



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΜΑΘΗΜΑ:

Ασφάλεια Δικτύων Υπολογιστών & Τεχνολογίες Προστασίας της
Ιδιωτικότητας

2^η Ομαδική

Θέμα

Δημιουργία ασφαλούς διαύλου διαχείρισης botnet (C&C) με
αξιοποίηση Tor και χρήση πρωτοκόλλου διασφάλισης
εμπιστευτικότητας και ακεραιότητας

Πέππας Κωνσταντίνος, icsd11134
Σωτηρέλης Χρήστος, icsd12182
Χαϊκάλης Νικόλαος, icsd12200

29/11/2015

Περιεχόμενα

1. Εκτελέσιμα προγράμματα (project Netbeans κτλ).
2. Τεκμηρίωση προγραμμάτων και επεξήγηση τυχόν δικών σας παραδοχών.
3. Στιγμιότυπα εκτέλεσης προγράμματος (screenshots).
4. Περιγραφή και τρόπος δημιουργίας πιστοποιητικών.
5. Ψηφιακά Πιστοποιητικά.
6. Ανάλυση των αποτελεσμάτων χρήσης του εργαλείου σύλληψης πακέτων (sniffer) και ενδεικτικά στιγμιότυπα εκτέλεσης (screenshots). Επεξήγηση των φίλτρων που χρησιμοποιήθηκαν.
7. Θεωρητικές Ερωτήσεις.
 - 1) Πιστεύετε πως το custom πρωτόκολλο που δημιουργήσατε στο κανάλι C&C Server – Bots διασφαλίζει από Replay Attacks; Αν όχι, τροποποιήστε το πρωτόκολλο που δημιουργήσατε για τη διασφάλιση από τέτοιου είδους επιθέσεις. Υλοποιήστε το αντίμετρο που σκεφτήκατε.
 - 2) Η εμπιστευτικότητα και ακεραιότητα της πληροφορίας που μεταδίδεται στο κανάλι BotMaster – C&C Server διασφαλίζεται από τους ενδιάμεσους κόμβους του TOR και του RP; Αν ναι, με ποιόν τρόπο;
 - 3) Το πρωτόκολλο που έχουμε δημιουργήσει είναι ευάλωτο σε clogging attacks (επιθέσεις πνιγμού). Στις επιθέσεις αυτές, ένας επιτιθέμενος που υποδύεται μία άλλη μηχανή (IP Spoofing) αποσκοπεί σε επίθεση Άρνησης Εξυπηρέτησης (DoS) δημιουργώντας “half-open sessions”, δηλαδή sessions που δεν καταλήγουν ποτέ στην ανταλλαγή πληροφορίας. Αφού συμβουλευτείτε τις διαφάνειες του μαθήματος, προτείνετε ένα τρόπο για την αντιμετώπιση αυτού του είδους των επιθέσεων.
8. Βιβλιογραφία

1) Εκτελέσιμα προγράμματα (project Netbeans κτλ)

Ο κώδικας των projects εμπεριέχεται στο φάκελο με όνομα «java projects».
C&C Server είναι server και για τον BotMaster και για τα Bots.

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.

Για την υλοποίηση της εργασίας χρησιμοποιήσαμε:




1 x Desktop με Windows 10 Pro x64

1 x VMware με Windows 10 Pro x86


















Ακόμη για την δημιουργία AES 256 λόγω του ότι το security της java από default έχει μέχρι 128 bit χρησιμοποιήσαμε τα παρακάτω .jar τα οποία τα αντιγράψαμε στον path:

C:\Program Files\Java\jdk1.8.0_65\jre\lib\security

Τα αρχεία που κατεβάσαμε είναι τα εξής:

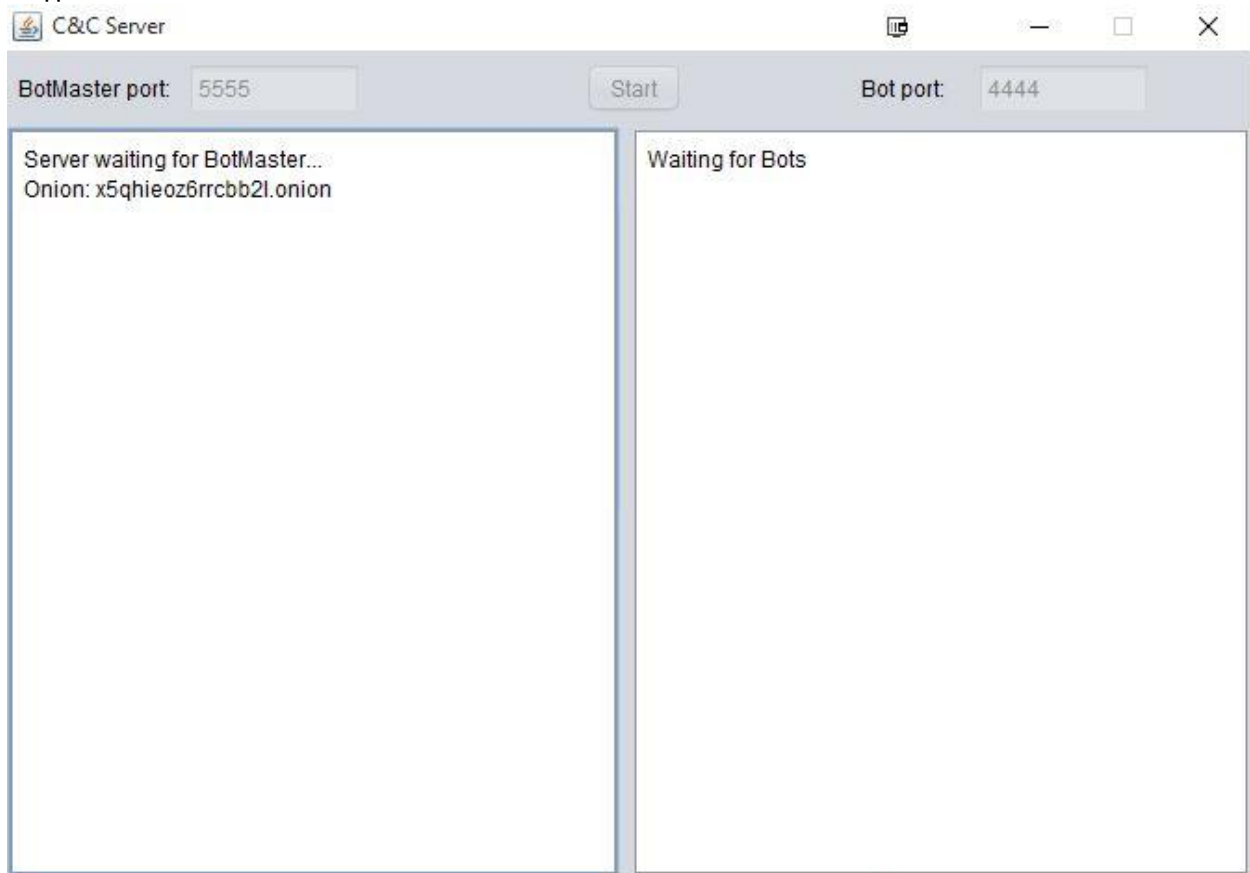
 local_policy	20-Dec-13 10:54 AM	Executable Jar File	3 KB
 README	20-Dec-13 10:54 AM	Text Document	8 KB
 US_export_policy	20-Dec-13 10:54 AM	Executable Jar File	3 KB

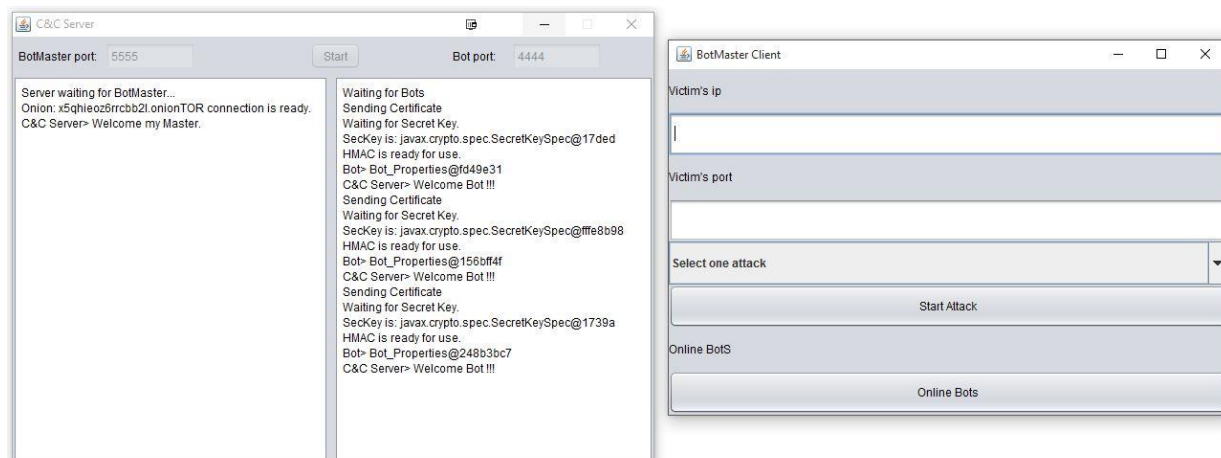
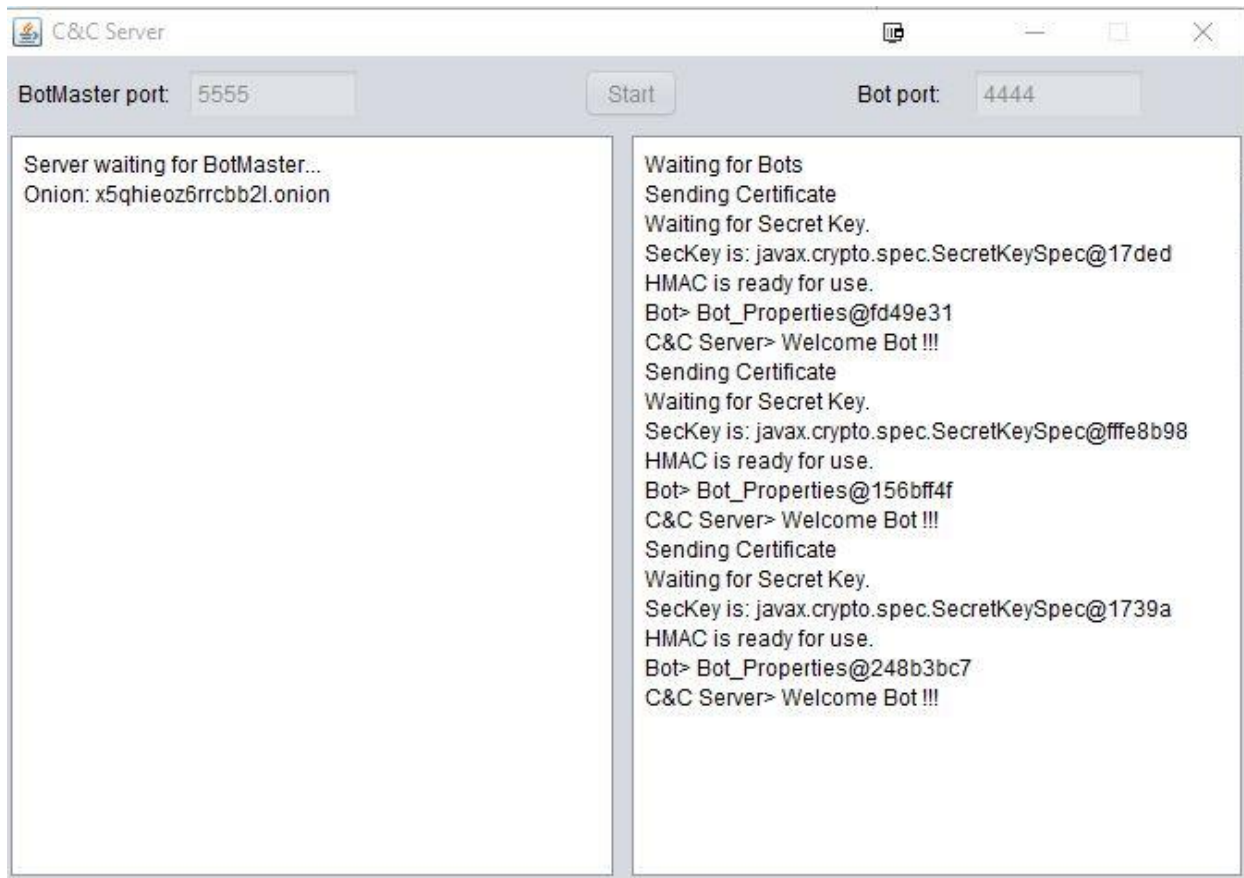
Επίσης τα Libraries που χρησιμοποιούνται από τον C&C Server (Server) και τον BotMaster(Client) για την επικοινωνία με το TOR είναι τα παρακάτω:

-  httpclient-4.4
-  httpclient-android-4.3.5.1
-  httpmime-4.4
-  netlib-0.0.4
-  netlib-0.0.4.jar.asc
-  netlib-0.0.4.pom
-  netlib-0.0.4.pom.asc
-  netlib-0.0.4-javadoc
-  netlib-0.0.4-javadoc.jar.asc
-  netlib-0.0.4-sources
-  netlib-0.0.4-sources.jar.asc
-  org.apache.httpcomponents.httpcore-4.1.2
-  sc-core-1.51.0.0
-  sc-pkix-1.51.0.0
-  sc-prov-1.51.0.0
-  silvertunnel.org_browser-0.15-beta_javawebstart
-  slf4j-api-1.7.12

4) Στιγμιότυπα εκτέλεσης προγράμματος (screenshots)

Στιγμιότυπα από τον C&C Server:



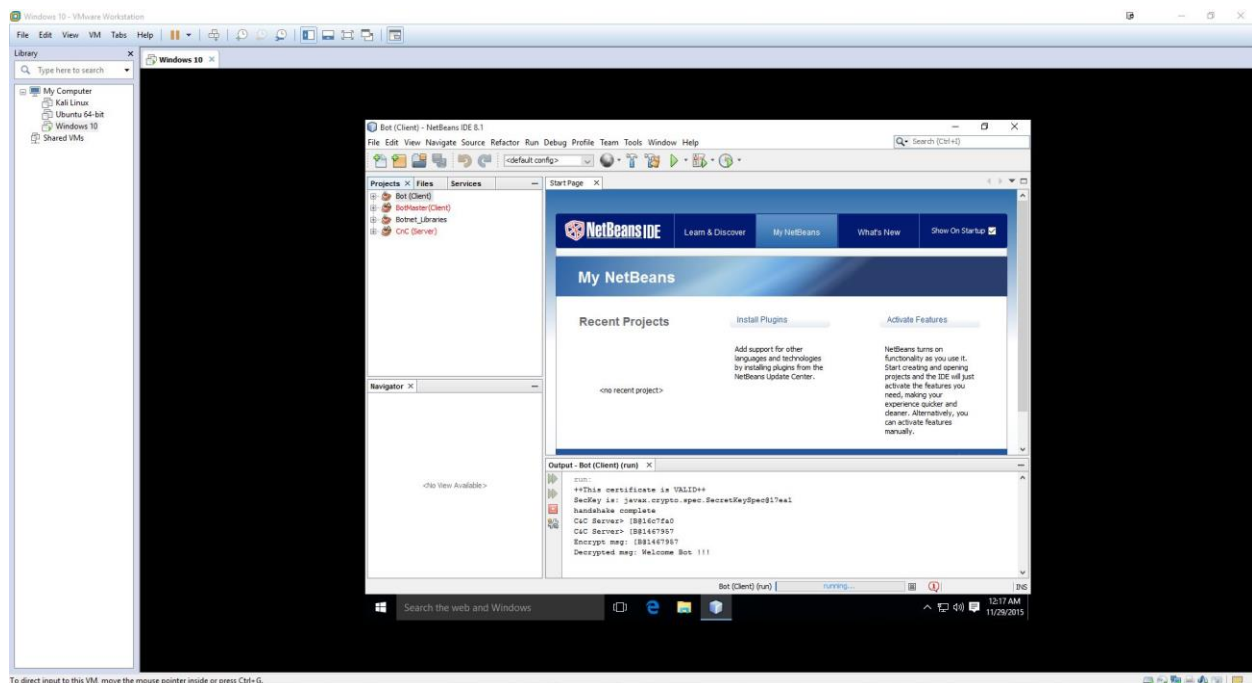


Ο server αφού κάνει Start δημιουργεί τα παρακάτω αρχεία για το TOR όπου το hostname το «τραβάει» ο BotMaster για να διαβάσει το .onion ώστε να ξεκινήσει το connection με τον server.

This PC > Documents > NetBeansProjects > Botnet TOR > CnC (Server) > onion				
	Name	Date modified	Type	Size
✦	hostname	29-Nov-15 1:01 AM	File	1 KB
✦	private_key	29-Nov-15 1:01 AM	File	1 KB
✦				
✦				

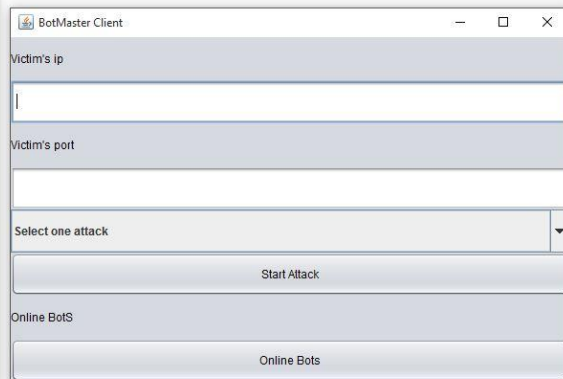
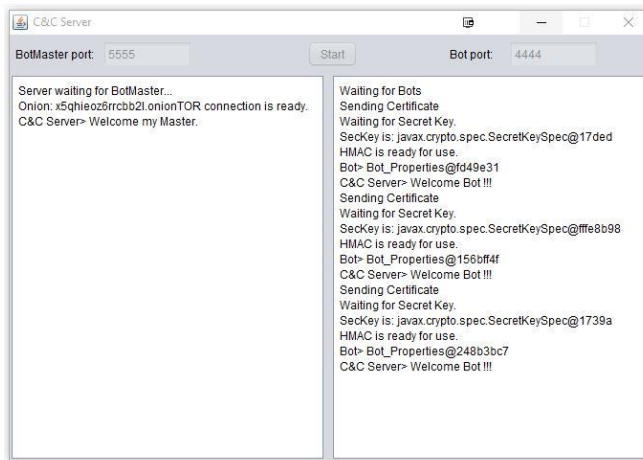
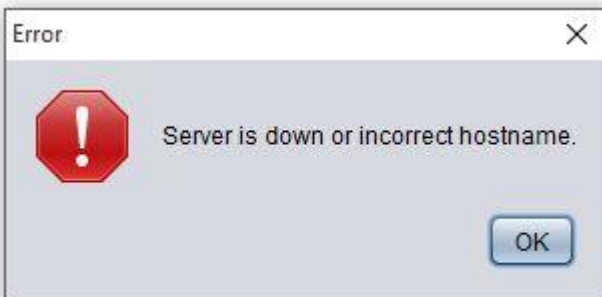
Bots:

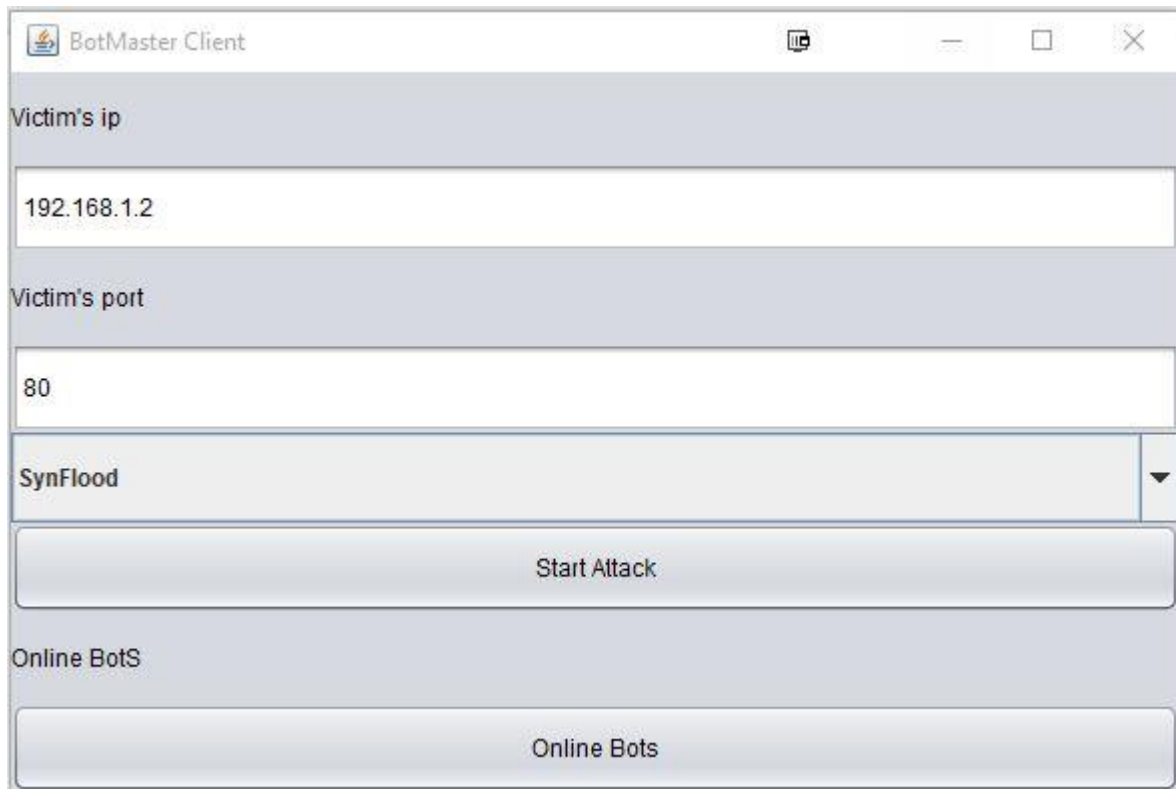
```
CnC (Server) (run) x Bot (Client) (run) x Bot (Client) (run) #2 x Bot (Client) (run) #3 x
run:
++This certificate is VALID++
SecKey is: javax.crypto.spec.SecretKeySpec@1739a
handshake complete. HMAC is ready.
C&C Server> [B@1cc3aff7
C&C Server> [B@774becc9
Encrypt msg: [B@774becc9
Decrypted msg: Welcome Bot !!!
```



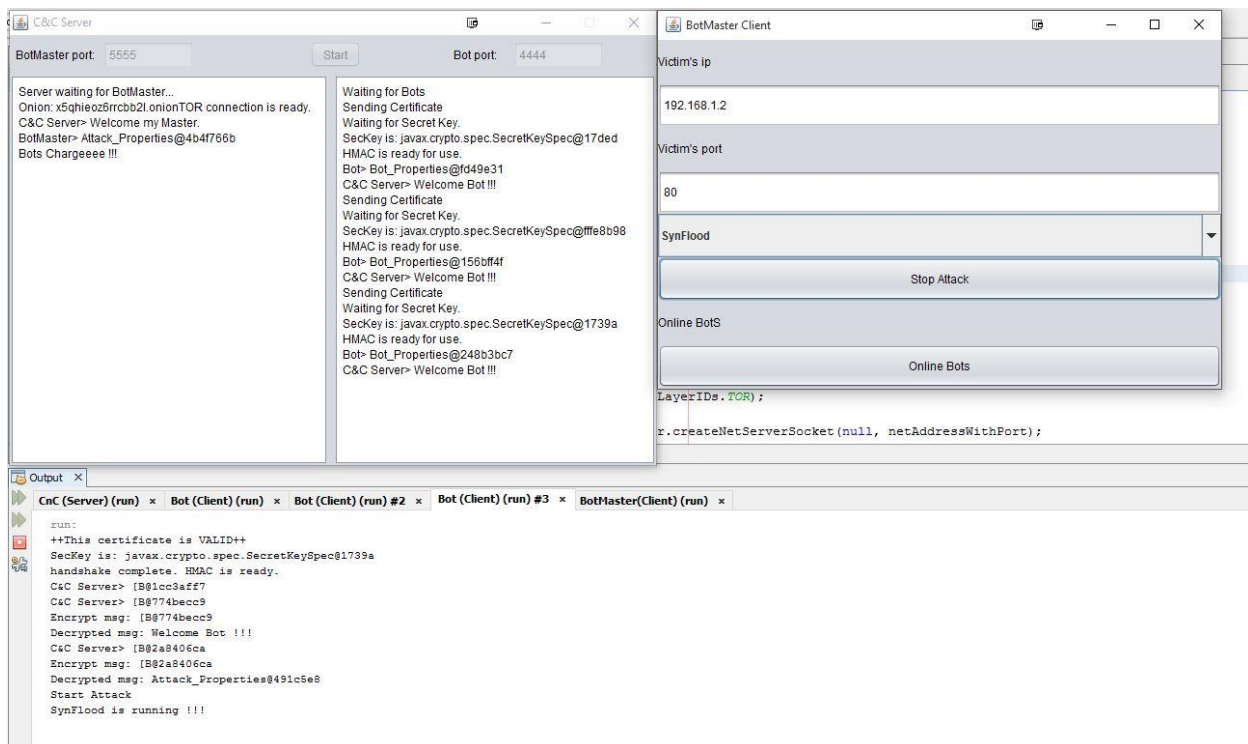
Στιγμιότυπα από τον BotMaster:

Σε περίπτωση ανύπαρκτου hostname ή ο C&C είναι κάτω.





CnC Server & BotMaster & Bots εδώ ο Master έχει δώσει εντολή για attack:



Stop Attack:

The screenshot displays two windows from a Java-based botnet framework. The 'C&C Server' window on the left shows the server's status and logs. It has a 'BotMaster port' of 5555 and a 'Bot port' of 4444. The logs indicate the server is waiting for bots, sending certificates, and receiving secret keys from three different bots. The 'BotMaster Client' window on the right shows the configuration for a specific bot. It has a 'Victim's ip' of 192.168.1.2 and a 'Victim's port' of 80. The 'SynFlood' dropdown is set to 'SynFlood'. There are buttons for 'Start Attack' and 'Online Bots'. Below these windows is an 'Output' window showing the execution of the attack. It includes the command 'run:', a certificate validation message, secret key exchange for three bots, and the execution of the 'Start Attack' and 'SynFlood' commands. The output also shows the receipt of 'STOP_ATTACK' and 'Retreat my Bots!!!!' commands from the bots.

C&C Server

BotMaster port: 5555 Start Bot port: 4444

Server waiting for BotMaster...
Onion: x5qhieoz6rrcbb2l.onionTOR connection is ready.
C&C Server> Welcome my Master.
BotMaster> Attack_Properties@4b4f766b
Bots Chargeeeee !!!
BotMaster> STOP_ATTACK
Retreat my Bots ...

Waiting for Bots
Sending Certificate
Waiting for Secret Key.
SecKey is: javax.crypto.spec.SecretKeySpec@17de
HMAC is ready for use.
Bot> Bot_Properties@fd49e31
C&C Server> Welcome Bot !!!
Sending Certificate
Waiting for Secret Key.
SecKey is: javax.crypto.spec.SecretKeySpec@ffe8b98
HMAC is ready for use.
Bot> Bot_Properties@156bff4f
C&C Server> Welcome Bot !!!
Sending Certificate
Waiting for Secret Key.
SecKey is: javax.crypto.spec.SecretKeySpec@1739a
HMAC is ready for use.
Bot> Bot_Properties@248b3bc7
C&C Server> Welcome Bot !!!

BotMaster Client

Victim's ip: 192.168.1.2
Victim's port: 80
SynFlood
Start Attack
Online BotS
Online Bots
LayerIDs.TOR);
r.createNetServerSocket(null, netAddressWithPort);

Output

run:
++This certificate is VALID++
SecKey is: javax.crypto.spec.SecretKeySpec@1739a
handshake complete. HMAC is ready.
C&C Server> [B@1cc3aff7
C&C Server> [B@774becc9
Encrypt msg: [B@774becc9
Decrypt msg: Welcome Bot !!!
C&C Server> [B@2a8406ca
Encrypt msg: [B@2a8406ca
Decrypt msg: Attack_Properties@491c5e8
Start Attack
SynFlood is running !!!
C&C Server> [B@e2931e6
Encrypt msg: [B@e2931e6
Decrypt msg: STOP_ATTACK
Retreat my Bots!!!!

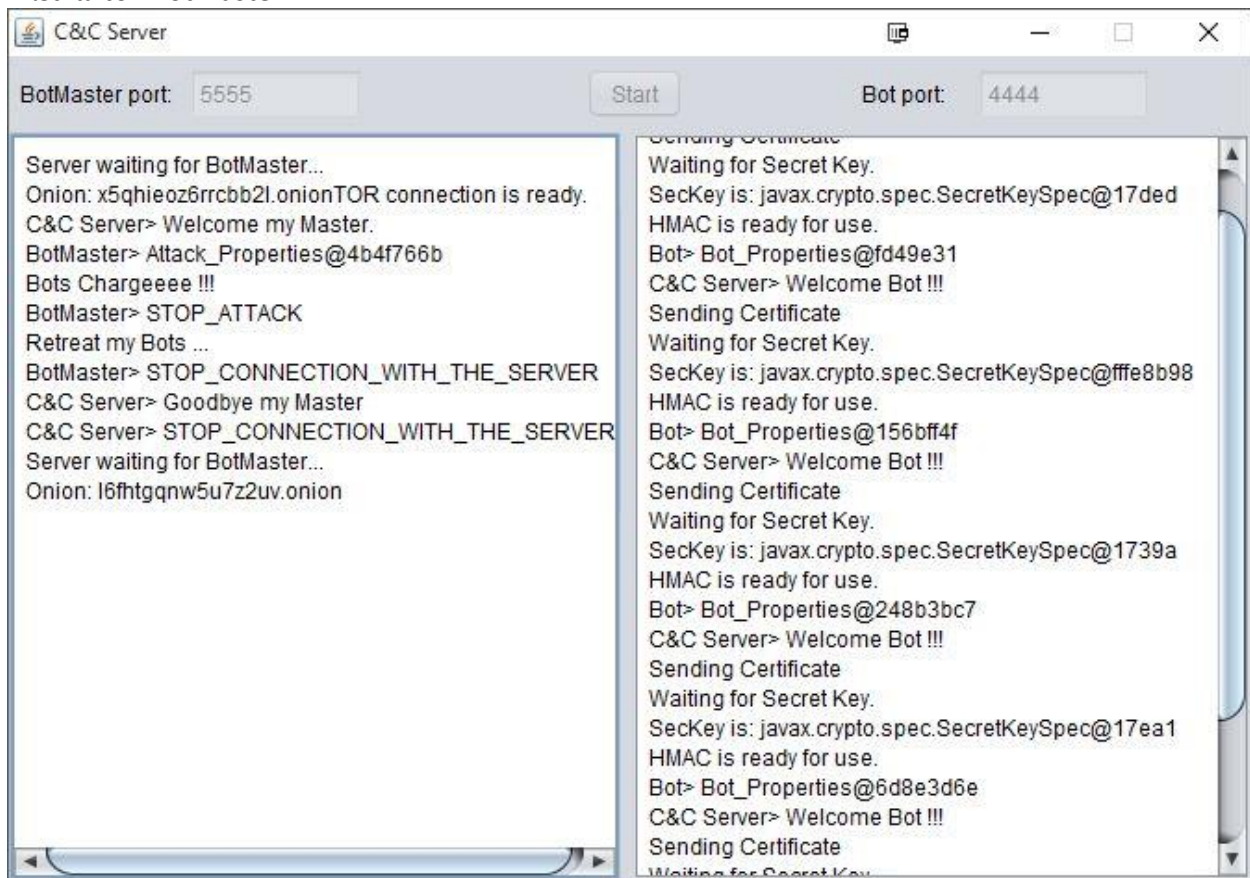
Online Bots:

Όπως βλέπουμε στο Desktop (MarvinTheMartian) υπάρχουν 3 clients (Bots) συνδεδεμένοι όπως φαίνεται και στο netbeans. Το Bot με host name DEKSTOP-2AVT004 είναι το Bot στο VMware.

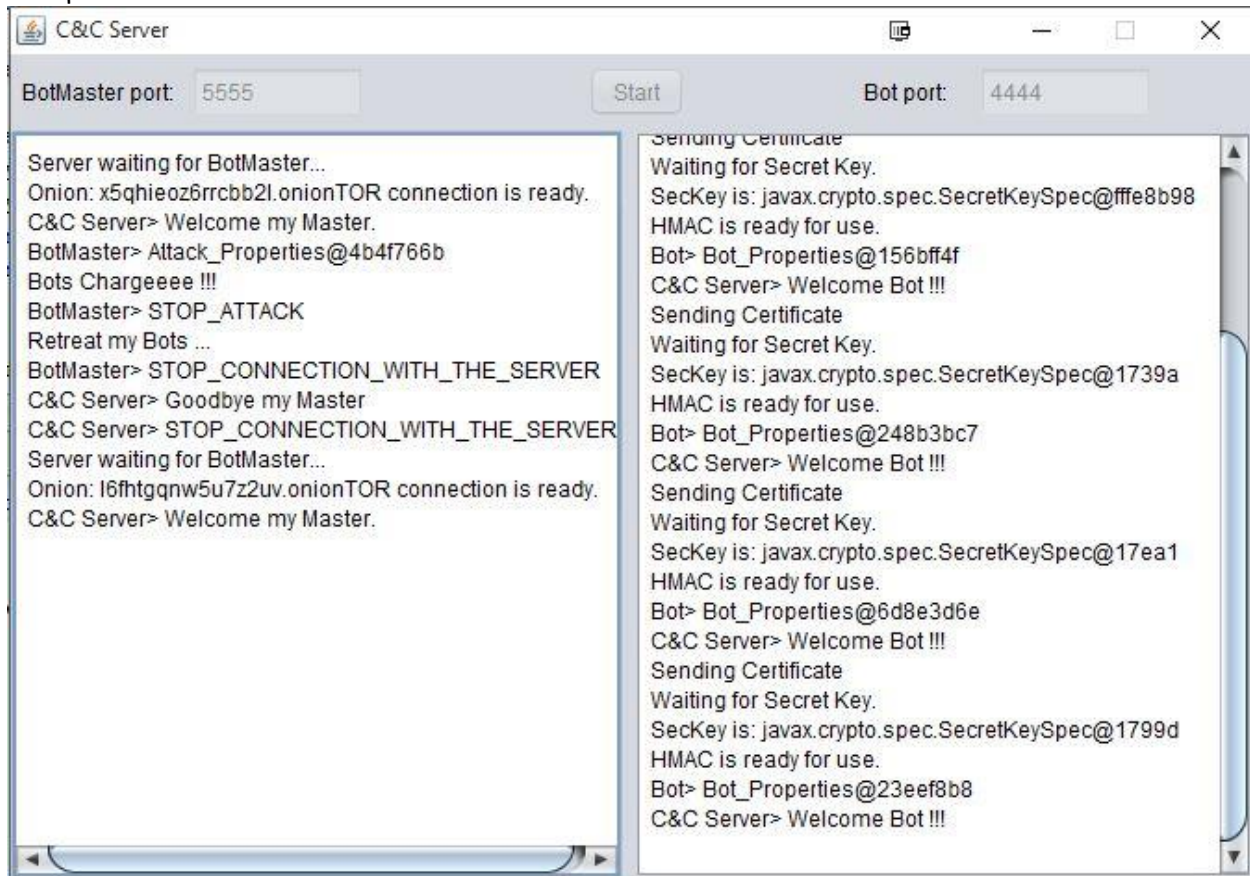
The screenshot shows a NetBeans window with a table listing online bots. The table has two columns: 'Host Name' and 'Host Address'. There are four rows of data, including three instances of 'MarvinTheMartian' and one instance of 'DESKTOP-2AVT004'.

Host Name	Host Address
MarvinTheMartian	169.254.80.80
MarvinTheMartian	169.254.80.80
MarvinTheMartian	169.254.80.80
DESKTOP-2AVT004	192.168.245.132

Κλείνω τον BotMaster:



Ανοίγω άλλον BotMaster:



4) Περιγραφή και τρόπος δημιουργίας πιστοποιητικών / 5) Ψηφιακά Πιστοποιητικά

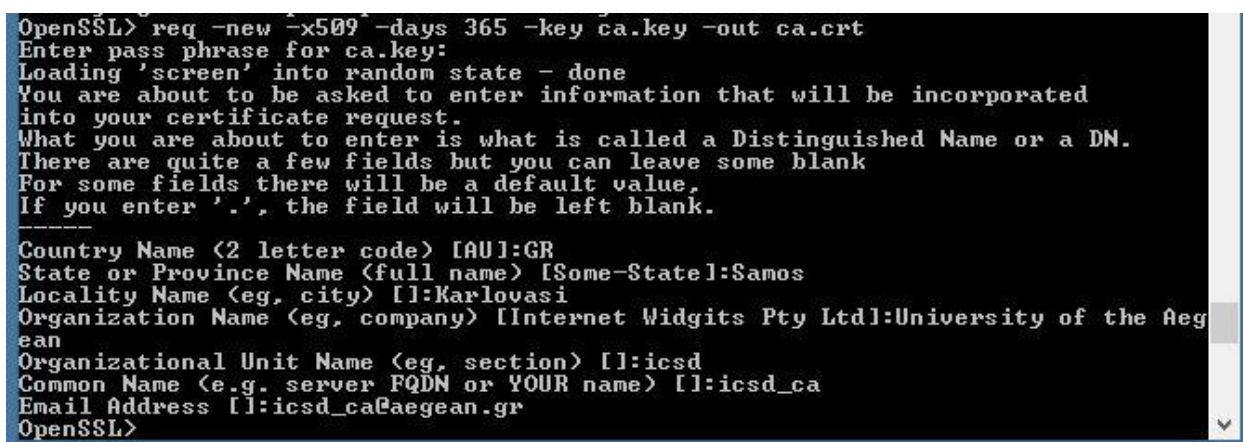
Χρησιμοποιήσαμε την CA που είχαμε δημιουργήσει στην πρώτη εργαστηριακή άσκηση. Με την διαφορά πως σε αυτό το project χρησιμοποιήσαμε μόνο την αρχή-πιστοποίησης – certification authority (ca):

Δημιουργία της 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 .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>
```

Για το σκοπό της εργασίας χρησιμοποιούμε τα παρακάτω (κίτρινο χρώμα):

ca	03-Nov-15 8:06 PM	Security Certificate	3 KB
ca.jks	04-Nov-15 12:18 A...	JKS File	4 KB
ca.key	03-Nov-15 8:04 PM	KEY File	4 KB
ca	03-Nov-15 8:10 PM	Personal Informati...	6 KB

Στέλνει ο C&C τα bytes της CA στα Bots τα Bots κρυπτογραφούν με Public Key της CA το Secret Key. Στην συνέχεια το Bot στέλνει το encrypt message στον C&C και ο C&C χρησιμοποιεί το .jks για τα διαβάσει το private key και να αποκρυπτογραφήσει το message ώστε να βρει το SecretKey.

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

Αρχικά εντοπίσαμε την IP που μιλάμε με το TOR. Στην συγκεκριμένη περίπτωση στο VMware τρέχει ο BotMaster και στα windows ο C&CServer. Το wireshark είναι εγκατεστημένο στα windows (C&C Server) οπότε η IP που θα δούμε είναι η τελική IP που μας στέλνει τα δεδομένα:

No.	Time	Source	Destination	Protocol	Length	Info
22581	345.100945	192.168.1.2	89.46.100.162	TCP	66	443->19683 [SYN] Seq=0 win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
22589	345.178185	89.46.100.162	192.168.1.2	TCP	66	443->19683 [SYN, ACK] Seq=0 Ack=1 win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=32
22590	345.178296	192.168.1.2	89.46.100.162	TCP	54	19683->443 [ACK] Seq=1 Ack=1 win=65536 Len=0
22591	345.180702	192.168.1.2	89.46.100.162	SSL	143	Client Hello
22599	345.246153	89.46.100.162	192.168.1.2	TCP	60	443->19683 [ACK] Seq=1 Ack=90 win=29216 Len=0
22620	345.264114	89.46.100.162	192.168.1.2	TLSv1.2	1463	Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
22621	345.264605	192.168.1.2	89.46.100.162	TCP	54	19683->443 [ACK] Seq=90 Ack=1410 win=64256 Len=0
22623	345.271940	192.168.1.2	89.46.100.162	TLSv1.2	200	Certificate, Client Key Exchange
22632	345.376878	89.46.100.162	192.168.1.2	TCP	60	443->19683 [ACK] Seq=1410 Ack=236 win=29216 Len=0
22633	345.377036	192.168.1.2	89.46.100.162	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
22644	345.438629	89.46.100.162	192.168.1.2	TCP	60	443->19683 [ACK] Seq=1410 Ack=311 win=29216 Len=0
22645	345.442759	89.46.100.162	192.168.1.2	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
22646	345.442875	192.168.1.2	89.46.100.162	TCP	54	19683->443 [ACK] Seq=311 Ack=1485 win=65536 Len=0
22647	345.443853	192.168.1.2	89.46.100.162	TLSv1.2	619	Application Data
22664	345.517334	89.46.100.162	192.168.1.2	TLSv1.2	619	Application Data
22665	345.517433	192.168.1.2	89.46.100.162	TCP	54	19683->443 [ACK] Seq=876 Ack=2050 win=65024 Len=0
22666	345.518270	192.168.1.2	89.46.100.162	TLSv1.2	619	Application Data
22684	345.587062	89.46.100.162	192.168.1.2	TLSv1.2	619	Application Data
22685	345.587139	192.168.1.2	89.46.100.162	TCP	54	19683->443 [ACK] Seq=1441 Ack=2615 win=64512 Len=0
22686	345.588892	192.168.1.2	89.46.100.162	TLSv1.2	619	Application Data
22687	345.590203	192.168.1.2	89.46.100.162	TLSv1.2	1514	Application Data, Application Data
22688	345.590221	192.168.1.2	89.46.100.162	TLSv1.2	289	Application Data
22689	345.590337	192.168.1.2	89.46.100.162	TLSv1.2	1514	Application Data, Application Data
22690	345.590374	192.168.1.2	89.46.100.162	TLSv1.2	289	Application Data
22691	345.590592	192.168.1.2	89.46.100.162	TLSv1.2	1514	Application Data, Application Data
22692	345.590601	192.168.1.2	89.46.100.162	TLSv1.2	289	Application Data
22693	345.591293	192.168.1.1	192.168.1.2	ICMP	590	Destination unreachable (Fragmentation needed)
22694	345.591340	192.168.1.2	89.46.100.162	TLSv1.2	941	[TCP out-of-order] Application Data
22695	345.592020	192.168.1.1	192.168.1.2	ICMP	590	Destination unreachable (Fragmentation needed)
22696	345.592831	192.168.1.1	192.168.1.2	ICMP	590	Destination unreachable (Fragmentation needed)
22724	345.661270	89.46.100.162	192.168.1.2	TCP	66	443->19683 [ACK] Seq=2615 Ack=2006 win=33728 Len=0 SLE=3466 SRE=3701
22726	345.664818	89.46.100.162	192.168.1.2	TCP	74	[TCP window update] 443->19683 [ACK] Seq=2615 Ack=2006 win=34880 Len=0 SLE=5161 SRE=5396 SLE=3466 SRE=3701
22727	345.671963	89.46.100.162	192.168.1.2	TCP	82	[TCP window update] 443->19683 [ACK] Seq=2615 Ack=2006 win=36000 Len=0 SLE=6856 SRE=7091 SLE=5161 SRE=5396 SLE=3466 SRE=3701
22728	345.675228	89.46.100.162	192.168.1.2	TCP	82	443->19683 [ACK] Seq=2615 Ack=2093 win=37760 Len=0 SLE=6856 SRE=7091 SLE=5161 SRE=5396 SLE=3466 SRE=3701
22729	345.675668	192.168.1.2	89.46.100.162	TLSv1.2	1506	[TCP Retransmission] Continuation Data
22747	345.756416	89.46.100.162	192.168.1.2	TCP	82	443->19683 [ACK] Seq=2615 Ack=4023 win=40032 Len=0 SLE=3466 SRE=4023 SLE=6856 SRE=7091 SLE=5161 SRE=5396
22748	345.756458	192.168.1.2	89.46.100.162	TLSv1.2	1506	[TCP Retransmission] Continuation Data
22749	345.756459	192.168.1.2	89.46.100.162	TCP	783	[TCP Retransmission] 19683->443 [PSH, ACK] Seq=6927 Ack=2615 win=64512 Len=729 [assembly error: protocol TCP: New fragment overlaps old data (retransmission)]
22750	345.761431	89.46.100.162	192.168.1.2	TCP	74	443->19683 [ACK] Seq=2615 Ack=5475 win=42944 Len=0 SLE=5161 SRE=5475 SLE=6856 SRE=7091
22776	345.840931	89.46.100.162	192.168.1.2	TCP	66	443->19683 [ACK] Seq=2615 Ack=7091 win=45824 Len=0 SLE=6856 SRE=6927

0000 ac 22 0b 7f 5d e0 a4 7e 39 c3 73 18 08 00 45 80 ...-...-...9...E.
0010 05 a9 cc 98 40 00 33 06 f5 b0 59 2e 64 a2 c0 a8 ...8.j...Y.d..
0020 01 02 01 b0 4c e3 4a 33 ad 7f 4b 4c 3a c8 50 18j...l..P..
0030 01 91 f4 f6 00 00 16 03 03 00 31 02 00 00 2d 03i.....
0040 03 4f c3 33 a6 b0 c3 71 64 1c 69 0b 67 03 1a 11 ..0.3...q d.i.g..
0050 01 38 2a 08 f1 a9 82 16 21 bc ad 2b 19 43 f3 c8 a8.....#...C..
0060 33 00 00 33 00 00 05 ff 01 00 01 00 16 03 03 03 3...3.....
0070 7d 0b 00 03 79 00 03 76 00 01 b8 30 82 01 ba 30 j...Y...-...0..
0080 82 01 1d a0 03 02 01 02 02 09 00 bc 27 5d 7e c4-...J..
0090 01 6a 83 30 04 06 09 2a 86 48 86 f7 0d 01 01 05H.....
00a0 05 09 30 1c 31 1a 30 18 06 03 55 04 03 13 11 77 ..0.1.0...U...w
00b0 77 77 2e 75 63 6b 62 63 61 61 75 2e 65 6f 6d ww.Uckcb cabv.com
00c0 30 1e 17 0d 31 35 31 31 31 32 30 30 30 30 30 0...1311 12000000
00d0 5a 17 0d 31 38 30 31 30 38 30 30 30 30 30 5a ..14010 60000002
00e0 30 1c 31 1a 30 18 06 03 55 04 03 13 11 77 77 77 0.1.0...U...www
00f0 2e 60 71 6c 34 75 65 32 61 32 2e 6e 65 74 30 81 ..kol4ue2 a2.net0.

File: C:\Users\ALCME-T\appdata\Local\T... Packets: 2205 - Displayed: 46 (0.2%) - Display 1 (0.2%) Profile: Default

Εντοπίσαμε την IP αυτή στην Ευρώπη πιο συγκεκριμένα στην Romania. Να τονίσουμε πως αυτή την external IP δεν έπαθε καμία ζημία, ο μόνος ρόλος της ήταν να μας στέλνει της εντολές του BotMaster μέσω TOR στον C&C Server χωρίς να υπάρχει κάποιος κακόβουλος στόχος προς την external IP αυτή.

Geolocation for **89.46.100.162**.

[Hide your IP with VPN](#)


IP Location Finder

IP Address:

Here are the results from a few Geolocation providers. Accuracy of geolocation data may vary from a provider to provider. Test drive yourself, and decide on the provider that you like.


Do you have a problem with IP location lookup? Report a [problem](#).

Geolocation data from **IP2Location** (Product: DB4 updated on 11/2/2015)

IP Address	Country	Region	City	ISP
89.46.100.162	Romania 	Vaslui	Vaslui	M247 Europe Srl


[Google Map for Vaslui, Vaslui, Romania \(New window\)](#)

Geolocation data from **EurekAPI** (Product: Pro On-demand API)

IP Address	Country	Region	City	ISP
89.46.100.162	Romania 	Salaj	Fabrica	M247 Europe Srl
		Continent	Latitude	Longitude
		Europe	47.2	23.4
				Organization
				M247 Europe SRL


[Google Map for Fabrica, Salaj, Romania \(New window\)](#)

Geolocation data from **DB-IP** (Product: Full updated on 11/3/2015)


IP Address	Country	Region	City	ISP
89.46.100.162	RO 	Bucharest	Bucharest	Clues Ips
		Time Zone	Latitude	Longitude
		Europe/bucharest	44.4325	26.1039
				Organization
				M247 Europe SRL

[Google Map for Bucharest, Bucharest, RO \(New window\)](#)

Geolocation data from **ipinfo.io** (On-demand API)

IP Address	Country	Region	City	Postal Code
89.46.100.162	RO 	Judetul Salaj	Fabrica	457154
		ISP	Hostname	Organization
		AS9009 M247 Ltd	no-rdns-yet	M247 Europe SRL

Geolocation data from **MaxMind** (Product: GeoLiteCity updated on 11/3/2015)

IP Address	Country	Region	City	Postal Code	Area Code
89.46.100.162	Romania 	31	Fabrica	457154	

[Google Map for Fabrica, 31, ROU \(New window\)](#)

Συνομιλία C&C Server με Bots:

Filter: tcp.port == 4444						Expression...	Clear	Apply	Save
No.	Time	Source	Destination	Protocol	Length	Info			
408	3.86811100	192.168.245.132	192.168.56.1	TCP	66	50595→4444 [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1			
409	3.86848300	192.168.56.1	192.168.245.132	TCP	58	4444→50595 [SYN, ACK] Seq=0 Ack=1 win=64240 Len=0 MSS=1460			
410	3.86857600	192.168.245.132	192.168.56.1	TCP	54	50595→4444 [ACK] Seq=1 Ack=1 win=64240 Len=0			
411	3.86948300	192.168.56.1	192.168.245.132	TCP	58	4444→50595 [PSH, ACK] Seq=1 Ack=1 win=64240 Len=4			
412	3.86951700	192.168.245.132	192.168.56.1	TCP	58	50595→4444 [PSH, ACK] Seq=1 Ack=1 win=64240 Len=4			
413	3.86960300	192.168.56.1	192.168.245.132	TCP	54	4444→50595 [ACK] Seq=5 Ack=5 win=64240 Len=0			
414	3.87266000	192.168.56.1	192.168.245.132	TCP	71	4444→50595 [PSH, ACK] Seq=5 Ack=5 win=64240 Len=17			
415	3.87269500	192.168.56.1	192.168.245.132	TCP	1514	4444→50595 [ACK] Seq=22 Ack=5 win=64240 Len=1460			
416	3.87270400	192.168.56.1	192.168.245.132	TCP	143	4444→50595 [PSH, ACK] Seq=1482 Ack=5 win=64240 Len=89			
417	3.87292400	192.168.245.132	192.168.56.1	TCP	54	50595→4444 [ACK] Seq=5 Ack=1571 win=64240 Len=0			
440	4.12390200	192.168.245.132	192.168.56.1	TCP	71	50595→4444 [PSH, ACK] Seq=5 Ack=1571 win=64240 Len=17			
441	4.12403500	192.168.56.1	192.168.245.132	TCP	54	4444→50595 [ACK] Seq=1571 Ack=22 win=64240 Len=0			
442	4.12416800	192.168.245.132	192.168.56.1	TCP	572	50595→4444 [PSH, ACK] Seq=22 Ack=1571 win=64240 Len=518			
443	4.12421400	192.168.56.1	192.168.245.132	TCP	54	4444→50595 [ACK] Seq=1571 Ack=540 win=64240 Len=0			
444	4.13972800	192.168.245.132	192.168.56.1	TCP	288	50595→4444 [PSH, ACK] Seq=540 Ack=1571 win=64240 Len=234			
445	4.13984300	192.168.56.1	192.168.245.132	TCP	54	4444→50595 [ACK] Seq=1571 Ack=774 win=64240 Len=0			
446	4.16272200	192.168.56.1	192.168.245.132	TCP	96	4444→50595 [PSH, ACK] Seq=1571 Ack=774 win=64240 Len=42			
451	4.22047400	192.168.245.132	192.168.56.1	TCP	54	50595→4444 [ACK] Seq=774 Ack=1613 win=64198 Len=0			

<	
Frame 446: 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface 2	
Ethernet II, Src: Vmware_e3:4b:c1 (00:50:56:e3:4b:c1), Dst: Vmware_17:ef:9d (00:0c:29:17:ef:9d)	
Internet Protocol Version 4, Src: 192.168.56.1 (192.168.56.1), Dst: 192.168.245.132 (192.168.245.132)	
Transmission Control Protocol, Src Port: 4444 (4444), Dst Port: 50595 (50595), Seq: 1571, Ack: 774, Len: 42	
Data (42 bytes)	
0000	00 0c 29 17 ef 9d 00 50 56 e3 4b c1 08 00 45 00 ...).P V.K...E.
0010	00 52 70 35 00 00 80 06 1b 9a c0 a8 38 01 c0 a8 .Rp5... ..8...
0020	f5 84 11 5c c5 a3 70 94 67 d4 5d 45 14 2d 50 18 ...\.p. g.]E.-P.
0030	fa f0 de 84 00 00 75 71 00 7e 00 00 00 00 00 20uq .~.....
0040	86 3c f3 58 9f 9c 80 6b 2b 55 6a a0 77 23 6f c7 .<.X...k +Uj.w#o.
0050	cd c6 6c fd d4 3b 0c 65 0e b1 af 70 6e 2c 32 3a ..l...;e ...pn,2:

7.1 Πιστεύετε πως το custom πρωτόκολλο που δημιουργήσατε στο κανάλι C&C Server – Bots διασφαλίζει από Replay Attacks; Αν όχι, τροποποιήστε το πρωτόκολλο που δημιουργήσατε για τη διασφάλιση από τέτοιου είδους επιθέσεις. Υλοποιήστε το αντίμετρο που σκεφτήκατε.

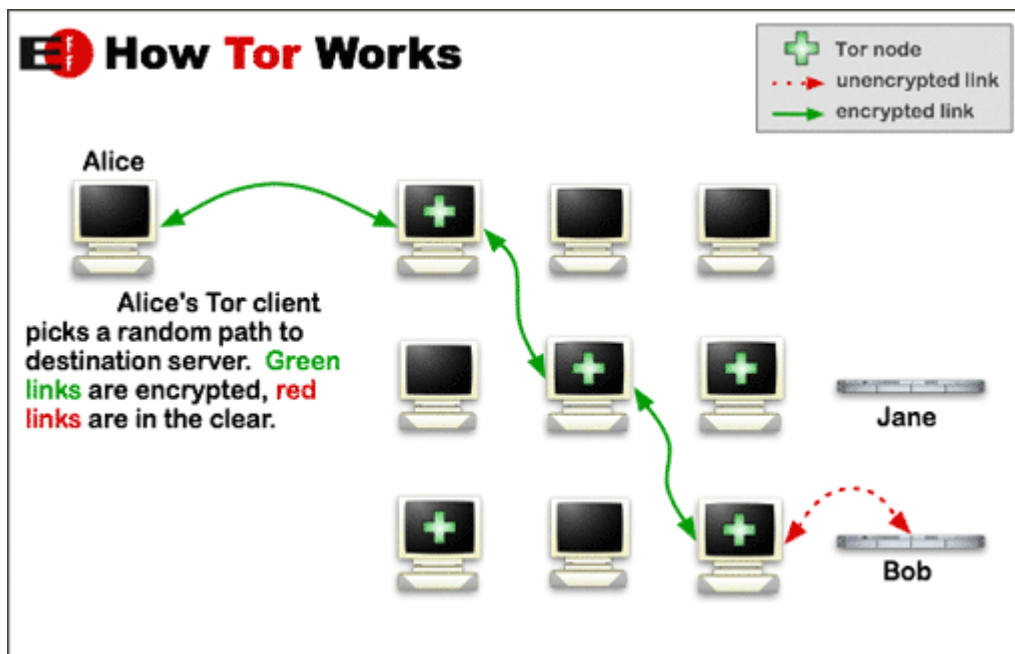
Replay Attack είναι η επίθεση όπου όταν 2 οντότητες μοιράζονται το ίδιο κλειδί μία τρίτη 'κρυφακούει' την συνομιλία και αποκτά στη κατοχή της το ίδιο κλειδί.

Το ένα αντίμετρο που μπορούμε να πάρουμε είναι μετά την κρυπτογράφηση να χρησιμοποιήσουμε έναν κωδικό αυθεντικοποίησης (HMAC). Το συγκεκριμένο αντίμετρο το υλοποιήσαμε στο δικό μας custom ssl πρωτόκολλο (encrypt then mac).

Το πρωτόκολλο που δημιουργήσαμε για να διασφαλίσει την ασφάλεια από replay attacks πρέπει να στην αρχή κάθε συνομιλίας να κρυπτογραφούμε μαζί με το password και μια timestamp. Έτσι διασφαλίζουμε την επικοινωνία μεταξύ δύο οντοτήτων για μικρό βέβαια χρονικό διάστημα.

7.2 Η εμπιστευτικότητα και ακεραιότητα της πληροφορίας που μεταδίδεται στο κανάλι BotMaster – C&C Server διασφαλίζεται από τους ενδιάμεσους κόμβους του TOR και του RP; Αν ναι, με ποιόν τρόπο;

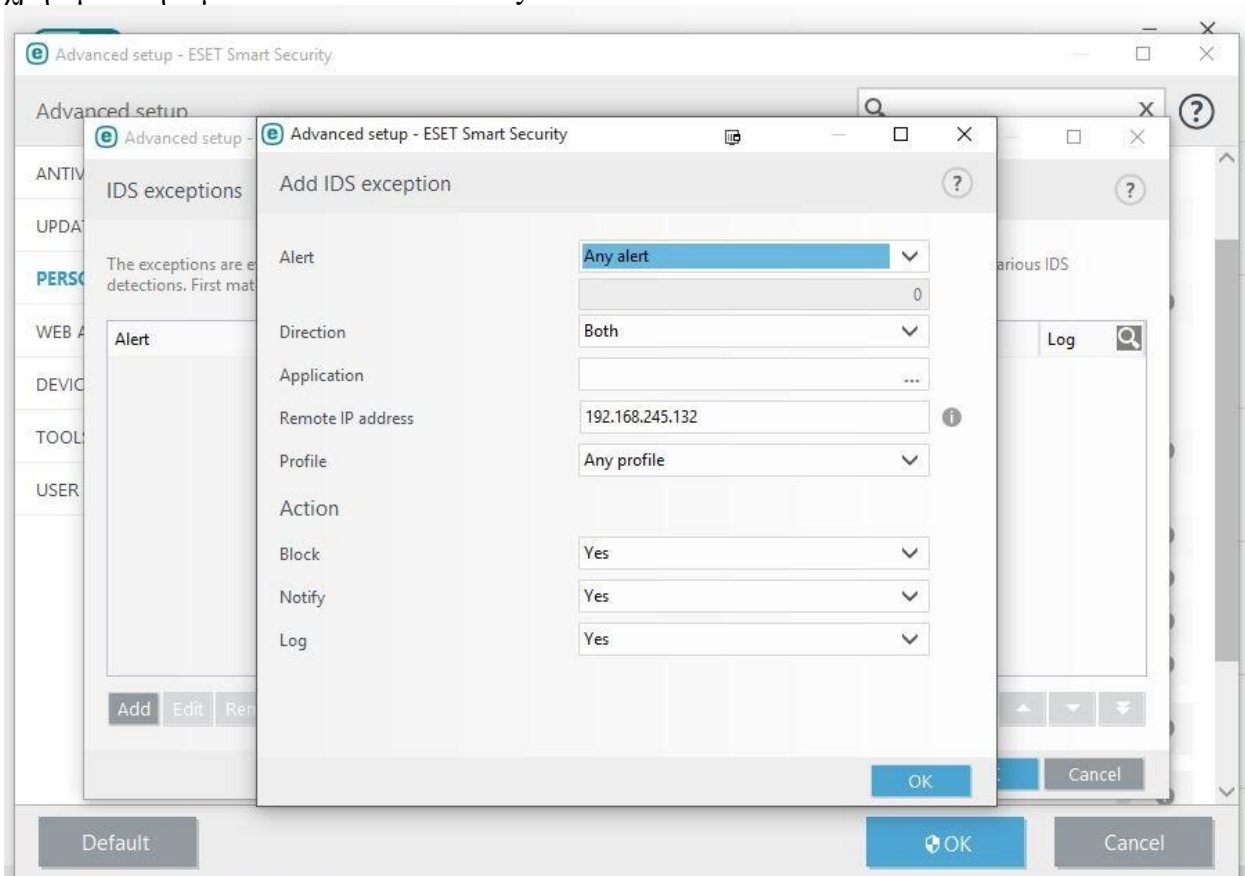
Η εμπιστευτικότητα και ακεραιότητα της πληροφορίας που μεταδίδεται στο κανάλι BotMaster– C&C Server διασφαλίζεται από τους ενδιάμεσους κόμβους του TOR και του RP καθώς το Tor μετά την υλοποίηση του onion routing, το οποίο κρυπτογραφεί και δρομολογεί τυχαία την επικοινωνία μέσω ενός δικτύου από κόμβους που το λειτουργούν εθελοντές ανά την υφήλιο. Οι συγκεκριμένοι δρομολογητές onion (κρεμμύδι) εφαρμόζουν κρυπτογράφιση πολλαπλών στρωμάτων (εξ ου και η μεταφορά του κρεμμυδιού) για να εξασφαλίσουν τέλεια μυστικότητα προς τα εμπρός (perfect forward secrecy) μεταξύ των κόμβων, και γι' αυτό προσφέρει ανωνυμία της δικτυακής τοποθεσίας. Κατά μήκος της διαδρομής τα κομμάτια της πληροφορίας είναι κρυπτογραφημένα. Επίσης στον παραλήπτη φαίνεται ότι ο τελευταίος κόμβος Tor (κόμβος εξόδου) είναι ο δημιουργός της επικοινωνίας.



7.3 Το πρωτόκολλο που έχουμε δημιουργήσει είναι ευάλωτο σε clogging attacks (επιθέσεις πνιγμού). Στις επιθέσεις αυτές, ένας επιτιθέμενος που υποδύεται μία άλλη μηχανή (IP Spoofing) αποσκοπεί σε επίθεση Άρνησης Εξυπηρέτησης (DoS) δημιουργώντας “half-open sessions”, δηλαδή sessions που δεν καταλήγουν ποτέ στην ανταλλαγή πληροφορίας. Αφού συμβουλευτείτε τις διαφάνειες του μαθήματος, προτείνετε ένα τρόπο για την αντιμετώπιση αυτού του είδους των επιθέσεων.

Για αντιμετωπίσουμε τις clogging attacks πρέπει να χρησιμοποιήσουμε τη βοήθεια του πρωτοκόλλου ike (internet key exchange) για την ανταλλαγή των κλειδών. Το ike αντιμετωπίζει τις clogging attacks με cookie exchange μεταξύ των δύο μερών. Μια αίτηση για την δημιουργία ενός δημόσιου κλειδιού πρέπει να έπεται μιας ανταλλαγής cookie. Ο responder στέλνει ένα cookie στον Initiator ο οποίος πρέπει να επιβεβαιώσει την παραλαβή του. Έτσι εάν αυτός που ζήτησε το κλειδί παριστάνει έναν άλλο δεν θα λάβει ποτέ το αντίστοιχο cookie.

Block μιας ip μέσω ενός antivirus ή του Windows Firewall with Advanced Security. Εμείς χρησιμοποιήσαμε το Eset Smart Security 9.



8) Βιβλιογραφία

<https://www.iplocation.net/>

https://en.wikipedia.org/wiki/Replay_attack

<https://el.wikipedia.org/wiki/Tor>

<http://stackoverflow.com/questions/14814749/connection-to-tor-java>

<https://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#impl>

<http://www.oracle.com/technetwork/java/javase/downloads/ice8-download-2133166.html>

<http://netnix.org/2015/04/19/aes-encryption-with-hmac-integrity-in-java/>

<http://support.eset.com/kb3343/>

Διαφάνιες Θεωρίας του μαθήματος.

Τέλος