



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**

**UNIVERSITY OF PIRAEUS**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΜΑΘΗΜΑ: «ΑΣΦΑΛΕΙΑ ΔΙΚΤΥΩΝ (8<sup>ο</sup> ΕΞΑΜΗΝΟ)»**

**ΚΑΘΗΓΗΤΕΣ: ΚΟΤΖΑΝΙΚΟΛΑΟΥ ΠΑΝΑΓΙΩΤΗΣ**

**ΟΜΑΔΑ ΕΡΓΑΣΙΑΣ:**

**ΔΗΜΗΤΡΕΛΛΟΣ ΠΑΝΑΓΙΩΤΗΣ, Π17026**

**ΚΑΡΑΜΠΟΪΚΗΣ ΝΙΚΟΛΑΟΣ, Π17040**

**ΡΟΥΝΤΟΥ ΑΝΝΑ-ΦΑΝΗ, Π17113**

**3η Άσκηση - Υλοποίηση IPsec με το strongswan**

## ΠΕΡΙΕΧΟΜΕΝΑ

1. Δημιουργία και εγκατάσταση κλειδιών.....	3
2. Δημιουργία και δοκιμή συνδέσεων.....	8
A. Σύνδεση host-to-host (κόμβος-με-κόμβο) με IKE2 και με τη χρήση των παραπάνω πιστοποιητικών.....	8
B. Παραλλαγή του προηγούμενου παραδείγματος με χρήση AH και αλγόριθμο hash SHA256 ή μεγαλύτερο.....	12

## 1. Δημιουργία και εγκατάσταση κλειδιών.

### - Δημιουργία δοκιμαστικής Αρχής Πιστοποίησης (Certification Authority).

10.0.2.15 το μηχάνημα που παίζει τον ρόλο της ΑΠ και του αριστερού άκρου:

```
(root@kali1)-[/etc]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe68:d8e4 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:68:d8:e4 txqueuelen 1000 (Ethernet)
    RX packets 21 bytes 7090 (6.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 31 bytes 4993 (4.8 KiB)
    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
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 400 (400.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 400 (400.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Δημιουργία φακέλου για τη ΑΠ.

Δημιουργία κλειδιού της ΑΠ (σε pem format).

```
(root@kali1)-[/etc]
# cd ipsec.d

(root@kali1)-[/etc/ipsec.d]
# mkdir myTestCA

(root@kali1)-[/etc/ipsec.d]
# ls
aacerts  acerts  cacerts  certs  crls  myTestCA  ocspcerts  policies  private  reqs

(root@kali1)-[/etc/ipsec.d]
# cd myTestCA

(root@kali1)-[/etc/ipsec.d/myTestCA]
# ipsec pki --gen --outform pem > myTestCAKey.pem

(root@kali1)-[/etc/ipsec.d/myTestCA]
# ls
myTestCAKey.pem
```

Δημιουργία αυτουπογεγραμμένου πιστοποιητικού για την ΑΠ.

```
(root@kali1)-[/etc/ipsec.d/myTestCA]
# ipsec pki --self --in myTestCAKey.pem --dn "O=cs.unipi, CN=testCA" --ca --outform pem > myTestCACert.pem
```

## - Δημιουργία ζεύγους κλειδιών και πιστοποιητικού για το αριστερό άκρο.

Δημιουργία μυστικού κλειδιού αριστερού άκρου.

```
(root@kali1)-[/etc/ipsec.d/myTestCA]
# ipsec pki --gen --type rsa --size 2048 --outform pem >leftKey.pem
```

Εξαγωγή δημόσιου κλειδιού και Δημιουργία/Υπογραφή πιστοποιητικού για το αριστερό άκρο, από την ΑΠ.

```
(root@kali1)-[/etc/ipsec.d/myTestCA]
# ipsec pki --pub --in leftKey.pem | ipsec pki --issue --cacert myTestCACert.pem --cakey myTestCAKey.pem --dn "O=cs.unipi, CN=left side" --flag ikeIntermediate --flag serverAuth --outform pem > leftCert.pem
```

Έτσι λοιπόν έχουμε:

```
(root@kali1)-[/etc/ipsec.d/myTestCA]
# ls
leftCert.pem leftKey.pem myTestCACert.pem myTestCAKey.pem
```

## - Δημιουργία ζεύγους κλειδιών και πιστοποιητικού για το δεξί άκρο.

Δημιουργία μυστικού κλειδιού δεξιού άκρου.

```
(root@kali1)-[/etc/ipsec.d/myTestCA]
# ipsec pki --gen --type rsa --size 2048 --outform pem >rightKey.pem
```

Εξαγωγή δημόσιου κλειδιού και Δημιουργία/Υπογραφή πιστοποιητικού για το δεξί άκρο, από την ΑΠ.

```
(root@kali1)-[/etc/ipsec.d/myTestCA]
# ipsec pki --pub --in rightKey.pem | ipsec pki --issue --cacert myTestCACert.pem --cakey myTestCAKey.pem --dn "O=cs.unipi, CN=right side" --flag ikeIntermediate --flag serverAuth --outform pem > rightCert.pem
```

Έτσι λοιπόν έχουμε:

```
(root@kali1)-[/etc/ipsec.d/myTestCA]
# ls
leftCert.pem leftKey.pem myTestCACert.pem myTestCAKey.pem rightCert.pem rightKey.pem
```

**- Αντιγραφή κλειδιών και πιστοποιητικών στα δύο μέρη της σύνδεσης.**

Αριστερό άκρο	Δεξί άκρο
Στο φάκελο /etc/ipsec.d/private αντιγράφουμε το ιδιωτικό κλειδί leftKey.pem	Στο φάκελο /etc/ipsec.d/private αντιγράφουμε το ιδιωτικό κλειδί rightKey.pem
Στο φάκελο /etc/ipsec.d/certs αντιγράφουμε το πιστοποιητικό leftCert.pem	Στο φάκελο /etc/ipsec.d/certs αντιγράφουμε το πιστοποιητικό rightCert.pem
Στο φάκελο /etc/ipsec.d/cacerts αντιγράφουμε το πιστοποιητικό της ΑΠ myTestCACert.pem	Στο φάκελο /etc/ipsec.d/cacerts αντιγράφουμε το πιστοποιητικό της ΑΠ myTestCACert.pem

Αριστερό άκρο:

```
(root@kali1)-[/etc/ipsec.d/myTestCA]
# cp leftCert.pem /etc/ipsec.d/certs

(root@kali1)-[/etc/ipsec.d/myTestCA]
# cp leftKey.pem /etc/ipsec.d/private

(root@kali1)-[/etc/ipsec.d/myTestCA]
# cp myTestCACert.pem /etc/ipsec.d/cacerts
```

Δεξί άκρο:

```
(root@kali2)-[/home/panos2]
# cp rightCert.pem /etc/ipsec.d/certs

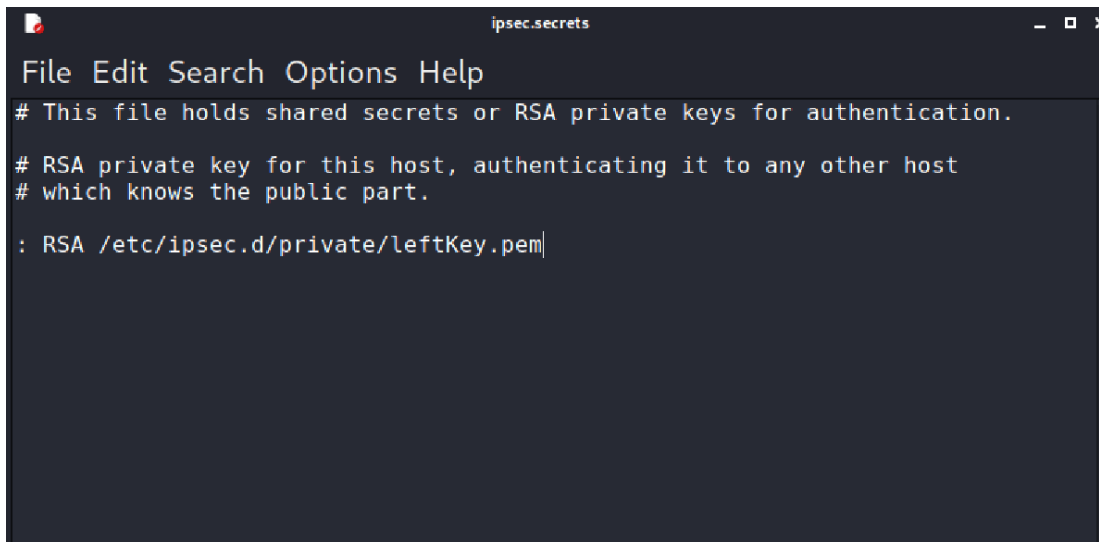
(root@kali2)-[/home/panos2]
# cp rightKey.pem /etc/ipsec.d/private

(root@kali2)-[/home/panos2]
# cp myTestCACert.pem /etc/ipsec.d/cacerts
```

### - Διαμόρφωση αρχείου /etc/ipsec.secrets

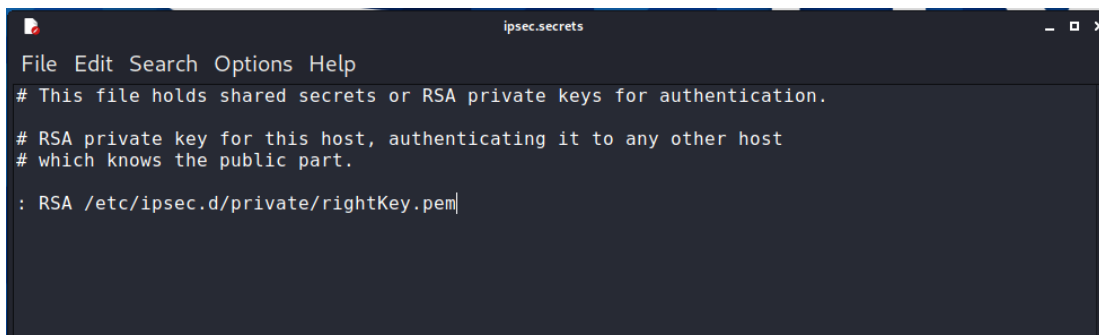
Εκτελείται αντίστοιχα και από τα δύο μέρη της σύνδεσης.

Στο αρχείο /etc/ipsec.secrets προσθέτουμε ένα δείκτη προς το RSA ιδιωτικό κλειδί του αριστερού άκρου:



```
File Edit Search Options Help
# This file holds shared secrets or RSA private keys for authentication.
# RSA private key for this host, authenticating it to any other host
# which knows the public part.
: RSA /etc/ipsec.d/private/leftKey.pem
```

Στο αρχείο /etc/ipsec.secrets προσθέτουμε ένα δείκτη προς το RSA ιδιωτικό κλειδί του δεξιού άκρου:

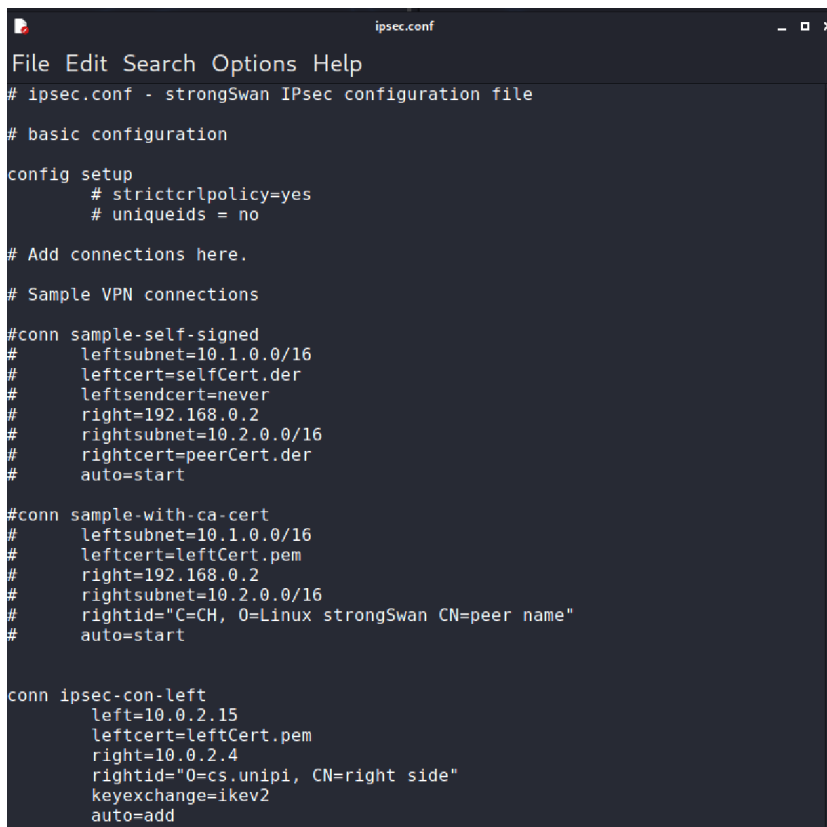


```
File Edit Search Options Help
# This file holds shared secrets or RSA private keys for authentication.
# RSA private key for this host, authenticating it to any other host
# which knows the public part.
: RSA /etc/ipsec.d/private/rightKey.pem
```

## - Διαμόρφωση αρχείου /etc/ipsec.conf.

Εκτελείται αντίστοιχα και από τα δύο μέρη της σύνδεσης.

Από την αριστερή μεριά (10.0.2.15):



```
File Edit Search Options Help
# ipsec.conf - strongSwan IPsec configuration file

# basic configuration

config setup
    # strictcrpolicies=yes
    # uniqueids = no

# Add connections here.

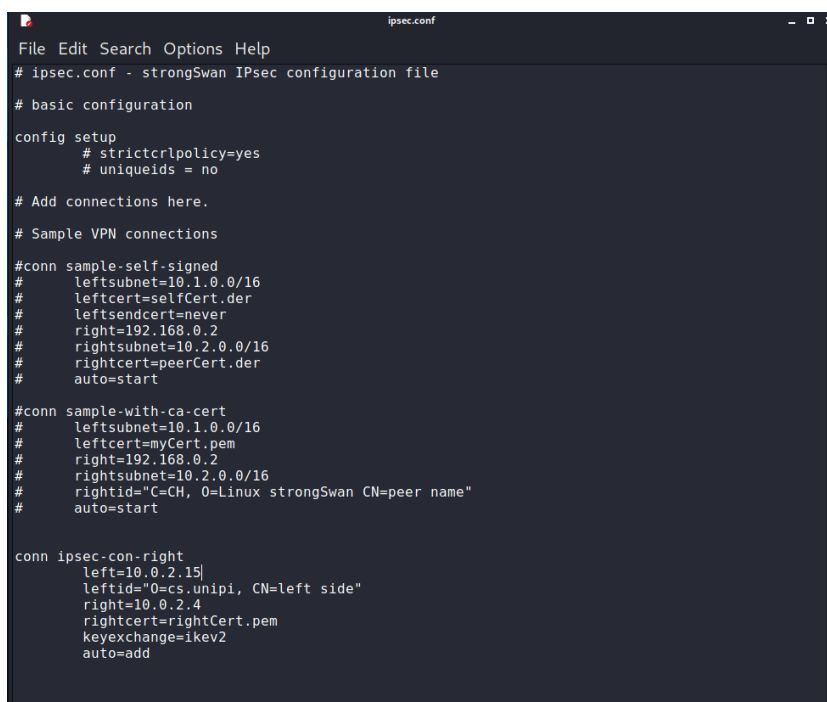
# Sample VPN connections

#conn sample-self-signed
#    leftsubnet=10.1.0.0/16
#    leftcert=selfCert.der
#    leftsendcert=never
#    right=192.168.0.2
#    rightsubnet=10.2.0.0/16
#    rightcert=peerCert.der
#    auto=start

#conn sample-with-ca-cert
#    leftsubnet=10.1.0.0/16
#    leftcert=leftCert.pem
#    right=192.168.0.2
#    rightsubnet=10.2.0.0/16
#    rightid="C=CH, O=Linux strongSwan CN=peer name"
#    auto=start

conn ipsec-con-left
    left=10.0.2.15
    leftcert=leftCert.pem
    right=10.0.2.4
    rightid="O=cs.unipi, CN=right side"
    keyexchange=ikev2
    auto=add
```

Από την δεξιά μεριά (10.0.2.4):



```
File Edit Search Options Help
# ipsec.conf - strongSwan IPsec configuration file

# basic configuration

config setup
    # strictcrpolicies=yes
    # uniqueids = no

# Add connections here.

# Sample VPN connections

#conn sample-self-signed
#    leftsubnet=10.1.0.0/16
#    leftcert=selfCert.der
#    leftsendcert=never
#    right=192.168.0.2
#    rightsubnet=10.2.0.0/16
#    rightcert=peerCert.der
#    auto=start

#conn sample-with-ca-cert
#    leftsubnet=10.1.0.0/16
#    leftcert=myCert.pem
#    right=192.168.0.2
#    rightsubnet=10.2.0.0/16
#    rightid="C=CH, O=Linux strongSwan CN=peer name"
#    auto=start

conn ipsec-con-right
    left=10.0.2.15
    leftid="O=cs.unipi, CN=left side"
    right=10.0.2.4
    rightcert=rightCert.pem
    keyexchange=ikev2
    auto=add
```

## 2. Δημιουργία και δοκιμή συνδέσεων.

A. Σύνδεση host-to-host (κόμβος-με-κόμβο) με IKE2 και με τη χρήση των παραπάνω πιστοποιητικών.

### - Εκκίνηση σύνδεσης

Εκτελείται αντίστοιχα και από τα δύο μέρη της σύνδεσης.

Αριστερό άκρο (10.0.2.15):

Αφού κάνουμε ipsec restart..

```
(root@kali) [/etc/ipsec.d]
# ipsec up ipsec-con-left
Initiating IKE_SA ipsec-con-left[1] to 10.0.2.4
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) N(HASH_ALG) N(REDIR_SUP) ]
sending packet: from 10.0.2.15[500] to 10.0.2.4[500] (710 bytes)
received packet: from 10.0.2.4[500] to 10.0.2.15[500] (623 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ N(FRAG_SUP) N(HASH_ALG) N(CHDLESS_SUP) N(MULT_AUTH) ]
selected proposal: IKE:AES_CBC_128/HMAC_SHA2_256_128/PRF_AES128_XCBC/MODP_3072
received cert request for "O=cs.unipi, CN=testCA"
sending cert request for "O=cs.unipi, CN=testCA"
authentication of "O=cs.unipi, CN=left side" (myself) with RSA_EMSA_PKCS1_SHA2_256 successful
sending end entity cert "O=cs.unipi, CN=left side"
establishing CHILD_SA ipsec-con-left[1]
generating IKE_AUTH request 1 [ IDi CERT N(INIT_CONTACT) CERTREQ IDr AUTH SA TSi TSr N(MOBIKE_SUP) N(NO_ADD_ADDR) N(MULT_AUTH) N(EAP_ONLY) N(MSG_ID_SYN_SUP) ]
splitting IKE message (1520 bytes) into 2 fragments
generating IKE_AUTH request 1 [ EF(1/2) ]
generating IKE_AUTH request 1 [ EF(2/2) ]
sending packet: from 10.0.2.15[4500] to 10.0.2.4[4500] (1236 bytes)
sending packet: from 10.0.2.15[4500] to 10.0.2.4[4500] (356 bytes)
received packet: from 10.0.2.4[4500] to 10.0.2.15[4500] (1236 bytes)
parsed IKE_AUTH response 1 [ EF(1/2) ]
received fragment #1 of 2, waiting for complete IKE message
received packet: from 10.0.2.4[4500] to 10.0.2.15[4500] (148 bytes)
parsed IKE_AUTH response 1 [ EF(2/2) ]
received fragment #2 of 2, reassembled fragmented IKE message (1312 bytes)
parsed IKE_AUTH response 1 [ IDr CERT AUTH SA TSi TSr N(AUTH_LFT) N(MOBIKE_SUP) N(NO_ADD_ADDR) ]
received end entity cert "O=cs.unipi, CN=right side"
using certificate "O=cs.unipi, CN=right side"
using trusted ca certificate "O=cs.unipi, CN=testCA"
checking certificate status of "O=cs.unipi, CN=right side"
certificate status is not available
reached self-signed root ca with a path length of 0
authentication of "O=cs.unipi, CN=right side" with RSA_EMSA_PKCS1_SHA2_256 successful
IKE_SA ipsec-con-left[1] established between 10.0.2.15[O=cs.unipi, CN=left side] ... 10.0.2.4[O=cs.unipi, CN=right side]
scheduling reauthentication in 18094s
maximum IKE_SA lifetime 10634s
selected proposal: ESP:AES_CBC_128/HMAC_SHA2_256_128/NO_EXT_SEQ
CHILD_SA ipsec-con-left[1] established with SPIs cfdde7b3_i c2468853_o and TS 10.0.2.15/32 == 10.0.2.4/32
received AUTH_LIFETIME of 9965s, scheduling reauthentication in 9425s
connection "ipsec-con-left" established successfully
```

```
(root@kali) [/etc/ipsec.d]
# ipsec status
Security Associations (1 up, 0 connecting):
ipsec-con-left[1]: ESTABLISHED 3 minutes ago, 10.0.2.15[O=cs.unipi, CN=left side] ... 10.0.2.4[O=cs.unipi, CN=right side]
ipsec-con-left{1}: INSTALLED, TUNNEL, reqid 1, ESP SPIs: cfdde7b3_i c2468853_o
ipsec-con-left{1}: 10.0.2.15/32 == 10.0.2.4/32
```

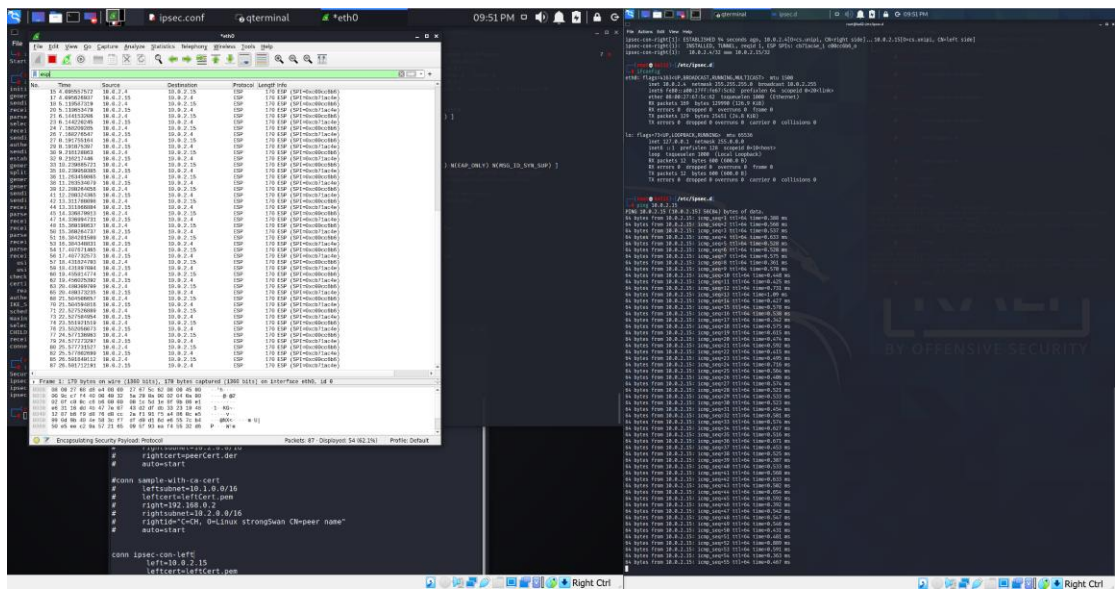


Δεξί άκρο (10.0.2.4):

Αφού κάνουμε ipsec restart..

```
root@kali:~# /etc/ipsec.d
# ipsec up ipsec-con-right
initiating IKE_SA ipsec-con-right[1] to 10.0.2.15
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) N(HASH_ALG) N(REDIR_SUP) ]
sending packet: from 10.0.2.4[500] to 10.0.2.15[500] (710 bytes)
received packet: from 10.0.2.15[500] to 10.0.2.4[500] (623 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ N(FRAG_SUP) N(HASH_ALG) N(CHDLESS_SUP) N(MULT_AUTH) ]
selected proposal: IKE:AES_CBC_128/HMAC_SHA2_256_128/PRF_AES128_XCBC/MOOP_3072
received cert request for "O=cs.unipl, CN=testCA"
sending cert request for "O=cs.unipl, CN=testCA"
authentication of "O=cs.unipl, CN=right side" (myself) with RSA_EMSA_PKCS1_SHA2_256 successful
sending end entity cert "O=cs.unipl, CN=right side"
establishing CHILD_SA ipsec-con-right[1]
generating IKE_AUTH request 1 [ IDi CERT N(INIT_CONTACT) CERTREQ IDr AUTH SA TSi TSr N(MOBIKE_SUP) N(NO_ADD_ADDR) N(MULT_AUTH) N(EAP_ONLY) N(MSG_ID_SYN_SUP) ]
splitting IKE message (1520 bytes) into 2 fragments
generating IKE_AUTH request 1 [ EF(1/2) ]
generating IKE_AUTH request 1 [ EF(2/2) ]
sending packet: from 10.0.2.4[4500] to 10.0.2.15[4500] (1236 bytes)
sending packet: from 10.0.2.4[4500] to 10.0.2.15[4500] (356 bytes)
received packet: from 10.0.2.15[4500] to 10.0.2.4[4500] (1236 bytes)
parsed IKE_AUTH response 1 [ EF(1/2) ]
received fragment #1 of 2, waiting for complete IKE message
received packet: from 10.0.2.15[4500] to 10.0.2.4[4500] (148 bytes)
parsed IKE_AUTH response 1 [ EF(2/2) ]
received fragment #2 of 2, reassembled fragmented IKE message (1312 bytes)
parsed IKE_AUTH response 1 [ IDr CERT AUTH SA TSi TSr N(AUTH_LFT) N(MOBIKE_SUP) N(NO_ADD_ADDR) ]
received end entity cert "O=cs.unipl, CN=left side"
using certificate "O=cs.unipl, CN=left side"
using trusted ca certificate "O=cs.unipl, CN=testCA"
checking certificate status of "O=cs.unipl, CN=left side"
certificate status is not available
reached self-signed root ca with a path length of 0
authentication of "O=cs.unipl, CN=left side" with RSA_EMSA_PKCS1_SHA2_256 successful
IKE_SA ipsec-con-right[1] established between 10.0.2.4[O=cs.unipl, CN=right side]... 10.0.2.15[O=cs.unipl, CN=left side]
scheduling reauthentication in 10031s
maximum IKE_SA lifetime 10571s
selected proposal: ESP:AES_CBC_128/HMAC_SHA2_256_128/NO_EXT_SEQ
CHILD_SA ipsec-con-right[1] established with SPIs cb71ac4e_1_c00cc6b6_o and TS 10.0.2.4/32 == 10.0.2.15/32
received AUTH_LIFETIME of 10074s, scheduling reauthentication in 9534s
peer supports MOBIKE
connection 'ipsec-con-right' established successfully
```

Έτσι λοιπόν για να ελέγξουμε την σύνδεση κάνουμε ping το δεξιο μηχάνημα στο αριστερό και παρατηρούμε την κίνηση:



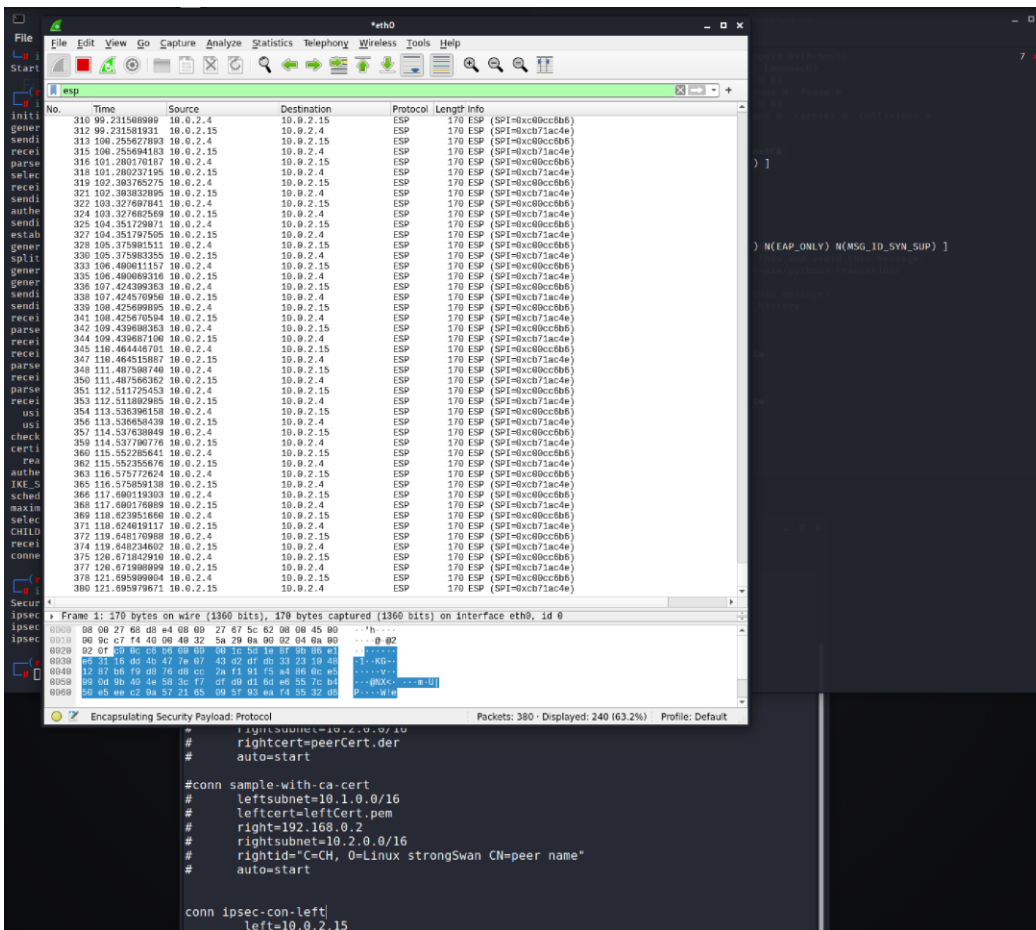
Πιο συγκεκριμένα εκτελούμε το ping από το δεξιό άκρο όπως φαίνεται παρακάτω:

```
root@kali2:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.4 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe07:5c62 prefixlen 64 scopeid 0<20<link>
    ether 08:00:27:07:5c:62 txqueuelen 1000 (Ethernet)
    RX packets 169 bytes 329996 (320.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 129 bytes 25451 (24.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

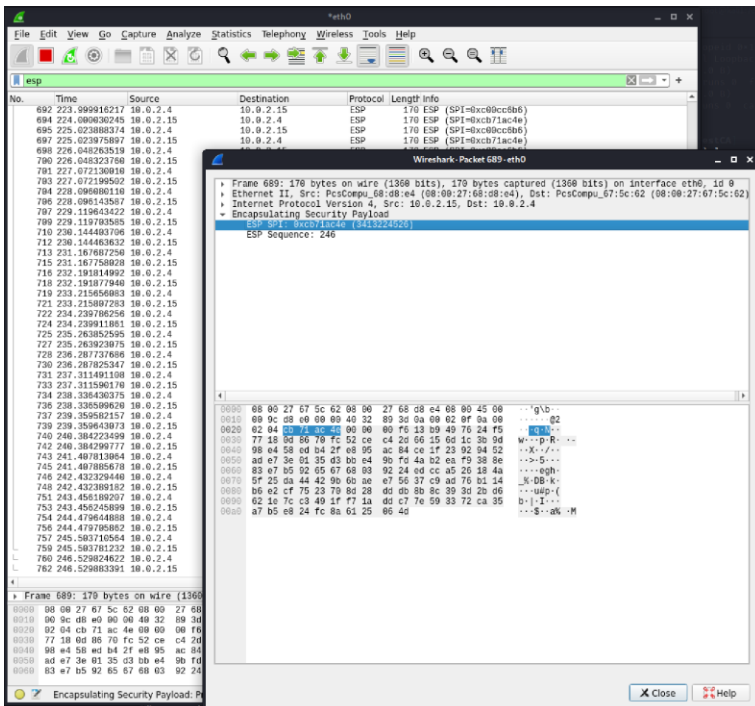
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 12 bytes 600 (600.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 600 (600.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali2:~# ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data:
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.388 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.368 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.537 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.633 ms
64 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=0.528 ms
64 bytes from 10.0.2.15: icmp_seq=6 ttl=64 time=0.528 ms
64 bytes from 10.0.2.15: icmp_seq=7 ttl=64 time=0.575 ms
64 bytes from 10.0.2.15: icmp_seq=8 ttl=64 time=0.361 ms
64 bytes from 10.0.2.15: icmp_seq=9 ttl=64 time=0.578 ms
64 bytes from 10.0.2.15: icmp_seq=10 ttl=64 time=0.448 ms
64 bytes from 10.0.2.15: icmp_seq=11 ttl=64 time=0.425 ms
64 bytes from 10.0.2.15: icmp_seq=12 ttl=64 time=0.731 ms
64 bytes from 10.0.2.15: icmp_seq=13 ttl=64 time=1.09 ms
64 bytes from 10.0.2.15: icmp_seq=14 ttl=64 time=0.427 ms
64 bytes from 10.0.2.15: icmp_seq=15 ttl=64 time=0.578 ms
64 bytes from 10.0.2.15: icmp_seq=16 ttl=64 time=0.538 ms
64 bytes from 10.0.2.15: icmp_seq=17 ttl=64 time=0.342 ms
64 bytes from 10.0.2.15: icmp_seq=18 ttl=64 time=0.574 ms
64 bytes from 10.0.2.15: icmp_seq=19 ttl=64 time=0.615 ms
64 bytes from 10.0.2.15: icmp_seq=20 ttl=64 time=0.474 ms
64 bytes from 10.0.2.15: icmp_seq=21 ttl=64 time=0.592 ms
64 bytes from 10.0.2.15: icmp_seq=22 ttl=64 time=0.415 ms
64 bytes from 10.0.2.15: icmp_seq=23 ttl=64 time=0.495 ms
64 bytes from 10.0.2.15: icmp_seq=24 ttl=64 time=0.716 ms
64 bytes from 10.0.2.15: icmp_seq=25 ttl=64 time=0.564 ms
64 bytes from 10.0.2.15: icmp_seq=26 ttl=64 time=0.486 ms
64 bytes from 10.0.2.15: icmp_seq=27 ttl=64 time=0.574 ms
64 bytes from 10.0.2.15: icmp_seq=28 ttl=64 time=0.523 ms
64 bytes from 10.0.2.15: icmp_seq=29 ttl=64 time=0.532 ms
64 bytes from 10.0.2.15: icmp_seq=30 ttl=64 time=0.523 ms
64 bytes from 10.0.2.15: icmp_seq=31 ttl=64 time=0.454 ms
64 bytes from 10.0.2.15: icmp_seq=32 ttl=64 time=0.581 ms
64 bytes from 10.0.2.15: icmp_seq=33 ttl=64 time=0.574 ms
64 bytes from 10.0.2.15: icmp_seq=34 ttl=64 time=0.627 ms
64 bytes from 10.0.2.15: icmp_seq=35 ttl=64 time=0.516 ms
64 bytes from 10.0.2.15: icmp_seq=36 ttl=64 time=0.671 ms
64 bytes from 10.0.2.15: icmp_seq=37 ttl=64 time=0.453 ms
64 bytes from 10.0.2.15: icmp_seq=38 ttl=64 time=0.525 ms
64 bytes from 10.0.2.15: icmp_seq=39 ttl=64 time=0.387 ms
64 bytes from 10.0.2.15: icmp_seq=40 ttl=64 time=0.568 ms
64 bytes from 10.0.2.15: icmp_seq=41 ttl=64 time=0.654 ms
64 bytes from 10.0.2.15: icmp_seq=42 ttl=64 time=0.633 ms
64 bytes from 10.0.2.15: icmp_seq=43 ttl=64 time=0.582 ms
64 bytes from 10.0.2.15: icmp_seq=44 ttl=64 time=0.654 ms
64 bytes from 10.0.2.15: icmp_seq=45 ttl=64 time=0.592 ms
64 bytes from 10.0.2.15: icmp_seq=46 ttl=64 time=0.392 ms
64 bytes from 10.0.2.15: icmp_seq=47 ttl=64 time=0.542 ms
64 bytes from 10.0.2.15: icmp_seq=48 ttl=64 time=0.547 ms
64 bytes from 10.0.2.15: icmp_seq=49 ttl=64 time=0.546 ms
64 bytes from 10.0.2.15: icmp_seq=50 ttl=64 time=0.431 ms
64 bytes from 10.0.2.15: icmp_seq=51 ttl=64 time=0.481 ms
64 bytes from 10.0.2.15: icmp_seq=52 ttl=64 time=0.889 ms
64 bytes from 10.0.2.15: icmp_seq=53 ttl=64 time=0.591 ms
64 bytes from 10.0.2.15: icmp_seq=54 ttl=64 time=0.363 ms
64 bytes from 10.0.2.15: icmp_seq=55 ttl=64 time=0.467 ms
64 bytes from 10.0.2.15: icmp_seq=56 ttl=64 time=0.337 ms
64 bytes from 10.0.2.15: icmp_seq=57 ttl=64 time=0.529 ms
64 bytes from 10.0.2.15: icmp_seq=58 ttl=64 time=0.551 ms
64 bytes from 10.0.2.15: icmp_seq=59 ttl=64 time=0.568 ms
```

Στην συνέχεια ανοίγουμε στο αριστερό άκρο το Wireshark και με την βοήθεια του φίλτρου «esp» μας εμφανίζονται όλα τα πακέτα με το πρωτόκολλο αυτό. Βλέπουμε λοιπόν ότι τα δύο άκρα μιλάνε μεταξύ τους χρησιμοποιώντας IPSEC.



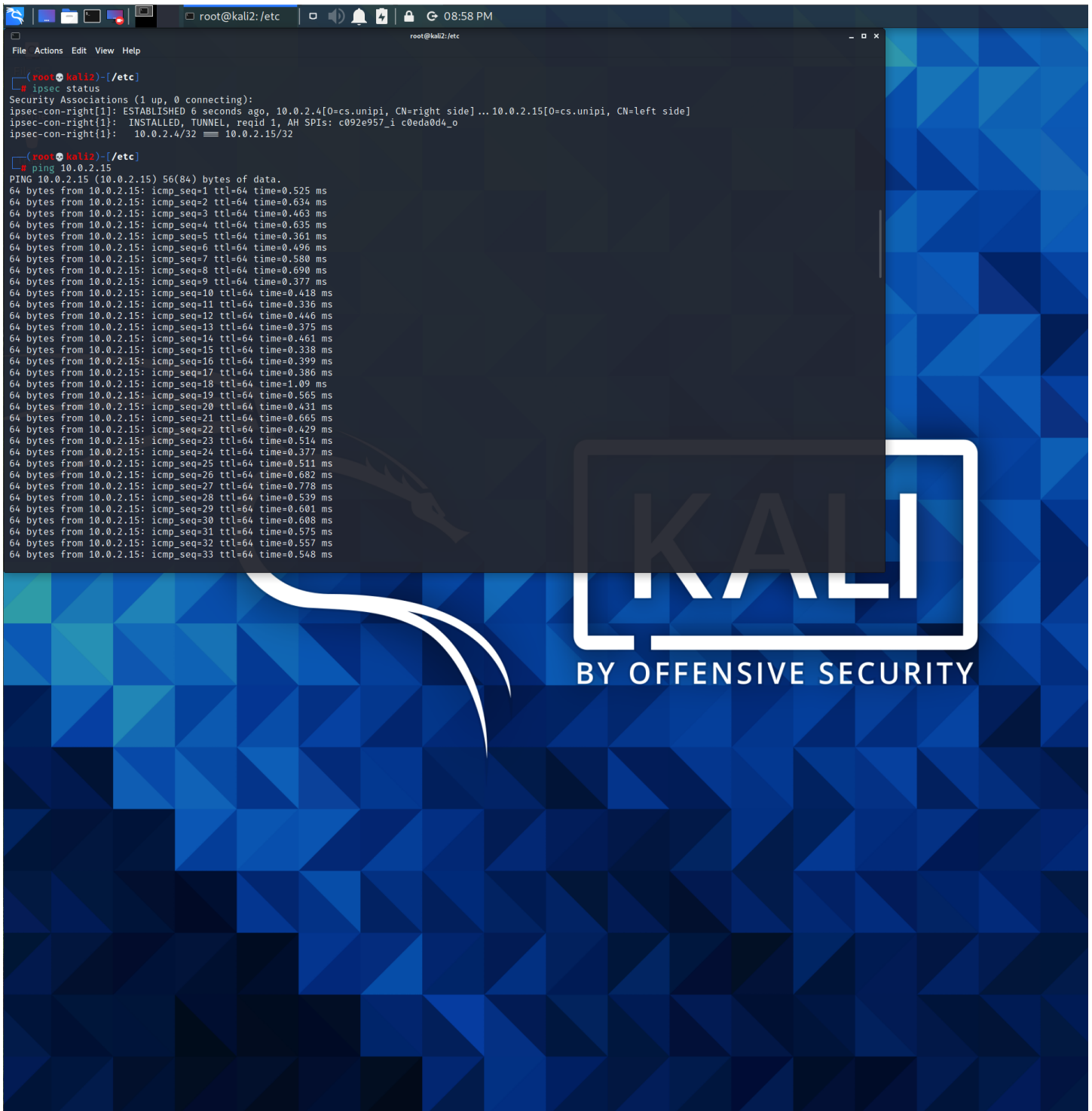
Καθώς επίσης βλέπουμε και το SPI επιλέγοντας ένα πακέτο:



## B. Παραλλαγή του προηγούμενου παραδείγματος με χρήση AH και αλγόριθμο hash SHA256 ή μεγαλύτερο.

Για να χρησιμοποιήσουμε Authentication Header πρέπει να πάμε στο αρχείο `ipsec.conf` που φτιάξαμε παραπάνω και να γράψουμε «`ah=sha256`» μιας και αυτός είναι ο αλγόριθμος που θέλουμε να χρησιμοποιήσουμε. Αξίζει να σημειωθεί ότι default στο `ipsec.conf` είναι η επικεφαλίδα ESP. Αυτό το κάνουμε και για τα δύο άκρα του δικτύου.

Κάνουμε ping το αριστερό μηχάνημα(10.0.2.15) από το δεξί (10.0.2.4).



```
root@kali2:/etc
File Actions Edit View Help

root@kali2:~# ipsec status
Security Associations (1 up, 0 connecting):
ipsec-con-right[1]: ESTABLISHED 6 seconds ago, 10.0.2.4[0=cs.unipi, CN=right side] ... 10.0.2.15[0=cs.unipi, CN=left side]
ipsec-con-right[1]:  INSTALLED, TUNNEL, reqid 1, AH SPIs: c092e957_i c0eda0d4_o
ipsec-con-right[1]:  10.0.2.4/32 == 10.0.2.15/32

root@kali2:~# ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data:
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.525 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.634 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.463 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.635 ms
64 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=0.361 ms
64 bytes from 10.0.2.15: icmp_seq=6 ttl=64 time=0.496 ms
64 bytes from 10.0.2.15: icmp_seq=7 ttl=64 time=0.580 ms
64 bytes from 10.0.2.15: icmp_seq=8 ttl=64 time=0.690 ms
64 bytes from 10.0.2.15: icmp_seq=9 ttl=64 time=0.377 ms
64 bytes from 10.0.2.15: icmp_seq=10 ttl=64 time=0.418 ms
64 bytes from 10.0.2.15: icmp_seq=11 ttl=64 time=0.336 ms
64 bytes from 10.0.2.15: icmp_seq=12 ttl=64 time=0.446 ms
64 bytes from 10.0.2.15: icmp_seq=13 ttl=64 time=0.375 ms
64 bytes from 10.0.2.15: icmp_seq=14 ttl=64 time=0.461 ms
64 bytes from 10.0.2.15: icmp_seq=15 ttl=64 time=0.338 ms
64 bytes from 10.0.2.15: icmp_seq=16 ttl=64 time=0.399 ms
64 bytes from 10.0.2.15: icmp_seq=17 ttl=64 time=0.386 ms
64 bytes from 10.0.2.15: icmp_seq=18 ttl=64 time=1.09 ms
64 bytes from 10.0.2.15: icmp_seq=19 ttl=64 time=0.565 ms
64 bytes from 10.0.2.15: icmp_seq=20 ttl=64 time=0.431 ms
64 bytes from 10.0.2.15: icmp_seq=21 ttl=64 time=0.665 ms
64 bytes from 10.0.2.15: icmp_seq=22 ttl=64 time=0.429 ms
64 bytes from 10.0.2.15: icmp_seq=23 ttl=64 time=0.514 ms
64 bytes from 10.0.2.15: icmp_seq=24 ttl=64 time=0.377 ms
64 bytes from 10.0.2.15: icmp_seq=25 ttl=64 time=0.511 ms
64 bytes from 10.0.2.15: icmp_seq=26 ttl=64 time=0.682 ms
64 bytes from 10.0.2.15: icmp_seq=27 ttl=64 time=0.778 ms
64 bytes from 10.0.2.15: icmp_seq=28 ttl=64 time=0.539 ms
64 bytes from 10.0.2.15: icmp_seq=29 ttl=64 time=0.601 ms
64 bytes from 10.0.2.15: icmp_seq=30 ttl=64 time=0.608 ms
64 bytes from 10.0.2.15: icmp_seq=31 ttl=64 time=0.575 ms
64 bytes from 10.0.2.15: icmp_seq=32 ttl=64 time=0.557 ms
64 bytes from 10.0.2.15: icmp_seq=33 ttl=64 time=0.548 ms
```



Στην συνέχεια πηγαίνουμε στο αριστερό άκρο και ανοίγουμε το Wireshark για να δούμε τα πακέτα που ανταλλάσσονται. Με την βοήθεια του φίλτρου «ah» βλέπουμε όλα τα πακέτα με AH καθώς και το SPI τους όπως φαίνεται στην παρακάτω φωτογραφία.

The screenshot shows the Wireshark network protocol analyzer interface. The main window displays a list of captured packets, filtered by 'ah'. The packet list shows various ICMP Echo (ping) requests and replies between 10.0.2.4 and 10.0.2.15. The selected packet (No. 794) is an ICMP Echo (ping) request. The detailed view pane on the right shows the structure of the selected packet, which is an Authentication Header (AH) packet. The AH packet contains an Internet Protocol (IP) header, an Internet Control Message Protocol (ICMP) header, and an Internet Protocol (IP) payload. The AH packet is 146 bytes long. The AH header fields are: Next header: IPIP (4), Length: 5 (28 bytes), Reserved: 0000, AH SPI: 0xc0eda0d4, AH Sequence: 21, and AH ICV: 4c188d025741751dd02e4998f8b78e28. The packet is captured on interface eth0, id 0. The packet is an Ethernet II, Src: PcsCompu\_67:5c:62 (08:00:27:67:5c:62), Dst: PcsCompu\_68:d8:e4 (08:00:27:68:d8:e4). The packet is an Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.15. The packet is an Internet Control Message Protocol.

Wireshark - Packet 1 - eth0

Frame 1: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits) on interface eth0, id 0

Ethernet II, Src: PcsCompu\_67:5c:62 (08:00:27:67:5c:62), Dst: PcsCompu\_68:d8:e4 (08:00:27:68:d8:e4)

Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.15

Authentication Header

Next header: IPIP (4)

Length: 5 (28 bytes)

Reserved: 0000

AH SPI: 0xc0eda0d4

AH Sequence: 21

AH ICV: 4c188d025741751dd02e4998f8b78e28

Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.15

Internet Control Message Protocol

0000 08 00 27 68 d8 e4 08 00 27 67 5c 62 08 00 45 00 ...h...

0010 00 84 3c 79 40 00 40 33 e5 bb 0a 00 02 04 0a 00 ...y0003

0020 02 0f 04 05 00 00 c0 ed a0 d4 00 00 00 15 4c 18 ...

0030 8d 02 57 41 75 1d d0 2e 49 98 f8 b7 8e 28 45 00 ...wAu...

0040 00 54 96 42 40 00 40 01 8c 54 0a 00 02 04 0a 00 ...T-B00@

0050 02 0f 08 00 17 c2 bb 43 00 15 97 43 6f 60 00 00 ...C

0060 00 00 56 6e 09 00 00 00 00 00 10 11 12 13 14 15 ...Vm...

0070 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ...! "\$

0080 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 ...&'()\*+,-./

0090 36 37 67