## 1) CONFIGURATION OF APACHE

Αρχικά έκανα εγκατάσταση τον **apache** (yum install httpd), εκκίνηση **(systemctl start httpd)** και ενεργοποίηση ώστε να ξεκινά κάθε φορά που ανοίγω το vm **(systemctl enable httpd)**. Στη συνέχεια ενεργοποίησα MONIMA τα πρωτόκολλα **http** και **https** ώστε να είναι προσπελάσιμα από παντού και να ανοίγει η σελίδα μου στο διαδίκτυο. Τέλος πηγαίνοντας στο **/etc/httpd/conf.d** και γράφοντας **nano ssl.conf** πραγματοποίησα αλλαγές στο configuration. Πιο συγκεκριμένα:

- **SSLCertificateFile** αλλαγή του υπογεγραμμένου πιστοποιητικού του server μου -> **server.crt**
- **SSLCertificateKeyFile** αλλαγή του προσωπικού κλειδιού του server μου -> **server.key**
- **SSLCertificateChainFile** τοποθέτηση στο chain το path του root πιστοποιητικού -> **wrx.crt**
- Στο τέλος του αρχείου δημιουργήθηκε το εξής :
  <VirtualHost *:80>
  ServerName 83.212.106.137
  Redirect / https://83.212.106.137/

  </VirtualHost>

  Το οποίο επιτυγχάνει με την εντολή redirect, την ανακατεύθυνση των **http port 80** σε **https port 443**.

## 2) RULES OF D)

Εντολές :

- firewall-cmd --permanent --add-service=http
- firewall-cmd --permanent --add-service=https
- firewall-cmd --permanent --remove-service=ssh
- firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="195.251.255.75" service name="ssh" accept'
- firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="195.251.255.77" service name="ssh" accept'

```
[root@snf-890528 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0 eth1
  sources:
  services: dhcpv6-client http https
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule family="ipv4" source address="195.251.255.75" service name="ssh" accept
        rule family="ipv4" source address="195.251.255.77" service name="ssh" accept
```

**3) (E)**

**4)**

Εντολές:

1. cd ..
2. cd etc
3. cd pki
4. cd CA
5. openssl genrsa -out wrx.key 4096
6. openssl req -new -x509 -days 365 -key wrx.key -out wrx.crt
7. openssl genrsa -out server.key 4096
8. openssl req -new -key server.key -out server.csr
9. openssl x509 -req -days 365 -in server.csr -CA wrx.crt -CAkey wrx.key -set_serial 01 -out server.crt

Πηγαίνουμε στο **/etc/pki/CA** και δημιουργούμε νέο private κλειδί 4096 Bit για το δικό μας Certificate Authority **(wrx.key-εντολή 5)**. Έπειτα δημιουργούμε SSL πιστοποιητικό για το CA μας, με διάρκεια ζωής 365 ημέρες **(wrx.crt-εντολή 6)**. Στη συνέχεια προχωράμε σε δημιουργία private κλειδιού για τον server και αυτό 4096 bit **(server.key-εντολή 7)** και φτιάχνουμε και το CSR του server **(server.csr-εντολή 8)**. Τέλος υπογράφουμε το CSR με το δικό μας CA εκδίδοντας SSL πιστοποιητικό διάρκειας 365 ημερών **(server.crt-εντολή 9)**.

**5) (G)**

Μεταβαίνουμε στο path **/var/www/html** και δημιουργούμε νέο αρχείο **index.html.**
Χρησιμοποιήθηκε **html** και **javascript** για την υλοποίηση της σελίδας.
Στο αρχείο ορίζουμε ένα form το οποίο περιέχει τίτλο για το κουτάκι που θα συμπληρώνουμε την απάντηση **(label tag)** και ακολουθούν δύο **inputs**, το ένα text που δέχεται την απάντηση μας και το άλλο **submit** (κουμπί) για να υποβάλλουμε την φόρμα (ορίζονται και τα ids-values όπου χρειάζεται ώστε να χρησιμοποιηθεί συνάρτηση της javascript για τον έλεγχο της απάντησης). Προχωράμε στην υλοποίηση της συνάρτησης μέσα στο **<script>…</script> (function check())** όπου έχουμε 2 μεταβλητές, η **user_answer** η οποία λαμβάνει κάθε φορά τα δεδομένα που πληκτρολογούμε **(getElementById("answer").value)** και η **correct_answer** που είναι ο αριθμός μητρώου μας. Στη συνέχεια πραγματοποιούμε τον έλεγχο με **if-else** και ανάλογα εμφανίζουμε μήνυμα σε πλαίσιο με **success** ή **fail**. **(alert)**