0.0/16'                  # VPN Network

#vpn_data=/var/lib/swarmlab/openvpn/openvpn-services/   # Dir
to save data ** this must exist **
vpn_data=$PWD/openvpn-services/
if [ ! -d $vpn_data ]; then
 mkdir -p $vpn_data
fi

NAME=swarmlab-vpn-services                                  # name
of docker service
DOCKERnetwork=swarmlab-vpn-services-network                 #
docker network
docker=registry.vlabs.uniwa.gr:5080/myownvpn                #
docker image

docker stop  $NAME                          #stop container
sleep 1
docker container rm  $NAME                        #rm container

# rm config files
rm -f $vpn_data/openvpn.conf.*.bak
rm -f $vpn_data/openvpn.conf
rm -f $vpn_data/ovpn_env.sh.*.bak
rm -f $vpn_data/ovpn_env.sh

# create network
sleep 1
docker network create --attachable=true --driver=bridge --
subnet=172.50.0.0/16 --gateway=172.50.0.1 $DOCKERnetwork

#run container        see ovpn_genconfig
docker run --net=none -it -v $vpn_data:/etc/openvpn  -p
1194:1194 --rm $docker ovpn_genconfig  -u udp://$IP:1194 \
-N -d -c -p "route 172.50.20.0 255.255.255.0" -e "topology
subnet" -s $OVPN_SERVER

# create pki          see ovpn_initpki
docker run --net=none -v $vpn_data:/etc/openvpn  --rm -it
$docker ovpn_initpki
```

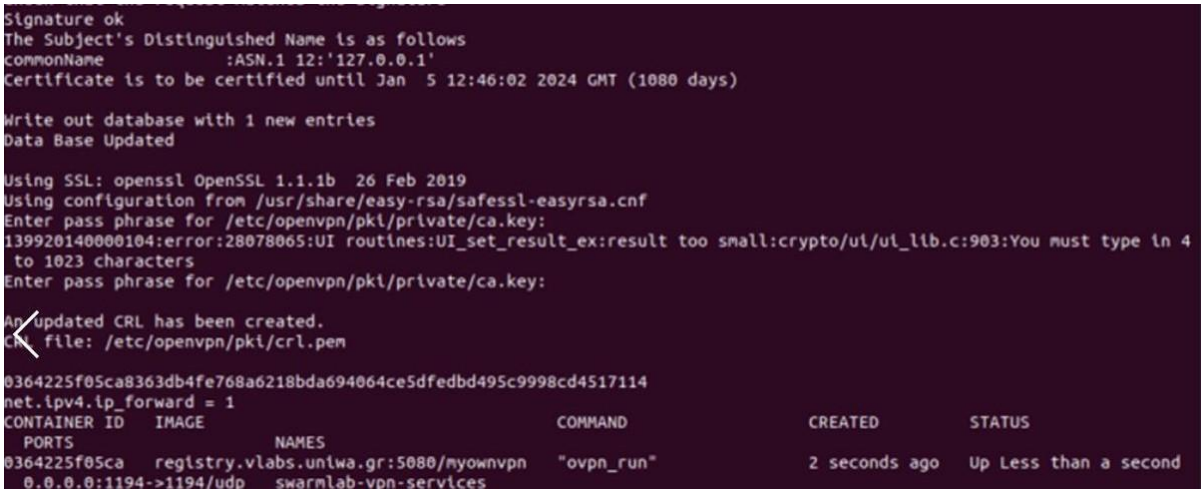
```
#                     see ovpn_copy_server_files
#docker run --net=none -v $vpn_data:/etc/openvpn  --rm $docker
ovpn_copy_server_files

#create vpn           see --cap-add=NET_ADMIN
sleep 1
docker run --detach --name $NAME -v $vpn_data:/etc/openvpn --
net=$DOCKERnetwork --ip=172.50.0.2 -p $P:1194/udp --cap-
add=NET_ADMIN $docker

sudo sysctl -w net.ipv4.ip_forward=1

#show created
docker ps
```

- IP=127.0.0.1 : localhost μέσα σε ένα κοντέινερ θα επιλυθεί στη στοίβα δικτύου αυτού του κοντέινερ
- P=1194 :port
- OVPN_SERVER='10.80.0.0/16' :καθορισμος διευθύνσεων και μασκών για τους Vpn clients
- vpn_data=$PWD/openvpn-services/ :κατάλογος για προσάρτηση δεδομένων
- NAME=swarmlab-vpn-services : ονομα του docker services
- docker run --net=none -it -v $vpn_data:/etc/openvpn -p 1194:1194 --rm $docker ovpn_genconfig -u udp://$IP:1194 \ -N -d -c -p "route 172.50.20.0 255.255.255.0" -e "topology subnet" -s $OVPN_SERVER :create config
- # create pki see ovpn_initpki docker run --net=none -v $vpn_data:/etc/openvpn --rm -it $docker ovpn_initpki :κλειδιά
- docker run --detach --name $NAME -v $vpn_data:/etc/openvpn --net=$DOCKERnetwork --ip=172.50.0.2 -p $P:1194/udp --cap-add=NET_ADMIN $docker : Run docker vpn service

Πριν τρέξουμε το αρχείο αλλάζουμε τα δικαιώματα ως εξής

```
chmod 700 create-vpn.sh
```

Πέρνουμε τα παρακάτω αποτελέσματα αφού τρέξουμε την εντολή **./create-vpn.sh.**



4.2. Δημιουργία χρήστη

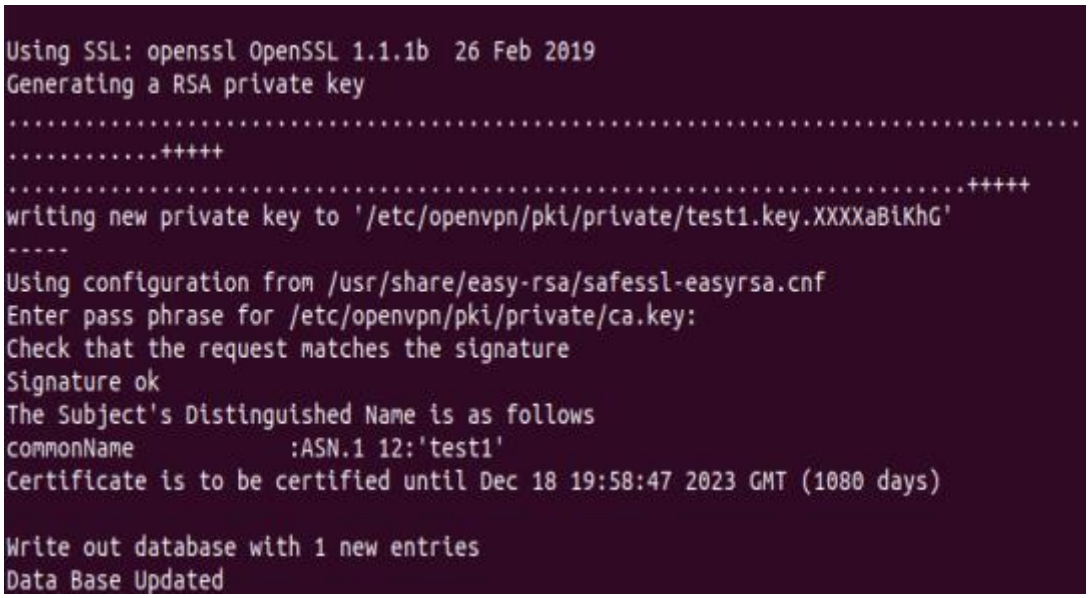
create-user.sh

```
USERNAME=test1
vpn_data=$PWD/openvpn-services/
docker=registry.vlabs.uniwa.gr:5080/myownvpn

docker run -v $vpn_data:/etc/openvpn --rm -it $docker easyrsa
build-client-full $USERNAME nopass
docker run -v $vpn_data:/etc/openvpn --log-driver=none --rm
$docker ovpn_getclient $USERNAME  > $USERNAME.ovpn
```

Πριν τρέξουμε το αρχείο αλλάζουμε τα δικαιώματα ως εξής

```
chmod 700 create-user.sh
```

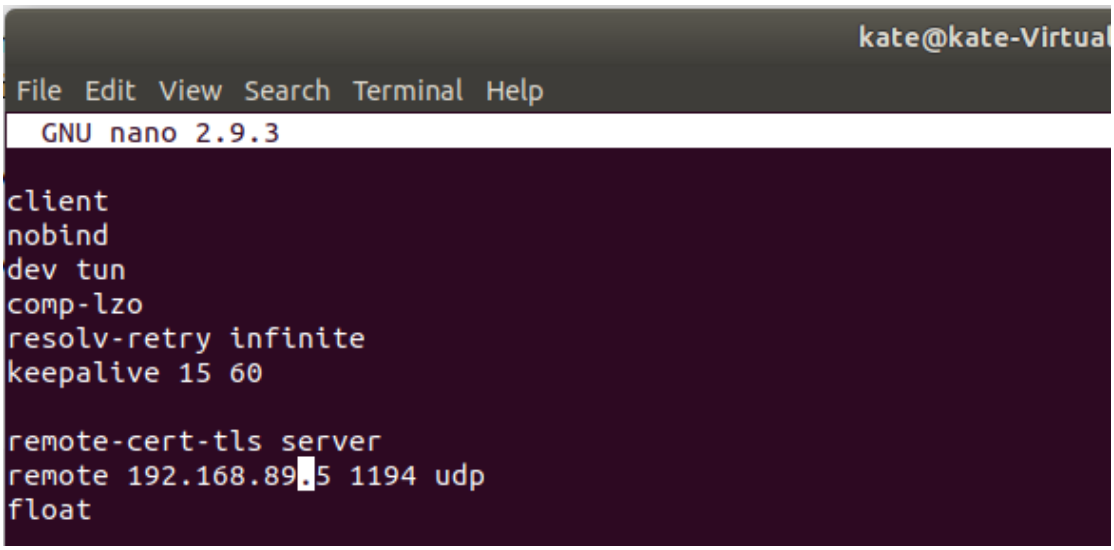
Πέρνουμε τα παρακάτω αποτελέσματα αφού τρέξουμε την εντολή **./create-user.sh.**



4.3. Τροποποίηση αρχείου test1.ovpn και σύνδεση

Τροποποιούμε το αρχείο test1.ovpn ,όπως φαίνεται στη συνέχεια

```
client
nobind
dev tun
comp-lzo
resolv-retry infinite
keepalive 15 60
remote-cert-tls server
remote 192.168.1.5 1194 udp
float
```



αντιγράφουμε το αρχείο

```
cp test1.ovpn ../myproj/project/test1.vpn
```

Στη συνέχεια εκτελούμε την παρακάτω εντολή για να γίνει η σύνδεση

```
openvpn --config ./test1.ovpn
```