Εργασία ΑΔΕ - Καμπεράκης Ιωάννης 47254 & Χριστοφορίδου Μαρία 47131

Πίνακας περιεχομένων

1. Μέρος Α: Δημιουργία και ρύθμιση των δικτύων.

Το **Docker** είναι μια πλατφόρμα που πακετάρει μια εφαρμογή και όλα της τα dependencies μαζί σε μορφή containers, δηλαδή απομονωμένες Περιοχές Χρήστη (User Spaces). Το πακετάρισμα αυτό σε containers που παρέχει το docker δίνει στην εφαρμογή τη δυνατότητα να δουλεύει σε οποιοδήποτε περιβάλλον. Έτσι αποφεύγεται η χρήση επιπλέον υπολογιστικών πόρων που θα απαιτούσε μια Εικονική Μηχανή (Virtual Machine).

1.1. Δημιουργία Docker Container

```
swarmalb-sec create
swarmlab-sec up size=5
swarmlab-sec login
sudo su
ssh docker@<master_IP>
sudo su
```

Έτσι έχουμε συνδεθεί στο master.

1.2. Δημιουργία Εικονικών Δικτύων στο Docker

Δημιουργούμε τα δίκτυα που μας χρειάζονται.

```
docker network create --driver=bridge --subnet=<network_IP/mask> <name_of_network>
```

1.3. Σύνδεση των Εικονικών Hosts στα Παραπάνω Δίκτυα

Χρησιμοποιούμε την εντολή όσες φορές χρειάζεται για όσα δίκτυα και hosts θέλουμε.

```
docker network connect <network_name> <host_name>
```

1.4. Σύνδεση Δικτύων με το Internet

Συνδεόμαστε σε όλους τους hosts των δικτύων μας για να τους δώσουμε σαν default gateway τον master.

```
swarmlab-sec login
sudo su
ssh docker@<host_IP>
sudo su
ip route add default via <master_ip_of_virt-net> dev <interface_of_virt-net>
```

1.5. Επικοινωνία Δικτύων A και B

Ανοίγουμε το αρχείο /etc/shorewall/rules και προσθέτουμε τις ανάλογες γραμμές έτσι ώστε τα δίκτυα μας να έχουν επικοινωνία στην πόρτα 80 ως εξής:

/etc/shorewall/rules

```
ACCEPT  netA                    netB      tcp     80
ACCEPT  netB                    netA      tcp     80
```

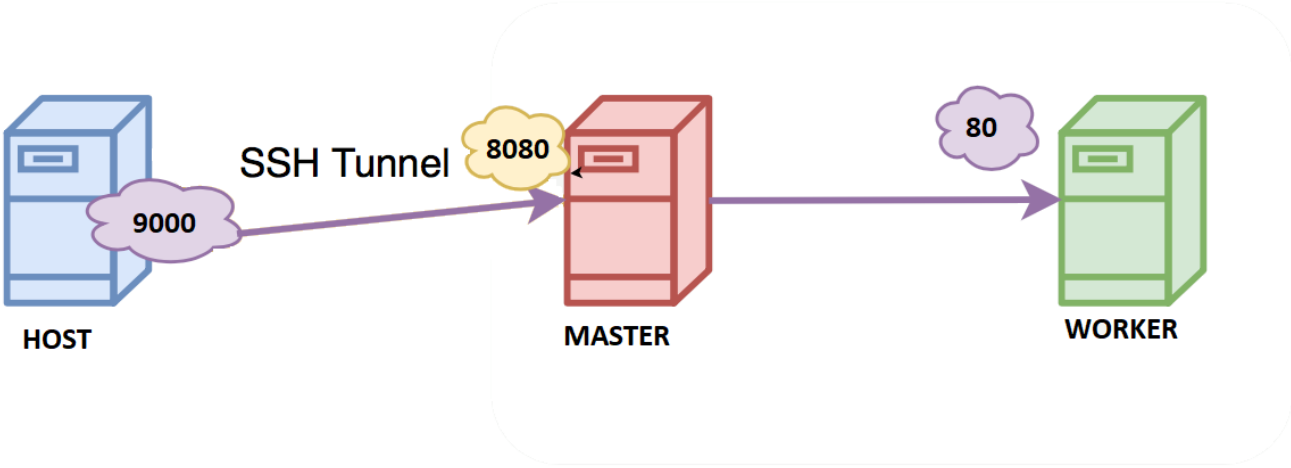
1.6. Ρύθμιση των Περιεχομένων των Rules, Interfaces, Zones, Policy, Masq και Resolv.conf

```
nano /etc/shorewall/rules
nano /etc/shorewall/interfaces
nano /etc/shorewall/zones
nano /etc/shorewall/policy
nano /etc/resolv.con (ορίζουμε το nameserver)
```

2. Μέρος Β: Tunneling

Το **SSH Tunneling** είναι μία μέθοδος δημιουργίας ασφαλών απομακρυσμένων συνδέσεων και μεταφοράς αρχείων πάνω από μη-έμπιστα δίκτυα με μια κρυπτογραφημένη SSH σύνδεση. Προσφέρει έναν τρόπο να μεταφερθεί με ασφάλεια κίνηση οποιασδήποτε εφαρμογής με χρήση port forwarding(tunneling TCP/IP ports over SSH). Αυτό σημαίνει πως η κίνηση δεδομένων της εφαρμογής κατευθύνεται σε μία ασφαλή SSH σύνδεση ώστε να μην μπορεί να υποκλαπεί κατά τη διάρκεια της μεταφοράς. Επίσης, προσφέρει πρόσθετη ασφάλεια σε εφαρμογές legacy που δεν υποστηρίζουν κρυπτογράφηση.

Για να δημιουργήσουμε αυτή τη σηράγγωση πρέπει πρώτα να γίνει σύνδεση του υπολογιστή μας (HOST) από τη θύρα 9000 με τη θύρα 8080 του MASTER και στη συνέχεια άλλη μια σύνδεση του MASTER με τη θύρα 80 του WORKER ο οποίος θα παρέχει το service που χρειαζόμαστε.




```
ssh -tL 9000:localhost:8080 <user>@<master_IP> ssh -L 8080:localhost:80
<user>@<worker_IP>
```

3. Μέρος Γ: VPN

Ένα **Εικονικό Ιδιωτικό Δίκτυο** (συνήθως αναφέρεται σαν **VPN**, Virtual Private Network) είναι ένα δίκτυο που χρησιμοποιεί κατά κύριο λόγο δημόσια τηλεπικοινωνιακή υποδομή, όπως το Διαδίκτυο, και δίνει τη δυνατότητα σε απομακρυσμένους χρήστες να έχουν πρόσβαση σε ένα κεντρικό οργανωτικό δίκτυο. Συνήθως απαιτεί από τους απομακρυσμένους χρήστες του δικτύου πιστοποίηση, και συχνά ασφαλίζει τα δεδομένα με τεχνολογίες κρυπτογράφησης για να εμποδιστεί η διάδοση των ιδιωτικών πληροφοριών σε μη εξουσιοδοτημένους τρίτους. Έτσι, δίνει τη δυνατότητα στους χρήστες να στέλνουν και να λαμβάνουν δεδομένα σε κοινόχρηστα ή δημόσια δίκτυα σαν να ήταν απευθείας συνδεδεμένες οι υπολογιστικές τους συσκευές με το ιδιωτικό δίκτυο.

Δημιουργούμε ένα VPN από την πλευρά του worker και φτιάχνουμε ένα χρήστη με τον οποίο θα συνδεθεί ο host μας στο VPN.

3.1. Δημιουργία Δικτύου που θα Παρέχει το VPN

```
docker network create --attachable=true --driver=bridge --subnet=<network_IP>
--gateway=<gateway_IP> <docker_nw_name>
```

3.2. Εκτέλεση του Container

```
docker run --net=none -it -v <physical_dir_path_to_save_data>:<virtual_path> --rm
<docker_image> ovpn_genconfig -u udp://<server_IP:server_port> \ -N -d -c -p "route
<network_IP> <network_mask>" -e "topology subnet" -s <VPN_network_IP>
```

3.3. Δημιουργία Κλειδιών για Σύνδεση Χρήστη

```
docker run --net=none -v <physical_dir_path_to_save_data>:<virtual_path> --rm -it
<docker_image> ovpn_initpki
```

3.4. Δημιουργία VPN

```
docker run --detach --name <VPN_name> -v
<physical_dir_path_to_save_data>:<virtual_path> --net=<docker_nw_name>
--ip=<VPN_host_IP> -p <physical_port>:<virtual_port>/udp --cap-add=NET_ADMIN
<docker_image>
```

3.5. Δημιουργία Χρήστη

```
docker run -v <physical_dir_path_to_save_data>:<virtual_path> --rm -it <docker_image>
easyrsa build-client-full <client_name> nopass
```

3.6. Αποστολή Ρυθμίσεων Χρήστη σε Αρχείο

```
docker run -v <physical_dir_path_to_save_data>:<virtual_path> --log-driver=none --rm
<docker_image> ovpn_getclient <client_name> > <path/filename.ovpn>
```

Το παραπάνω αρχείο πρέπει να αποσταλεί από τον host στον χρήστη του VPN ώστε να μπορεί να συνδεθεί. Για την πραγματοποίηση της σύνδεσης, ο χρήστης θα πρέπει να εκτελέσει την εντολή openvpn --config <filename.ovpn>.