ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

# Ομαδική Εργασία - Ασφάλεια Δικτύων και Επικοινωνιών

## Δημιουργία συνδέσεων IPSec

## ***ΕΚΦΩΝΗΣΗ ΕΡΓΑΣΙΑΣ***

Σε αυτή την εργασία, καλείστε να εγκαταστήσετε και να παραμετροποιήσετε IPSec συνδέσεις σε περιβάλλον linux, χρησιμοποιώντας το λογισμικό strongswan.

Η δημιουργία των συνδέσεων προτείνεται να γίνει με τη χρήση του Versatile IKE Control Interface vici plugin και του εργαλείου εντολών swanctl με το πρωτόκολλο IKEv2.

### (I) Δημιουργία και εγκατάσταση κλειδιών

Δημιουργήστε μία Αρχή Πιστοποίησης (ΑΠ - CA) η οποία θα χρησιμοποιήσει τον αλγόριθμο RSA για τη δημιουργία του ιδιωτικού κλειδιού της μήκους 4096 bit. Η ΑΠ μπορεί να δημιουργηθεί στον έναν από τους δύο κόμβους που θα χρησιμοποιήσετε στην άσκηση.

Δημιουργήστε ένα self-signed πιστοποιητικό για την ΑΠ.

Μέσω της ΑΠ να δημιουργήστε, για κάθε άκρο της σύνδεσης τα ιδιωτικά κλειδιά και τα αντίστοιχα πιστοποιητικά. Για τα δύο άκρα της σύνδεσης τα κλειδιά να είναι κλειδιά RSA μήκους 4096 bit.

Αντιγράψτε σε κάθε άκρο της σύνδεσης, τα απαιτούμενα πιστοποιητικά και κλειδιά στον αντίστοιχο φάκελο που απαιτείται (/etc/swanctl/{x509,x509ca,private,pubkey} ). Διαμορφώστε αντίστοιχα τις συνδέσεις σε κάθε άκρο, στο αρχείο /etc/swanctl/swanctl.conf και το μυστικό κλειδί τους στο αντίστοιχο αρχείο /etc/swanctl/secrets εφόσον απαιτείται (εναλλακτικά το μυστικό κλειδί μπορεί να οριστεί στο swanctl.conf αρχείο). **Σημείωση:** Μπορείτε να δημιουργήσετε τα απαραίτητα κλειδιά και πιστοποιητικά, είτε χρησιμοποιώντας την υπηρεσία pki του strongswan είτε χρησιμοποιώντας το openssl.

### (II) Δημιουργία και δοκιμή συνδέσεων

Δημιουργήστε και δοκιμάστε διαδοχικά τις παρακάτω συνδέσεις (connections) που περιγράφονται στα βήματα (Α) και (Β). Εκκινήστε διαδοχικά κάθε μία από τις παρακάτω συνδέσεις και επαληθεύστε την με τη βοήθεια ενός packet sniffer (πχ wireshark). Στο τελικό σας παραδοτέο να περιλαμβάνονται τα αρχεία διαμόρφωσης του ipsec από τα δύο άκρα, με όλες τις παραπάνω συνδέσεις και οποιοδήποτε άλλο αρχείο πιθανώς απαιτείται

**(Α) Σύνδεση host-to-host** (κόμβος-με-κόμβο) με IKE2 και με τη χρήση των παραπάνω πιστοποιητικών. Σε αυτή την περίπτωση θα χρειαστεί να δημιουργήσετε 2 VM.

**(Β) Σύνδεση site-to-site** (δίκτυο-με-δίκτυο) με IKE2 και με τη χρήση των παραπάνω πιστοποιητικών. Σε αυτή την περίπτωση θα χρειαστεί να δημιουργήσετε 4 VM. Θα πρέπει να ρυθμίσετε το κάθε VM που έχει το ρόλο του εσωτερικού host, να βλέπει τον αντίστοιχο Gateway ως default gateway και επίσης οι gateway να δρομολογούν τα πακέτα που λαμβάνουν.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

# 1 Εισαγωγή

Σκοπός της παρούσας εργασίας είναι η υλοποίηση ασφαλών συνδέσεων IPSec σε περιβάλλον Linux, με χρήση του λογισμικού strongSwan και του πρωτοκόλλου IKEv2.
Το IPSec αποτελεί βασική τεχνολογία ασφάλειας δικτύων αφού αντιμετωπίζει επιθέσεις στο επίπεδο δικτύου (μοντέλο TCP/IP) και ειδικά σε επιθέσεις που είναι ευάλωτο το IP. Σε αντίθεση με πρωτόκολλα ανώτερων επιπέδων το IPSec προστατεύει τα IP πακέτα ανεξάρτητα εφαρμογής. Το IPSec καλύπτει στόχους ασφαλείας όπως: Εμπιστευτικότητα, Ακεραιότητα, Αυθεντικοποίηση και Προστασία από επιθέσεις επανάληψης. Βασικά συστατικά του IPSec:

- Μηχανισμοί προστασίας των πακέτων: Το IPSec έχει 2 πρωτόκολλα για αυτή τη χρήση. 1) Authenticator Header (AH) που παρέχει αυθεντικοποίηση του αποστολέα και ακεραιότητα δεδομένων. 2) Encapsulation Security Payload (ESP). Είναι το πρωτόκολλο που χρησιμοποιήται περισσότερο και παρέχει κρυπτογράφηση δεδομένων, ακεραιότητα, αυθεντικοποίηση.
- Συσχετισμοί Ασφαλείας(SA): Ένα SA είναι μια “συμφωνία”, μιας κατεύθυνσης, μεταξύ των κόμβων και ορίζει το σύνολο των παραμέτρων ασφαλείας που θα εφαρμοστούν.
- Πρωτόκολλο ανταλλαγής Κλειδιών (IKE): Για την ασφαλή ανταλλαγή κοινών κλειδιών μεταξύ των κόμβων.

Το IPSec μπορεί να λειτουργήσει σε δύο καταστάσεις.

- Transport mode (Κατάσταση μεταγωγής):Η προστασία εφαρμόζεται μόνο στο payload του IP πακέτου, ενώ η IP επικεφαλίδα παραμένει ορατή. Χρησιμοποιείται κυρίως για άμεση επικοινωνία μεταξύ δύο τελικών συστημάτων (end-to-end).
- Tunnel mode (Κατάσταση διόδου): Ολόκληρο το IP πακέτο (επικεφαλίδα και payload) κρυπτογραφείται και τοποθετείται μέσα σε νέο IP πακέτο με νέα επικεφαλίδα. Χρησιμοποιείται συνήθως για επικοινωνία μέσω IPsec gateways και είναι η βασική λειτουργία για υλοποίηση VPN, καθώς αποκρύπτει πλήρως τις εσωτερικές διευθύνσεις δικτύου.

Για την υλοποίηση της εργασίας πραγματοποιήθηκε η κατάλληλη εγκατάσταση και παραμετροποίηση των απαιτούμενων εργαλείων, ενώ οι συνδέσεις

διαμορφώθηκαν και διαχειρίστηκαν μέσω του εργαλείου swanctl. Τέλος η ορθή λειτουργία των συνδέσεων επαληθεύτηκε με δοκιμές επικοινωνίας και ανάλυση της δικτυακής κίνησης.

# 2 Δημιουργία και εγκατάσταση κλειδιών

## 2.1 Προετοιμασία περιβάλλοντος

Στο πλαίσιο της άσκησης, η διαδικασία συνεχίστηκε σε περιβάλλον Kali Linux, όπου ο πρώτος κόμβος εγκαταστάθηκε από ISO στο VirtualBox, ενώ ο δεύτερος κόμβος δημιουργήθηκε ως clone του πρώτου, ώστε να έχουμε δύο διακριτά μηχανήματα για τη δοκιμή των συνδέσεων, όπως προβλέπεται στην άσκηση. Πριν ξεκινήσουμε με τη διαμόρφωση του IPSec, εκτελέσαμε τις παρακάτω εντολές για ενημέρωση του συστήματος και εγκατάσταση των απαραίτητων εργαλείων και στους δύο κόμβους:

- `apt-get update`
- `sudo apt install -y strongswan-swanctl charon-systemd libcharon-extra-plugins strongswan-pki`

Επεξήγηση εντολών:

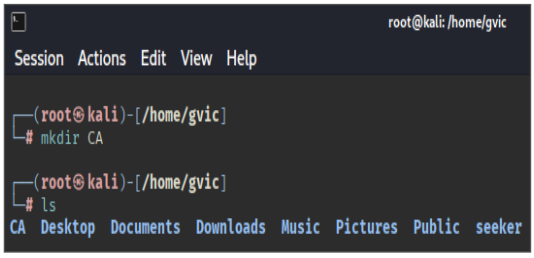
- **apt-get update**: Ενημερώνει τη λίστα των διαθέσιμων πακέτων από τα αποθετήρια, ώστε να γνωρίζει το σύστημα τις πιο πρόσφατες εκδόσεις.
- **sudo apt install -y strongswan-swanctl:** Εγκαθιστά το κύριο λογισμικό strongSwan και το εργαλείο **swanctl** για διαχείριση των IPSec/IKEv2 συνδέσεων.
- **charon-systemd:** Εγκαθιστά τον daemon Charon που εκτελεί τις IKEv2 διαδικασίες ως υπηρεσία systemd.
- **libcharon-extra-plugins:** Εγκαθιστά πρόσθετα plugins κρυπτογράφησης και αυθεντικοποίησης που χρησιμοποιούνται από το strongSwan.
- **strongswan-pki:** Παρέχει εργαλεία για τη δημιουργία και διαχείριση ψηφιακών πιστοποιητικών και κλειδιών (Public Key Infrastructure).

Με την εγκατάσταση αυτή εξασφαλίσαμε ότι το περιβάλλον ήταν έτοιμο για τη δημιουργία και παραμετροποίηση των IPSec συνδέσεων.

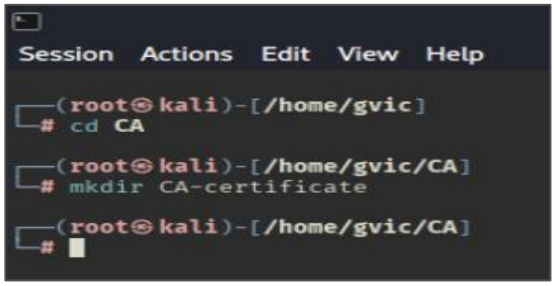
## 2.2 Δημιουργία Αρχής Πιστοποίησης (CA)

### 2.2.1 Δομή αρχείων

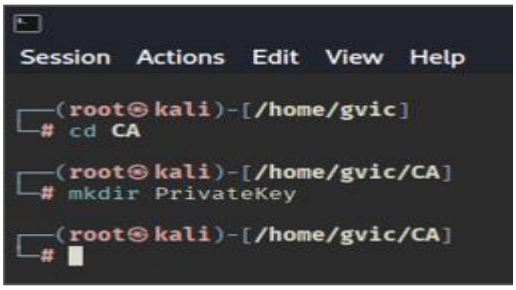
Για τη δημιουργία της Αρχής Πιστοποίησης (Certificate Authority – CA), αρχικά οργανώθηκε κατάλληλα η δομή των αρχείων στο σύστημα. Συγκεκριμένα, στον home φάκελο δημιουργήθηκε ένας κατάλογος με όνομα *CA* (Εικόνα 1) , μέσα στον οποίο δημιουργήθηκαν δύο επιπλέον υποκατάλογοι: ο *CA-certificate* (Εικόνα 2) για την αποθήκευση του πιστοποιητικού της CA και ο *PrivateKey* (Εικόνα 3) για την αποθήκευση του ιδιωτικού της κλειδιού.




*Εικόνα 1: Δημιουργία καταλόγου CA*




*Εικόνα 2: Δημιουργία υποκαταλόγου CA-certificate*



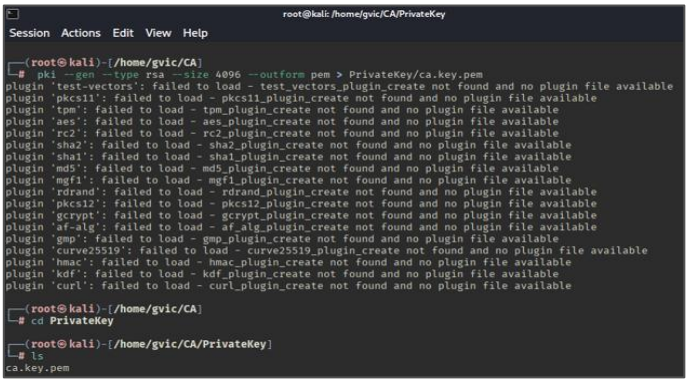

*Εικόνα 3: Δημιουργία υποκαταλόγου PrivateKey*

### 2.2.2 Δημιουργία ιδιωτικού κλειδιού της Αρχής Πιστοποίησης

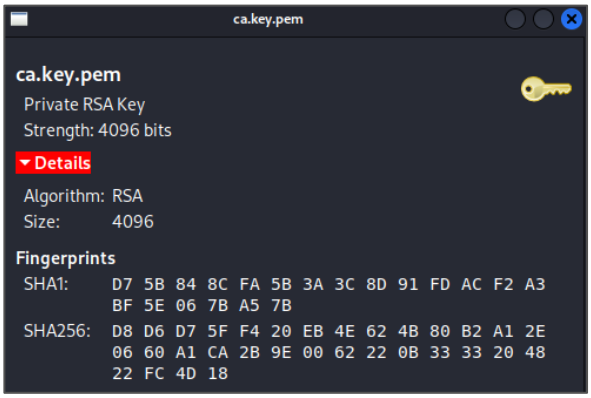
Για τη δημιουργία του ιδιωτικού κλειδιού της Αρχής Πιστοποίησης χρησιμοποιήθηκε το εργαλείο pki του strongSwan και εκτελέστηκε η παρακάτω εντολή: (Εικόνα 4)

- `pki --gen --type rsa --size 4096 --outform pem > PrivateKey/ca.key.pem`

Η παραπάνω εντολή δημιουργεί ένα ιδιωτικό κλειδί τύπου RSA μήκους 4096 bit, σε μορφή PEM. Το κλειδί αυτό αποτελεί το βασικό και απόρρητο στοιχείο της Αρχής Πιστοποίησης και χρησιμοποιείται για την υπογραφή των πιστοποιητικών που θα εκδοθούν για τα άκρα των IPSec συνδέσεων.




*Εικόνα 4: Δημιουργία ιδιωτικου κλειδιού Αρχής Πιστοποίησης*




*Εικόνα 5: Ιδιωτικό κλειδί Αρχής Πιστοποίησης*

### 2.2.3 Δημιουργία self-signed πιστοποιητικού της Αρχής Πιστοποίησης

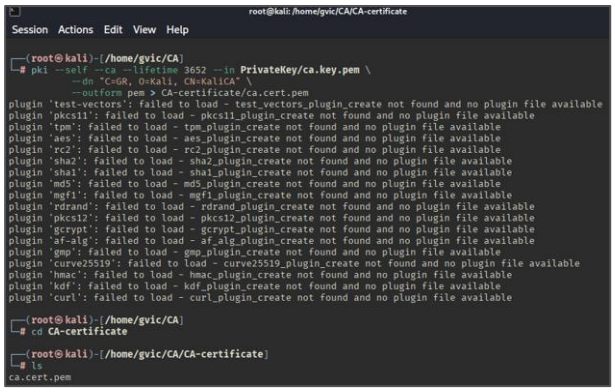
Μετά τη δημιουργία του ιδιωτικού κλειδιού της Αρχής Πιστοποίησης, ακολούθησε η δημιουργία του ψηφιακού πιστοποιητικού της, το οποίο είναι self-signed. Το πιστοποιητικό αυτό χρησιμοποιείται ώστε τα άκρα των IPSec συνδέσεων να μπορούν να εμπιστευτούν την Αρχή Πιστοποίησης. Για τον σκοπό αυτό εκτελέστηκε η παρακάτω εντολή: (Εικόνα 6)

- ```
  pki --self --ca --lifetime 3652 \
      --in PrivateKey/ca.key.pem \
      --dn "C=CA, O=Kali, CN=KaliCA" \
      --outform pem > CA-certificate/ca.cert.pem
  ```

Επεξήγηση εντολής:

- **pki –self**: Δημιουργεί ένα πιστοποιητικό το οποίο υπογράφεται από το ίδιο το ιδιωτικό κλειδί (self-signed).
- **--ca**: Δηλώνει ότι το πιστοποιητικό που δημιουργείται ανήκει σε Αρχή Πιστοποίησης.
- **--lifetime 3652**: Ορίζει τη διάρκεια ισχύος του πιστοποιητικού σε ημέρες (περίπου 10 έτη).
- **--in PrivateKey/ca.key.pem**: Χρησιμοποιεί το ιδιωτικό κλειδί της Αρχής Πιστοποίησης για την υπογραφή του πιστοποιητικού.
- **--dn "C=CA, O=Kali, CN=KaliCA"**: Καθορίζει το Distinguished Name του πιστοποιητικού, δηλαδή τα στοιχεία ταυτότητας της CA.
  ```

- **--outform pem**: Ορίζει τη μορφή εξόδου του πιστοποιητικού σε PEM.
- **> CA-certificate/ca.cert.pem**: αποθηκεύει το παραγόμενο πιστοποιητικό στο αντίστοιχο αρχείο.




*Εικόνα 6: Δημιουργία πιστοποιητικού της CA*

## 2.3 Δημιουργία και προετοιμασία κόμβων επικοινωνίας

### 2.3.1 Δομή αρχείων

Για τη δημιουργία των κόμβων επικοινωνίας, χρησιμοποιήθηκαν δύο μηχανήματα με ονομασίες Kali1 και Kali2. Η αρχική δημιουργία όλων των απαραίτητων αρχείων και καταλόγων πραγματοποιήθηκε στον πρώτο κόμβο (Kali1). Για λόγους ομοιομορφίας και ευκολίας διαχείρισης, οργανώθηκε κατάλληλα η δομή των αρχείων στον home κατάλογο του συστήματος.

Συγκεκριμένα, δημιουργήθηκαν οι φάκελοι Kali1 και Kali2, οι οποίοι αντιστοιχούν στους δύο κόμβους επικοινωνίας. Εντός κάθε καταλόγου δημιουργήθηκε ξεχωριστός υποφάκελος για την αποθήκευση των ιδιωτικών κλειδιών και των πιστοποιητικών του αντίστοιχου κόμβου, ως εξής:

- **Kali1/PrivateKeyk1** για την αποθήκευση του ιδιωτικού κλειδιού του κόμβου Kali1
- **Kali1/k1certificate** για την αποθήκευση του ψηφιακού πιστοποιητικού του κόμβου Kali1

- **Kali2/PrivateKeyk2** για την αποθήκευση του ιδιωτικού κλειδιού του κόμβου Kali2

- **Kali2/k2certificate** για την αποθήκευση του ψηφιακού πιστοποιητικού του κόμβου Kali2

### 2.3.2 Δημιουργία ιδιωτικών κλειδιών

Αφού δημιουργήθηκε η κατάλληλη δομή φακέλων για τους κόμβους, ακολούθησε η δημιουργία των ιδιωτικών κλειδιών που θα χρησιμοποιηθούν για την αυθεντικοποίηση των κόμβων κατά τη δημιουργία των IPSec συνδέσεων.

Για τον πρώτο κόμβο (Kali1) εκτελέστηκε η παρακάτω εντολή:

- `pki --gen --type rsa --size 4096 --outform pem > Kali1/PrivateKeyK1/k1.key.pem`

Αντίστοιχα, για τον δεύτερο κόμβο (Kali2) εκτελέστηκε η εντολή:

- `pki --gen --type rsa --size 4096 --outform pem > Kali2/PrivateKeyK2/k2.key.pem`

Η εντολή pki --gen δημιουργεί ένα νέο ιδιωτικό κλειδί, χρησιμοποιώντας τον αλγόριθμο RSA με μήκος 4096 bit. Τα παραγόμενα κλειδιά αποθηκεύονται σε μορφή PEM και αποτελούν τα απόρρητα κλειδιά των αντίστοιχων κόμβων, τα οποία θα χρησιμοποιηθούν στη συνέχεια για την έκδοση των ψηφιακών πιστοποιητικών τους και τη διαδικασία αυθεντικοποίησης μέσω IKEv2.

### 2.3.3 Δημιουργία πιστοποιητικών

Αρχικά, για κάθε κόμβο δημιουργήθηκαν αιτήσεις πιστοποιητικών (Certificate Signing Requests – CSR), οι οποίες δεν είναι από μόνες τους έγκυρα πιστοποιητικά, αλλά περιέχουν το δημόσιο κλειδί του κόμβου και τα στοιχεία ταυτότητάς του. Οι αιτήσεις αυτές υποβάλλονται στην Αρχή Πιστοποίησης (KaliCA) ώστε να υπογραφούν και να εκδοθούν τα αντίστοιχα ψηφιακά

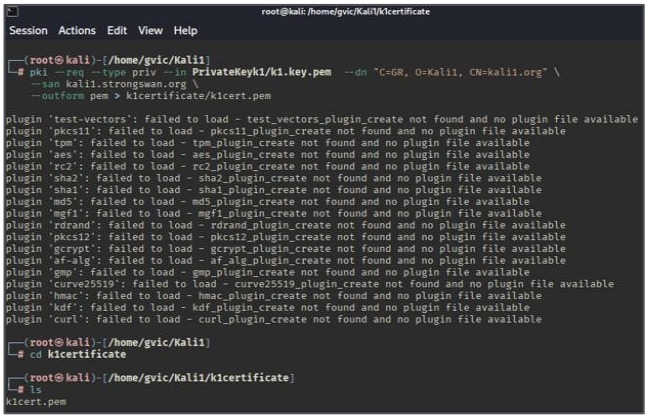
πιστοποιητικά. Η διαδικασία εκτελέστηκε στον κόμβο Kali1 και για τους δυο κόμβους αντίστοιχα .

Η παρακάτω εντολή δημιουργεί το αρχείο αίτησης πιστοποιητικού k1cert.pem, χρησιμοποιώντας το ιδιωτικό κλειδί του κόμβου Kali1 (Εικόνα 7) :

- ```
  pki --req --type priv --in PrivateKeyk1/k1.key.pem \
  --dn "C=GR, O=Kali, CN=kali1.org" \
  --san kali1.strongswan.org \
  --outform pem > k1certificate/k1cert.pem
  ```

Επεξήγηση εντολής:

- **pki --req**: δημιουργεί αίτηση πιστοποιητικού (CSR)
- **--type priv**: δηλώνει ότι χρησιμοποιείται ιδιωτικό κλειδί
- **--in**: το ιδιωτικό κλειδί του κόμβου
- **--dn**: τα στοιχεία ταυτότητας του κόμβου (Distinguished Name)
- **--san**: εναλλακτικό όνομα (Subject Alternative Name)
- **--outform pem**: μορφή εξόδου PEM



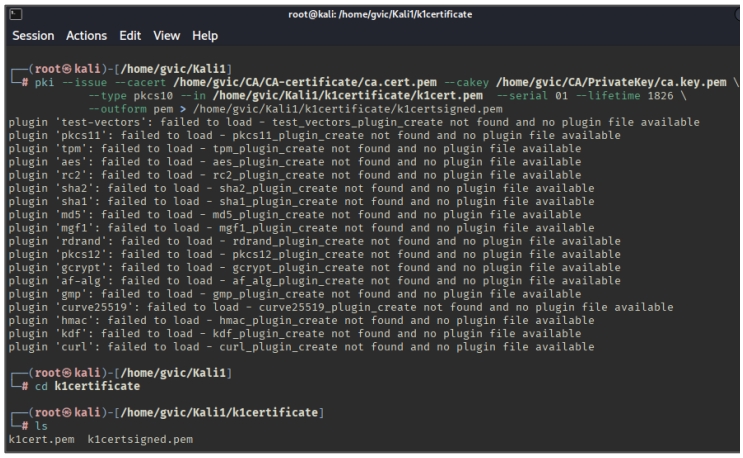

*Εικόνα 7: Δημιουργία αίτησης πιστοποιητικού*

Στη συνέχεια, η αίτηση πιστοποιητικού υπογράφεται από την Αρχή Πιστοποίησης KaliCA, με αποτέλεσμα τη δημιουργία ενός έγκυρου και επαληθεύσιμου ψηφιακού πιστοποιητικού, με διάρκεια ισχύος δέκα (10) ετών.
  ```

- ```
  pki --issue --cacert /home/gvic/CA/CA-certificate/ca.cert.pem \
  --cakey /home/gvic/CA/PrivateKey/ca.key.pem \
  --type pkcs10 –in /home/gvic/Kali1/k1certificate/k1cert.pem \
  --serial 01 --lifetime 3652 \
  --outform
  pem >/home/gvic/Kali1/k1certificate/k1certsigned.pem
  ```

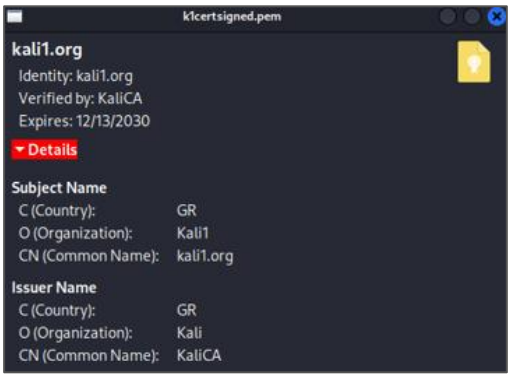
Επεξήγηση εντολής:

- **pki --issue**: Υπογράφει αίτηση πιστοποιητικού
- **--cacert**: Πιστοποιητικό της Αρχής Πιστοποίησης
- **--cakey**: Ιδιωτικό κλειδί της Αρχής Πιστοποίησης
- **--type pkcs10**: Δηλώνει ότι η είσοδος είναι CSR
- **--serial**: Σειριακός αριθμός πιστοποιητικού
- **--lifetime**: Διάρκεια ισχύος σε ημέρες
- **--outform pem**: Μορφή εξόδου PEM




*Εικόνα 8: Δημιουργία υπογεγραμμένου πιστοποιητικού*

Το παραγόμενο αρχείο k1certsigned.pem αποτελεί το τελικό, υπογεγραμμένο πιστοποιητικό του κόμβου Kali1 και μπορεί να επαληθευτεί από οποιονδήποτε εμπιστεύεται την KaliCA. (Εικόνα 9)
  ```




*Εικόνα 9: Υπογεγραμμένο πιστοποιητικό του κόμβου Kali1*

Η ίδια ακριβώς διαδικασία ακολουθήθηκε και για τον κόμβο Kali2, με τη δημιουργία αίτησης πιστοποιητικού k2cert.pem και υπογεγραμμένου πιστοποιητικού k2certsigned.pem. (Εικόνα 10)


*Εικόνα 10: Υπογεγραμμένο πιστοποιητικό του κόμβου Kali2*

Αφού ολοκληρώθηκε η δημιουργία των αρχείων που αφορούν τον δεύτερο κόμβο (Kali2), ο αντίστοιχος φάκελος μεταφέρθηκε στο δεύτερο μηχάνημα μέσω shared folder. Με τον τρόπο αυτό διασφαλίστηκε ότι κάθε κόμβος διαθέτει αποκλειστικά τα αρχεία που τον αφορούν, ενώ διατηρήθηκε η ίδια δομή αρχείων και στα δύο συστήματα.

## 2.4 Εγκατάσταση πιστοποιητικών και κλειδιών στο swanctl

Μετά τη δημιουργία των ιδιωτικών κλειδιών και των υπογεγραμμένων πιστοποιητικών, πραγματοποιήθηκε η αντιγραφή των απαραίτητων αρχείων σε κάθε άκρο της σύνδεσης, στους καταλόγους που χρησιμοποιεί το swanctl για τη διαχείριση πιστοποιητικών και κλειδιών.

### 2.4.1 Τοποθέτηση αρχείων

Σε κάθε κόμβο πραγματοποιήθηκαν τα εξής:

- Στον κατάλογο **x509** τοποθετήθηκε το υπογεγραμμένο πιστοποιητικό του ίδιου του κόμβου
- Στον κατάλογο **x509ca** τοποθετήθηκε το πιστοποιητικό της Αρχής Πιστοποίησης (KaliCA)
- Στον κατάλογο **private** τοποθετήθηκε το ιδιωτικό κλειδί του κόμβου
- Στον κατάλογο **pubkey** τοποθετήθηκε το δημόσιο κλειδί του κόμβου

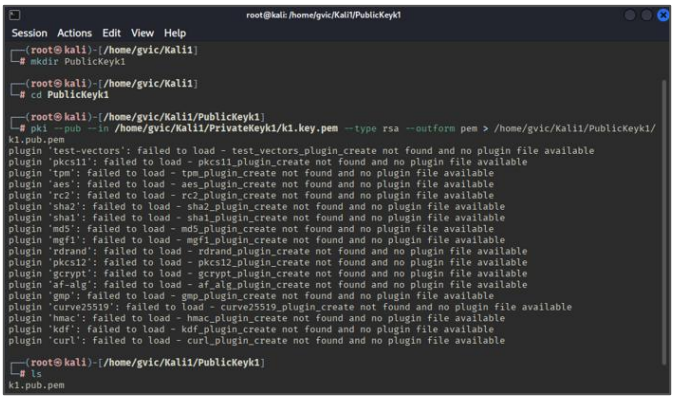
### 2.4.2 Δημιουργία / εξαγωγή δημόσιου κλειδιού

Το δημόσιο κλειδί κάθε κόμβου εξήχθη από το αντίστοιχο ιδιωτικό κλειδί με την παρακάτω εντολή.(Εικόνα 11) Η διαδικασία εκτελέστηκε ξεχωριστά σε κάθε κόμβο.

```
pki --pub --in /home/gvic/Kali1/PrivateKeyk1/k1.key.pem \
  --type rsa --outform pem > \
  /home/gvic/Kali1/PublicKeyk1/k1.pub.pem
```

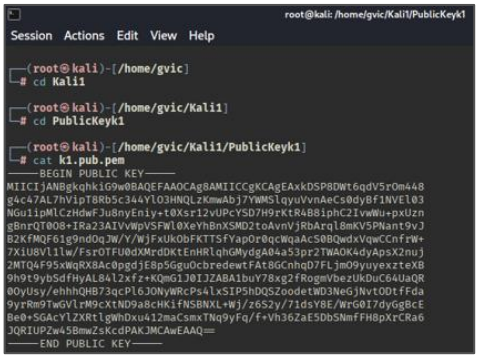
Επεξήγηση εντολής:

- **pki --pub**: Εξάγει το δημόσιο κλειδί
- **--in**: Το ιδιωτικό κλειδί από το οποίο προκύπτει
- **--type rsa**: Τύπος αλγορίθμου
- **--outform pem**: Μορφή εξόδου PEM

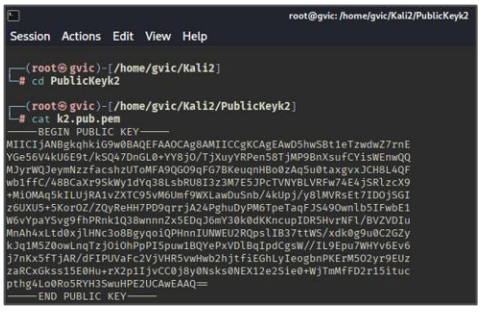



*Εικόνα 11: Δημιουργία δημόσιου κλειδιού για Kali1*

Η ίδια εντολή εκτελέστηκε και για τον κόμβο Kali2, χρησιμοποιώντας το αντίστοιχο ιδιωτικό κλειδί και κατάλογο.




*Εικόνα 12: Εμφάνιση δημόσιου κλειδιού Kali1 με εντολή cat*




*Εικόνα 13: Εμφάνιση δημόσιου κλειδιού Kali2*

Με την ολοκλήρωση των παραπάνω βημάτων, κάθε κόμβος διαθέτει όλα τα απαραίτητα κλειδιά και πιστοποιητικά στις σωστές τοποθεσίες, ώστε να μπορεί να προχωρήσει η παραμετροποίηση των IPSec συνδέσεων μέσω του swanctl.

# 3 Συνδέσεις IPSec

## 3.1 Σύνδεση Host-to-Host με IKEv2

Στο παρόν στάδιο υλοποιήθηκε σύνδεση τύπου host-to-host μεταξύ δύο κόμβων Kali Linux (Kali1 και Kali2), χρησιμοποιώντας το πρωτόκολλο IKEv2 και πιστοποιητικά X.509 που εκδόθηκαν από την εσωτερική Αρχή Πιστοποίησης (KaliCA).

Και τα δύο εικονικά μηχανήματα συνδέθηκαν στο ίδιο Host-Only Adapter του VirtualBox, εξασφαλίζοντας άμεση επικοινωνία σε απομονωμένο δίκτυο χωρίς πρόσβαση στο Internet.

Αρχικά, με την εντολή **ip a** επιβεβαιώθηκαν οι διευθύνσεις IP των δύο κόμβων:

- Kali1: 192.168.56.102
- Kali2: 192.168.56.101

## 3.2 Διαμόρφωση strongSwan (swanctl)

### 3.2.1 Αρχεία διαμόρφωσης swanctl.conf

Σε κάθε κόμβο δημιουργήθηκε το αρχείο ρυθμίσεων, στο οποίο ορίστηκε η σύνδεση host-to-host. Για τη διασφάλιση ισχυρής κρυπτογράφησης, χρησιμοποιήθηκαν σύγχρονοι και ασφαλείς αλγόριθμοι, όπως:

- **IKE: aes256-sha256-modp2048**

Αυτό αφορά το IKE Security Association (IKE SA), δηλαδή το ασφαλές κανάλι που χρησιμοποιείται για τη διαπραγμάτευση παραμέτρων ασφάλειας, την ανταλλαγή κλειδιών και τη δημιουργία των IPsec tunnels. Αποτελείται από τρία μέρη:

AES256: Αλγόριθμος συμμετρικής κρυπτογράφησης, 256-bit κλειδί. Χρησιμοποιείται για προστασία των μηνυμάτων κατά την ανταλλαγή κλειδιών.

SHA256: Αλγόριθμος κατακερματισμού (hash), χρησιμοποιείται για integrity/authentication των μηνυμάτων.

MODP2048: Diffie-Hellman group για ανταλλαγή κλειδιών (2048-bit modulus). Παράγει κοινό μυστικό που χρησιμοποιείται για κρυπτογράφηση δεδομένων.

- **ESP: aes256-sha256**

Αυτό αφορά το CHILD SA / ESP tunnel, δηλαδή το κανάλι που στέλνει τα πραγματικά δεδομένα (π.χ. ICMP, TCP, UDP).

AES256: Κρυπτογράφηση δεδομένων.

SHA256: Integrity/authentication για να διασφαλίσει ότι τα πακέτα δεν αλλοιώθηκαν.

Το παρακάτω αρχείο δημιουργήθηκε στο /etc/swanctl/swanctl.conf του κόμβου Kali1 (192.168.56.102).

```
connections {

   host-to-host {

      version = 2                    # Χρήση του πρωτοκόλλου IKEv2
      proposals = aes256-sha256-modp2048 # Αλγόριθμοι για το IKE SA

      local_addrs  = 192.168.56.102      # IP του τοπικού κόμβου (Kali1)
      remote_addrs = 192.168.56.101      # IP του απομακρυσμένου κόμβου (Kali2)

      local {
         auth  = pubkey              # Αυθεντικοποίηση με δημόσιο κλειδί
         certs = k1certsigned.pem       # Υπογεγραμμένο πιστοποιητικό Kali1
         id    = "C=GR, O=Kali1, CN=kali1.org"      # Ταυτότητα που δηλώνεται στο πιστοποιητικό
      }

      remote {
         auth = pubkey               # Αυθεντικοποίηση του απομακρυσμένου κόμβου
         id   = "C=GR, O=Kali2, CN=kali2.org"       # Αναμενόμενη ταυτότητα Kali2
      }

      children {
         net-net {

            local_ts  = 192.168.56.102/32  # Traffic selector για Kali1
            remote_ts = 192.168.56.101/32  # Traffic selector για Kali2

            esp_proposals = aes256-sha256  # Αλγόριθμοι για το ESP tunnel
         }
      }
   }
}
```

Το αντίστοιχο αρχείο στον κόμβο Kali2 (192.168.56.101).

```
connections {

   host-to-host {

      version = 2
      proposals = aes256-sha256-modp2048

      local_addrs  = 192.168.56.101
      remote_addrs = 192.168.56.102

      local {
         auth  = pubkey
         certs = k2certsigned.pem
         id    = "C=GR, O=Kali2, CN=kali2.org"
      }

      remote {
         auth = pubkey
         id   = "C=GR, O=kali1, CN=kali1.org"
      }

      children {
         net-net {

            local_ts  = 192.168.56.101/32
            remote_ts = 192.168.56.102/32

            esp_proposals = aes256-sha256
         }
      }
   }
}
```

### 3.2.2 Φόρτωση πιστοποιητικών και συνδέσεων

Μετά τη διαμόρφωση των αρχείων, εκτελέστηκαν οι παρακάτω εντολές σε κάθε κόμβο:

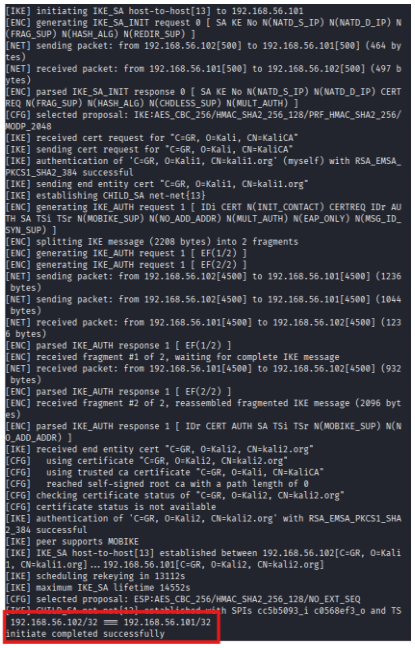
- **sudo swanctl --load-cred**: Φορτώνει τα πιστοποιητικά, τα ιδιωτικά και δημόσια κλειδιά από τους φακέλους
- **sudo swanctl --load-conns:** Φορτώνει τις συνδέσεις IPSec που έχουν οριστεί στο swanctl.conf

## 3.3 Εκκίνηση της σύνδεσης

Η σύνδεση εκκινήθηκε από τον κόμβο Kali1 με την εντολή:

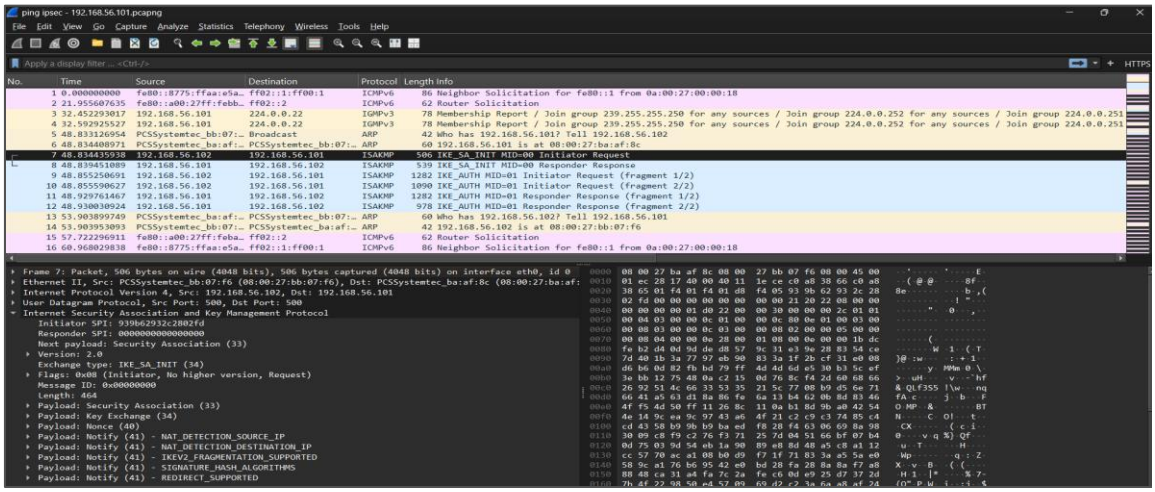
- `sudo swanctl --initiate --child net-net`

Η επιτυχής δημιουργία του IKE SA και του CHILD SA επιβεβαιώνει ότι το tunnel έχει δημιουργηθεί σωστά. (Εικόνα 14)




*Εικόνα 14: Επιτυχής δημιουργία tunnel*

Το IKEv2 τρέχει μόνο στην αρχή για να δημιουργηθεί το tunnel. Μόλις δημιουργηθεί το tunnel, όλο το traffic περνάει μέσα στο ESP, και δεν χρειάζεται ξανά IKE μέχρι να γίνει rekeying ή επανασύνδεση. (Εικόνα 15)



*Εικόνα 15: Εμφάνιση πρωτόκολλου IKE*

Κατά την ανάλυση της κίνησης στο Wireshark παρατηρήθηκε η διαδικασία εγκαθίδρυσης ενός IPsec VPN μέσω του πρωτοκόλλου IKEv2 (Internet Key Exchange v2) μεταξύ των hosts 192.168.56.102 (Initiator) και 192.168.56.101 (Responder). Η διαδικασία περιλαμβάνει τρία βασικά ανταλλασσόμενα μηνύματα (Request/Response), τα οποία περιγράφονται παρακάτω.

**1. IKE_SA_INIT (Request / Response)**

Το **IKE_SA_INIT** αποτελεί το πρώτο στάδιο του IKEv2 και χρησιμοποιείται για τη δημιουργία του αρχικού ασφαλούς καναλιού επικοινωνίας (IKE Security Association).

- Ο Initiator αποστέλλει αίτημα (**IKE_SA_INIT Request**) με προτάσεις κρυπτογραφικών αλγορίθμων (encryption, integrity, Diffie-Hellman group).
- Ο Responder απαντά με **IKE_SA_INIT Response**, αποδεχόμενος μία από τις προτεινόμενες παραμέτρους.

Κατά τη φάση αυτή:

- Πραγματοποιείται ανταλλαγή **public keys**

- Ανταλλάσσονται **nonces**
- Δημιουργείται το **IKE SA**

Σημειώνεται ότι σε αυτό το στάδιο δεν πραγματοποιείται ταυτοποίηση των δύο πλευρών.

**2. IKE_AUTH (Initiator Request)**

Στο δεύτερο στάδιο, ο Initiator αποστέλλει μήνυμα **IKE_AUTH Request**, το οποίο στο capture εμφανίζεται τμηματοποιημένο (fragmented) λόγω του μεγάλου μεγέθους των δεδομένων.

Το μήνυμα περιλαμβάνει:

- Ταυτοποιητικά στοιχεία του Initiator (**IDi**)
- Δεδομένα αυθεντικοποίησης (**AUTH**), με χρήση pre-shared key ή ψηφιακού πιστοποιητικού
- Πρόταση για δημιουργία **CHILD SA**, το οποίο θα χρησιμοποιηθεί για τη μεταφορά των δεδομένων

Με το μήνυμα αυτό, ο Initiator αποδεικνύει την ταυτότητά του στον Responder.

**3. IKE_AUTH (Responder Response)**

Ο Responder απαντά με **IKE_AUTH Response**, επίσης τμηματοποιημένο, ολοκληρώνοντας τη διαδικασία αυθεντικοποίησης.

Το μήνυμα περιλαμβάνει:

- Ταυτοποιητικά στοιχεία του Responder (**IDr**)
- Δεδομένα αυθεντικοποίησης (**AUTH**)
- Επιβεβαίωση της δημιουργίας του **CHILD SA**

Με την ολοκλήρωση αυτού του σταδίου:

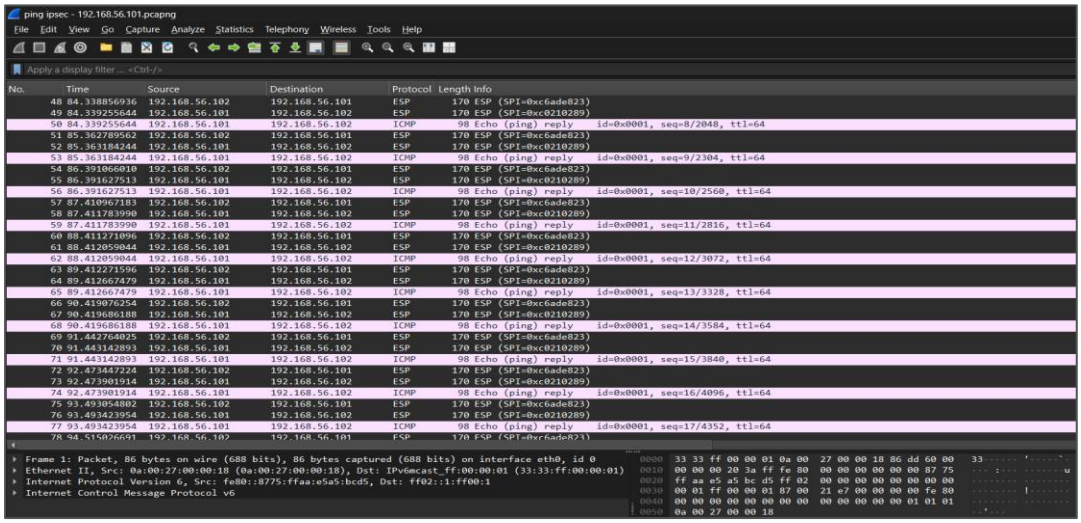
- Έχει δημιουργηθεί επιτυχώς το **IKE SA**
- Έχει δημιουργηθεί το **CHILD SA**
- Το IPsec tunnel είναι ενεργό και έτοιμο για ασφαλή μεταφορά δεδομένων μέσω ESP

Για την επαλήθευση της σύνδεσης πραγματοποιήθηκε αποστολή ICMP πακέτων:

- `ping 192.168.56.101`

Κατά την ανάλυση της κίνησης με το Wireshark παρατηρήθηκαν πακέτα τόσο ESP όσο και ICMP. Αυτό οφείλεται στο γεγονός ότι το sniffing πραγματοποιήθηκε τοπικά στον κόμβο Kali1, ο οποίος αποτελεί άκρο του IPsec tunnel.

Συγκεκριμένα, το ICMP πακέτο (ping) δημιουργείται αρχικά από το λειτουργικό σύστημα και στη συνέχεια ενθυλακώνεται από το IPsec σε πακέτο ESP πριν σταλεί στο δίκτυο. Επειδή το capture έγινε στον ίδιο τον host, το Wireshark είναι σε θέση να καταγράψει, το ICMP πακέτο πριν την ενθυλάκωση, και το αντίστοιχο ESP πακέτο που μεταφέρεται μέσω του IPsec tunnel. Η παρατήρηση αυτή επιβεβαιώνει ότι το IPsec tunnel λειτουργεί σωστά και ότι η κίνηση προστατεύεται μέσω ESP. (Εικόνα 16)

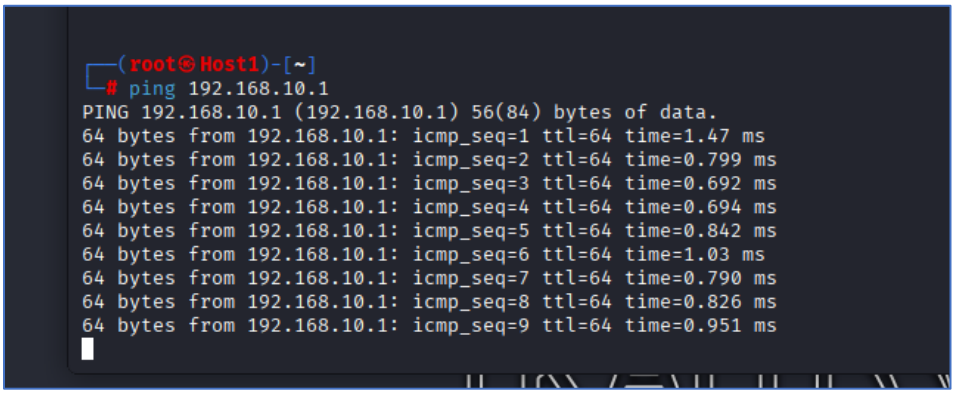



*Εικόνα 16: Αποστολή ESP πακέτων*

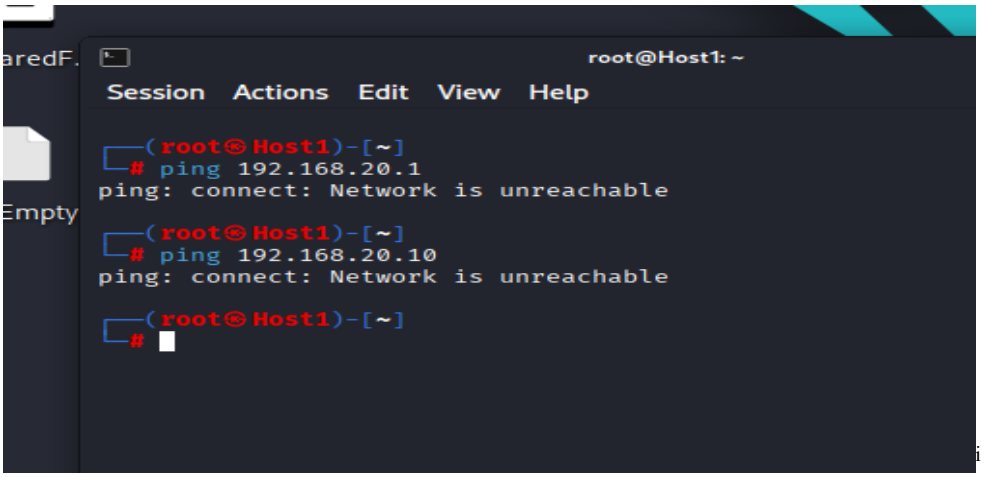
### 3.4 Σύνδεση Site-to-Site με IKEv2

Οι δικτυακές συνδέσεις των VMs ρυθμίστηκαν στο VirtualBox έτσι ώστε να υποστηρίζεται η υλοποίηση IPsec site-to-site. Συγκεκριμένα, στα VMs KaliGateWay1 και KaliGateWay2 ορίστηκαν δύο κάρτες δικτύου, προκειμένου να λειτουργούν ως δρομολογητές μεταξύ των τοπικών δικτύων και του ενδιάμεσου δικτύου.
Το KaliGateWay1 και το KaliHost1 βρίσκονται στο κοινό τοπικό δίκτυο 192.168.10.0/24, με διευθύνσεις IP 192.168.10.1 και 192.168.10.2. Αντίστοιχα, το KaliGateWay2 και το KaliHost2 βρίσκονται στο κοινό τοπικό δίκτυο 192.168.20.0/24, με διευθύνσεις IP 192.168.20.1 και 192.168.20.2.
Παράλληλα, τα KaliGateWay1 και KaliGateWay2 συνδέθηκαν μεταξύ τους μέσω ξεχωριστού κοινού δικτύου 10.0.0.0/24, με διευθύνσεις IP 10.0.0.1 και 10.0.0.2, το οποίο χρησιμοποιείται για την ανταλλαγή της IPsec κίνησης.
Με τη συγκεκριμένη αρχιτεκτονική, το KaliHost1 έχει δικτυακή πρόσβαση μόνο μέσω του KaliGateWay1, το KaliHost2 μόνο μέσω του KaliGateWay2, ενώ η επικοινωνία μεταξύ των δύο απομακρυσμένων δικτύων πραγματοποιείται αποκλειστικά μέσω των gateways και του IPsec tunnel.

Τέλος δοκιμάσαμε με ICMP (ping) να δούμε ότι όντως το Host1 έχει άμεση επικοινωνία μόνο με το GateWay1 και όχι με το Host2.



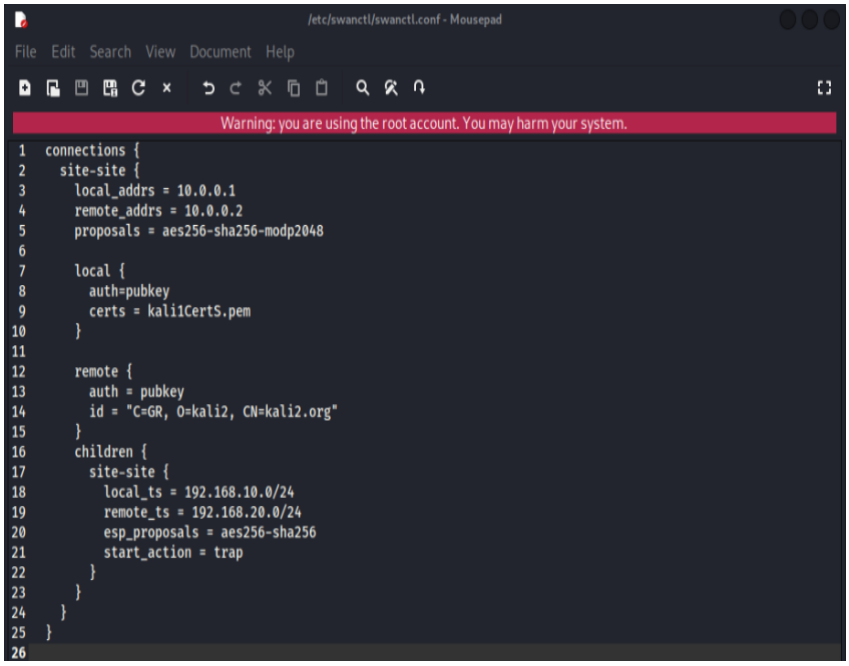

*Εικόνα 17: Επιτυχής αποστολή και λήψη ICMP πακέτων (ping) από τον Host1 προς τον Gateway1.*



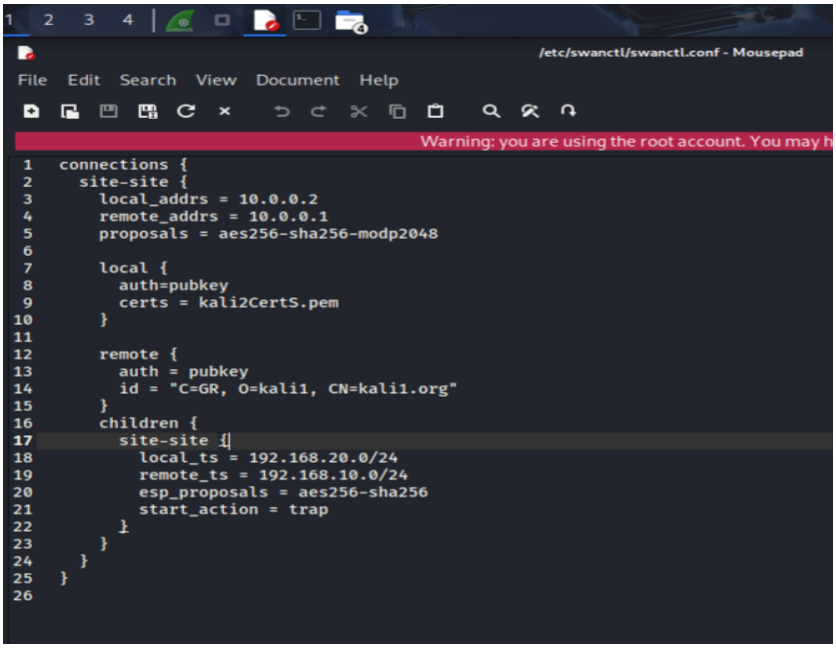

*Εικόνα 18 : Ανεπιτυχής αποστολή και λήψη ICMP πακέτων (ping) από τον Host1 προς τον Gateway1*

### 3.4.1 Διαμόρφωση strongSwan (swanctl)

Στην συνέχεια διαμορφώσαμε τα αρχεία swanctl.conf για τους δύο κόμβους KaliGateWay1 και KaliGateWay2 χρησιμοποιώντας τα πιστοποιητικά που είχαμε δημιουργήσει στο προηγούμενο μέρος της άσκησης.




*Εικόνα 19: swanctl.conf Gateway1*

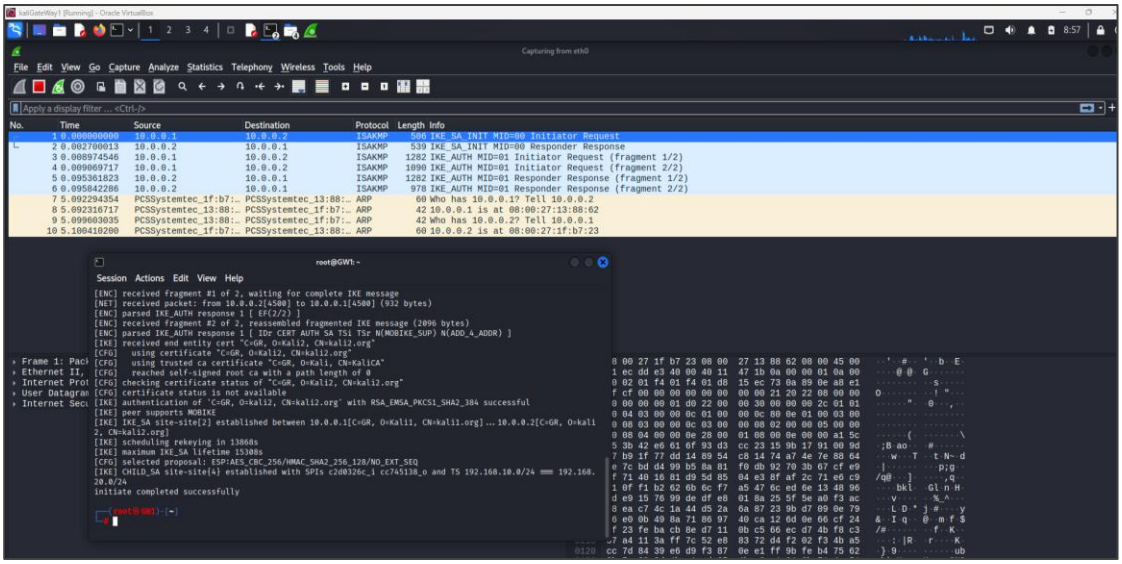



*Εικόνα 20: swanctl.conf Gateway2*

### 3.4.2 Σύνδεση Site-to-Site με IKEv2

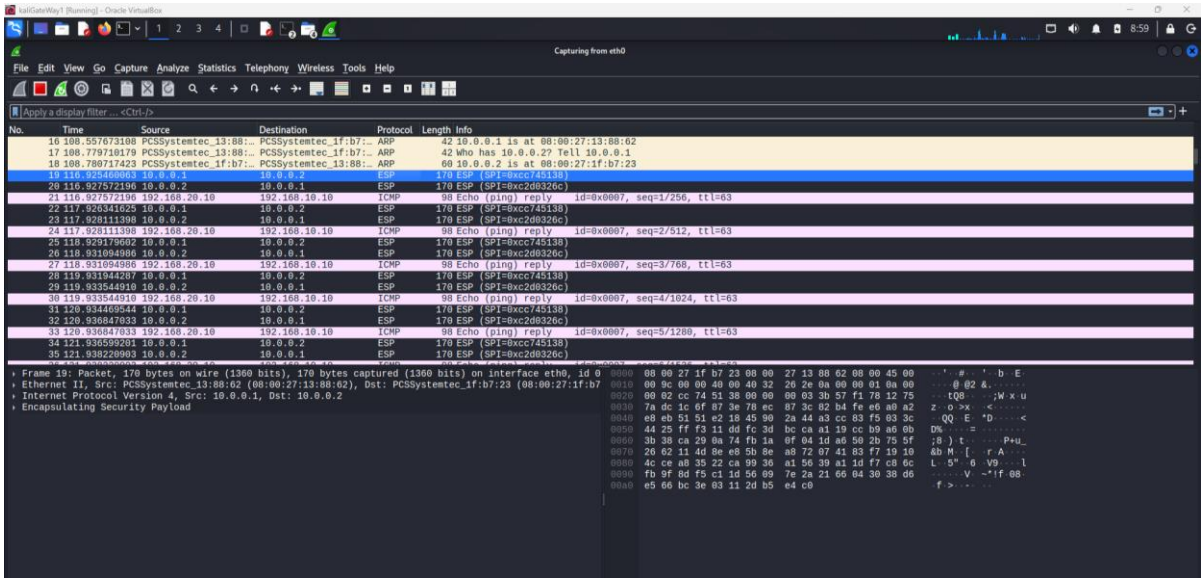
Στην συνέχεια ορίσαμε ως defaultgateway στο Host1 το Gateway1 και στο Host2 το Gateway2 επίσης ενεργοποιήσαμε ipforwarding στα GateWay1 και GateWay2 ώστε να λειτουργούν ως δρομολογητές. Τέλος κάναμε load τα credentials και τα connections και αρχίσαμε την ipsec σύνδεση με τις παρακάτω εντολές.

- **sudo swanctl --load-cred**
- **sudo swanctl --load-conns**
- **sudo swanctl --initiate --child site-site**

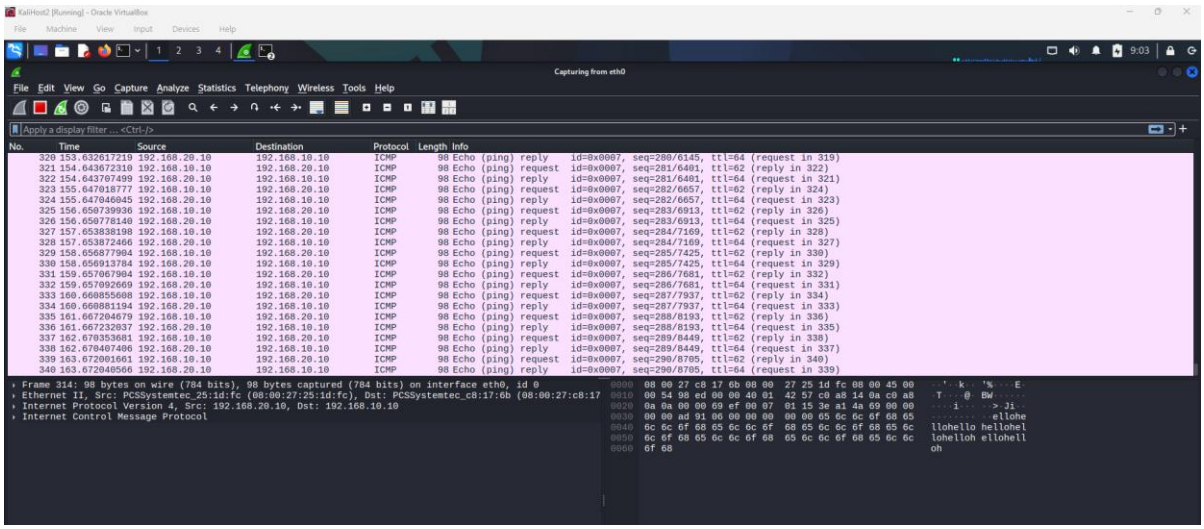


*Εικόνα 21: ipsec site to site.*

Για να επαληθεύσουμε ότι όλα έχουν γίνει σωστά στείλαμε ping με payload από το Host1 στο Host2.



*Εικόνα 22: Το payload από τον GW1 στον GW2 είναι κρυπτογραφημένο.*



*Εικόνα 23: Ο Host2 βλέπει το μύνημα Hello του Host1.*

Χρησιμοποιώντας wireshark αρχικά στο GateWay1 (εικόνα 22) βλέπουμε την ανταλλαγή πακέτων κατά τη διάρκεια ενεργής σύνδεσης IPsec site-to-site μεταξύ των Gateways. Παρατηρούμε ότι τα πακέτα ESP μεταφέρονται μεταξύ των διευθύνσεων 10.0.0.1 και 10.0.0.2, οι οποίες αντιστοιχούν στα δύο Gateways. Τα πακέτα αυτά περιέχουν κρυπτογραφημένη κίνηση που προέρχεται από τα εσωτερικά δίκτυα (192.168.10.0/24 και 192.168.20.0/24). Στην επικεφαλίδα των ESP πακέτων βλέπουμε το Security Parameters Index (SPI), το οποίο χρησιμοποιείται για την ταυτοποίηση του αντίστοιχου IPsec Security Association (SA). Στο συγκεκριμένο capture παρατηρούνται διαφορετικές τιμές SPI (π.χ. 0x4c715138, 0x2cd326c), οι οποίες αντιστοιχούν σε διαφορετικά μονόδρομα SAs για κάθε κατεύθυνση επικοινωνίας μεταξύ των gateways.

Τέλος, παρουσιάζεται η καταγραφή πακέτων στο Wireshark από το Host2, όπου παρατηρούνται τα μηνύματα ICMP σε αποκρυπτογραφημένη μορφή. Η εμφάνιση των πακέτων με διευθύνσεις πηγής και προορισμού των εσωτερικών hosts επιβεβαιώνει ότι η κρυπτογράφηση και αποκρυπτογράφηση της κίνησης πραγματοποιείται στα IPsec gateways, ενώ τα τελικά συστήματα λαμβάνουν την κίνηση σε καθαρή μορφή.(Εικόνα 23)