ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Διδάσκων: Επίκουρος Καθηγητής Γεώργιος Καμπουράκης
Εργαστηριακοί Συνεργάτες: Μάριος Αναγνωστόπουλος (ΥΔ), Δημήτρης Παπαμαρτζιβάνος (ΥΔ)

Ασφάλεια Δικτύων Υπολογιστών

και Τεχνολογίες Προστασίας της Ιδιωτικότητας

2η Εργαστηριακή Άσκηση

Νοέμβριος 2015

ΑΣΚΗΣΗ 2

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

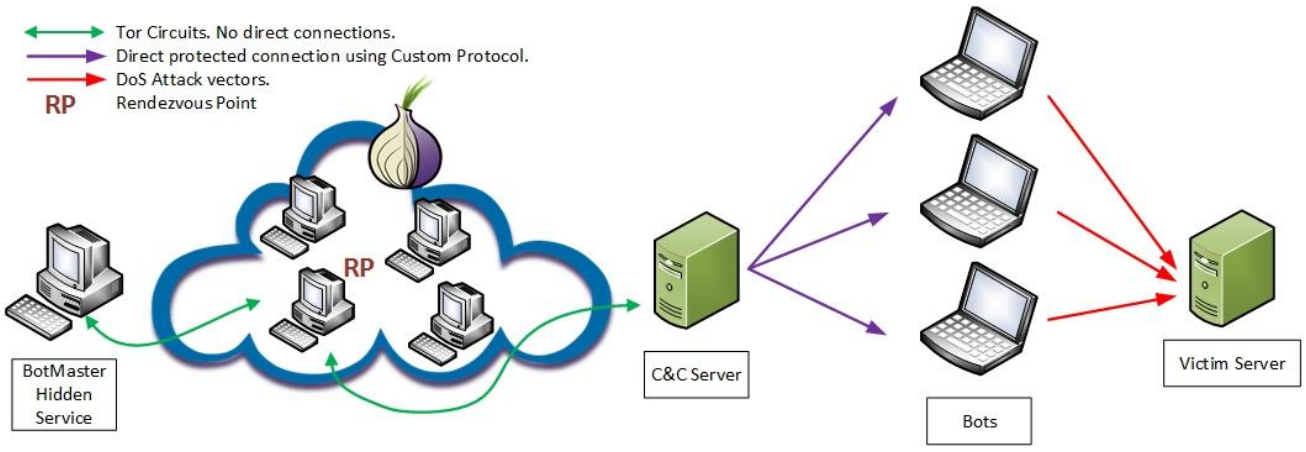
Περιγραφή

Στη 2η εργαστηριακή άσκηση καλείστε να επεκτείνετε την υποδομή διαχείρισης botnet που δημιουργήσατε στην 1η, αξιοποιώντας τεχνικές για την ανωνυμοποίηση των οντοτήτων της υποδομής και τη διασφάλιση της εμπιστευτικότητας και της ακεραιότητας των μεταδιδόμενων δεδομένων.

Πιο συγκεκριμένα, στο πλαίσιο αυτής στη εργασίας καλείστε να αξιοποιήσετε το δίκτυο ανωνυμοποίησης Tor [1] για την ανωνυμοποίηση του BotMaster και να αναπτύξετε ένα Custom πρωτόκολλο ασφαλείας μεταξύ των Bots και του Command & Control Server για τη διασφάλιση της εμπιστευτικότητας και ακεραιότητας της μεταδιδόμενης πληροφορίας.

Στην πραγματικότητα, η υποδομή που υλοποιήσατε στην προηγούμενη εργασία παραμένει η ίδια ως προς την υλοποίηση των επιθέσεων άρνησης εξυπηρέτησης και το πρωτόκολλο επικοινωνίας των οντοτήτων. Η διαφορά παρουσιάζεται στον τρόπο εγκαθίδρυσης των καναλιών επικοινωνίας μεταξύ των οντοτήτων.

Η νέα τοπολογία που καλείστε να υλοποιήσετε παρουσιάζεται στο παρακάτω σχήμα.



BotMaster – C&C Server: Η επικοινωνία του BotMaster με το C&C Server, τον οποίο διαχειρίζεται, θα πρέπει να προωθείται μέσω του δικτύου ανωνυμοποίησης Tor. Σκοπός αυτής της υλοποίησης είναι να παραμένουν κρυφά η ταυτότητα και η IP του BotMaster. Για την υλοποίηση του BotMaster θα κάνετε χρήση της βιβλιοθήκης silvertunnel [2], η οποία παρέχει τη δυνατότητα δημιουργίας Server ως Hidden Service (HS) [3] μέσα στο δίκτυο Tor. Μία HS στο Tor έχει μία διεύθυνση της μορφής π.χ. vw4qq5htd3hgohro.onion [4]. Η βιβλιοθήκη και οι υποστηρικτικές βιβλιοθήκες για την λειτουργία του silvertunnel υπάρχουν διαθέσιμα στο [5].

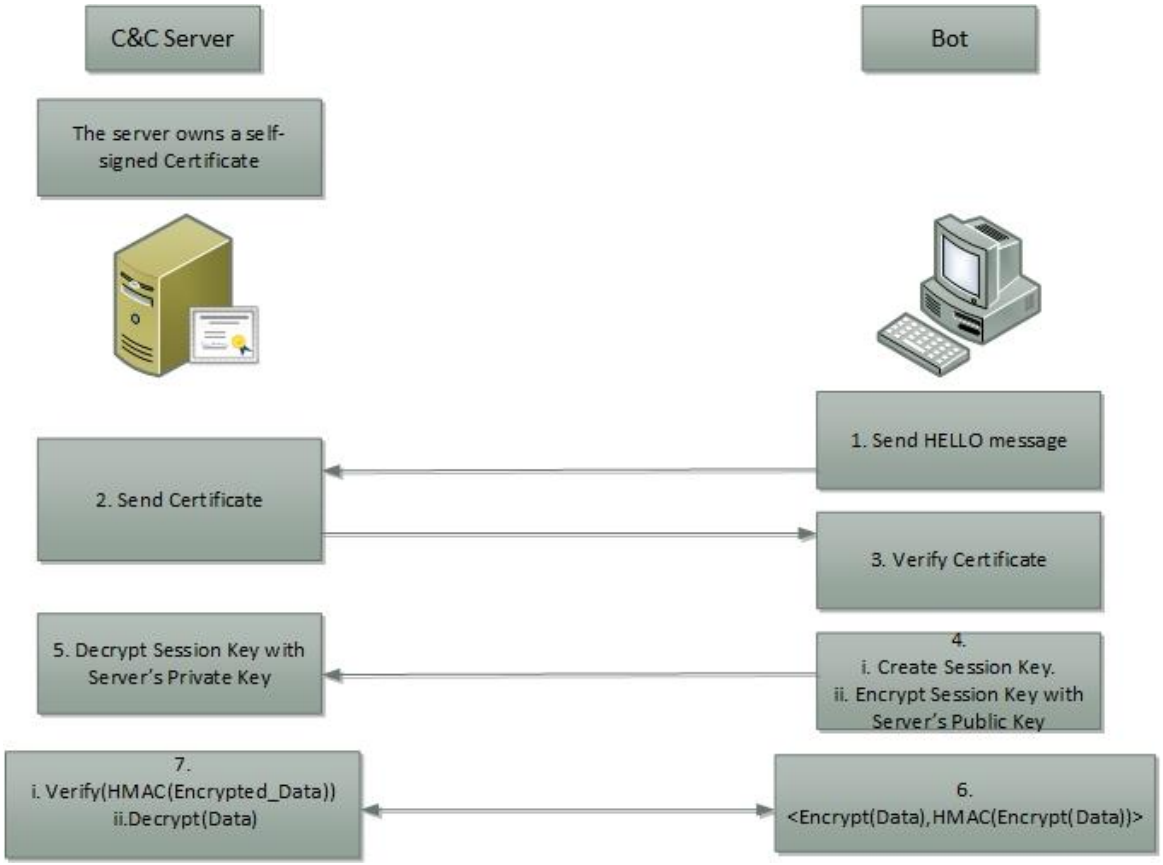
Ο C&C Server συμπεριφέρεται ως client προς την πλευρά του BotMaster και συνδέεται σε αυτόν για να μπορεί να λαμβάνει εντολές χρησιμοποιώντας την .onion διεύθυνση του. Στο πλαίσιο της εργασίας ΔΕΝ μας ενδιαφέρει ο τρόπος με τον οποίο ο C&C Server ενημερώνεται για πρώτη φορά με την .onion διεύθυνση στην οποία πρέπει να συνδεθεί.

Από τη στιγμή που ο C&C Server συνδεθεί με τον BotMaster, τον ενημερώνει ανά χρονικά διαστήματα για τα διαθέσιμα συνδεδεμένα bots ή ενημερώνεται για εντολές έναρξης/τερματισμού μιας επίθεσης προς κάποιο στόχο. Όπως και στην πρώτη εργασία, ο C&C Server διατηρεί μία λίστα με τα διαθέσιμα bots με σκοπό να ενημερώνει τον BotMaster.

C&C Server - Bots: Η επικοινωνία μεταξύ του C&C Server και των Bots θα κάνει χρήση ενός custom πρωτοκόλλου για την διασφάλιση της εμπιστευτικότητας και της ακεραιότητας των μεταδιδόμενων δεδομένων. Καλείστε να διασφαλίσετε το κανάλι C&C Server – Bot προσομοιώνοντας τον τρόπο λειτουργίας ενός πρωτοκόλλου ασφαλείας για τη συμφωνία ενός συμμετρικού κλειδιού κρυπτογράφησης, με την χρήση του οποίου θα διαφυλάξετε την ακεραιότητα και την εμπιστευτικότητα των μεταδιδόμενων δεδομένων.

Η διαφορά στην παρούσα εργασία είναι ότι δεν αξιοποιείτε τις έτοιμες, από την Java, κλάσεις (SSLSocket κτλ.) για την εγκαθίδρυση ενός ασφαλούς καναλιού επικοινωνίας αλλά χρησιμοποιείτε απλά sockets και διασφαλίζετε την ασφάλεια των δεδομένων με τη χρήση του συμμετρικού κλειδιού που θα παραχθεί μέσω του custom πρωτοκόλλου. Άρα, η ασφάλεια παρέχεται στο επίπεδο εφαρμογής.

Τα βήματα του πρωτοκόλλου παρουσιάζονται στο παρακάτω σχήμα.



Ο C&C Server έχει στην κατοχή του ένα αυθυπόγραφο (self-singed) ψηφιακό πιστοποιητικό, το οποίο χρησιμοποιείται για τη μονομερή αυθεντικοποίηση του. Για τη δημιουργία του πιστοποιητικού θα χρησιμοποιήσετε το εργαλείο openssl [6]. Το πρωτόκολλο ξεκινάει ουσιαστικά στο βήμα 1, όπου ο client στέλνει στο server το μήνυμα HELLO. Στη συνέχεια στο βήμα 2, ο server απαντάει αποστέλλοντας το πιστοποιητικό του στο client ο οποίος επιβεβαιώνει το ψηφιακά υπογεγραμμένο πιστοποιητικό στο βήμα 3. Αφού ο client επιβεβαιώσει την γνησιότητα του πιστοποιητικού, δημιουργεί ένα συμμετρικό κλειδί κρυπτογράφησης συνόδου AES-256 bits και το κρυπτογραφεί με το δημόσιο κλειδί του server πριν το αποστείλει. Από αυτή τη στιγμή και έπειτα ο server και ο client έχουν στην κατοχή τους ένα κοινό μυστικό με το οποίο μπορούν να εξασφαλίσουν την ακεραιότητα και εμπιστευτικότητα των δεδομένων. Για τη διασφάλιση της ακεραιότητας και της αυθεντικότητας (authenticity) των ανταλλασσόμενων μηνυμάτων θα χρησιμοποιήσετε τη συνάρτηση HMAC-SHA256 για τη δημιουργία του MAC [10, 12] των μηνυμάτων. Κάθε μήνυμα που ανταλλάσσεται μεταξύ client – server πρέπει να είναι κρυπτογραφημένο με το συμμετρικό κλειδί και κάθε οντότητα που παραλαμβάνει το μήνυμα να κάνει τις απαραίτητες ενέργειες για την επιβεβαίωση της ακεραιότητας των δεδομένων, όπως φαίνεται στα βήματα 7 και 8. Το σχήμα ακεραιότητας που πρέπει να υλοποιήσετε είναι το Encrypt-then-MAC [11].

Όσον αφορά τις λειτουργίες του C&C Server, αυτός λειτουργεί ως server για τα bots, τα οποία υπακούν σε εντολές επίθεσης προς στόχους που ορίζει ο BotMaster και προωθούνται στα bots μέσω του C&C Server.

Bots: Και σε αυτή την εργασία, τα bots, εξαπολύουν τις επιθέσεις που τους υποδεικνύονται προς τους στόχους εκτελώντας τα scripts επίθεσης που δημιουργήσατε στην 1η εργαστηριακή άσκηση.

Καλείστε να επιβεβαιώσετε την ασφάλεια της επικοινωνίας, κάνοντας χρήση κάποιου προγράμματος σύλληψης πακέτων (packet sniffer). Τέτοια προγράμματα είναι τα Wireshark [7], Tcpdump [8], Ettercap [9] κλπ. Χρησιμοποιείστε κατάλληλα φίλτρα για να εμφανίσετε την κίνηση στο κανάλι C&C Server – Bots. Επιβεβαιώσετε ότι τα ανταλλασσόμενα μηνύματα είναι κρυπτογραφημένα.

Για τη διασύνδεση των οντοτήτων του παραπάνω δικτύου μπορείτε να χρησιμοποιήσετε τοπικό δίκτυο και να θεωρήσετε ότι η IP διεύθυνση τους είναι γνωστή. Η γλώσσα προγραμματισμού που θα χρησιμοποιηθεί για τη δημιουργία του γραφικού περιβάλλοντος (GUI) καθώς και την ασφαλή επικοινωνία μεταξύ των οντοτήτων του συστήματος είναι η JAVA. Τα κανάλια επικοινωνίας θα υλοποιηθούν αξιοποιώντας τα Sockets της JAVA και κάνοντας χρήση ροών αντικειμένων (Object Streams). Να μη χρησιμοποιηθούν εργαλεία για την αυτόματη δημιουργία GUI (π.χ., Netbeans GUI Building).

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.

Bonus:
Υλοποιείστε το αντίμετρο που προτείνατε στην Ερώτηση 3. (1 μονάδα)

Προσοχή!
Στην αξιολόγηση της εργασίας θα ληφθούν υπόψη η εκτέλεση των scripts εναντίων των στόχων, το πρωτόκολλο επικοινωνίας των οντοτήτων και γενικότερα η λειτουργικότητα που ορίζεται στις απαιτήσεις της 1ης εργαστηριακής άσκησης.

Παραδοτέα

Καθ' όλη τη διάρκεια εκπόνησης της εργασίας θα πρέπει να χρησιμοποιείτε την πλατφόρμα Gitlab για το διαμοιρασμό του πηγαίου κώδικα μεταξύ των μελών κάθε ομάδας και τον έλεγχο της πορείας της εργασίας σας από τους διδάσκοντες. Είναι απαραίτητο όλα τα μέλη της ομάδας να συμμετέχουν στην παραπάνω διαδικασία. Κατά την παράδοση όλος ο πηγαίος κώδικας (εκτός από το eclass) θα πρέπει να έχει αναρτηθεί και στο Gitlab σύμφωνα με τις οδηγίες που παρουσιάστηκαν κατά την παρουσίαση.

Η **αναφορά** σας θα περιέχει τα ακόλουθα:

[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.

Η εργασία πρέπει να παραδοθεί μέχρι τις **29/11** μέσω της πλατφόρμας ηλεκτρονικής μάθησης **e-class**.
Το τελικό παραδοτέο πρέπει να είναι αρχείο .zip (ή .rar) με όνομα:

ΑριθμόςΜητρώου1_ΑριθμόςΜητρώου2_ΑριθμόςΜητρώου3_Lab02.zip
(π.χ. icsd12001_icsd12002_icsd12003_Lab02.zip), του κάθε μέλους της ομάδας.

Τεχνικό παράρτημα

Για την υλοποίηση του Hidden Service και του Client που συνδέεται σε αυτή θα κάνετε χρήση του κώδικα που δίνεται στο [3]. Οι απαραίτητες βιβλιοθήκες που χρειάζονται για την υποστήριξη των λειτουργιών του Hidden Service βρίσκονται στην ενότητα έγγραφα στο eclass. Οι βιβλιοθήκες έχουν δοκιμαστεί από τους διδάσκοντες στις εκδόσεις JDK 1.7 και 1.8.

Συμπληρωματική Βιβλιογραφία

[1] The Tor Project – Anonymity Online, https://www.torproject.org/
[2] Silvertunnel - easy-to-use secure and anonymous communication, https://silvertunnel.org/browser-for-all.html
[3] Silvertunnel -Netlib Direct API Usage, https://silvertunnel.org/doc/netlib-direct-api-usage.html
[4] Tor: Hidden Service Protocol, https://www.torproject.org/docs/hidden-services.html.en
[5] SilverTunnel-NG - Java library for easy accessing Tor network, http://sourceforge.net/projects/silvertunnel-ng/
[6] Openssl - Cryptography and SSL/TLS Toolkit, https://www.openssl.org/
[7] Wireshark network protocol analyzer, https://www.wireshark.org/
[8] Tcpdump - command-line packet analyzer, http://www.tcpdump.org/
[9] Ettercap, https://ettercap.github.io/ettercap/
[10] MAC – Message Authentication Code, https://en.wikipedia.org/wiki/Message_authentication_code
[11] Διαφάνειες Μαθήματος - 4_NetSec_SSL_U.pdf – Διαφάνεια #89
[12] Διαφάνειες Μαθήματος - 4_NetSec_SSL_U.pdf