



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ  
ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

## ΜΑΘΗΜΑ:

Ασφάλεια Δικτύων Υπολογιστών & Τεχνολογίες Προστασίας της  
Ιδιωτικότητας

### 1<sup>η</sup> Ομαδική Εργασία

#### Θέμα

*Δημιουργία ασφαλούς διαύλου διαχείρισης botnet (C&C) με χρήση  
IPsec & TLS/SSL πρωτοκόλλων.*

Πέππας Κωνσταντίνος 321/2011134  
Σωτηρέλης Χρήστος 321/2012182  
Χαϊκάλης Νικόλαος 321/2012200  
08/11/2015

## Περιεχόμενα

1. Πηγαίος κώδικας προγραμμάτων.
2. Τεκμηρίωση προγραμμάτων και επεξήγηση τυχόν δικών σας παραδοχών.
3. Εκτελέσιμα προγράμματα (project Netbeans κτλ)
4. Στιγμιότυπα εκτέλεσης προγράμματος (screenshots)
5. Περιγραφή και τρόπος δημιουργίας πιστοποιητικών
6. Ψηφιακά Πιστοποιητικά
7. Ανάλυση των αποτελεσμάτων χρήσης του εργαλείου σύλληψης πακέτων (sniffer) και ενδεικτικά στιγμιότυπα εκτέλεσης (screenshots). Επεξήγηση των φίλτρων που χρησιμοποιήθηκαν.
8. Περιγραφή των τροποποιήσεων του συστήματος για την χρήση του πρωτοκόλλου IPsec
9. Σύγκριση των τεχνολογιών TLS/SSL και IPv6/IPsec
10. Βιβλιογραφία

## 1) Πηγαίος κώδικας προγραμμάτων

Ο κώδικας των projects εμπεριέχεται στο φάκελο με όνομα «java projects».

## 2) Τεκμηρίωση προγραμμάτων και επεξήγηση τυχόν δικών σας παραδοχών.

Μέσα στις classes που έχουμε δημιουργήσει υπάρχουν αναλυτικά σχόλια για την επεξήγηση του κώδικα καθώς και για τις παραδοχές που έχουμε κάνει. Παράλληλα να εξηγήσουμε πως για το Project του C&C Server δημιουργήσαμε γραφικό περιβάλλον για την δική μας ευκολία. Το γραφικό αυτό δεδομένου πως δεν ζητήθηκε από την εκφώνηση το κάναμε χρησιμοποιώντας έτοιμο γραφικό περιβάλλον από το netbeans. Στο project του BotMaster τα γραφικά έγιναν από εμάς όπως ζητήθηκε. Επίσης για την υλοποίηση της εργασίας χρησιμοποιήσαμε δύο υπολογιστές (1xDesktop 1xLaptop) με λειτουργικό σύστημα Windows 8.1 Pro x64 Bit και ένα virtual machine Windows 8.1 Pro x32 bit εγκατεστημένο στο Desktop.

IP Συστημάτων:

Desktop -> 192.168.1.69

Laptop -> 192.168.1.65

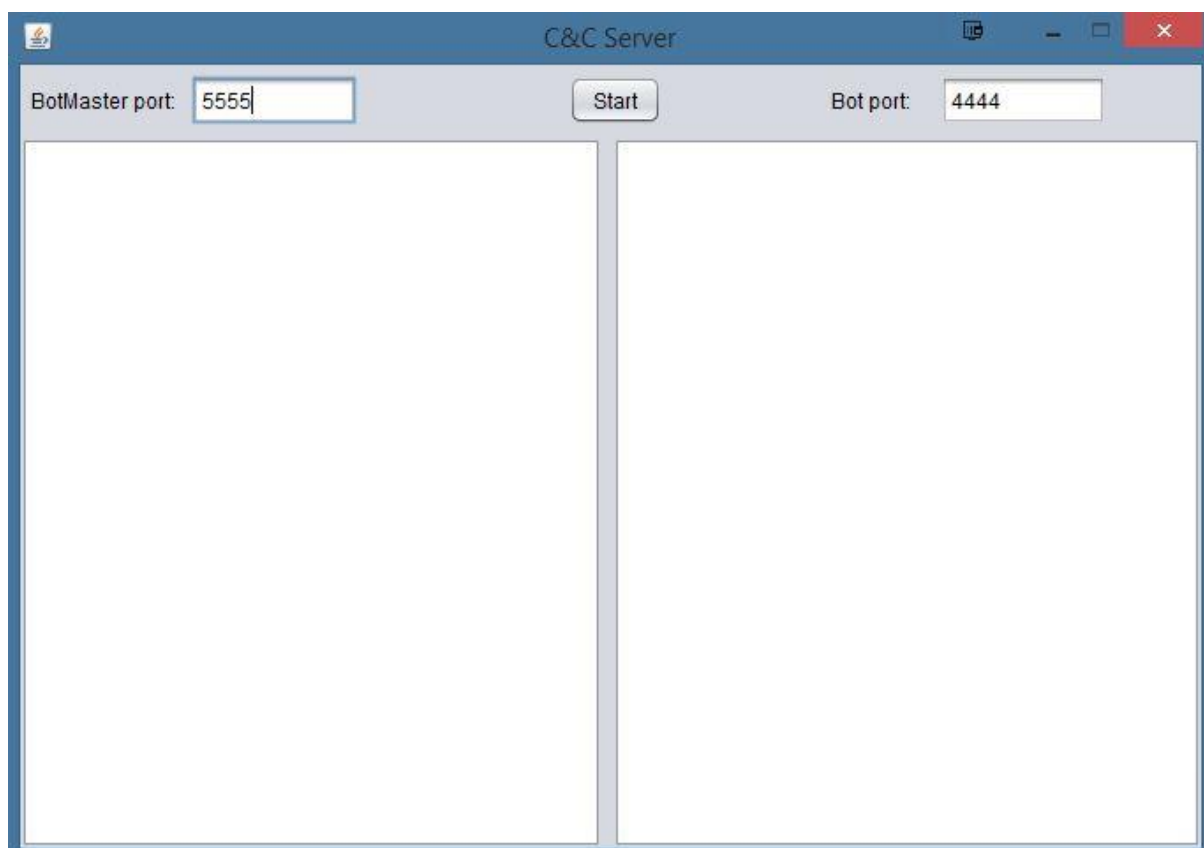
VM -> 10.0.2.15

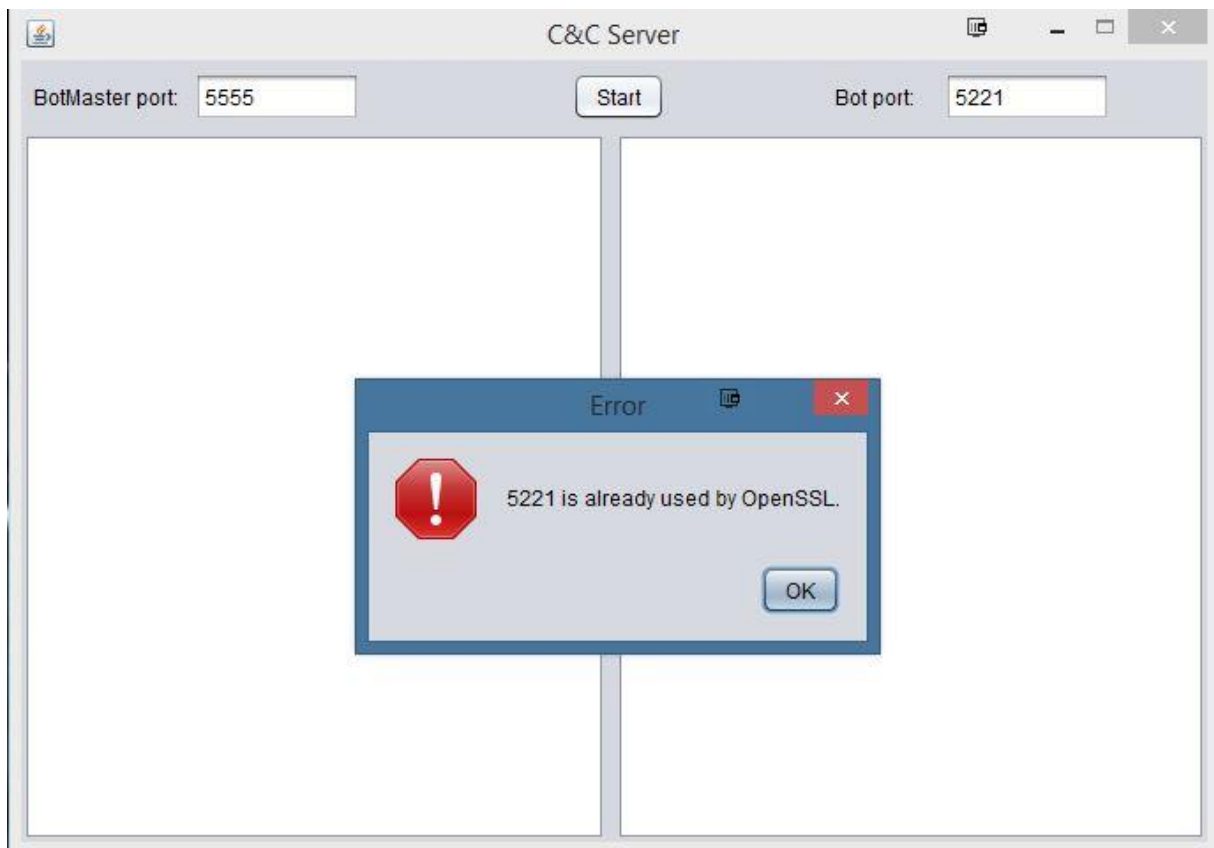
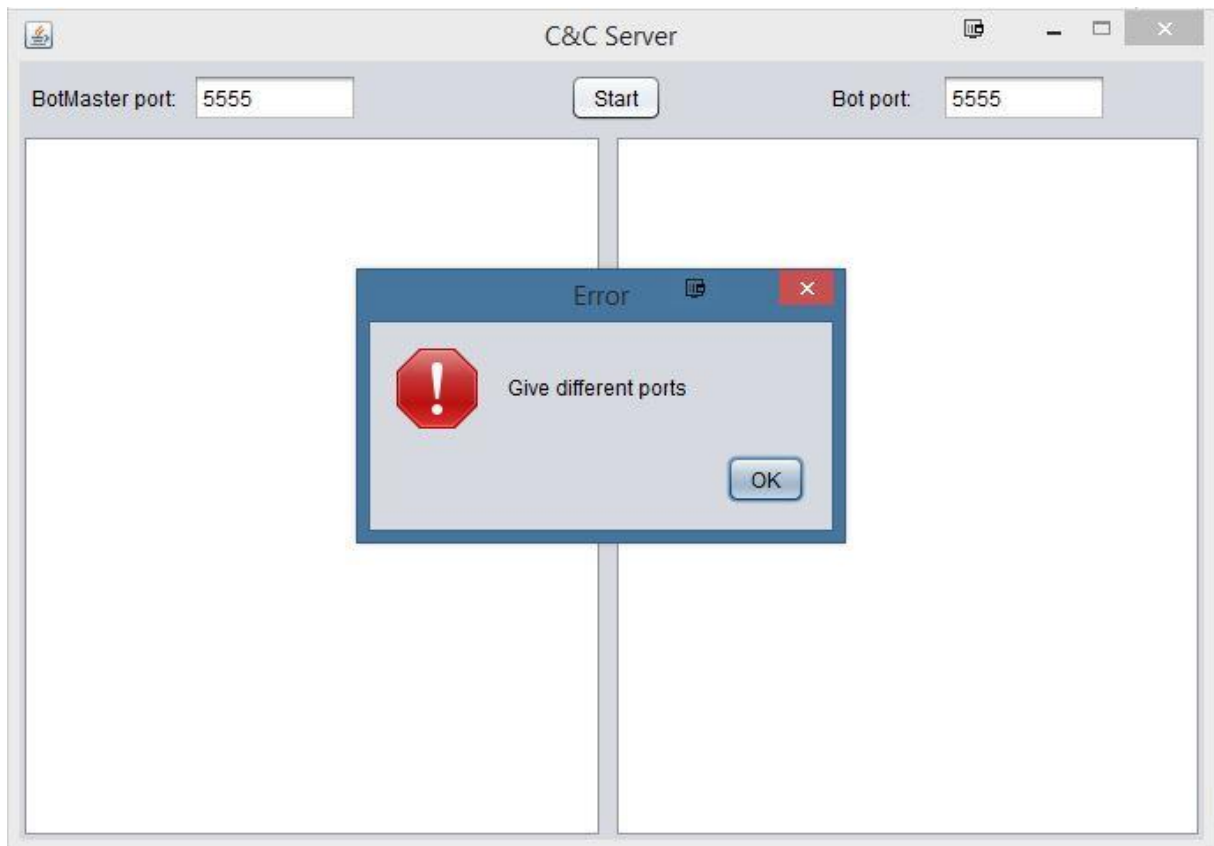
## 3) Εκτελέσιμα προγράμματα (project Netbeans κτλ)

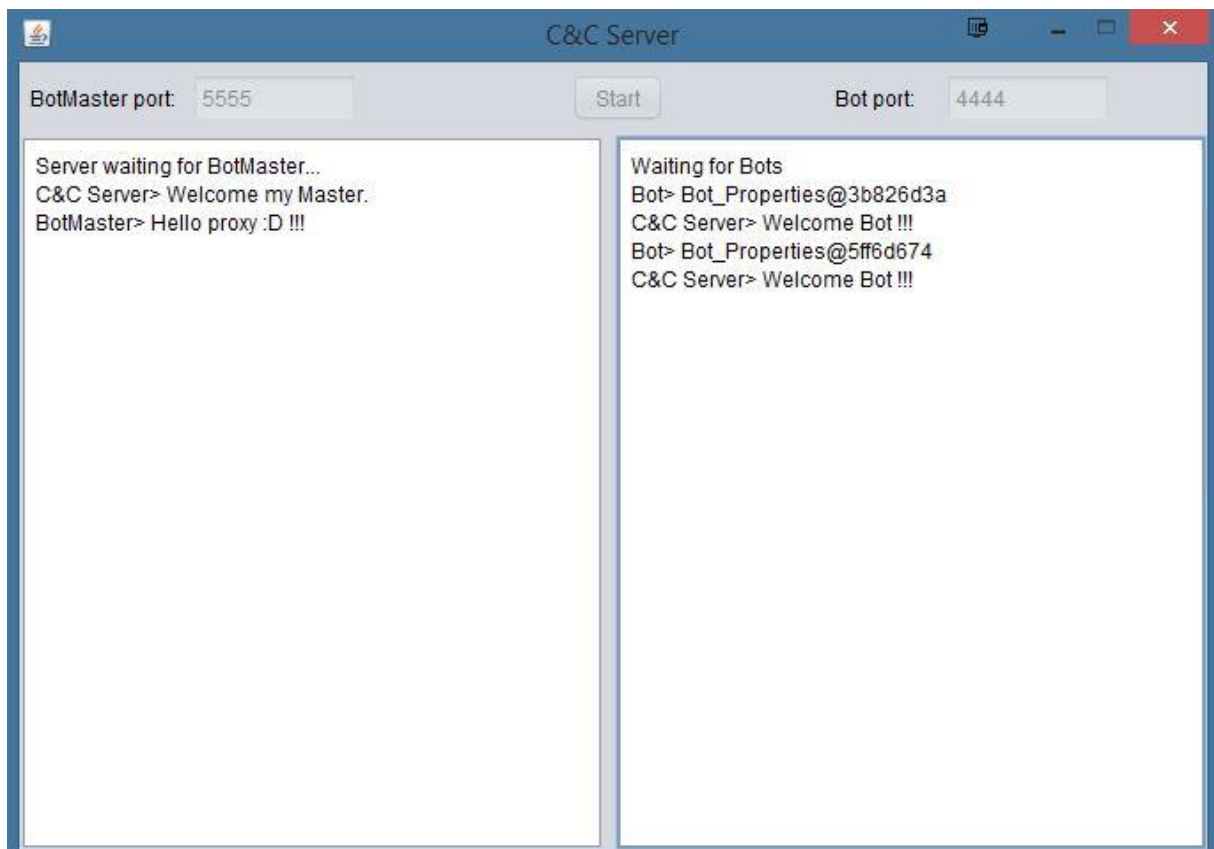
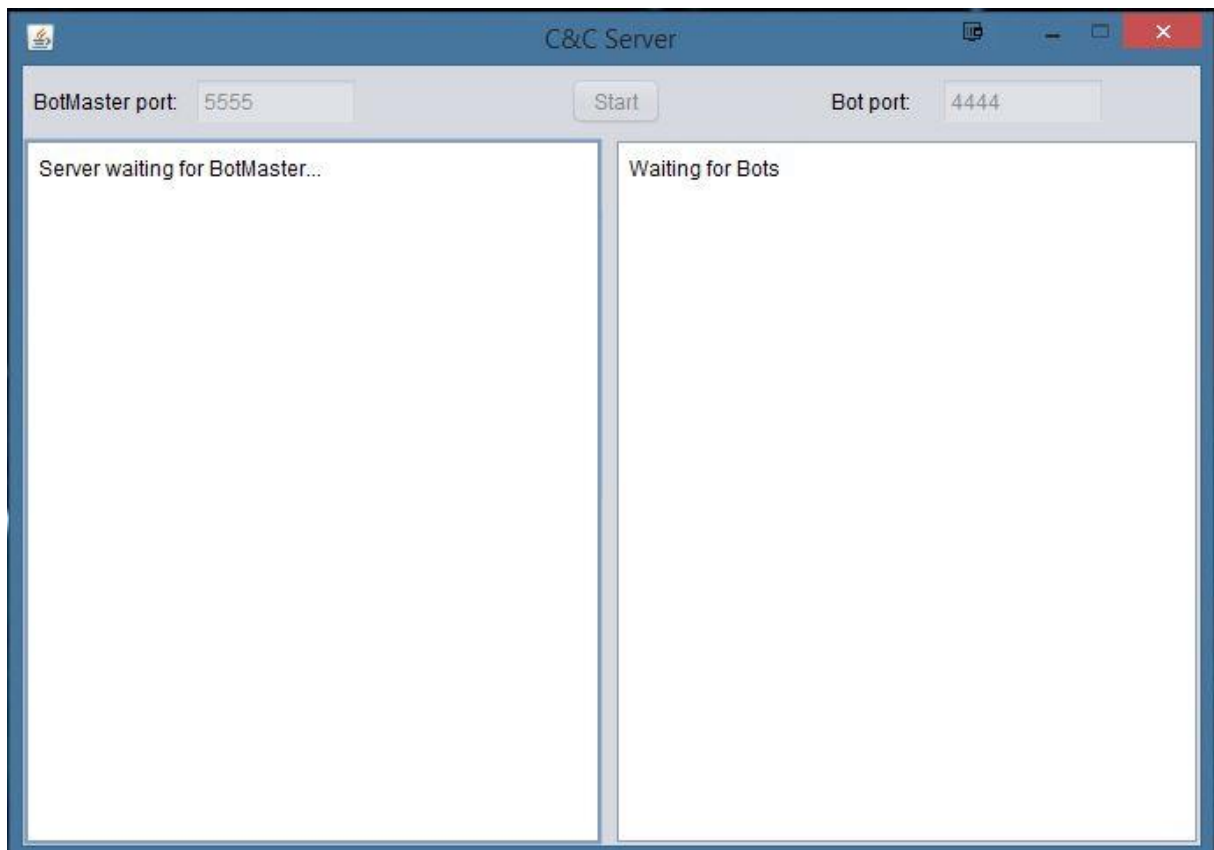
Τα projects εμπεριέχεται στο φάκελο με όνομα «java projects». Τα certificates υπάρχουν στο φάκελο certificates. Επίσης τα .jks αρχεία και το SynFlood.py υπάρχουν μέσα στα projects της java, ανάλογα με το ποιο χρηζόμασταν κάθε φορά.

#### 4) Στιγμιότυπα εκτέλεσης προγράμματος (screenshots)

Στιγμιότυπα από των Server:



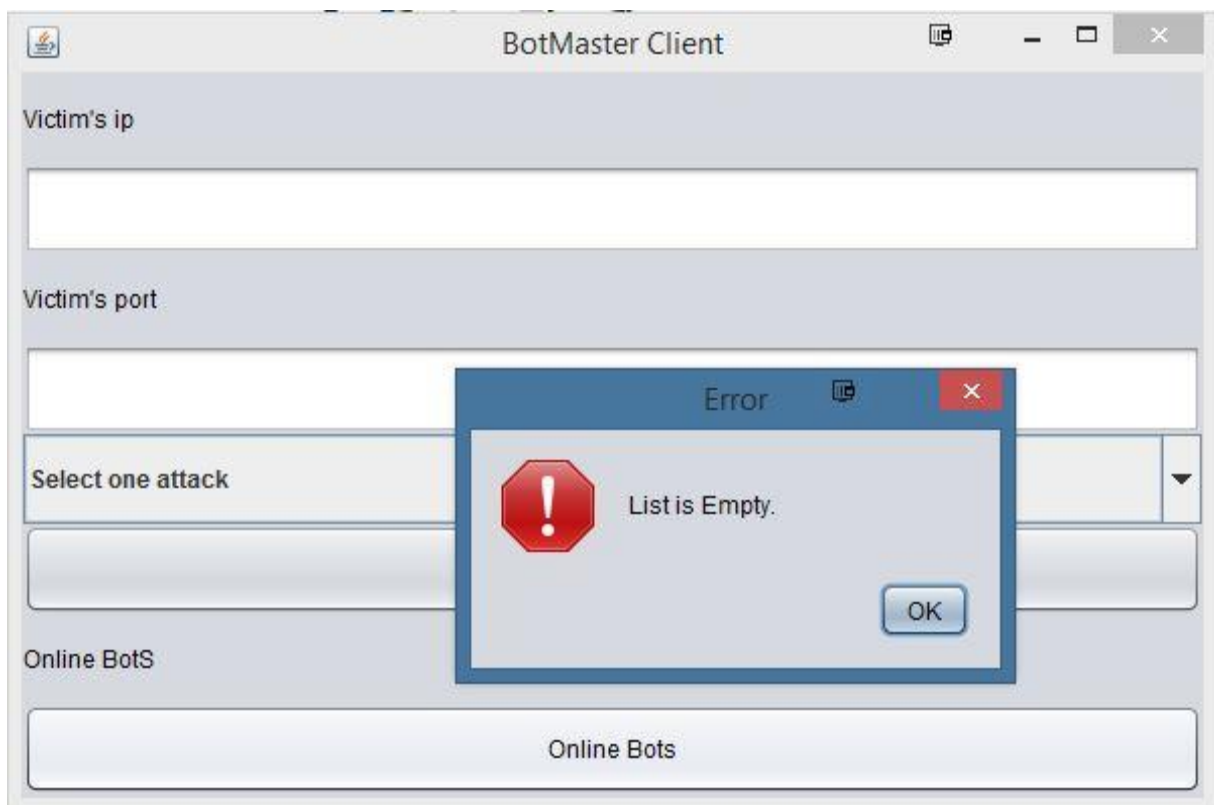




BotMaster:



Πατώντας το JButton Online Bots (αν δεν υπάρχουν Bots) :



BotMaster Client

Victim's ip

192.165.1.65

Victim's port

80

SynFlood

Start Attack

Online BotS

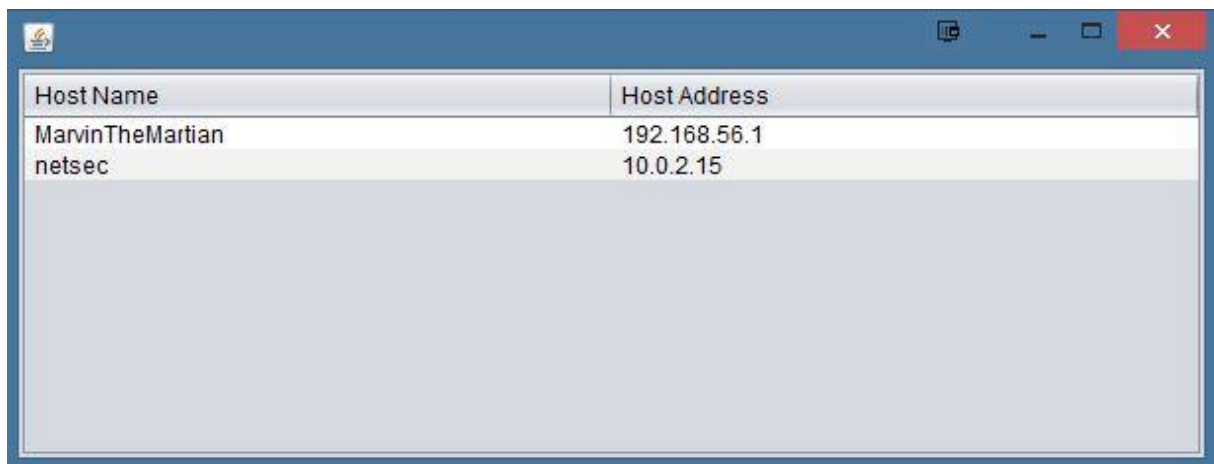
Online Bots

Πατώντας το JButton Online Bots (αν υπάρχουν bots):

Host Name	Host Address
Stelios	192.168.1.65
MarvinTheMartian	192.168.56.1
netsec	10.0.2.15



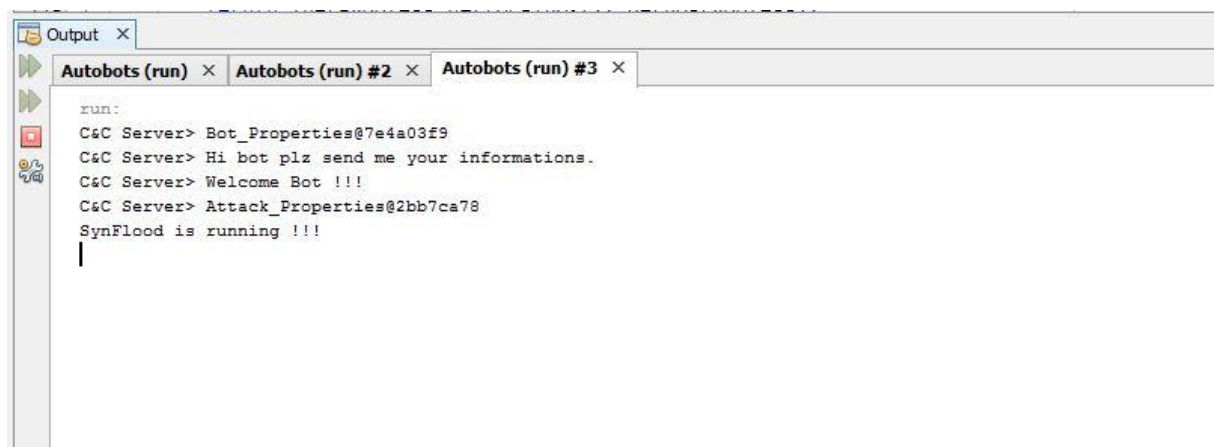
Αν αφαιρέσουμε ένα bot:



Host Name	Host Address
MarvinTheMartian	192.168.56.1
netsec	10.0.2.15

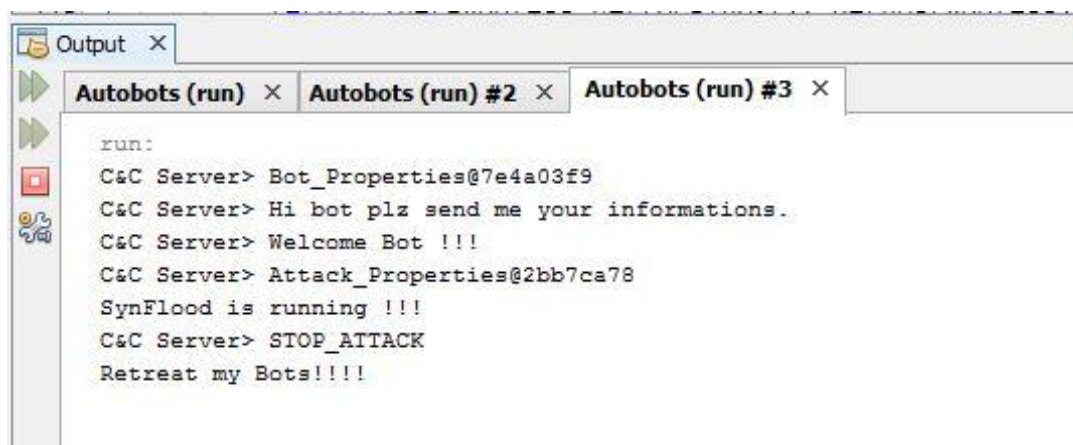
Bots:

Start Attack:



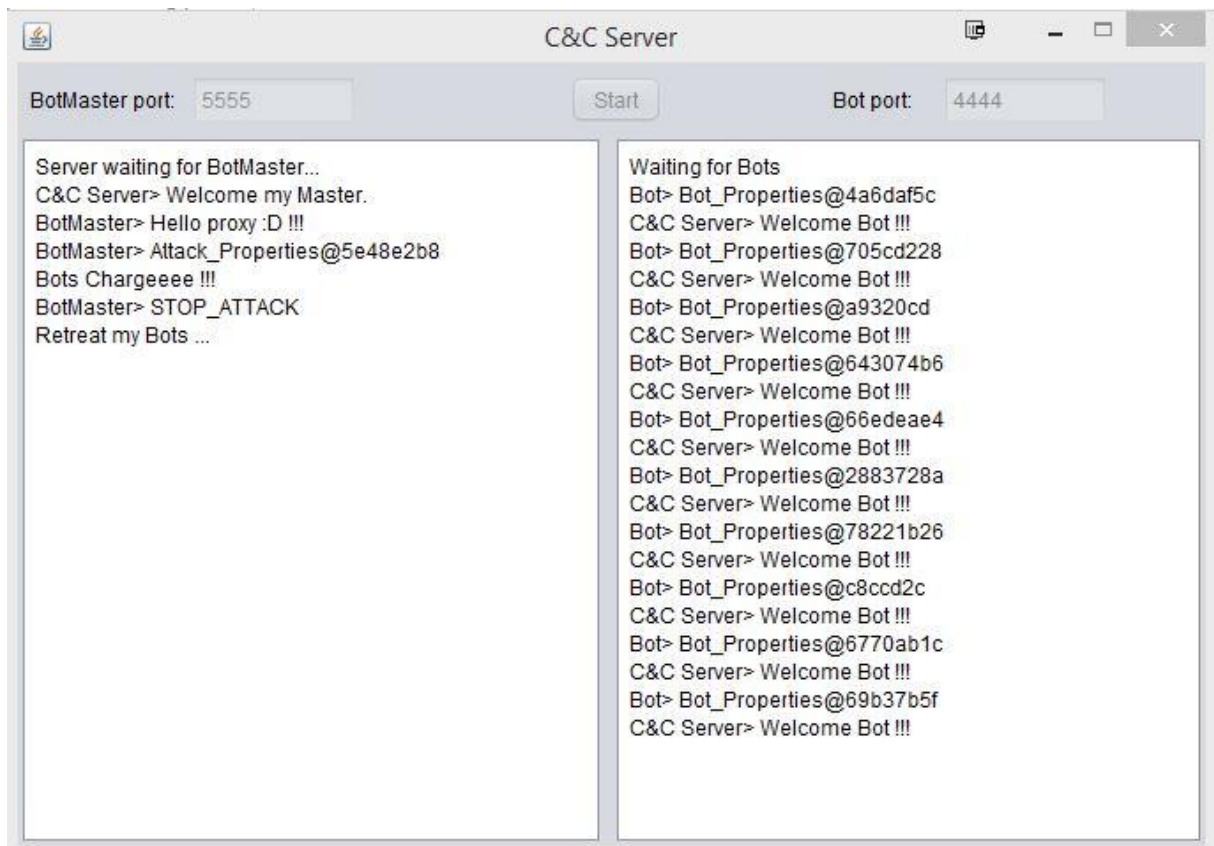
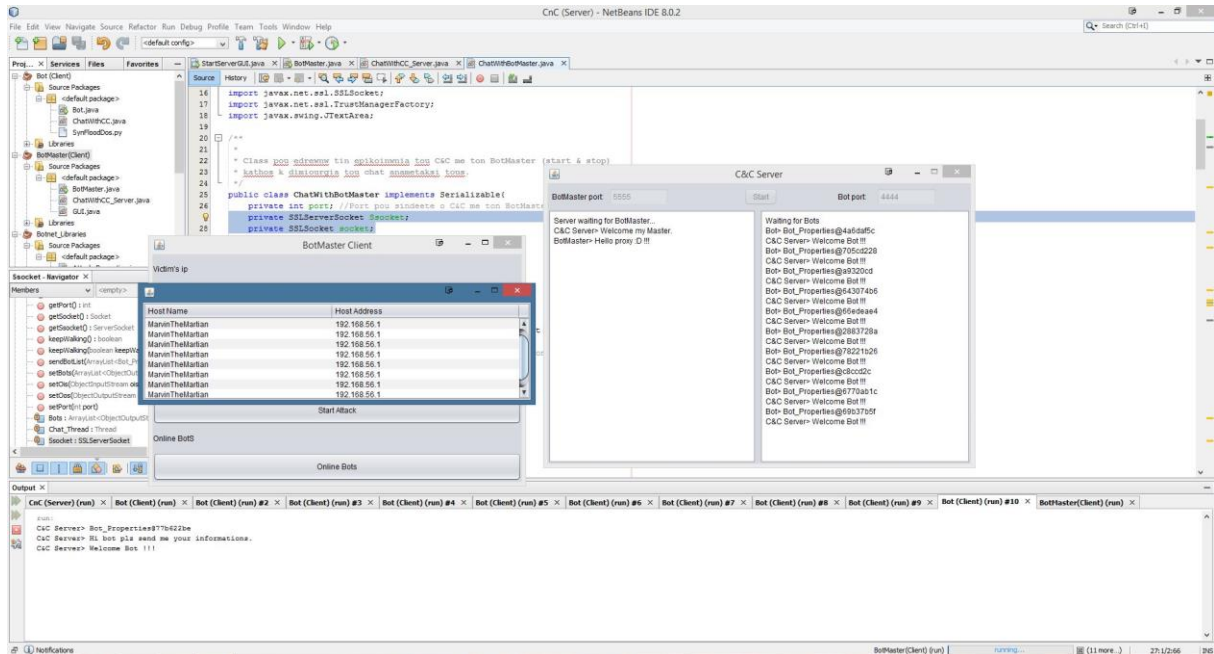
```
Output x
Autobots (run) x Autobots (run) #2 x Autobots (run) #3 x
run:
C&C Server> Bot_Properties@7e4a03f9
C&C Server> Hi bot plz send me your informations.
C&C Server> Welcome Bot !!!
C&C Server> Attack_Properties@2bb7ca78
SynFlood is running !!!
|
```

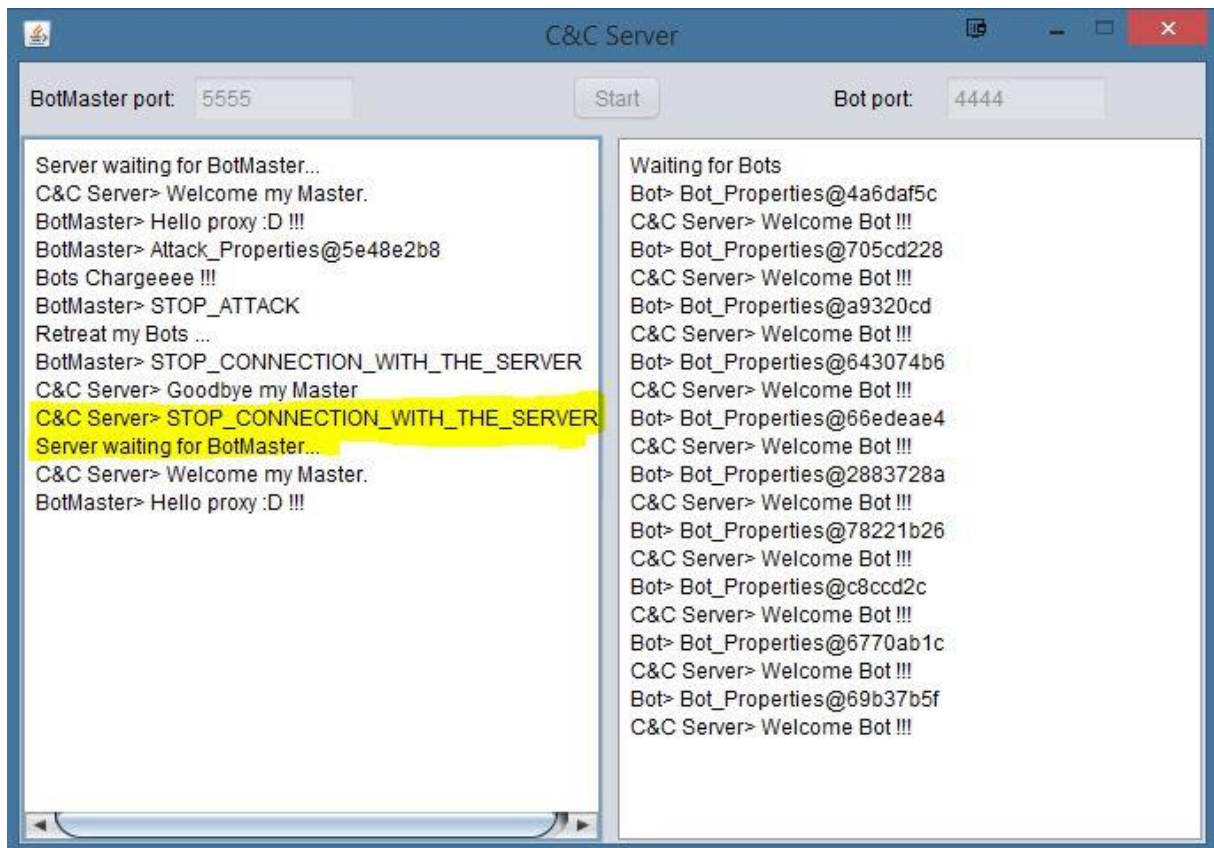
Stop Attack:



```
Output x
Autobots (run) x Autobots (run) #2 x Autobots (run) #3 x
run:
C&C Server> Bot_Properties@7e4a03f9
C&C Server> Hi bot plz send me your informations.
C&C Server> Welcome Bot !!!
C&C Server> Attack_Properties@2bb7ca78
SynFlood is running !!!
C&C Server> STOP_ATTACK
Retreat my Bots!!!!
```

- Test σε local host του botnet μας. Να τονίσουμε πως το test αυτό είναι με τα Projects που σας έχουμε ανεβάσει στο eclass δηλαδή είναι με secure connection μέσω του OpenSSL. Άρα όπως βλέπουμε το botnet μας δεν περιορίζεται σε μόνο 2-3 bots αλλά μπορούμε να τρέξουμε όσα θέλουμε.





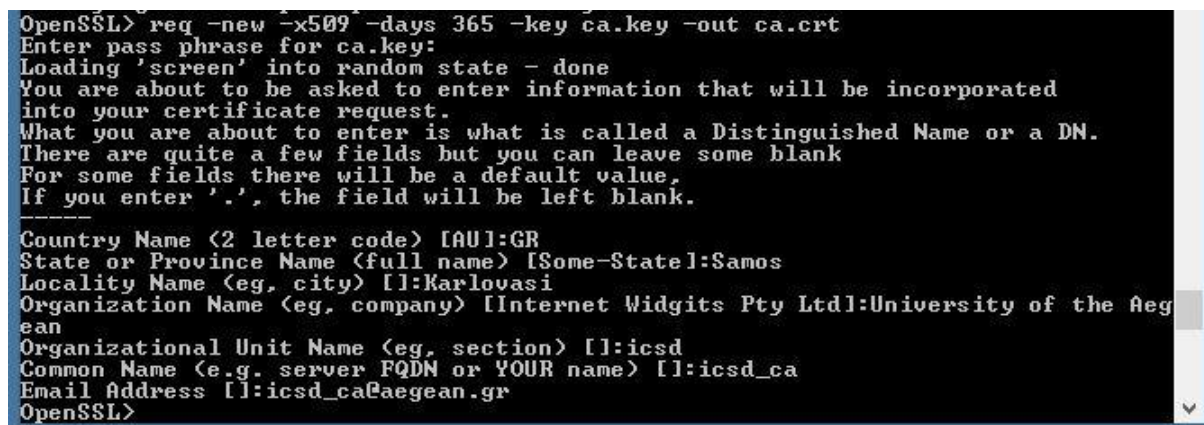
## 5) Περιγραφή και τρόπος δημιουργίας πιστοποιητικών / 6) Ψηφιακά Πιστοποιητικά

Δημιουργία αρχή πιστοποίησης - *certification authority (ca)* :



```
Command Prompt - C:\OpenSSL-Win64\bin\openssl
unable to write 'random state'
OpenSSL> genrsa -aes256 -out ca.key 4096
Loading 'screen' into random state - done
Generating RSA private key, 4096 bit long modulus
.....++
....++
unable to write 'random state'
e is 65537 (0x10001)
Enter pass phrase for ca.key:
Verifying - Enter pass phrase for ca.key:
OpenSSL>
```

Create ca certificate (.crt):



```
OpenSSL> req -new -x509 -days 365 -key ca.key -out ca.crt
Enter pass phrase for ca.key:
Loading 'screen' into random state - done
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.
-----
Country Name (2 letter code) [AU]:GR
State or Province Name (full name) [Some-State]:Samos
Locality Name (eg, city) []:Karlovasi
Organization Name (eg, company) [Internet Widgits Pty Ltd]:University of the Aegean
Organizational Unit Name (eg, section) []:icsd
Common Name (e.g. server FQDN or YOUR name) []:icsd_ca
Email Address []:icsd_ca@aegean.gr
OpenSSL>
```

Create ca Personal Information Exchange (.pfx) :

```
Administrator: Command Prompt - C:\OpenSSL-Win64\bin\openssl
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Ossl

C:\Ossl>C:\OpenSSL-Win64\bin\openssl
OpenSSL> pkcs12 -export -out ca.pfx -inkey ca.key -in ca.crt -certfile ca.crt
Loading 'screen' into random state - done
Enter pass phrase for ca.key:
Enter Export Password:
Verifying - Enter Export Password:
OpenSSL>
```

Create Server Key (private key):

```
Verifying - Enter Export Password:
OpenSSL> genrsa -aes256 -out Server.key 4096
Loading 'screen' into random state - done
Generating RSA private key, 4096 bit long modulus
.....++
e is 65537 (0x10001)
Enter pass phrase for Server.key:
Verifying - Enter pass phrase for Server.key:
OpenSSL>
```

Create Server certificate request (.csr) :

```
OpenSSL> req -new -key Server.key -out Server.csr
Enter pass phrase for Server.key:
Loading 'screen' into random state - done
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.
-----
Country Name (2 letter code) [AU]:GR
State or Province Name (full name) [Some-State]:Samos
Locality Name (eg, city) []:Karlovasi
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CCServer
Organizational Unit Name (eg, section) []:proxy
Common Name (e.g. server FQDN or YOUR name) []:proxyServer
Email Address []:proxy@aegean.gr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
OpenSSL>
```



Create Server Certificate (.crt):

```
OpenSSL> x509 -req -days 365 -in Server.csr -CA ca.crt -CAkey ca.key -set_serial 02 -out Server.crt
Loading 'screen' into random state - done
Signature ok
subject=/C=GR/ST=Samos/L=Karlovasi/O=CCServer/OU=proxy/CN=proxyServer/emailAddress=proxy@aegean.gr
Getting CA Private Key
Enter pass phrase for ca.key:
OpenSSL>
```

Create Server Java Keystore File (.jks):

```
C:\Ossl>"C:\Program Files\Java\jre1.8.0_65\bin\keytool" -importkeystore -srckeystore Server.pfx -srcstoretype pkcs12 -destkeystore Server.jks -deststoretype JKS
Enter destination keystore password:
Re-enter new password:
keytool error: java.io.FileNotFoundException: -Server.pfx (The system cannot find the file specified)

C:\Ossl>"C:\Program Files\Java\jre1.8.0_65\bin\keytool" -importkeystore -srckeystore Server.pfx -srcstoretype pkcs12 -destkeystore Server.jks -deststoretype JKS
Enter destination keystore password:
Re-enter new password:
Enter source keystore password:
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
C:\Ossl>
```

Create Bot1 key:

```
OpenSSL> genrsa -aes256 -out Bot1.key 4096
Loading 'screen' into random state - done
Generating RSA private key, 4096 bit long modulus
.....++
.....++
e is 65537 (0x10001)
Enter pass phrase for Bot1.key:
Verifying - Enter pass phrase for Bot1.key:
OpenSSL>
```

Create Bot1 crt:

```
OpenSSL> x509 -req -days 365 -in Bot1.csr -CA ca.crt -CAkey ca.key -set_serial 03 -out Bot1.crt
Loading 'screen' into random state - done
Signature ok
subject=/C=FR/ST=France/L=Lion/O=Bot/OU=Bot/CN=icsd12200/emailAddress=icsd12200@aegean.gr
Getting CA Private Key
Enter pass phrase for ca.key:
OpenSSL>
```

Create Bot1 csr:

```
OpenSSL> req -new -key Bot1.key -out Bot1.csr
Enter pass phrase for Bot1.key:
Loading 'screen' into random state - done
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.

-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:France
Locality Name (eg, city) []:Lion
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Bot
Organizational Unit Name (eg, section) []:Bot
Common Name (e.g. server FQDN or YOUR name) []:icsd12200
Email Address []:icsd12200@aegean.gr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1234
An optional company name []:
OpenSSL>
```

Create Bots pfx:

```
OpenSSL> pkcs12 -export -out Bot1.pfx -inkey Bot1.key -in Bot1.crt -certfile ca.
crt
Loading 'screen' into random state - done
Enter pass phrase for Bot1.key:
Enter Export Password:
Verifying - Enter Export Password:
OpenSSL> pkcs12 -export -out Bot2.pfx -inkey Bot2.key -in Bot2.crt -certfile ca.
crt
Loading 'screen' into random state - done
Enter pass phrase for Bot2.key:
Enter Export Password:
Verifying - Enter Export Password:
OpenSSL> pkcs12 -export -out Bot3.pfx -inkey Bot3.key -in Bot3.crt -certfile ca.
crt
Loading 'screen' into random state - done
Enter pass phrase for Bot3.key:
Enter Export Password:
Verifying - Enter Export Password:
OpenSSL> pkcs12 -export -out BotMaster.pfx -inkey BotMaster.key -in BotMaster.cr
t -certfile ca.crt
Loading 'screen' into random state - done
Enter pass phrase for BotMaster.key:
Enter Export Password:
Verifying - Enter Export Password:
OpenSSL>
```

Ομοίως για όλα τα Bots & BotMaster φτιάξαμε τα .key , .crt, .csr, .pfx, .jks. Ποιο αναλυτικά τα αρχεία έχουν ως εξής:

This PC > Local Disk (C:) > Ossl				
Name	Date modified	Type	Size	
Bot1	03-Nov-15 8:29 PM	Security Certificate	2 KB	
Bot1.csr	03-Nov-15 8:25 PM	CSR File	2 KB	
Bot1.jks	03-Nov-15 10:03 P...	JKS File	6 KB	
Bot1	03-Nov-15 8:22 PM	KEY File	4 KB	
Bot1	03-Nov-15 8:54 PM	Personal Information Exchange	6 KB	
Bot2	03-Nov-15 8:34 PM	Security Certificate	2 KB	
Bot2.csr	03-Nov-15 8:33 PM	CSR File	2 KB	
Bot2.jks	03-Nov-15 10:03 P...	JKS File	6 KB	
Bot2	03-Nov-15 8:31 PM	KEY File	4 KB	
Bot2	03-Nov-15 8:55 PM	Personal Information Exchange	6 KB	
Bot3	03-Nov-15 8:40 PM	Security Certificate	2 KB	
Bot3.csr	03-Nov-15 8:39 PM	CSR File	2 KB	
Bot3.jks	03-Nov-15 10:04 P...	JKS File	6 KB	
Bot3	03-Nov-15 8:36 PM	KEY File	4 KB	
Bot3	03-Nov-15 8:55 PM	Personal Information Exchange	6 KB	
BotMaster	03-Nov-15 8:44 PM	Security Certificate	2 KB	
BotMaster.csr	03-Nov-15 8:43 PM	CSR File	2 KB	
BotMaster.jks	03-Nov-15 10:03 P...	JKS File	6 KB	
BotMaster	03-Nov-15 8:40 PM	KEY File	4 KB	
BotMaster	03-Nov-15 8:56 PM	Personal Information Exchange	6 KB	
ca	03-Nov-15 8:06 PM	Security Certificate	3 KB	
ca	03-Nov-15 8:04 PM	KEY File	4 KB	
ca	03-Nov-15 8:10 PM	Personal Information Exchange	6 KB	
Server	03-Nov-15 9:33 PM	Security Certificate	2 KB	
Server.csr	03-Nov-15 9:31 PM	CSR File	2 KB	
Server.jks	03-Nov-15 10:01 P...	JKS File	6 KB	
Server	03-Nov-15 9:30 PM	KEY File	4 KB	
Server	03-Nov-15 9:34 PM	Personal Information Exchange	6 KB	

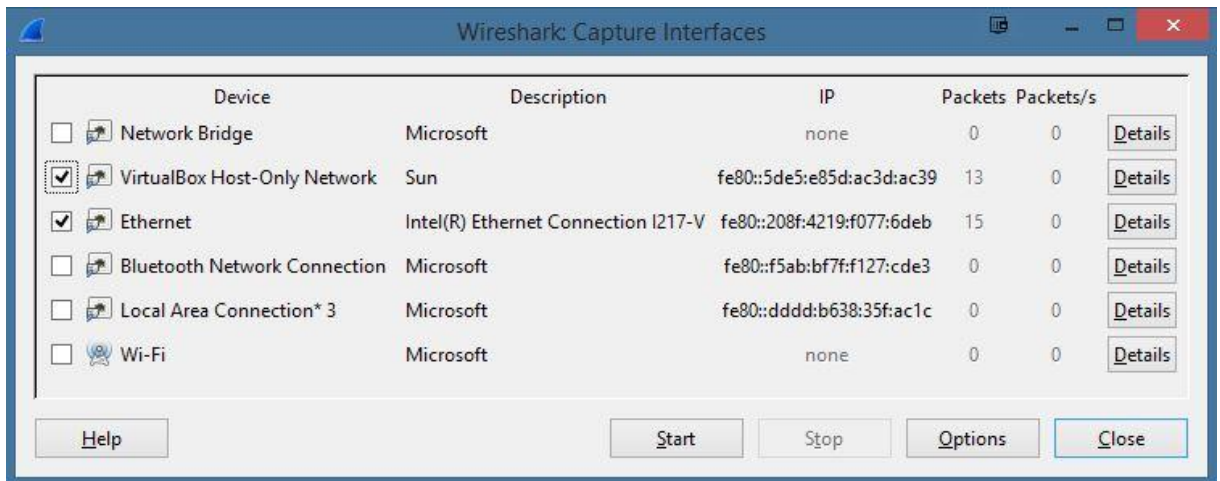
Το .jks το χρησιμοποιούμε για να καλούμε τα πιστοποιητικά μέσω της java. Για την εργασία δημιουργήσαμε 3 crt για τα Bots αλλά μέσα στον κώδικα χρησιμοποιούμε κάθε φορά μόνο το ένα πιστοποιητικό αφού δεν μας χρειάζεται κάτι άλλο. Το .pfx το χρησιμοποιούμε για την δημιουργία του .jks.

Για την επιτυχή σύνδεση του Server με τους Clients (Bots & BotMaster). Κάθε οντότητα χρησιμοποιεί το πιστοποιητικό της συν το πιστοποιητικό της ca.

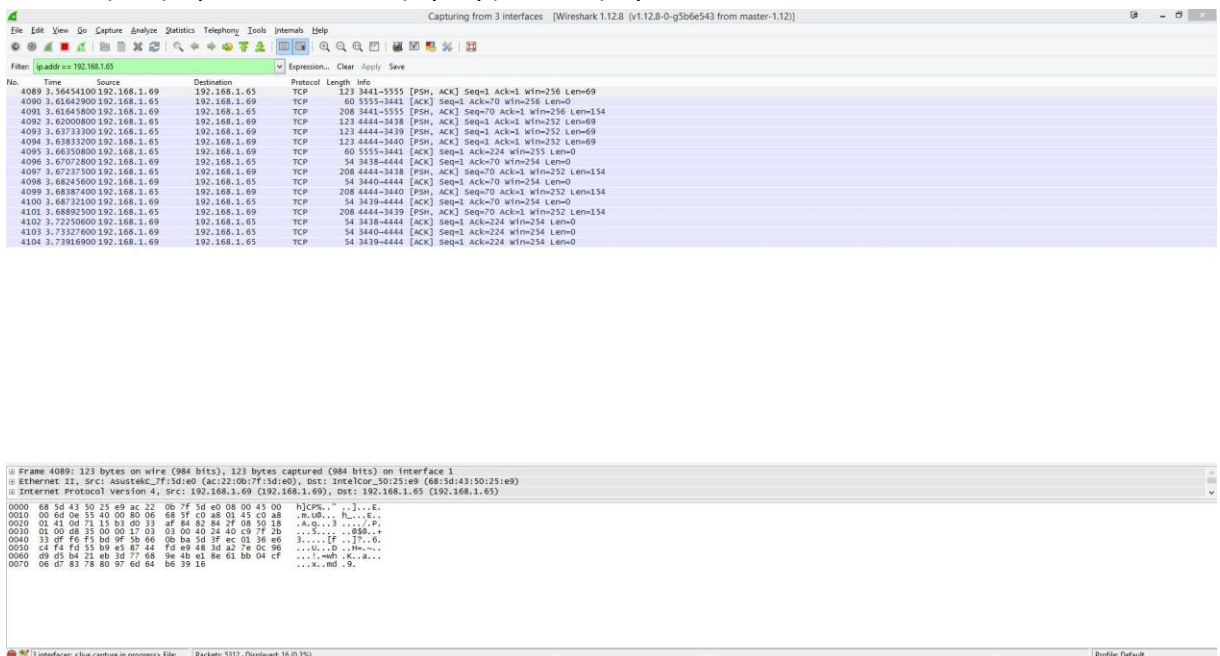


## 7) Ανάλυση των αποτελεσμάτων χρήσης του εργαλείου σύλληψης πακέτων (sniffer) και ενδεικτικά στιγμιότυπα εκτέλεσης (screenshots). Επεξήγηση των φίλτρων που χρησιμοποιήθηκαν.

- Ρυθμίσεις στο WireShark:



- Επίθεση στην ip 192.168.1.65 (laptop) από την ip 192.168.1.69:



Όπως φαίνεται στο screenshot η συνομιλία είναι κρυπτογραφημένη μέσω του OpenSSL. Χρησιμοποιήσαμε το φίλτρο `ip.addr == 192.168.1.65` (από την ip 192.168.1.69 (Desktop)).

Για την SynFlood Attack χρησιμοποιήσαμε το script της Python που μας δόθηκε κάνοντας εγκατάσταση τη Python, την scapy librarie και το PyCharm IDE για να τρέξουμε/τεστάρουμε τον κώδικα.

his PC > Local Disk (C:) > Python34				Search Python34
Name	Date modified	Type	Size	
DLLs	27-Jun-15 5:35 PM	File folder		
Doc	27-Jun-15 5:35 PM	File folder		
fysom-2.1.2	25-Oct-15 11:07 AM	File folder		
include	27-Jun-15 5:35 PM	File folder		
Lib	27-Jun-15 5:35 PM	File folder		
libs	27-Jun-15 5:35 PM	File folder		
pcapy-0.10.8	25-Oct-15 11:36 AM	File folder		
scapy	25-Oct-15 2:48 AM	File folder		
Scripts	25-Oct-15 11:51 AM	File folder		
tcl	27-Jun-15 5:35 PM	File folder		
Tools	27-Jun-15 5:35 PM	File folder		
LICENSE	24-Feb-15 9:50 PM	Text Document	31 KB	
NEWS	24-Feb-15 9:38 PM	Text Document	376 KB	
python	24-Feb-15 9:43 PM	Application	27 KB	
pythonw	24-Feb-15 9:43 PM	Application	27 KB	
pywin32-wininst	25-Oct-15 11:51 AM	Text Document	121 KB	
README	24-Feb-15 9:38 PM	Text Document	7 KB	
Removepywin32	25-Oct-15 11:51 AM	Application	187 KB	

- Stop Attack στην ip 192.168.1.65:

Filter: ip.addr == 192.168.1.65

No.	Time	Source	Destination	Protocol	Length	Info
4089	3.56454100	192.168.1.69	192.168.1.65	TCP	123	3441-5555 [PSH, ACK] Seq=1 Ack=1 Wln=256 Len=0
4090	3.61642900	192.168.1.65	192.168.1.69	TCP	60	5555-3441 [ACK] Seq=1 Ack=70 Wln=256 Len=0
4091	3.61645800	192.168.1.69	192.168.1.65	TCP	208	3441-5555 [PSH, ACK] Seq=70 Ack=1 Wln=256 Len=154
4092	3.62008000	192.168.1.65	192.168.1.69	TCP	123	4444-3438 [PSH, ACK] Seq=1 Ack=1 Wln=252 Len=0
4093	3.63733300	192.168.1.65	192.168.1.69	TCP	123	4444-3439 [PSH, ACK] Seq=1 Ack=1 Wln=252 Len=0
4094	3.63812000	192.168.1.65	192.168.1.69	TCP	123	4444-3440 [PSH, ACK] Seq=1 Ack=1 Wln=252 Len=0
4095	3.66350800	192.168.1.65	192.168.1.69	TCP	60	5555-3441 [ACK] Seq=1 Ack=224 Wln=255 Len=0
4096	3.67072800	192.168.1.69	192.168.1.65	TCP	54	3438-4444 [ACK] Seq=1 Ack=70 Wln=254 Len=0
4097	3.67237500	192.168.1.65	192.168.1.69	TCP	208	4444-3438 [PSH, ACK] Seq=70 Ack=1 Wln=252 Len=154
4098	3.68245600	192.168.1.69	192.168.1.65	TCP	54	3440-4444 [ACK] Seq=1 Ack=70 Wln=254 Len=0
4099	3.68387400	192.168.1.65	192.168.1.69	TCP	208	4444-3440 [PSH, ACK] Seq=70 Ack=1 Wln=252 Len=154
4100	3.68732200	192.168.1.69	192.168.1.65	TCP	54	3439-4444 [ACK] Seq=1 Ack=70 Wln=254 Len=0
4101	3.68892500	192.168.1.65	192.168.1.69	TCP	208	4444-3439 [PSH, ACK] Seq=70 Ack=1 Wln=252 Len=154
4102	3.72210600	192.168.1.69	192.168.1.65	TCP	54	3438-4444 [ACK] Seq=1 Ack=224 Wln=254 Len=0
4103	3.73376000	192.168.1.69	192.168.1.65	TCP	54	3440-4444 [ACK] Seq=1 Ack=224 Wln=254 Len=0
4104	3.73516900	192.168.1.69	192.168.1.65	TCP	54	3439-4444 [ACK] Seq=1 Ack=224 Wln=254 Len=0
5574	37.13277900	192.168.1.69	192.168.1.65	TCP	123	3421-5555 [PSH, ACK] Seq=224 Ack=1 Wln=256 Len=0
5575	37.33880400	192.168.1.65	192.168.1.69	TCP	123	4444-3438 [PSH, ACK] Seq=224 Ack=1 Wln=252 Len=0
5576	37.33882900	192.168.1.65	192.168.1.69	TCP	123	4444-3439 [PSH, ACK] Seq=224 Ack=1 Wln=252 Len=0
5577	37.33972500	192.168.1.65	192.168.1.69	TCP	123	4444-3440 [PSH, ACK] Seq=224 Ack=1 Wln=252 Len=0
5578	37.38397900	192.168.1.65	192.168.1.69	TCP	60	5555-3441 [ACK] Seq=1 Ack=293 Wln=255 Len=0
5579	37.38931200	192.168.1.69	192.168.1.65	TCP	54	3439-4444 [ACK] Seq=1 Ack=293 Wln=254 Len=0
5580	37.38931500	192.168.1.69	192.168.1.65	TCP	54	3438-4444 [ACK] Seq=1 Ack=293 Wln=254 Len=0
5581	37.38931800	192.168.1.69	192.168.1.65	TCP	54	3440-4444 [ACK] Seq=1 Ack=293 Wln=254 Len=0

Ανά την IP: 192.168.1.69 που στέλνει τα Bots κάτω SYNflood στην IP: 192.168.1.65

- Chat BotMaster με Server (via OpenSSL) :

Filter: tcp.port == 4444

No.	Time	Source	Destination	Protocol	Length	Info
476	31.94207800	192.168.1.69	192.168.1.65	TCP	60	2637-4444 [SYN] Seq=0 Wln=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
478	31.94484900	192.168.1.69	192.168.1.65	TCP	60	4444-2637 [ACK] Seq=0 Ack=1 Wln=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
480	31.94850200	192.168.1.69	192.168.1.65	TCP	54	2637-4444 [ACK] Seq=1 Ack=1 Wln=5536 Len=0
482	31.95354900	192.168.1.69	192.168.1.65	TCP	260	2637-4444 [PSH, ACK] Seq=1 Ack=1 Wln=5536 Len=212
483	32.01510000	192.168.1.65	192.168.1.69	TCP	60	4444-2637 [ACK] Seq=1 Ack=213 Wln=5536 Len=0
485	32.26043500	192.168.1.65	192.168.1.69	TCP	1514	4444-2637 [ACK] Seq=1 Ack=213 Wln=5536 Len=1460
486	32.26043600	192.168.1.65	192.168.1.69	TCP	1514	4444-2637 [ACK] Seq=1461 Ack=213 Wln=5536 Len=1460
487	32.26043700	192.168.1.65	192.168.1.69	TCP	995	4444-2637 [PSH, ACK] Seq=2921 Ack=213 Wln=5536 Len=841
488	32.26047300	192.168.1.69	192.168.1.65	TCP	54	2637-4444 [ACK] Seq=213 Ack=3862 Wln=5536 Len=0
489	32.27002000	192.168.1.69	192.168.1.65	TCP	1514	2637-4444 [ACK] Seq=213 Ack=3862 Wln=5536 Len=1460
490	32.27002700	192.168.1.69	192.168.1.65	TCP	1514	2637-4444 [ACK] Seq=1873 Ack=3862 Wln=5536 Len=1460
491	32.27003000	192.168.1.69	192.168.1.65	TCP	191	2637-4444 [PSH, ACK] Seq=213 Ack=3862 Wln=5536 Len=137
492	32.27286300	192.168.1.69	192.168.1.65	TCP	60	4444-2637 [ACK] Seq=3862 Ack=3270 Wln=5536 Len=0
493	32.32748000	192.168.1.69	192.168.1.65	TCP	570	2637-4444 [PSH, ACK] Seq=3270 Ack=3862 Wln=5536 Len=525
494	32.38958800	192.168.1.65	192.168.1.69	TCP	60	4444-2637 [ACK] Seq=3862 Ack=3795 Wln=55024 Len=0
495	32.38963800	192.168.1.69	192.168.1.65	TCP	145	2637-4444 [PSH, ACK] Seq=3795 Ack=3862 Wln=5536 Len=91
496	32.39274400	192.168.1.65	192.168.1.69	TCP	60	4444-2637 [PSH, ACK] Seq=3862 Ack=3886 Wln=55024 Len=0
499	32.44314800	192.168.1.69	192.168.1.65	TCP	54	2637-4444 [ACK] Seq=3886 Ack=3868 Wln=5536 Len=0
500	32.44527400	192.168.1.65	192.168.1.69	TCP	208	4444-2637 [PSH, ACK] Seq=3868 Ack=3886 Wln=55024 Len=154
503	32.44667600	192.168.1.69	192.168.1.65	TCP	123	2637-4444 [PSH, ACK] Seq=3886 Ack=4022 Wln=5536 Len=69
502	32.45103400	192.168.1.65	192.168.1.69	TCP	155	4444-2637 [PSH, ACK] Seq=4022 Ack=3955 Wln=5536 Len=103
503	32.45163000	192.168.1.69	192.168.1.65	TCP	267	2637-4444 [PSH, ACK] Seq=3955 Ack=4123 Wln=55280 Len=213
504	32.51453300	192.168.1.65	192.168.1.69	TCP	60	4444-2637 [ACK] Seq=4123 Ack=4168 Wln=55280 Len=0
505	32.51457600	192.168.1.69	192.168.1.65	TCP	224	2637-4444 [PSH, ACK] Seq=4168 Ack=4123 Wln=55280 Len=170
506	32.52111900	192.168.1.65	192.168.1.69	TCP	139	4444-2637 [PSH, ACK] Seq=4123 Ack=4338 Wln=5512 Len=85
507	32.57284400	192.168.1.69	192.168.1.65	TCP	54	2637-4444 [ACK] Seq=4338 Ack=4208 Wln=55280 Len=0
850	54.02361300	192.168.1.69	192.168.1.65	TCP	60	2636-4444 [SYN] Seq=0 Wln=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
851	54.03139200	192.168.1.65	192.168.1.69	TCP	66	4444-2636 [SYN, ACK] Seq=0 Ack=1 Wln=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
854	54.03144100	192.168.1.69	192.168.1.65	TCP	54	2636-4444 [ACK] Seq=1 Ack=1 Wln=5536 Len=0
855	54.03698700	192.168.1.69	192.168.1.65	TCP	260	2636-4444 [PSH, ACK] Seq=1 Ack=1 Wln=5536 Len=212
857	54.10967700	192.168.1.65	192.168.1.69	TCP	60	4444-2636 [ACK] Seq=1 Ack=213 Wln=5536 Len=0
858	54.38297900	192.168.1.65	192.168.1.69	TCP	1514	4444-2636 [ACK] Seq=1461 Ack=213 Wln=5536 Len=1460
860	54.38298000	192.168.1.65	192.168.1.69	TCP	995	4444-2636 [PSH, ACK] Seq=2921 Ack=213 Wln=5536 Len=841
861	54.38300700	192.168.1.69	192.168.1.65	TCP	54	2636-4444 [ACK] Seq=213 Ack=3862 Wln=5536 Len=0

Frame 858: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 1  
 Ethernet II, Src: IntelCor\_50:25:0e (68:5d:43:50:25:0e), Dst: Asustek\_7f:5d:e0 (a2:22:0b:7f:5d:e0)  
 Internet Protocol Version 4, Src: 192.168.1.65 (192.168.1.65), Dst: 192.168.1.69 (192.168.1.69)  
 Version: 4  
 Header Length: 20 bytes

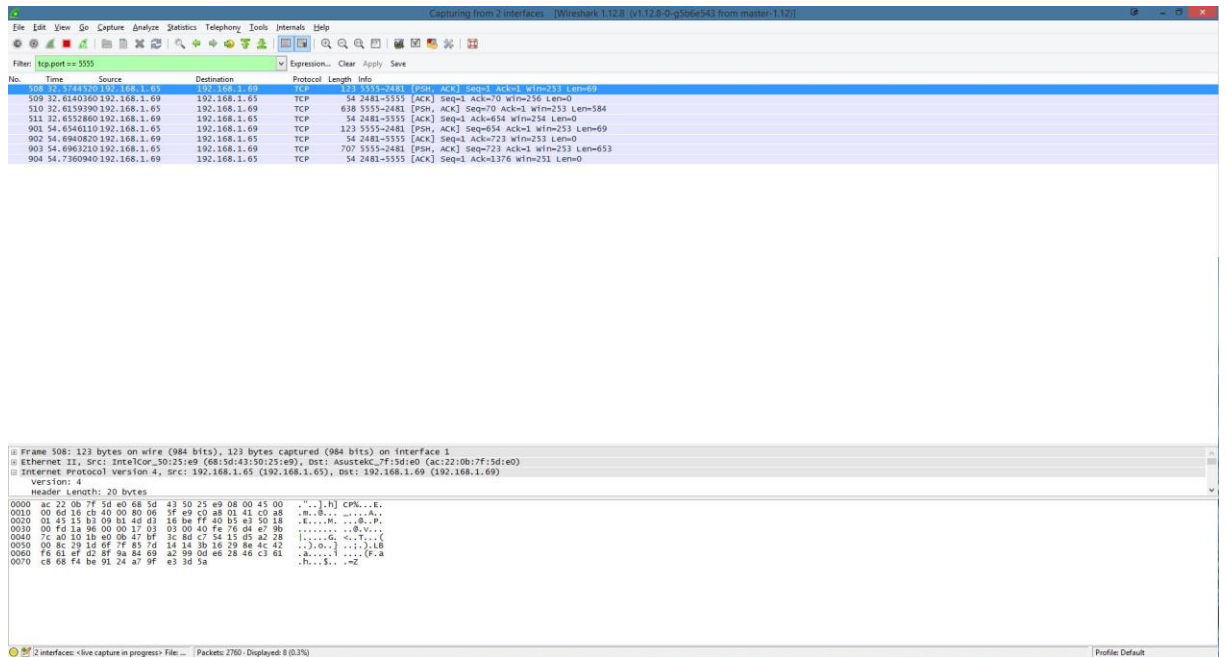
```

0000  ac 22 0b 7f 5d e0 68 5d 43 50 25 e9 08 00 45 00  "...].h] CPK...E.
0010  05 0c 16 ff 40 00 80 06 5a 66 c0 a8 01 41 c0 a8  "P...A...
0020  01 45 11 3c 0a 60 87 98 3b c9 5e bc 8c 07 50 10  "E...A...P.
0030  01 00 22 00 00 16 03 03 0f 10 02 00 00 4d 03 0f  "V...A...e.3.#
0040  03 56 3c 05 5e ee a0 9e f8 f4 73 65 aa aa e8 23  "V...A...e.3.#
0050  01 97 c4 87 e8 64 9e 60 a7 07 f4 83 41 83 13 0f  "d...e...A...
0060  50 20 56 3c 06 3b 8b 9e 7a 19 8d 8c ad 55 74 ff  "P Ve...A...z...MUT.
0070  83 61 a8 0b 01 8c 02 cc ca 3b 76 1c 05 0b 0d c7  "A...e...A...
0080  d1 e9 c9 27 00 00 05 ff 01 00 01 00 0b 0b aa  "....A...
0090  00 a7 00 05 8a 10 42 05 86 10 82 03 47 02 01  "....A...
00a0  02 30 d0 06 09 2a 8e 48 86 f7 d0 01 01 0b 05 00  "O...A...
00b0  10 87 11 0b 08 06 15 54 08 0c 05 53 61 8f 8f 73  "O...A...
00c0  31 04 00 0c 06 03 55 04 08 0c 05 53 61 8f 8f 73  "I...O...samps
00d0  12 10 10 06 03 55 04 07 0c 09 4b 03 72 0c ff  "I...O...KAP To
00e0  76 61 73 69 31 21 30 3f 06 03 55 04 0a c0 18 55  "v8t110...U
00f0  4e 69 76 65 73 69 31 21 30 3f 06 03 55 04 0a c0  "ntverse y of the
0100  20 41 65 67 65 61 6a 31 0d 30 0b 06 03 55 04 0b  "Aegant...O...U...
0110  0a 08 65 73 69 31 21 30 3f 06 03 55 04 0b 06  "T...O...H...

```

Filter: tcp.port == 5555 όπου είναι το port του SSLSocket στον κώδικα.

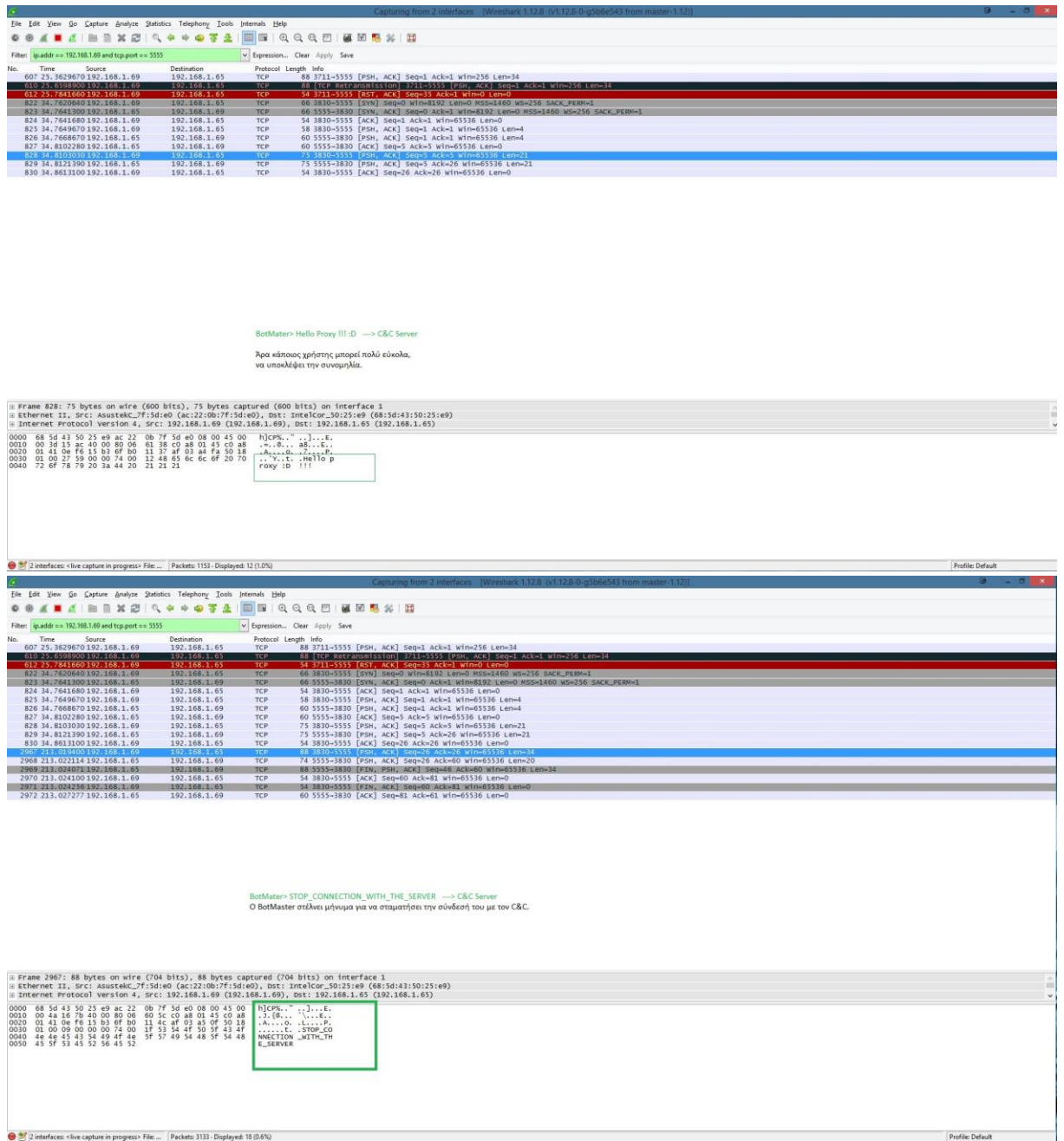
- Server chat with BotMaster (via OpenSSL):



Filter: tcp.port == 5555 όπου είναι το port του SSLSocket στον κώδικα.

Ομοίως και για τα Bots στο Port 4444

- Unsecure επικοινωνία του Botnet:



Wireshark 1.12.4 (v1.12.4-0-g750e543 from master-1.12.4)

Filter: `ip.addr == 192.168.1.69 and tcp.port == 5555`

No.	Time	Source	Destination	Protocol	Length	Info
607	23.3829670	192.168.1.69	192.168.1.65	TCP	88	3711->5555 [PSH, ACK] Seq=1 Ack=1 Win=256 Len=34
610	23.6998900	192.168.1.69	192.168.1.65	TCP	88	[TCP Retransmission] 3711->5555 [PSH, ACK] Seq=1 Ack=1 Win=256 Len=34
612	23.7641660	192.168.1.69	192.168.1.65	TCP	54	3721->5555 [RST, ACK] Seq=35 Ack=1 Win=0 Len=0
613	34.7620640	192.168.1.69	192.168.1.65	TCP	66	5555->5555 [SYN] Seq=0 Win=1460 Len=0 MSS=1460 WS=256 SACK_PERM=1
823	34.7641300	192.168.1.65	192.168.1.69	TCP	66	5555->3830 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
824	34.7641680	192.168.1.69	192.168.1.65	TCP	54	3830->5555 [ACK] Seq=1 Ack=1 Win=65536 Len=0
825	34.7649670	192.168.1.69	192.168.1.65	TCP	58	3830->5555 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=4
826	34.7668670	192.168.1.65	192.168.1.69	TCP	60	5555->3830 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=4
827	34.8022800	192.168.1.65	192.168.1.69	TCP	60	5555->3830 [ACK] Seq=1 Ack=1 Win=65536 Len=0
828	34.8103030	192.168.1.69	192.168.1.65	TCP	74	5555->5555 [PSH, ACK] Seq=5 Ack=5 Win=65536 Len=21
829	34.8121300	192.168.1.65	192.168.1.69	TCP	75	5555->3830 [PSH, ACK] Seq=5 Ack=26 Win=65536 Len=21
830	34.8813100	192.168.1.69	192.168.1.65	TCP	54	3830->5555 [ACK] Seq=26 Ack=26 Win=65536 Len=0

BotMaster: Hello Proxy !!! :D -> C&C Server

¶ Raw data: 0000 68 5d 43 50 25 e9 ac 22 0b 7f 5d e0 08 00 45 00 h|CPK...E.  
0010 00 3d 15 ac 40 00 80 06 01 38 c0 a8 01 45 c0 a8 ..S...8...E.  
0020 01 41 0e f6 15 b3 ef 00 11 37 af 03 a8 fa 50 18 ..A...o...P.  
0030 01 00 27 59 00 00 74 00 12 48 65 6c 6f 20 70 ...v...t...Hello p  
0040 72 6f 78 20 3a 44 20 11 21 21 Proxy ID !!!

2 interfaces: <live capture in progress> File... (Packets: 1153) Displayed: 12 (1.0%)

Wireshark 1.12.4 (v1.12.4-0-g750e543 from master-1.12.4)

Filter: `ip.addr == 192.168.1.69 and tcp.port == 5555`

No.	Time	Source	Destination	Protocol	Length	Info
607	23.3829670	192.168.1.69	192.168.1.65	TCP	88	3711->5555 [PSH, ACK] Seq=1 Ack=1 Win=256 Len=34
610	23.6998900	192.168.1.69	192.168.1.65	TCP	88	[TCP Retransmission] 3711->5555 [PSH, ACK] Seq=1 Ack=1 Win=256 Len=34
612	23.7641660	192.168.1.69	192.168.1.65	TCP	54	3721->5555 [RST, ACK] Seq=35 Ack=1 Win=0 Len=0
613	34.7620640	192.168.1.69	192.168.1.65	TCP	66	5555->5555 [SYN] Seq=0 Win=1460 Len=0 MSS=1460 WS=256 SACK_PERM=1
823	34.7641300	192.168.1.65	192.168.1.69	TCP	66	5555->3830 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
824	34.7641680	192.168.1.69	192.168.1.65	TCP	54	3830->5555 [ACK] Seq=1 Ack=1 Win=65536 Len=0
825	34.7649670	192.168.1.69	192.168.1.65	TCP	58	3830->5555 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=4
826	34.7668670	192.168.1.65	192.168.1.69	TCP	60	5555->3830 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=4
827	34.8022800	192.168.1.65	192.168.1.69	TCP	60	5555->3830 [ACK] Seq=1 Ack=1 Win=65536 Len=0
828	34.8103030	192.168.1.69	192.168.1.65	TCP	75	3830->5555 [PSH, ACK] Seq=5 Ack=5 Win=65536 Len=21
829	34.8121300	192.168.1.65	192.168.1.69	TCP	75	5555->3830 [PSH, ACK] Seq=5 Ack=26 Win=65536 Len=21
830	34.8813100	192.168.1.69	192.168.1.65	TCP	54	3830->5555 [ACK] Seq=26 Ack=26 Win=65536 Len=0
2967	213.0194000	192.168.1.69	192.168.1.65	TCP	88	5555->5555 [PSH, ACK] Seq=26 Ack=26 Win=65536 Len=34
2968	213.0212144	192.168.1.65	192.168.1.69	TCP	74	5555->3830 [PSH, ACK] Seq=26 Ack=60 Win=65536 Len=20
2969	213.0240714	192.168.1.65	192.168.1.69	TCP	88	5555->3830 [FIN, PSH, ACK] Seq=46 Ack=60 Win=65536 Len=34
2970	213.0243000	192.168.1.69	192.168.1.65	TCP	54	3830->5555 [ACK] Seq=60 Ack=81 Win=65536 Len=0
2971	213.0245196	192.168.1.69	192.168.1.65	TCP	54	3830->5555 [FIN, ACK] Seq=60 Ack=81 Win=65536 Len=0
2972	213.0272772	192.168.1.65	192.168.1.69	TCP	60	5555->3830 [ACK] Seq=81 Ack=61 Win=65536 Len=0

BotMaster: STOP\_CONNECTION\_WITH\_THE\_SERVER -> C&C Server

Ο BotMaster στέλνει μήνυμα για να σταματήσει την σύνδεσή του με το C&C.

¶ Raw data: 0000 68 5d 43 50 25 e9 ac 22 0b 7f 5d e0 08 00 45 00 h|CPK...E.  
0010 00 4a 16 7b 40 00 80 06 60 5c c0 a8 01 45 c0 a8 ..J.(...E.  
0020 01 41 0e f6 15 b3 ef 00 11 4c af 03 a8 6f 50 18 ..A...o...P.  
0030 01 00 09 00 00 00 74 00 1f 53 54 4f 50 5f 43 4f .....t...STOP\_CO  
0040 4e 4e 45 43 34 49 4f 4e 5f 47 49 34 48 5f 34 48 E\_SERVER  
0050 45 5f 53 45 52 56 45 52

2 interfaces: <live capture in progress> File... (Packets: 3133) Displayed: 18 (0.6%)



Capturing from 2 interfaces [Winbox 1.12.0 - v1.12.0-g156a541 from master-1.12]

Filter: ip.addr == 192.168.1.69 and tcp.port == 5555

No.	Time	Source	Destination	Protocol	Length	Info
607	25.3630670	192.168.1.69	192.168.1.65	TCP	88	3711->5555 [PSH, ACK] Seq=1 Ack=1 wln=258 Len=34
610	25.3700000	192.168.1.69	192.168.1.65	TCP	54	3711->5555 [ACK] Seq=1 Ack=1 wln=258 Len=0
612	25.7641660	192.168.1.69	192.168.1.65	TCP	54	3711->5555 [RST, ACK] Seq=33 Ack=1 wln=0 Len=0
822	34.7070000	192.168.1.69	192.168.1.65	TCP	66	5555->5830 [SYN, ACK] Seq=0 Ack=1 wln=8192 Len=0 MSS=1460 w=256 SACK_PERM=1
823	34.7641300	192.168.1.65	192.168.1.69	TCP	66	5555->5830 [SYN, ACK] Seq=0 Ack=1 wln=8192 Len=0 MSS=1460 w=256 SACK_PERM=1
824	34.7641680	192.168.1.69	192.168.1.65	TCP	54	3830->5555 [ACK] Seq=1 Ack=1 wln=65536 Len=0
825	34.7649670	192.168.1.69	192.168.1.65	TCP	58	3830->5555 [PSH, ACK] Seq=1 Ack=1 wln=65536 Len=4
826	34.7668870	192.168.1.65	192.168.1.69	TCP	60	5555->3830 [PSH, ACK] Seq=1 Ack=1 wln=65536 Len=4
827	34.8102780	192.168.1.65	192.168.1.69	TCP	60	5555->3830 [ACK] Seq=5 Ack=5 wln=65536 Len=0
828	34.8103030	192.168.1.69	192.168.1.65	TCP	75	3830->5555 [PSH, ACK] Seq=5 Ack=5 wln=65536 Len=21
829	34.8121390	192.168.1.65	192.168.1.69	TCP	75	5555->3830 [PSH, ACK] Seq=5 Ack=26 wln=65536 Len=21
830	34.8613100	192.168.1.69	192.168.1.65	TCP	54	3830->5555 [ACK] Seq=26 Ack=26 wln=65536 Len=0
2967	213.0194000	192.168.1.69	192.168.1.65	TCP	88	3830->5555 [PSH, ACK] Seq=26 Ack=26 wln=65536 Len=34
2968	213.0221110	192.168.1.65	192.168.1.69	TCP	74	5555->3830 [PSH, ACK] Seq=26 Ack=60 wln=5536 Len=0
2969	213.0240070	192.168.1.65	192.168.1.69	TCP	88	5555->3830 [PSH, ACK] Seq=26 Ack=60 wln=65536 Len=34
2970	213.0241000	192.168.1.69	192.168.1.65	TCP	54	3830->5555 [ACK] Seq=60 Ack=81 wln=65536 Len=0
2971	213.0242560	192.168.1.69	192.168.1.65	TCP	54	3830->5555 [FIN, ACK] Seq=60 Ack=81 wln=65536 Len=0
2972	213.0272770	192.168.1.65	192.168.1.69	TCP	60	5555->3830 [ACK] Seq=81 Ack=61 wln=65536 Len=0

C&C Server> Goodbye my Master --> BotMaster

O C&C Server στέλνει μήνυμα Goodbye στον BotMaster και μετά κλείνει το connection.

Frame 2968: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 1  
 Ethernet II, Src: IntelCor\_50:25:e9 (68:5d:43:50:25:e9), Dst: Asustek\_7f:5d:e0 (ac:22:0b:7f:5d:e0)  
 Internet Protocol Version 4, Src: 192.168.1.65 (192.168.1.65), Dst: 192.168.1.69 (192.168.1.69)

```

0000  ac 22 0b 7f 5d e9 68 5d 43 50 25 e9 08 00 45 00  ...[..] CPH...E.
0010  00 3c 1c a8 40 80 80 9a 39 c9 a8 01 41 c3 a8  ...R...P...
0020  01 45 15 b3 de f6 a7 03 a3 0f 8f 80 11 8e 50 18  ...E.....G..RP.
0030  01 00 9e b2 00 00 74 00 11 47 6f 64 62 79 65  ...t.....Goodbye
0040  20 6d 79 20 4d 61 73 74 65 72                    My Mast er
  
```

2 interfaces: <live capture in progress> File... Packets: 4298, Displayed: 18 (0.4%) Profile: Default

Όπως παρατηρούμε τα δεδομένα που ανταλλάσσονται δεν είναι κρυπτογραφημένα και κάποιος κακόβουλος χρήστης μπορεί να μας υποκλέψει τις συνομιλίες μας.

- Secure Connection via IPsec χρησιμοποιώντας unsecure projects (απλό socket χωρίς openssl με χρήση της ca) :

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help						
Filter: esp Expression... Clear Apply Save						
No.	Time	Source	Destination	Protocol	Length	Info
71	5.35766000	192.168.1.69	192.168.1.65	ESP	102	ESP (SPI=0x273f8ec6)
72	5.35978000	192.168.1.65	192.168.1.69	ESP	102	ESP (SPI=0x85458972)
73	5.35986000	192.168.1.69	192.168.1.65	ESP	86	ESP (SPI=0x273f8ec6)
74	5.36247000	192.168.1.69	192.168.1.65	ESP	94	ESP (SPI=0x273f8ec6)
75	5.36654800	192.168.1.65	192.168.1.69	ESP	94	ESP (SPI=0x85458972)
76	5.37011900	192.168.1.69	192.168.1.65	ESP	110	ESP (SPI=0x273f8ec6)
77	5.38155500	192.168.1.65	192.168.1.69	ESP	110	ESP (SPI=0x85458972)
78	5.43776500	192.168.1.69	192.168.1.65	ESP	86	ESP (SPI=0x273f8ec6)

+ Frame 74: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 1	
+ Ethernet II, Src: AsustekC_7f:5d:e0 (ac:22:0b:7f:5d:e0), Dst: IntelCor_50:25:e9 (68:5d:43:50:25:e9)	
+ Internet Protocol Version 4, Src: 192.168.1.69 (192.168.1.69), Dst: 192.168.1.65 (192.168.1.65)	
0000	68 5d 43 50 25 e9 ac 22 0b 7f 5d e0 08 00 45 00 h]CP%.." ..]...E.
0010	00 50 6f 63 40 00 80 32 07 42 c0 a8 01 45 c0 a8 .Poc@..2 .B...E..
0020	01 41 27 3f 8e c6 00 00 00 11 83 97 d3 e5 94 8b .A'?..... .....

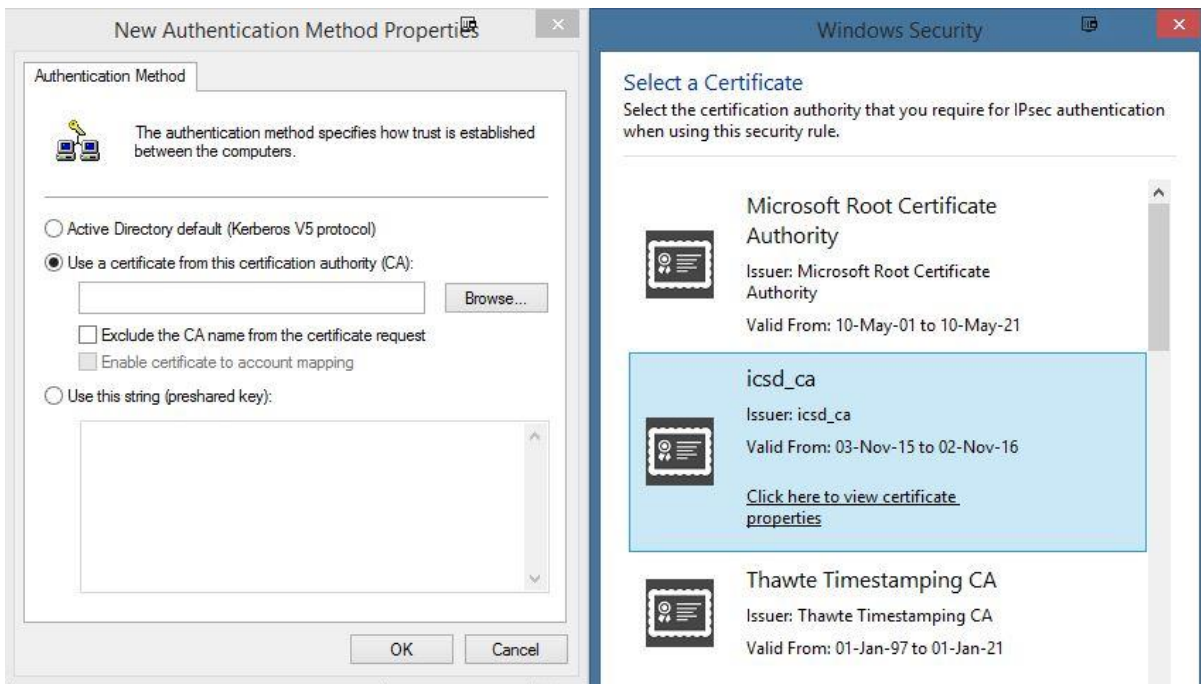
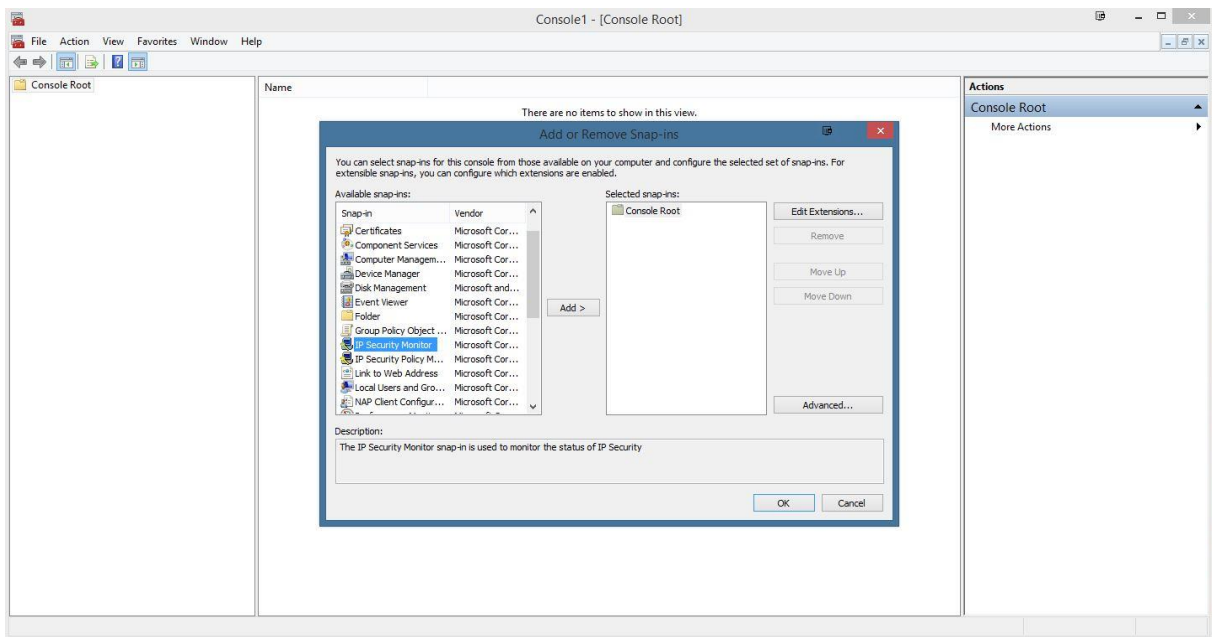
Filter: esp

Ο BotMaster κάνει login στον C&C Server. Όπως φαίνεται η συνομιλία είναι πλήρως κρυπτογραφημένη από το IPsec.

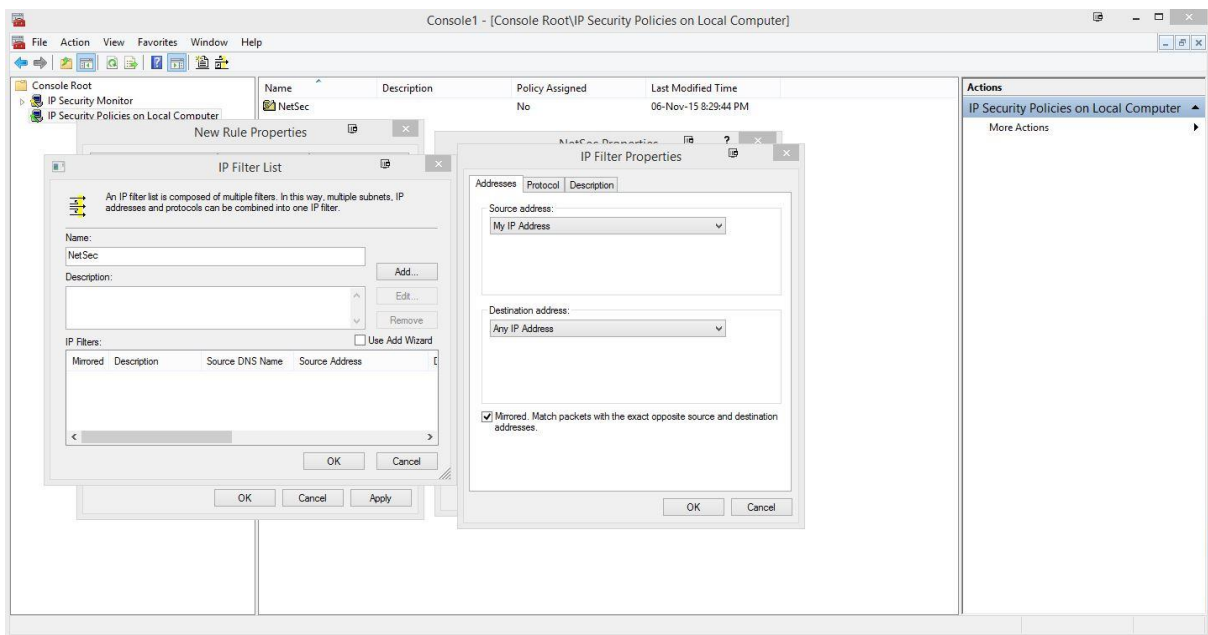
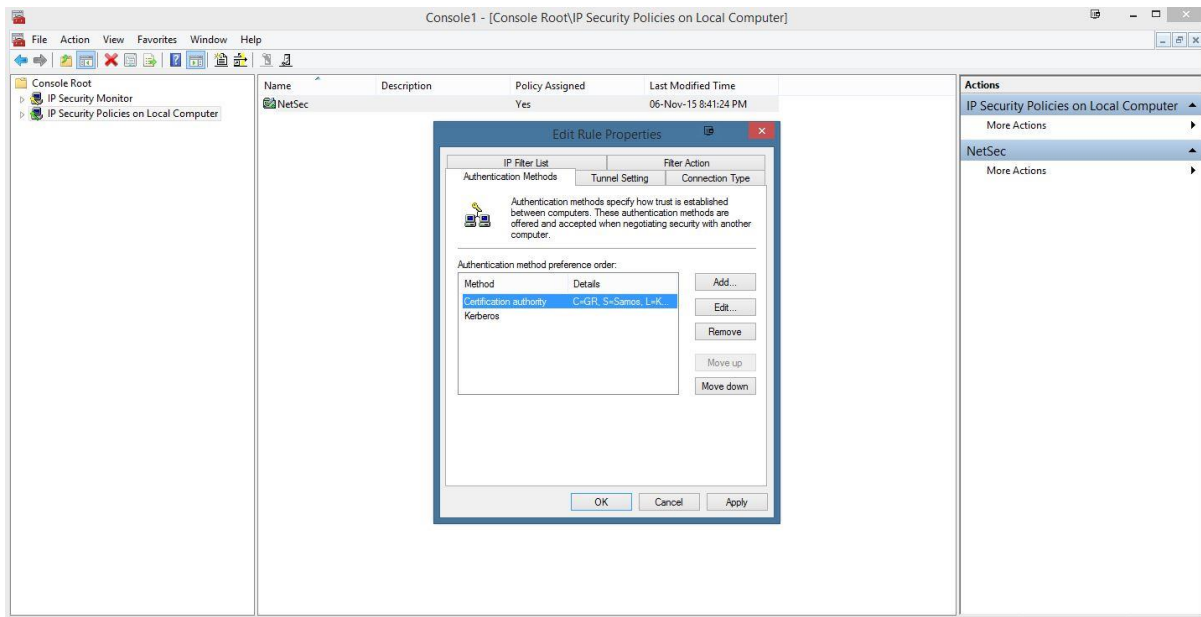
Ομοίως για όλες τις συνομιλίες με BotMaster & Bots με των C&C Server.

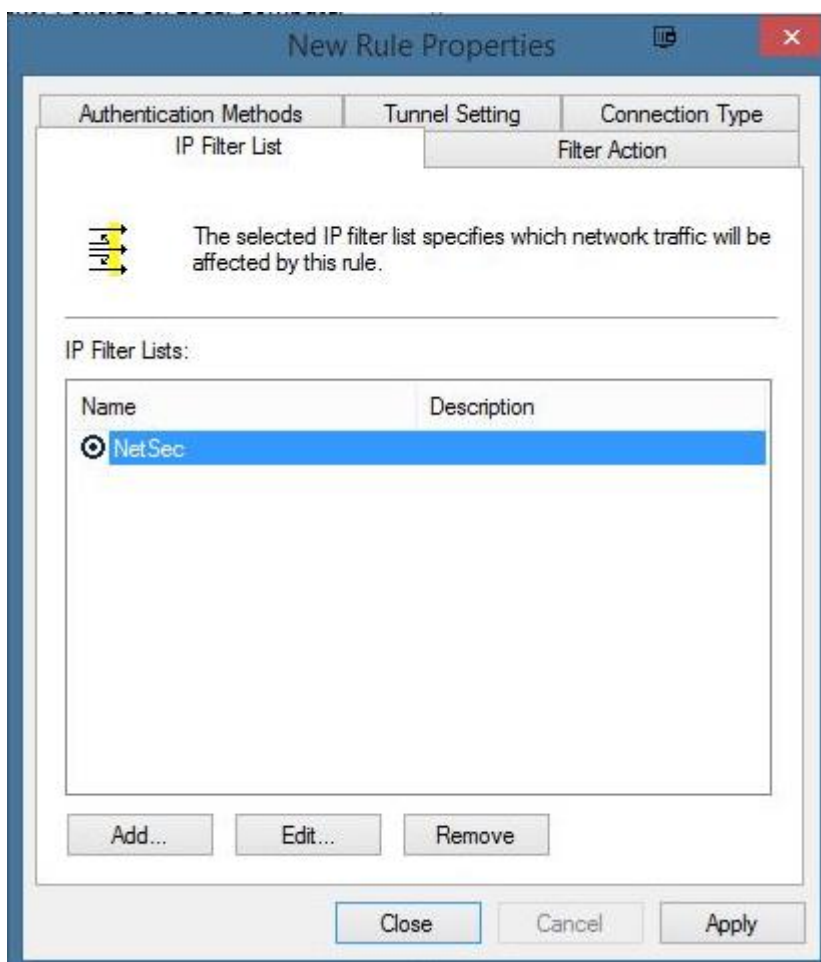
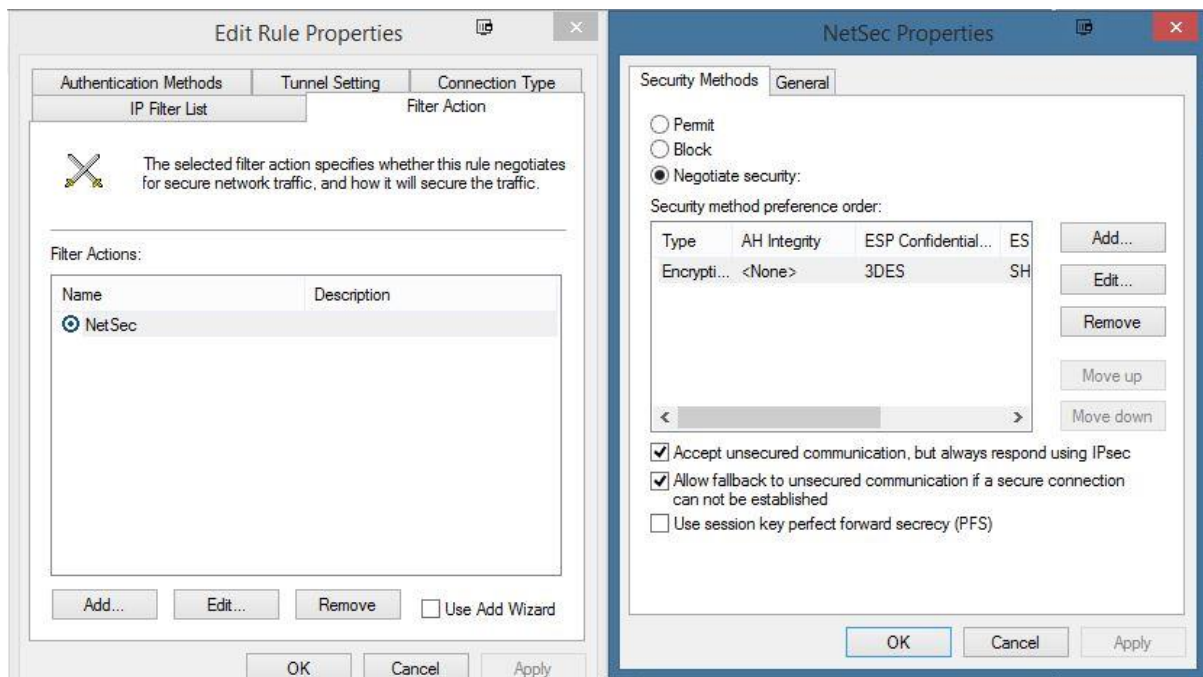
## 8)Περιγραφή των τροποποιήσεων του συστήματος για την χρήση του πρωτοκόλλου IPsec

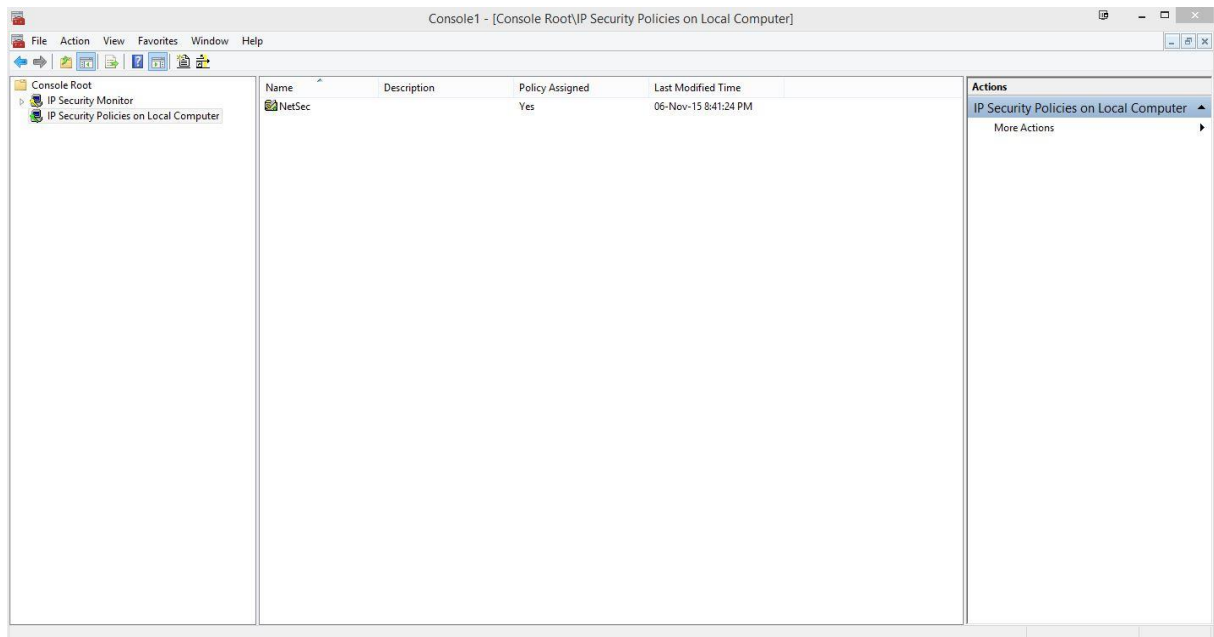
- Open mmc and add/remove snap-in (IP Security Monitor & IP Security Policies on Local Computer.



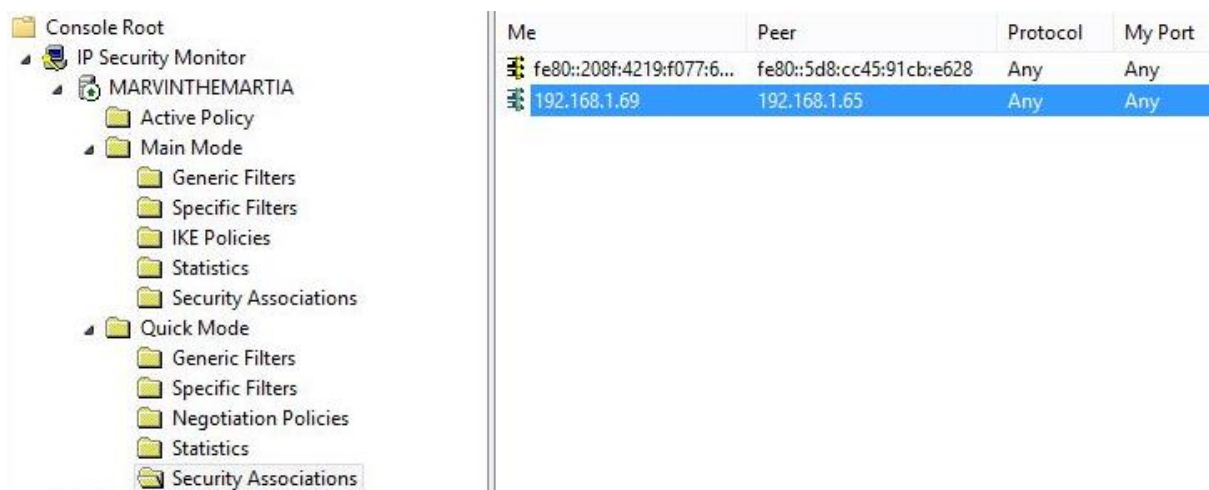








Εδώ βλέπουμε πως μέσα στον δίαυλου που δημιουργήσαμε επικοινωνούν δύο ip. Με αυτό τον τρόπο κάνουμε επαλήθευση πως το connection που έχουμε δημιουργήσει είναι secure. Το ίδιο παρατηρούμε και από το WireShark.



Το IPsec το ρυθμίσαμε στην ip 192.168.1.69 (Desktop) ομοίως ρυθμίσαμε το IPsec και στην ip 192.168.1.65 (Laptop). Το IPsec βλέπουμε πως δουλεύει στο 7) μέρος της εργασίας κάνοντας χρήση των unsecure projects.

## 9) Σύγκριση των τεχνολογιών TLS/SSL και IPv6/IPsec

### - SSL:

Το SSL (Secure Socket Layer), είναι ένα γενικού σκοπού πρωτόκολλο για την αποστολή κρυπτογραφημένης πληροφορίας μέσω του Internet. Αναπτύχθηκε από την Netscape και έγινε προσιτό από το ευρύ κοινό από τον web browser και server της Netscape.

Το SSL είναι ένα επίπεδο που υπάρχει ανάμεσα στη σειρά του TCP/IP πρωτοκόλλου και στο επίπεδο εφαρμογής. Ενώ το κανονικό TCP/IP πρωτόκολλο απλά στέλνει ένα ανώνυμο free-error ρεύμα πληροφοριών ανάμεσα σε δυο υπολογιστές, το SSL προσθέτει πολυάριθμες λειτουργίες σε αυτό το ρεύμα, περιλαμβάνοντας:

- ✓ Απόδειξη γνησιότητας και απαγόρευσης απάρνησης του server, χρησιμοποιώντας ψηφιακές υπογραφές
- ✓ Απόδειξη γνησιότητας και απαγόρευσης απάρνησης του client, χρησιμοποιώντας ψηφιακές υπογραφές
- ✓ Εμπιστοσύνη δεδομένων μέσω της χρήσης της κρυπτογραφίας
- ✓ Ακεραιότητα δεδομένων μέσω της χρήσης κωδικών απόδειξης γνησιότητας μηνυμάτων

Η κρυπτογραφία είναι ένας γρήγορα αναπτυσσόμενος τομέας, και τα κρυπτογραφικά πρωτόκολλα δεν δουλεύουν αν τα δυο μέρη της επικοινωνίας δεν χρησιμοποιούν τους ίδιους αλγόριθμους. Για το λόγο αυτό το SSL είναι ένα επεκτάσιμο και ένα πρωτόκολλο που μπορεί να προσαρμοστεί εύκολα. Όταν ένα πρόγραμμα που χρησιμοποιεί SSL προσπαθεί να επικοινωνήσει με ένα άλλο, τότε τα δυο προγράμματα ηλεκτρονικά συγκρίνουν στοιχεία και καθορίζουν ποιος είναι ο δυνατότερος κρυπτογραφικός αλγόριθμος που διαθέτουν από κοινού.

Το SSL κάνει εκτεταμένη χρήση των πιστοποιητικών δημόσιου κλειδιού για την απόδειξη γνησιότητας τόσο του client όσο και του server στις SSL συναλλαγές. Το SSL κάνει χρήση των X. 509v3 πιστοποιητικών για τον έλεγχο των RSA ζεύγος κλειδιών, και ένα τροποποιημένο X.509 πιστοποιητικό για τον έλεγχο δημόσιων κλειδιών που χρησιμοποιούνται από το US Department of Defense Fortezza/DMS πρωτόκολλο ανταλλαγής κλειδιών.

Το 1995, το IETF έκανε την πρώτη σκέψη για την υιοθεσία του SSL σαν μέρος ενός νέου προτύπου το Transport layer Security (TLS). Ένα σχέδιο πρωτόκολλου δημοσιεύτηκε στις 6 Μαρτίου 1997.

Το TLS είναι παρόμοιο με το SSL 3.0 με λίγες σημαντικές αλλαγές. Αντί της χρήσης του MD5, το TLS χρησιμοποιεί την HMAC ασφαλή συνάρτηση αποσύνθεσης κλειδιών. Το TLS επίσης έχει λίγο διαφορετικό τρόπο κρυπτογράφησης από το SSL 3.0.

## -IPv6/IPsec:

Το IPsec είναι ένα κρυπτογραφικό πρωτόκολλο σχεδιασμένο από το Internet Engineering Task Force για την παροχή εμπιστευτικότητας για τα πακέτα που “ταξιδεύουν” μέσα στο Internet. Το IPsec δουλεύει με το IPv4, την έκδοση του IP standard που χρησιμοποιείται σήμερα στο Internet. Το IPv6, είναι η επόμενη “γενιά” IP, που περιλαμβάνει το IPsec.

Το IPsec δεν προσφέρεται για την ακεραιότητα, την αναγνώριση ταυτότητας, ή την απαγόρευση απάρνησης, αλλά αφήνει αυτά τα

α χαρακτηριστικά για τα άλλα πρωτόκολλα. Πρόσφατα, η κύρια χρήση του IPsec φαίνεται να είναι ένα πρωτόκολλο για την δημιουργία εικονικών προσωπικών δικτύων (Virtual Private Networks - VPNs) μέσω του Internet. Αλλά το IPsec έχει την ικανότητα να παρέχει αναγνώριση ταυτότητας, ακεραιότητα, και προαιρετικά την εμπιστοσύνη των δεδομένων για όλες τις επικοινωνίες που παίρνουν μέρος πάνω στο Internet, έχοντας ευρέως διαδεδομένες εφαρμογές του πρωτοκόλλου και επίσης την άδεια χρήση αυτών από τις κυβερνήσεις.

## 10) Βιβλιογραφία

- <https://www.digicert.com/ssl-support/jks-import-export-java.htm>
- <https://www.digicert.com/code-signing/java-code-signing-guide.htm>
- <https://docs.oracle.com/cd/E19509-01/820-3503/ggfen/index.html>
- <https://www.youtube.com/watch?v=LHUbQtUeQ0o>
- <http://stackoverflow.com/questions/13874387/create-app-with-sslsocket-java>
- <http://download.java.net/jdk7/archive/b123/docs/api/javax/net/ssl/SSLSocket.html>
- [http://www.java2s.com/Tutorial/Java/0490\\_Security/SSLClientSession.htm](http://www.java2s.com/Tutorial/Java/0490_Security/SSLClientSession.htm)
- <https://www.youtube.com/watch?v=y4JNaegGtaY>
- <https://en.wikipedia.org/wiki/IPsec>
- [http://en.wikipedia.org/wiki/Secure\\_Sockets\\_Layer](http://en.wikipedia.org/wiki/Secure_Sockets_Layer)
- [https://en.wikipedia.org/wiki/Transport\\_Layer\\_Security](https://en.wikipedia.org/wiki/Transport_Layer_Security)

Προγράμματα/Λειτουργικά Συστήματα που χρησιμοποιήσαμε:

Oracle Netbeans IDE

Oracle VM

WireShark

Python IDLE

Jetbrains Pycharm IDE

Windows 8.1 pro x64 (1xDesktop & 1xLaptop)

Windows 8.1 pro x86 (1xVM)