Εργαστηριακή Άσκηση 12
Ασφάλεια

Ο σκοπός αυτής της εργαστηριακής άσκησης είναι η εξέταση θεμάτων σχετικών με την ασφαλή μετάδοση δεδομένων στο διαδίκτυο. Στα δίκτυα επικοινωνίας ένας εισβολέας μπορεί: να υποκλέπτει μηνύματα κρυφακούγοντας τις μεταδόσεις, να εισάγει αυθαίρετα μηνύματα στη σύνδεση, να υποδύεται άλλον παράγοντας μηνύματα με ψευδή διεύθυνση πηγής καθώς και να αποστερεί τη χρήση υπηρεσιών από άλλους (πχ. υπερφορτώνοντας το δίκτυο). Τα βασικά θέματα ασφαλείας, όσον αφορά την επικοινωνία πάνω από δίκτυα δεδομένων, περιλαμβάνουν την πιστοποίηση αυθεντικότητας (authentication), τον έλεγχο πρόσβασης (access control), την εμπιστευτικότητα (confidentiality), την ακεραιότητα (integrity), τη μη αποποίηση ευθύνης (non-repudiation), τη διαθεσιμότητα (availability) και το απόρρητο (privacy). Μέσω της πιστοποίησης αυθεντικότητας ο αποστολέας και ο παραλήπτης επιβεβαιώνουν ο ένας την ταυτότητα του άλλου. Μέσω του ελέγχου πρόσβασης οι υπηρεσίες είναι προσβάσιμες μόνο στους εξουσιοδοτημένους χρήστες. Μόνο ο αποστολέας και ο "κανονικός" παραλήπτης πρέπει να κατανοούν το περιεχόμενο του μηνύματος (εμπιστευτικότητα): ο αποστολέας κρυπτογραφεί το μήνυμα και ο παραλήπτης το αποκρυπτογραφεί. Επιπλέον, ο αποστολέας και ο παραλήπτης θέλουν να είναι βέβαιοι ότι το μήνυμα δεν τροποποιήθηκε κατά τη διαδρομή (ή μεταγενέστερα) χωρίς αυτό να γίνει αντιληπτό (ακεραιότητα). Ο χρήστης της υπηρεσίας δεν θα πρέπει να μπορεί να αρνηθεί τη χρήση της (μη αποποίηση ευθύνης) και ένα ελάχιστο επίπεδο υπηρεσίας πρέπει να είναι διαθέσιμο (διαθεσιμότητα). Τέλος το απόρρητο (privacy) περιλαμβάνει τα στοιχεία επικοινωνίας που πρέπει να είναι μυστικά στους τρίτους, δηλαδή, με ποιον επικοινωνεί ο χρήστης, σε ποια θέση, κλπ.

Όπως και στις προηγούμενες εργαστηριακές ασκήσεις θα εργασθείτε με τον αναλυτή πρωτοκόλλων Wireshark. Εδώ θα χρησιμοποιήσετε τη λειτουργία *Capture*, με φίλτρο, ώστε τα πλαίσια που καταγράφονται να περιέχουν κάποια συγκεκριμένα χαρακτηριστικά. Υπενθυμίζεται ότι το φίλτρο απεικόνισης *(Display)* που επιλέγετε από το μενού *Analyze* ενεργοποιείται **αφού** έχει ολοκληρωθεί η διαδικασία καταγραφής, ώστε να αποκρύψει κάποια από τα συλληφθέντα πλαίσια, ενώ το φίλτρο σύλληψης που επιλέγετε από το μενού *Capture* ενεργοποιείται **πριν** ξεκινήσει η διαδικασία καταγραφής, με αποτέλεσμα να καταγράφεται μόνο ένα μέρος των διερχόμενων πλαισίων.

Για τις παρακάτω ασκήσεις απαντήστε στο συνοδευτικό φυλλάδιο, το οποίο θα υποβάλλετε ως αρχείο pdf.

1. Πιστοποίηση αυθεντικότητας στο πρωτόκολλο HTTP

Κάποιες ιστοσελίδες στο διαδίκτυο προστατεύονται από συνθηματικά (password-protected) και δεν επιτρέπεται σε όλους τους χρήστες να τις επισκεφτούν παρά μόνο σε όσους διαθέτουν το συνθηματικό. Η διαδικασία της πιστοποίησης αυθεντικότητας (authentication) συνίσταται στην επαλήθευση της ψηφιακής ταυτότητας του χρήστη που αιτείται είσοδο στην ιστοθέση.

Με τη βοήθεια του Wireshark, καταγράψτε την κίνηση ενώ κάνετε χρήση της υπηρεσίας HTTP του υπολογιστή edu-dy.cn.ntua.gr (147.102.40.15). Εφαρμόστε φίλτρο σύλληψης `host 147.102.40.15` για να παρατηρείτε μόνο την κίνηση που σχετίζεται με αυτόν. Ξεκινήστε μία καταγραφή κίνησης και επισκεφτείτε τη σελίδα http://edu-dy.cn.ntua.gr/auth/. Η πρόσβαση σε αυτή τη σελίδα απαιτεί την επαλήθευση της ταυτότητάς σας. Δώστε `edu-dy` στο πεδίο του ονόματος χρήστη (user name) και `password` στο πεδίο του μυστικού κωδικού (password). Σταματήστε την καταγραφή.

1.1 Να καταγραφεί ο αριθμητικός κωδικός κατάστασης (status code) και η φράση που επιστρέφει ο εξυπηρετητής ως απόκριση στο αρχικό μήνυμα HTTP τύπου GET του πλοηγού ιστού.

1.2 Ο πλοηγός ιστού στέλνει και δεύτερο μήνυμα HTTP τύπου GET στον εξυπηρετητή. Συγκρίνοντας με το πρώτο μήνυμα HTTP GET να καταγραφεί το επιπλέον πεδίο που περιλαμβάνει η επικεφαλίδα HTTP του δεύτερου μηνύματος.

1.3 Καταγράψτε το περιεχόμενο του παραπάνω πεδίου όπως αυτό εμφανίζεται στο παράθυρο με τα περιεχόμενα του επιλεγμένου πλαισίου σε μορφή ASCII.

Τα στοιχεία πιστοποίησης αυθεντικότητας που καταχωρήσατε δεν κρυπτογραφήθηκαν για να αποσταλούν στον εξυπηρετητή, απλώς κωδικοποιήθηκαν σύμφωνα με μια πολύ γνωστή μέθοδο, τη Base64. Το αποτέλεσμα της κωδικοποίησης είναι η γραμματοσειρά που ακολουθεί τις λέξεις "Authorization: Basic" στην επικεφαλίδα του δεύτερου μηνύματος HTTP GET του πλοηγού.

1.4 Επισκεφτείτε την ιστοσελίδα https://www.motobit.com/util/base64-decoder-encoder.asp και εισάγετε στο δεύτερο παράθυρο που εμφανίζεται το περιεχόμενο του πεδίου που καταγράψατε στο ερώτημα 1.3. Αποκωδικοποιήστε το περιεχόμενο αυτό επιλέγοντας το "**decode** the data **from a Base64** string (base64 decoding)" και κάνοντας κλικ στο κουμπί "Convert the source data". Καταγράψτε το αποτέλεσμα της αποκωδικοποίησης[1].

1.5 Τι συμπεραίνετε για την ασφάλεια του βασικού μηχανισμού πιστοποίησης αυθεντικότητας που παρέχει το HTTP; *[Υπόδ.: https://en.wikipedia.org/wiki/Basic_access_authentication.]*

2. Υπηρεσία SSH – Secure SHell

Το Secure Shell ή SSH είναι ένα πρωτόκολλο που επιτρέπει την ανταλλαγή δεδομένων μεταξύ δύο δικτυακών οντοτήτων μέσω ενός ασφαλούς διαύλου επικοινωνίας. Χρησιμοποιείται αντί του TELNET ή άλλων μη ασφαλών προγραμμάτων πρόσβασης στον φλοιό (rlogin, κλπ), τα οποία στέλνουν πληροφορία, όπως π.χ. τους κωδικούς χρήστη, χωρίς κρυπτογράφηση. Συνήθως χρησιμοποιείται για εκτέλεση εντολών σε απομακρυσμένο υπολογιστή. Υποστηρίζει όμως σήραγγες (tunnels), συνδέσεις X11 καθώς και μεταφορά αρχείων με τη βοήθεια των συνδεδεμένων πρωτοκόλλων SFTP ή SCP. Η κρυπτογράφηση που χρησιμοποιείται στο SSH παρέχει εμπιστευτικότητα (confidentiality) και διασφαλίζει την ακεραιότητα (integrity) των μεταδιδόμενων δεδομένων. Το SSH χρησιμοποιεί κρυπτογράφηση δημόσιου κλειδιού για την πιστοποίηση αυθεντικότητας του απομακρυσμένου υπολογιστή και επιτρέπει την πιστοποίηση αυθεντικότητας του τοπικού υπολογιστή από τον απομακρυσμένο υπολογιστή.

Η έκδοση 2 του SSH αποτελείται από τρία συνεργαζόμενα πρωτόκολλα: το SSH Transport Layer Protocol (SSH-TRANS), το SSH Authentication Protocol (SSH-AUTH) και το SSH Connection Protocol (SSH-CONN). Ο πελάτης χρησιμοποιεί το SSH-AUTH πάνω από μια σύνδεση TCP που έχει εγκαταστήσει με τη βοήθεια του SSH-TRANS προκειμένου να ταυτοποιηθεί από τον εξυπηρετητή, ενώ το πρωτόκολλο SSH-CONN παρέχει μια ποικιλία υπηρεσιών (π.χ. γραμμή εντολών, σήραγγα, κλπ) μέσω της μοναδικής σύνδεσης που παρέχει το SSH-TRANS. Το πρωτόκολλο SSH-TRANS παρέχει την πιστοποίηση αυθεντικότητας του εξυπηρετητή καθώς και τις λειτουργίες εμπιστευτικότητας και ακεραιότητας των μεταδιδόμενων δεδομένων πάνω από τη σύνδεση TCP. Περισσότερες πληροφορίες για την αρχιτεκτονική του πρωτοκόλλου SSH θα βρείτε στο RFC 4251. Ακολουθεί μια σύντομη περιγραφή της λειτουργίας του πρωτοκόλλου SSH-TRANS.

Ο πελάτης και εξυπηρετητής SSH συμφωνούν για τη μεταφορά των δεδομένων πάνω από σύνδεση TCP ακολουθώντας μια διαδικασία διαπραγμάτευσης. Η σύνδεση TCP ξεκινά από την πλευρά του πελάτη προς τον εξυπηρετητή. Στο πρώτο βήμα, οι δύο πλευρές ανταλλάσουν πληροφορία για την έκδοση του πρωτοκόλλου. Αμφότερες στέλνουν ένα αναγνωριστικό κείμενο (string) της μορφής "SSH-protoversion-softwareversion comments", το οποίο περιλαμβάνει *υποχρεωτικά* την έκδοση του πρωτοκόλλου SSH (protoversion) και του αντίστοιχου λογισμικού (softwareversion) καθώς και *προαιρετικά* κάποια σχόλια (comments). Ακολουθεί η φάση της ανταλλαγής κλειδιών. Κάθε πλευρά

[1] Το Wireshark αυτομάτως εκτελεί την αποκωδικοποίηση αυτή. Μπορείτε να δείτε το αποτέλεσμα κάνοντας διπλό κλικ στο πεδίο "Authorization: Basic" της επικεφαλίδας HTTP.

στέλνει μια σειρά από λίστες ονομάτων αλγορίθμων για τις οποίες πρέπει να επέλθει συμφωνία. Η πρώτη λίστα περιέχει κατά σειρά προτίμησης τους αλγορίθμους ανταλλαγής κλειδιών (key exchange – kex) που υποστηρίζει η κάθε πλευρά. Εν γένει, επιλέγεται ο πρώτος αλγόριθμος του πελάτη που υποστηρίζεται και από την εξυπηρετητή. Ακολουθεί η λίστα αλγόριθμων παραγωγής κλειδιών (server host key). Ο εξυπηρετητής δηλώνει τους αλγορίθμους για τους οποίους διαθέτει ζεύγη δημόσιου-ιδιωτικού κλειδιού. Ο πελάτης δηλώνει τους αλγόριθμους που μπορεί να δεχθεί. Έπονται, οι λίστες αλγόριθμων κρυπτογράφησης (encryption) δεδομένων, πιστοποίησης αυθεντικότητας μηνυμάτων (MAC – Message Authentication Code) και συμπίεσης (compression) δεδομένων. Οι λίστες αυτές δίδονται κατά τις κατευθύνσεις πελάτης→εξυπηρετητής και εξυπηρετητής→πελάτης. Οι ροές δεδομένων είναι ανεξάρτητες και επιτρέπεται να χρησιμοποιηθούν διαφορετικοί αλγόριθμοι (π.χ 3DES+SHA1 και Blowfish+MD5). Εάν χρησιμοποιηθεί συμπίεση, τότε τα δεδομένα συμπιέζονται πρώτα και κατόπιν κρυπτογραφούνται.

Η φάση της ανταλλαγής κλειδιών καταλήγει στη δημιουργία ενός κοινού μυστικού *K* και μίας σύνοψης *H* που επιπλέον χρησιμεύει και ως ταυτότητα της συνόδου. Η μέθοδος ανταλλαγής κλειδιων που έχει επιλεχθεί αρχικά καθορίζει τον τρόπο με τον οποίο αυτά παράγονται. Στη συνέχεια εγκαθίστανται τα κλειδιά κρυπτογράφησης και ακεραιότητας που προκύπτουν από τα *K* και *H* με τη βοήθεια συνάρτησης κατακερματισμού (hash function). Κάθε μήνυμα που ακολουθεί κρυπτογραφείται με το κλειδί κρυπτογράφησης και η γνησιότητά του πιστοποιείται με την προσθήκη σύνοψης που παράγεται από τα περιεχόμενα του μηνύματος, τον αύξοντα αριθμό και το κλειδί ακεραιότητας του βάσει του επιλεχθέντος αλγόριθμου πιστοποίησης αυθεντικότητας μηνυμάτων. Περισσότερες πληροφορίες για το πρωτόκολλο SSH-TRANS θα βρείτε στο RFC 4253.

Για τη χρήση της υπηρεσίας SSH σε συστήματα Linux/Unix θα χρησιμοποιήσετε το πρόγραμμα ssh. Στα Windows 10 το πρόγραμμα ssh υπάρχει, αλλά δεν είναι ενοποιημένο. Μπορείτε να το ενεργοποιήσετε από το Manage optional features (*Settings → Apps → Optional features*). Στην άσκηση όμως θα χρησιμοποιήσετε το πρόγραμμα PuTTY που θα εγκαταστήσετε κατεβάζοντάς το από την ιστοσελίδα https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html.

Καταγράψτε με τη βοήθεια του Wireshark την κίνηση ενώ κάνετε χρήση της υπηρεσίας SSH του υπολογιστή edu-dy.cn.ntua.gr. Όπως πριν, εφαρμόστε φίλτρο σύλληψης `host 147.102.40.15` για να παρατηρείτε μόνο την κίνηση που σχετίζεται με αυτόν και ξεκινήστε την καταγραφή. Στο πεδίο *Host Name* του παραθύρου που ανοίγει όταν εκτελέσετε το PuTTY, πληκτρολογήστε `edu-dy.cn.ntua.gr`, στη συνέχεια κάνετε κλικ στο πρωτόκολλο *SSH* και τέλος στο κουμπί *Open*. Αν ενδεχομένως ανοίξει κάποιο παράθυρο διαλόγου, επιλέξτε *Yes* για να προχωρήσετε. Στην προτροπή `login:` πληκτρολογήστε `abcd` ως όνομα χρήστη ακολουθούμενο από `<Enter>`, ενώ στην προτροπή `Password:` πληκτρολογήστε `efgh` ως κωδικό ακολουθούμενο από `<Enter>`. Σε συστήματα Linux/Unix σε παράθυρο γραμμής εντολών γράψτε `ssh abcd@147.102.40.15` και συνεχίστε δίνοντας τον κωδικό. Σημειώνεται ότι ο χρήστης abcd δεν υπάρχει στον συγκεκριμένο εξυπηρετητή και η αναγνώριση του χρήστη θα αποτύχει. Πληκτρολογήστε `<Ctrl>+c` για να κλείσει το παράθυρο και σταματήσετε την καταγραφή κίνησης.

2.1 Ποιο πρωτόκολλο μεταφοράς χρησιμοποιεί το SSH (TCP ή UDP);

2.2 Καταγράψτε τις θύρες του πρωτοκόλλου μεταφοράς που χρησιμοποιούνται για την επικοινωνία μεταξύ του υπολογιστή σας και του edu-dy.cn.ntua.gr.

2.3 Ποια από τις παραπάνω θύρες αντιστοιχεί στο πρωτόκολλο εφαρμογής SSH; *[Προσδιορίστε τη ζητούμενη θύρα συμβουλευόμενοι τον κατάλογο πασίγνωστων θυρών στην ιστοσελίδα http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.]*

2.4 Εφαρμόστε φίλτρο ώστε να παραμείνουν μόνο τα μηνύματα SSH. Ποια είναι η σύνταξη του φίλτρου που χρησιμοποιήσατε;

2.5 Εντοπίσετε τα μηνύματα SSH τύπου *Protocol*. Αναλύοντας το αναγνωριστικό που στέλνει ο εξυπηρετητής στον πελάτη, ποια έκδοση του πρωτοκόλλου SSH και ποια έκδοση λογισμικού χρησιμοποιεί ο εξυπηρετητής; Περιλαμβάνονται σχόλια στο αναγνωριστικό αυτό; Αν ναι, να καταγραφούν.

2.6 Αναλύοντας το αναγνωριστικό που στέλνει ο πελάτης στον εξυπηρετητή, ποια έκδοση του πρωτοκόλλου SSH και ποια έκδοση λογισμικού χρησιμοποιεί ο πελάτης; Περιλαμβάνονται σχόλια στο αναγνωριστικό αυτό; Αν ναι, να καταγραφούν.

2.7 Εντοπίσετε το μήνυμα SSH τύπου *Key Exchange Init* που έστειλε ο πελάτης και βρείτε τη λίστα με τους αλγόριθμους ανταλλαγής κλειδιών (kex). Καταγράψτε το πλήθος τους και τους πρώτους δύο. *[Υπόδειξη: Κάντε δεξί κλικ στο σχετικό πεδίο στο παράθυρο με τις λεπτομέρειες και επιλέξτε Show Packet Bytes… προκειμένου να δείτε το πλήρες περιεχόμενό του.]*

2.8 Από τη λίστα των αλγορίθμων παραγωγής κλειδιών (server host key) που υποστηρίζει ο πελάτης καταγράψτε το πλήθος τους και τους πρώτους δύο εξ αυτών.

2.9 Από τις λίστες αλγόριθμων κρυπτογράφησης (encryption) που υποστηρίζει ο πελάτης καταγράψτε τους δύο πρώτους για την κατεύθυνση πελάτης → εξυπηρετητής.

2.10 Από τις λίστες αλγόριθμων πιστοποίησης αυθεντικότητας μηνυμάτων (mac) που υποστηρίζει ο πελάτης καταγράψτε τους δύο πρώτους για την κατεύθυνση πελάτης → εξυπηρετητής.

2.11 Από τις λίστες αλγόριθμων συμπίεσης (compression) που υποστηρίζει ο πελάτης καταγράψτε τους δύο πρώτους για την κατεύθυνση πελάτης → εξυπηρετητής.

2.12 Εντοπίσετε το μήνυμα SSH τύπου *Key Exchange Init* που έστειλε ο εξυπηρετητής και προσδιορίστε τον αλγόριθμο ανταλλαγής κλειδιών που θα ακολουθήσουν τα δύο μέρη. Τον εμφανίζει κάπου το Wireshark; *[Υπόδειξη: Όπως μπορείτε να βρείτε στο RFC 4253, είναι ο πρώτος της λίστας του πελάτη που υπάρχει και στη λίστα του εξυπηρετητή]*.

2.13 Από τις λίστες με τους αλγόριθμους κρυπτογράφησης που υποστηρίζει ο εξυπηρετητής, βρείτε αυτόν που τελικά θα χρησιμοποιηθεί στην κατεύθυνση πελάτης → εξυπηρετητής.

2.14 Από τις λίστες με τους αλγόριθμους πιστοποίησης αυθεντικότητας μηνυμάτων που υποστηρίζει ο εξυπηρετητής, βρείτε αυτόν που τελικά θα χρησιμοποιηθεί στην κατεύθυνση πελάτης → εξυπηρετητής.

2.15 Από τις λίστες με τους αλγόριθμων συμπίεσης που υποστηρίζει ο εξυπηρετητής, βρείτε αυτόν που τελικά θα χρησιμοποιηθεί στην κατεύθυνση πελάτης → εξυπηρετητής.

2.16 Εμφανίζει σε κάποιο σημείο το Wireshark τους επιλεχθέντες αλγόριθμους κρυπτογράφησης, πιστοποίησης αυθεντικότητας μηνυμάτων και συμπίεσης;

2.17 Ποιους άλλους τύπους μηνυμάτων SSH καταγράψατε; *[Υπόδειξη: σε ένα μήνυμα μπορεί να περιέχονται περισσότεροι του ενός τύποι.]*

2.18 Μπορείτε να εντοπίσετε τα πακέτα όπου μεταφέρεται η πληροφορία για την προτροπή login και password στην περίπτωση του SSH; Να δικαιολογήσετε την απάντησή σας.

2.19 Σχολιάστε την ασφάλεια της υπηρεσίας SSH όσον αφορά την πιστοποίηση της αυθεντικότητας, την εμπιστευτικότητα και την ακεραιότητα των δεδομένων συγκρίνοντας με άλλα πρωτόκολλα ανταλλαγής δεδομένων.

3. Υπηρεσία HTTPS

Το Hypertext Transfer Protocol Secure (HTTPS) συνήθως χρησιμοποιούνταν για τη διεξαγωγή χρηματικών συναλλαγών μέσω του παγκόσμιου ιστού καθώς και την πρόσβαση σε ευαίσθητα δεδομένα (π.χ. μηνύματα ηλεκτρονικού ταχυδρομείου). Σήμερα, για λόγους ασφάλειας, η πρόσβαση σε όλους τους δημοφιλής εξυπηρετητές ιστού γίνεται αποκλειστικά με αυτό. Για τη χρήση του HTTPS, αντί για http://, στο URI των ιστοσελίδων χρησιμοποιείται το https://. Το HTTPS είναι ένας συνδυασμός του Hypertext Transfer Protocol (HTTP) με ένα πρωτόκολλο για ασφαλή μετάδοση δεδομένων. Αμφότερα, το HTTP και το πρωτόκολλο ασφαλούς μετάδοσης, λειτουργούν πάνω από το στρώμα μεταφοράς TCP του διαδικτύου. Το πρωτόκολλο ασφαλούς μετάδοσης λειτουργεί ως υπόστρωμα *πάνω* από το πρωτόκολλο μεταφοράς και *κάτω* από το στρώμα εφαρμογής, κρυπτογραφώντας τα μηνύματα HTTP πριν τη μετάδοση και αποκρυπτογραφώντας τα κατά τη λήψη. Το HTTPS ήταν γνωστό και ως “Hypertext Transfer Protocol over Secure Socket Layer”, αλλά τώρα το πρωτόκολλο ασφαλούς μεταφοράς είναι το Transport Layer Security (TLS) αντί του Secure Sockets Layer (SSL).

Το SSL αναπτύχθηκε αρχικά από την εταιρεία Netscape το 1995 για χρήση από τους πλοηγούς ιστού κατά την κρυπτογράφηση των πληροφοριών που ανταλλάσσονται μέσω ιστού. Το TLS πρόκειται για μια βελτιωμένη έκδοση του πρωτοκόλλου SSL και συγκεκριμένα βασίστηκε στην έκδοση 3 αυτού (SSLv3). Περισσότερες πληροφορίες για τα πρωτόκολλα αυτά μπορείτε να βρείτε στην ιστοσελίδα https://en.wikipedia.org/wiki/Transport_Layer_Security.

Μια σύντομη περιγραφή της λειτουργίας του πρωτοκόλλου TLS ακολουθεί. Ο πελάτης και εξυπηρετητής TLS διαπραγματεύονται την εγκατάσταση σύνδεσης ακολουθώντας μια διαδικασία χειραψίας. Κατά τη χειραψία ο πελάτης και ο εξυπηρετητής συμφωνούν σε διάφορες παραμέτρους σχετικές με την ασφάλεια της σύνδεσης ακολουθώντας τα εξής βήματα: Client Hello, Server Hello, key exchange, Change cipher spec και encrypted handshake. Η χειραψία αρχίζει όταν ο πελάτης ζητά μια ασφαλή σύνδεση στέλνοντας στον εξυπηρετητή το μήνυμα *ClientHello*. Στο μήνυμα αυτό παρουσιάζει έναν τυχαίο αριθμό, τη μέγιστη έκδοση του πρωτοκόλλου SSL/TLS που υποστηρίζει, καθώς και μια λίστα με σουίτες κωδίκων (cipher suites) και αλγόριθμους συμπίεσης κατά σειρά προτεραιότητας. Η σουίτα κωδίκων είναι ένα σύνολο αλγορίθμων: ο αλγόριθμος για την *ανταλλαγή κλειδιών (key exchange)*, ο αλγόριθμος *πιστοποίησης ταυτότητας (authentication)*, ο αλγόριθμος *κρυπτογράφησης δεδομένων (bulk encryption)* και ο κωδικός *πιστοποίησης αυθεντικότητας μηνυμάτων (message authentication code - MAC)*. Ο αλγόριθμος για την παραγωγή και ανταλλαγή κλειδιών χρησιμεύει στην παραγωγή ενός κλειδιού (κοινό μυστικό) που θα χρησιμοποιηθεί για την κρυπτογράφηση και αποκρυπτογράφηση των μηνυμάτων μεταξύ των δύο μηχανημάτων. Είναι ασύμμετρος και έχει καλές επιδόσεις για μικρές ποσότητες δεδομένων. Ο αλγόριθμος για την κρυπτογράφηση των δεδομένων, που ανταλλάσσονται μεταξύ πελάτη και εξυπηρετητή, είναι συμμετρικός και έχει καλές επιδόσεις για μεγάλες ποσότητες δεδομένων. Με τον αλγόριθμο MAC ελέγχεται η ακεραιότητα των δεδομένων, δηλαδή, ότι αυτά δεν άλλαξαν κατά τη μετάδοση. Ο έλεγχος επιτυγχάνεται με την παραγωγή και σύγκριση συνόψεων των μηνύματος μέσω συναρτήσεων κατακερματισμού (hash functions). Τέλος, ο αλγόριθμος πιστοποίησης ταυτότητας, συνήθως βασισμένος σε κρυπτογραφία δημόσιου κλειδιού RSA, βοηθά στην επιβεβαίωση της ταυτότητας του πελάτη και/ή του εξυπηρετητή.

Ο εξυπηρετητής επιλέγει από τη λίστα του πελάτη τη σουίτα κωδίκων που θα χρησιμοποιηθεί και απαντά με το μήνυμα *ServerHello* υποδεικνύοντας την επιλογή του, την έκδοση του πρωτοκόλλου SSL/TLS που θα χρησιμοποιήσει και ένα τυχαίο αριθμό. Κατόπιν, ο εξυπηρετητής αποστέλλει[2] στον πελάτη μέσω του μηνύματος *Certificate* την ταυτότητά του με τη μορφή ενός ψηφιακού πιστοποιητικού (digital certificate). Το πιστοποιητικό περιέχει το όνομα του εξυπηρετητή, την έμπιστη αρχή πιστοποίησης (trusted certificate authority – CA) που το επικυρώνει και το δημόσιο κλειδί κρυπτογράφησης του εξυπηρετητή. Ο πελάτης μπορεί να επικοινωνήσει με την CA και να επιβεβαιώσει ότι το πιστοποιητικό είναι αυθεντικό προτού προχωρήσει στην εγκατάσταση κλειδιού κρυπτογράφησης για τη σύνοδο. Στη συνέχεια ο εξυπηρετητής στέλνει προαιρετικά (εξαρτάται από με τη σουίτα κωδικών που επιλέχθηκε) το μήνυμα *ServerKeyExchange*[3]. Τέλος, ο εξυπηρετητής αποστέλλει το μήνυμα *ServerHelloDone* υποδηλώνοντας ότι ολοκλήρωσε από την πλευρά του τη χειραψία.

Για την παραγωγή του κλειδιού συνόδου, με το οποίο θα γίνει η κρυπτογράφηση των δεδομένων, ο πελάτης μπορεί:

(α) στη διανομή κλειδιών με το σύστημα κρυπτογραφίας δημόσιου κλειδιού RSA, να παράγει ένα τυχαίο αριθμό (Pre-master secret), να τον κρυπτογραφήσει με το δημόσιο κλειδί του εξυπηρετητή και να στείλει το αποτέλεσμα στον εξυπηρετητή. Μόνο ο εξυπηρετητής μπορεί να το αποκρυπτογραφήσει χρησιμοποιώντας το μυστικό του κλειδί. Με τον τρόπο αυτό ο εξυπηρετητής και ο πελάτης μοιράζονται ένα κοινό μυστικό που δεν είναι προσβάσιμο από τρίτους. Από αυτό και τους τυχαίους αριθμούς που ανταλλάχθηκαν αρχικά με τα Client/Server Hello, θα παραχθεί το Master secret από το οποίο προκύπτουν τα κλειδιά συνόδου που θα χρησιμοποιηθούν κατά την

[2] Ανάλογα με τον επιλεχθέντα κώδικα, το βήμα αυτό μπορεί να παραληφθεί.
[3] Στην περίπτωση ανταλλαγής κλειδιών Diffie – Hellman, δείτε πιο κάτω, το στέλνει πάντα.

κρυπτογράφηση των δεδομένων στη συνέχεια. Για ένα πλήρες παράδειγμα ανταλλαγής μηνυμάτων SSL κατά τη σύνδεση πλοηγού ιστού σε εξυπηρετητή με HTTPS με τη μέθοδο αυτή ανατρέξτε στην ιστοθέση https://www.eventhelix.com/RealtimeMantra/Networking/SSL.pdf.

(β) να χρησιμοποιήσει την ανταλλαγή κλειδιών Diffie–Hellman. Στην ανταλλαγή Diffie-Hellman ο τρόπος παραγωγής του κοινού μυστικού έχει την επιπλέον ιδιότητα της εμπρόσθιας μυστικότητας (forward secrecy): εάν το μυστικό κλειδί του εξυπηρετητή αποκαλυφθεί στο μέλλον, αυτό δεν μπορεί να χρησιμοποιηθεί για την αποκρυπτογράφηση μιας συνόδου, ακόμη και εάν αυτή καταγραφεί από κάποιον τρίτο. Για τον σκοπό αυτό με το μήνυμα *ClientKeyExchange* και το προηγηθέν υποχρεωτικά σε αυτή την περίπτωση *ServerKeyExchange*, ο πελάτης και εξυπηρετητής ανταλλάσουν δημόσια κλειδιά που έχουν παράγει από τυχαία ιδιωτικά κλειδιά (δεν έχουν σχέση με δημόσια κλειδιά των Certificate). Αυτά θα χρησιμοποιηθούν για την παραγωγή του κοινού μυστικού (Pre-master key secret) και στη συνέχεια του Master secret και των κλειδιών συνόδου, όπως περιγράφθηκε πριν.

Ακολούθως, ο πελάτης στέλνει το μήνυμα *ChangeCipherSpec* λέγοντας στον εξυπηρετητή ότι η επικοινωνία από εδώ και πέρα θα είναι κρυπτογραφημένη. Κατόπιν, ο πελάτης στέλνει το μήνυμα *Finished* κρυπτογραφημένο (*EncryptedHandshakeMessage* στην καταγραφή) που περιέχει το αποτέλεσμα της συνάρτησης κατακερματισμού επί των προηγούμενων μηνυμάτων της χειραψίας. Ο εξυπηρετητής θα προσπαθήσει να το αποκρυπτογραφήσει και να επιβεβαιώσει την ορθότητα του αποτελέσματος της συνάρτησης κατακερματισμού. Εάν αυτό γίνει επιτυχώς, ο εξυπηρετητής στέλνει το δικό του *ChangeCipherSpec* καθώς και το μήνυμα *Finished* κρυπτογραφημένο (*EncryptedHandshakeMessage*). Ο πελάτης το αποκρυπτογραφεί και επιβεβαιώνει την ορθότητα του αποτελέσματος της συνάρτησης κατακερματισμού. Εάν κάποιο από τα προηγούμενα βήματα αποτύχει, η χειραψία αποτυγχάνει και δεν εγκαθίσταται σύνδεση. Από εδώ και πέρα τα μηνύματα εφαρμογής *ApplicationData* είναι κρυπτογραφημένα. Τυχόν λάθη ή οι προειδοποιήσεις κατά τη διάρκεια της χειραψίας ή της μεταφοράς δεδομένων σηματοδοτούνται με μηνύματα *Alert*. Σε περίπτωση θανάσιμου λάθους (fatal error), η σύνοδος θα διακοπεί αμέσως. Εάν πρόκειται για προειδοποίηση (warning), η πλευρά που το λαμβάνει αποφασίζει για το κατά πόσο θα συνεχίσει τη σύνοδο. Για ένα πλήρες παράδειγμα ανταλλαγής κλειδιών κατά Diffie-Hellman και μια πιο αναλυτική περιγραφή δείτε ιστοθέση www.eventhelix.com/Networking/ssl-tls/https-ssl-tls-session-for-spdy.pdf αγνοώντας μέρος περί SPDY.

Σε αυτή την άσκηση θα καταγραφούν τα μηνύματα που παράγονται κατά τη χρήση της υπηρεσίας HTTPS του υπολογιστή bbb2.cn.ntua.gr, αφού προηγουμένως αδειάσετε την προσωρινή μνήμη (cache) του πλοηγού ιστού που χρησιμοποιείτε, π.χ στον Mozilla Firefox από τη διαδρομή *History → Clear Recent History* είτε πιέζοντας ταυτόχρονα τα πλήκτρα Ctrl-Shift-Delete. Κατόπιν ξεκινήστε μια νέα καταγραφή εφαρμόζοντας φίλτρο σύλληψης ώστε να παρατηρείτε μόνο την κίνηση που σχετίζεται με τον bbb2.cn.ntua.gr. Πρώτα, επισκεφθείτε με τον πλοηγό ιστού την ιστοσελίδα http://bbb2.cn.ntua.gr/. Μόλις φορτωθεί η σελίδα, επισκεφθείτε την πάλι, χρησιμοποιώντας αυτή τη φορά το πρωτόκολλο HTTPS. Για το σκοπό αυτό, πληκτρολογήστε τη διεύθυνση https://bbb2.cn.ntua.gr/. Όταν φορτωθεί πλήρως η σελίδα περιμένετε λίγο, κλείστε τον πλοηγό ιστού και σταματήστε την καταγραφή. Να σημειωθεί ότι το Wireshark εμφανίζει τα πακέτα που μεταφέρουν τα μηνύματα του HTTPS ως TLS (Transport Layer Security).

3.1 Ποια είναι η σύνταξη του φίλτρου σύλληψης που χρησιμοποιήσατε;
3.2 Εφαρμόστε κατάλληλο φίλτρο απεικόνισης ώστε να παραμείνουν μόνο τα πρώτα τεμάχια TCP των τριμερών χειραψιών που διεξήχθησαν με τον εξυπηρετητή bbb2.cn.ntua.gr. Ποια είναι η σύνταξή του;
3.3 Σε ποιες (πασίγνωστες) θύρες του εξυπηρετητή bbb2.cn.ntua.gr γίνονται οι συνδέσεις;
3.4 Ποια από τις παραπάνω θύρες αντιστοιχεί στο πρωτόκολλο εφαρμογής HTTP και ποια στο HTTPS; *[Προσδιορίστε τις ζητούμενες θύρες συμβουλευόμενοι και την ιστοσελίδα http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.]*

3.5 Βρείτε πόσες συνδέσεις ανοίχθηκαν μεταξύ του υπολογιστή σας και του εξυπηρετητή ιστού bbb2.cn.ntua.gr στην περίπτωση HTTP και πόσες στην περίπτωση HTTPS.
3.6 Για τις συνδέσεις TCP της περίπτωσης HTTPS καταγράψτε τις θύρες πηγής.

Σε αυτό το σημείο πρέπει να αναφερθεί ότι το πρωτόκολλο TLS αποτελείται στην πραγματικότητα από δύο στρώματα. Στο κατώτερο επίπεδο και πάνω από κάποιο αξιόπιστο πρωτόκολλο μεταφοράς (π.χ. το TCP), είναι το Στρώμα Εγγραφών (Record Layer) TLS. Το στρώμα αυτό χρησιμοποιείται για την ενθυλάκωση κάποιου πρωτοκόλλου TLS ανώτερου επιπέδου, όπως είναι, το Πρωτόκολλο Χειραψίας (Handshake Protocol), το πρωτόκολλο συναγερμών (Alert Protocol), το πρωτόκολλο μετάβασης σε κρυπτογράφηση (ChangeCipherSpec) και το πρωτόκολλο εφαρμογής (Application). Πρέπει επίσης να τονιστεί ότι κάθε πλαίσιο Ethernet μπορεί να περιλαμβάνει μία ή περισσότερες εγγραφές TLS. Επιπλέον, στην περίπτωση που μια εγγραφή TLS δεν χωράει σε ένα πλαίσιο Ethernet, τότε θα χρειαστούν πολλαπλά πλαίσια για να τη μεταφέρουν.

Εφαρμόστε το φίλτρο απεικόνισης tls.record ώστε να παραμείνουν μόνο πλαίσια τα οποία περιλαμβάνουν εγγραφές TLS. Υπενθυμίζεται ότι οι εγγραφές αυτές δημιουργήθηκαν από τη χρήση του πρωτοκόλλου HTTPS με τον bbb2.cn.ntua.gr.

3.7 Αναπτύσσοντας τις επικεφαλίδες Στρώματος Εγγραφών TLS κάθε πλαισίου θα παρατηρήσετε ότι τα τρία πρώτα πεδία είναι κοινά. Ποια είναι αυτά και ποιο το μήκος τους;
3.8 Ένα από τα πεδία που καταγράψατε στο παραπάνω ερώτημα είναι ο *τύπος περιεχομένου* (*content type*). Να καταγραφούν οι διαφορετικές τιμές για όλες τις εγγραφές TLS που έχετε καταγράψει και τα αντίστοιχα πρωτόκολλα TLS (π.χ. Change Cipher Spec – 20). *[Υπόδειξη: Υπενθυμίζεται ότι ορισμένα πλαίσια μπορεί να περιλαμβάνουν περισσότερες από μία εγγραφές TLS.]*
3.9 Για το πρωτόκολλο χειραψίας (handshake protocol) καταγράψτε τους διαφορετικούς τύπους μηνυμάτων χειραψίας που παρατηρήσατε (π.χ. Client Hello – 0).
3.10 Πόσα μηνύματα *Client Hello* έστειλε ο πελάτης και ποια η σχέση τους με τις συνδέσεις TCP που καταγράψατε προηγουμένως;
3.11 Εντοπίστε το πρώτο μήνυμα *Client Hello* που στέλνει ο πελάτης κατά τη χειραψία του πρωτοκόλλου TLS. Ποια η μέγιστη έκδοση του TLS που υποστηρίζεται από τον πελάτη;
3.12 Ποιο είναι το μήκος σε byte του τυχαίου αριθμού που περιέχει; Καταγράψτε τα πρώτα 4 byte. Τι παριστάνουν;
3.13 Εξετάζοντας την επικεφαλίδα της παραπάνω εγγραφής TLS, αναπτύξτε τη λίστα με τις σουίτες κωδίκων (cipher suites) που υποστηρίζει ο πελάτης. Να καταγραφεί το πλήθος τους και οι δεκαεξαδικές τιμές των δύο πρώτων από αυτές.
3.14 Εντοπίστε το μήνυμα *Server Hello* με το οποίο απαντά ο εξυπηρετητής στη χειραψία που ξεκίνησε με το προηγούμενο *Client Hello*. Εξετάζοντας την επικεφαλίδα εγγραφής TLS του μηνύματος, καταγράψτε την έκδοση TLS που θα χρησιμοποιηθεί καθώς και το όνομα και τη δεκαεξαδική τιμή της σουίτας κωδίκων κρυπτογράφησης η οποία τελικά επιλέχθηκε.
3.15 Ποιο είναι το μήκος σε byte του τυχαίου αριθμού που περιέχει; Καταγράψτε τα πρώτα 4 byte. του τυχαίου μέρους.
3.16 Χρησιμοποιείται κάποια μέθοδος συμπίεσης από τον εξυπηρετητή και τον πελάτη;
3.17 Ποιοι είναι οι αλγόριθμοι ανταλλαγής κλειδιών, πιστοποίησης ταυτότητας, κρυπτογράφησης και η συνάρτηση κατακερματισμού που επιλέχτηκαν; *[Υπόδειξη: Συμβουλευθείτε την ιστοσελίδα https://en.wikipedia.org/wiki/Cipher_suite για τον τρόπο ονομασίας των σουιτών και τα ονόματα των υποστηριζόμενων αλγορίθμων στις εκδόσεις TLS 1.0–1.2]*.
3.18 Εντοπίστε την εγγραφή TLS που μεταφέρει το πιστοποιητικό (*Certificate*) του εξυπηρετητή. Ποιο είναι το μήκος αυτής σύμφωνα με το πεδίο length της επικεφαλίδας;
3.19 Πόσα πιστοποιητικά μεταφέρονται; Ποια τα ονόματά τους;
3.20 Πόσα πλαίσια Ethernet χρειάστηκαν ώστε να μεταφερθεί η παραπάνω εγγραφή TLS; *[Υπόδειξη: Δείτε πεδίο [Reassembled TCP Segments] στο παράθυρο με τις λεπτομέρειες.]*
3.21 Εντοπίστε τις εγγραφές TLS σχετικά με την ανταλλαγή κλειδιών Diffie–Hellman (*ClientKeyExchange, ServerKeyExchange*). Ποιο είναι το μήκος του δημόσιου κλειδιού που

αποστέλλει ο πελάτης και ποιο του εξυπηρετητή; Καταγράψτε τα 5 πρώτα γράμματα αμφότερων των κλειδιών.

3.22 Ποιο είναι το μήκος της εγγραφής TLS που μεταφέρει στον εξυπηρετητή την υπόδειξη (*ChangeCipherSpec*) ότι η επικοινωνία από εδώ και πέρα θα είναι κρυπτογραφημένη;

3.23 Ποιο είναι το μήκος σε byte της εγγραφής TLS που περιέχει από την πλευρά του πελάτη το αποτέλεσμα της συνάρτησης κατακερματισμού (*EncryptedHandshakeMessage*) επί των προηγούμενων μηνυμάτων της χειραψίας;

3.24 Παρατηρήσατε εγγραφές TLS με την υπόδειξη (*ChangeCipherSpec*) και το αποτέλεσμα της συνάρτησης κατακερματισμού (*EncryptedHandshakeMessage*) από την πλευρά του εξυπηρετητή;

3.25 Παρατηρήσατε εγγραφές TLS του πρωτοκόλλου Alert *(Encrypted Alert)*; Από ποια πλευρά στάλθηκαν;

3.26 Γιατί νομίζεται ότι υπάρχουν; *[Υπόδειξη: Απενεργοποιήστε το ισχύον φίλτρο απεικόνισης και δείτε τι ακολουθεί στην αντίστοιχη σύνδεση TCP.]*

3.27 Επιλέξτε από την ιστοσελίδα μια φράση με λατινικούς χαρακτήρες (π.χ. “BigBlueButton”). Προσπαθήστε να βρείτε το πακέτο που μεταφέρει αυτή την πληροφορία. Τι παρατηρείτε στην περίπτωση του πρωτοκόλλου HTTP σε σύγκριση με αυτή του HTTPS; *[Υπόδειξη: Για την εύρεση του ζητούμενου πακέτου, ακολουθήστε τη διαδρομή Edit → Find Packet… και πληκτρολογήστε τη φράση προς αναζήτηση αφού βεβαιωθείτε ότι η επιλογή String είναι ενεργή.]*

3.28 Σχολιάστε την ασφάλεια του πρωτοκόλλου HTTPS σε σύγκριση με το απλό HTTP, όσον αφορά την πιστοποίηση της αυθεντικότητας, την εμπιστευτικότητα και την ακεραιότητα των δεδομένων.

Όνοματεπώνυμο:	**Ομάδα:**
Όνομα PC/ΛΣ:	**Ημερομηνία: / /**
Διεύθυνση IP: . . .	**Διεύθυνση MAC: - - - - -**

Εργαστηριακή Άσκηση 12
Ασφάλεια

Απαντήστε στα ερωτήματα στον χώρο που σας δίνεται παρακάτω και στην πίσω σελίδα εάν δεν επαρκεί. Το φυλλάδιο αυτό θα παραδοθεί στον επιβλέποντα.

1

1.1 .....

1.2 .....

1.3 .....

1.4 .....

1.5 .....

.....

2

2.1 .....

2.2 .....

2.3 .....

2.4 .....

2.5 .....

.....

.....

2.6 .....

.....

2.7 .....

.....

.....

2.8 .....

.....

.....

2.9 .....

.....

2.10 .....

.....

2.11 .....

.....