אבטחת מערכות ויישומים ברשת - שיעור #4

נושא היום: Web Secure Cmmunication & SSL (secure socket layer):

Validating a certificate – סיכום:

צד אחד מקבל תעודה מצד שני, לרוב client מקבל מ-web server., התעודות מונפקים ע"י CAs, ובמבנה ה-certificate יש היררכיה שניתן לקנן ברמות שונות של CAs כך ש-CAs מנפיקים certificate ל-CAs אחרים, עד תחתית ההיררכיה בה CA מנפיק certificate לשרת המבקש את התעודה. ל-client מספיק למצוא trusted CA יחיד שממנו יחל התהליך של בדיקת ה-certificate.

צריך לבדוק שה-certificate עדכנית, כאשר CA מפרסמים Certificate Revocation List (CRL). כמו כן צריך לבדוק שהתעודה שאותה מאשרים מכילה את ה-public key של אותו domain שעבורו ביקשנו את התעודה – בדיקות אלו נעשות ע"י ה-browser.

SSL Overview:

מטרתו לאבטח את תעבורת הרשת מ-client ל-server והפוך. הדרישות ממנגנון אבטחה כזה הם:

- Data confidentiality: הבטחת סודיות המידע המועבר ברשת. הנחת העבודה: מידע מועבר ברשת חשוף להאזנה, ולכן יש צורך במנגנון הצפנה.
- Integrity: הבטחה שהמידע לא עבר שינוי במהלך ההעברה, ולשם כך יש צורך במנגנון חתימה דיגיטלית.
- Authentication: רוצים להבטיח שהתקשורת אכן נעשית בין הצדדים שטוענים שהם מי שהם.
- Non-repudiation: מניעת התכחשות; צד שמעביר מידע לצד שני לא יוכל להתכחש לעובדה שהוא העביר את המידע הזה. חתימות דיגיטליות אסימטריות (מפתח פרטי וציבורי ליתר דיוק) מאפשרות מניעת התכחשות (שכן חותם במפתח פרטי הוא היחיד שיודע את אותו מפתח, ולכן רק הוא יכול לחתום על מידע מסויים שהועבר חתום כך). דרישה זו מיושמת ברמה האפליקטיבית ומנגנון ה-SSL לא נותן לה פתרון.

Transport Layer Security:

פרוטוקול ה-TCP מעל ה-IP נותן את האבטחה שההודעה אכן מגיעה בצורה שלמה מהמקור אל היעד, כלומר דואג שכל ה-packets שנשלחו יגיעו, דואג להתעלמות מ-packets כפולים, דואג למצב בו חסרות פקטות (ע"י מספור הפקטות). מה שלא יכול לבדוק הוא שינוי של תוכן packet – של המידע עצמו. אם מישהו מצליח להתחזות לכתובת IP ברשת, ההודעה תגיע אליו והשולח לא ידע שהכתובת אליה הגיעה ההודעה אינה של הנמען המקורי. לבעיות אלו צריך לדאוג ברמה האפליקטיבית.

הקשר בין SSL ו-TCP:

פרוטוקול ה-SSL בא לפתור את הבעיות הללו לפני הרמה האפליקטיבית. ה-SSL יושב בשכבה נוספת בין הרמה האפליקטיבית לפרוטוקול ה-TCP. שכבה זו תתן, ללא תלות באפליקציה, את התכונות הבאות:

- Secure communication.
- שקיפות לשכבת ה-TCP (לא ידרשו שינויים ב-TCP)
- העבודה ברמה האפליקטיבית תהיה כמו מול ה-TCP, כלומר ה-SSL יספק API דומה ל-TCP, רק במקום socket יהיה secure socket.

בד"כ עבודה בפרוטוקול אפליקטיבי מעל SSL יהיה עם תוספת "s" לפרוטוקול, כמו https – המאזין מעל 443. לכל פרוטוקול אפליקטיבי יש מקבילה מעל SSL. כל מפתח יכול ליצוק פרוטוקול מעל SSL, ע"י שימוש ב-Open SSL API.

מה מספק SSL:

Authentication: ה-SSL פותח במקור ע"י netscape ולכן פותח web-oriented, והוגדר כי server-authentication הוא הכרחי, ולעומת זאת client-authentication הוא אופציונאלי בלבד. הזיהוי מבוסס על certificates, וכדי ש-we server יתמוך ב-SSL הוא צריך certificate, וזו דרישה סבירה. לעומת זאת דרישה זו מכל משתמש ב-web ל-certificate היא לא קבילה, ולכן הדרישה האחרונה היא אופציונאלית בלבד.

Data encryption: ה-SSL מבצע הצפנה סימטרית, כיוון שלא ניתן לעמוד ב-overhead של הצפנה אסימטרית עבור מידע רב. בעית העברת המפתח המשותף הסימטרי נפתרת ב-SSL ע"י העברת המפתח מצד אחד לשני באמצעות הצפנה אסימטרית ל-key exchange בלבד.

Data integity: חתימה דיגיטלית מורכבת משני שלבים – תהליך ה-crypto hash המייצר message digest ושלב הצפנת ה-digest. כאן נעשה שימוש בחתימה דיגיטלית סימטרית בה משתמשים במפתח סימטרי המתואם בין השולח למקבל ע"י הצפנה אסימטרית ל-key exchange. תהליך זה נקרא MAC – message authentication code, כאשר מנגון זה משמש כאן לבדיקת שלמות ואמינות ההודעה עצמה.

SSL Protocol Arcitecture:

ה-SSL הוא two-layered protocol:

השכבה הראשונה היא SSL Record Protocol בו כל שלושת התנאים – confidentiality, integrity, הבטחת ומניעת reply מסופקים בשכבה זו.

השכבה השניה היא שכבת מעבר לרמה האפליקטיבית. השכבה השניה מכילה:

- SSL handshake protocol – פרוטוקול החלפת המפתחות ואלגוריתמי הצפנה.
- SSL change cipher spec protocol: ביט אחד שאם הוא דולק הוא מציין את תחילת היישום של כל מה שהוחלט בפרוטוקול לעיל.
- SSL Alert protocol – יפורט בהמשך.
- HTTP, other apps

SSL Record Protocol:

אחראית להעביר מידע של האפליקציה ו-SSL management data. התנאים שמספק משתמשים ב:

- Integrity: MAC בשימוש באלגוריתם דומה ל-HMAC ; ... מפורט במצגת.
- Confidentiality: אלגוריתמי הצפנה סימטריים.

מדוע נקרא Record: לוקחים את המידע שמועבר מהשכבות מעל ומחלקים אותו ל-records, שגודלו עד $2^{14} = 16{,}384\ bytes$. כל record (או בלוק) יכול לעבור גם תהליך דחיסה לפני ההצפנה (אופציונאלי). לבלוק מוסיפים headers, חתימה דיגיטלית סימטרית וכל זה עובר הצפנה. תהליך זה קורה עבור כל record (בלוק) עד סוף ההודעה.

HMAC scheme: בחתימה דיגיטלית סימטרית מחשבים digest ומבצעים עליה הצפנה. כאן נעשה אחרת: משלבים את המפתח בתהליך חישוב ה-digest, ואז מי שלא יודע את המפתח לא יכול לחשב את ה-digest. אחת הדרישות מפונקציית hash קריפטוגרפית היא שבהינתן h לא ניתן לחזור ל-m כך ש-H(m)=h. לכן פרוטוקול ה-HMAC הוא:

$$HMAC(k,m) = H[(k \oplus opad)||H(k \oplus ipad||m)]$$

כאשר ipad, opad הם ריפודים, H היא פונקצית ה-digest ו-|| הוא שרשור. היישום ב-Record layer הוא לקחת את הבלוק אחרי הדחיסה האופציונאלית, מבצעים עליו את ה-MAC ו-(m, MAC) נשלח להצפנה, שהיא השלב הבא (הצפנה לצורכי data confidentiality).

MAC verification: הצד המקבל מבצע את ה-MAC על ההודעה שקיבל (m) ובודק התאמה ל-MAC שקיבל מהשולח (זאת אחרי ההצפנה/פענוח). ישנם שני ערוצי הצפנה – אחד משרת ללקוח, והשני מהלקוח לשרת, המשתמשים במפתחות נפרדים ל-MAC. כך, גם אם התגרלה סט מפתחות אחד, לא ניתן להוציא מכך את סט המפתחות של הערוץ השני.

ל-SSL record מוצמד header המכיל:

- סוג ההודעה – מה ה-data המוכל ב-record – change_cipher_spec, alert, handshake, app. Data.
- גרסת פרוטוקול – Major version.
- Minor version.
- אורך ה-record... עוד כמה דברים.

סיכום: מידע מחולק ל-records, לכל אחת מוסיפים version, length, kind, חלוקה זו עוברת דחיסה ולאחר מכן MAC ולבסוף הצפנה. משם המידע מועבר ל-transport.

SSL connection and Session:

ה-session הוא תולדה של ה-handshake. תהליך ה-handshake שמבצע את אימות המפתחות וכו' הוא כבד יחסית (מבוסס על הצפנה אסמיטרית), ולכן מאפשרים כמה connections המתבססים על session קיים – וכך בניית connection חדש ע"ב session קיים חוסכת את ה-handshake. מה מחזיק session:

- Session id – השרת אחראי על מתן id ל-/session
- Certificate – בדיקת תעודות הדדית.
- Compression method: מכיל האם לבצע דחיסה, ואם כן באיזה אלגוריתם.

- סט אלגוריתמים קריפוטוגרפיים עליהם הוסכם.
- Master secret: מפתח בן 48 bytes שהוא ה-shared secret, המפתח הסודי משלב ה-handshake. מפתח זה הוא "מסטר" כיוון שממנו יגזרו כל המפתחות שישומשו ב-session זה. מפתח זה מזוהה עם session ספציפי.
- "is resumable" – האם מאפשרים יותר מ-connection אחד או לא.

Connection: כעת על בסיס session רוצים לבנות connection, ולשם כך חסרים המפתחות עצמם (שיגזרו מה-master secret). בשלב ה-connection, כל אחד מהצדדים מגריל מידע ראנדומי ושולח לצד השני. כל צד לוקח את המידע הראנדומי שלו ושל הצד השני, ויחד עם ה-master secret הוא יוצר זוג מפתחות לחישוב ה-digest ואת זוג המפתחות לצורך ההצפנה – זוגות כיוון שיש מפתח אחד לכל ערוץ (client-server q server-client). בבניית connection על בסיס session אין שימוש בהצפנה אסימטרית וזה יעיל יותר (לעומת session). לכן מבחינה חישובית נעדיף לבנות כמה connections מעל session יחיד.

SSL is a Statefull Protocol:

- SSL מטפל בכמה sessions במקביל.
- כל session מכיל כמה connections על גביו.

The SSL Handshake Protocol:

מטרת ה-handshake היא לבנות את כל מידע ה-session וה-connection, או רק connection אם ה-session קיים. כדי שהתקשורת תוכל להתקיים על השרת והלקוח להסכים על גרסת הפרוטוקול, אלגוריתם דחיסה (והאם תתקיים) ותיאום session ID. בנוסף צריך להחליט על אלגוריתמי הצפנה: ל-key exchange, data encryption, message digest. אלו מאוחדים למה שנקרא cipher suits. לבסוף תריך להחליט על master secrets.

Cipher suits:

- SSL symmetric cipher selection: ניתן להחליט שלא יוצפן, stream ciphrs או block ciphers.
- Digest function choice: מחליטים על אלגוריתם החתימה – אם לא יהיה בכלל, MD-5 או SHA-1.
- כמו כן נעשית החלטה על מהו האלגוריתם לחישוב ה-MAC, לאימות הזהות.

SSL key exchange algorithm:

- הלקוח בוחר pre-master_secret, מצפין אותו עם המפתח הציבורי של השרת, ושולח לו. השרת מפענח אותו (היחיד שיכול) וכך מתקבל pre_master_secret מתואם.
- על בסיס ה-pre_master_secret יחושב המסטר: הלקוח והשרת מגרילים כל אחד random data. נסמן r_c, r_s מחרוזות ראנדומיות אלו, ו-s_p סימון ל-pre_master. חישוב ה-s_m שהוא המסטר יעשה ע"י:

$$s_m = MD5\left(s_p|SHA('A'|s_p|r_c|r_s)\right)|MD5\left(s_p|SHA('BB'|s_p|r_c|r_s)\right)|MD5\left(s_p|SHA('CCC'|s_p|r_c|r_s)\right) -$$

שרשור זה יתן מפתח בן 48 ביטים.

SSL handshake overview: פירוט מלא במצגת (סכמת דו שיח).

אין סיכום מסוף השעה השניה והלאה.