גירסה 1.00 – 1.8.2004

אוסף מושגים בתחום אבטחת מחשבים / ניר אדר

מקדם הבטיחות של סמא הינו תוחלת הזמן לפיצוח הסמא בחיפוש שיטתי.

צופן הוא פונקציה המקבלת כתב גלוי P ומפתח סודי K ומוציאה כפלט כתב סתר C.

צופן שטף הוא צופן בו בלוק C_i יכול להיות תלוי ב- M_i אותו מצפינים, מפתח ההצפנה K, מספר הבלוק i ובלוקים קודמים של כתב הסתר והכתב הגלוי. צפני שטף הינם בעלי זיכרון.
דוגמא לצופן שטף: RC4

צופן בלוקים הוא צופן בו כתב הסתר C_i תלוי ב- M_i אותו מצפינים ובמפתח ההצפנה K בלבד.
דוגמאות לצפני בלוקים: DES, AES

Replay attack - התקפה על ידי הקלטת שידור שנערך בעבר ושידורו שוב.

צופן מושלם: צופן שעבורו לא ניתן להסיק מידע מכתב הסתר בנוגע לכתב הגלוי.
ניתן להוכיח: בצופן מושלם אורך המפתח חייב להיות גדול או שווה לאורך ההודעה. כמו כן כל מפתח יכול לשמש להצפנה של לכל היותר הודעה אחת.

צופן קיסר: "הזז ב- k מקומות את האלף בית" כאשר k הוא המפתח הסודי. צופן זה אינו מושלם.

עקרון קרכהוף: הצופן ידוע ופיסת המידע היחידה שחסרה למתקיף היא המפתח בו השתמשו להצפנה.

צפנים סימטריים הם צפנים המבוססים על פעולות שנחשבות בטוחות, ולרוב אינם תלויים בהנחות סיבוכיות כלשהן. צפנים סימטריים הם צפנים בהם לשני הצדדים מפתח משותף בו הם משתמשים.

צפנים אסימטריים מבוססים לרוב על בעיות שחושדים שהן קשות, אך אין הוכחה לכך. צפנים אסימטריים למשל הם צפנים המבוססים על public key כאשר לכל צד יש מפתח פרטי משלו.
שימוש בצפנים סימטריים מהיר יותר מצפנים אסימטריים.

חתימה דיגיטלית: מתבצעת באמצעות מפתח ציבורי ומפתח פרטי. הודעה חתומה מעידה כי: החותם אכן כתב את ההודעה, ההודעה לא שונתה מאז שנשלחה (שלמות), וכן השולח לא יכול להכחיש את השליחה (אי הכחשה).

MAC – (Message Authentication Codes) לצדדים A, B מפתח k משותף. כדי לאמת שהודעה m הגיעה מאחד מהם, הם יחשבו פונקציה מסוימת התלויה ב-k, כלומר $MAC = f(m,k)$. הפרוטוקול מספק: אימות השולח, בדיקת שלמות המידע. הפרוטוקול אינו מספק את תכונת אי ההכחשה (ולכן איננו חתימה דיגיטלית!), כי המקבל יכול לייצר את ה-MAC בעצמו. דוגמא ל-MAC: שרשור המפתח מוצפן בסוף ההודעה, כאשר ההודעה מוצפנת ב-CBC MODE.

RSA – צופן מפתח פומבי, מבין הראשונים שפורסמו.

<u>סיכום RSA:</u>

יצירת מפתחות:

בחר זוג ראשוניים גדולים (512 סיביות או יותר) p,q.

מצא זוג מספרים e,d כך ש-$ed = 1 \pmod{(p-1)(q-1)}$.

פרסם את (n,e) כמפתח פומבי ושמור את d כפרטי.

הצפנה של הודעה m נעשית ע"י העלאת ההודעה בחזקת e מודולו n: $c = m^e \pmod{n}$

הפענוח של כתב סתר c נעשה ע"י העלאת כתב הסתר בחזקת d מודולו n: $m = c^d \pmod{n}$

נשים לב כי מתקיים:

$$D(E(M)) = D(m^e \pmod{n}) = m^{ed} \pmod{n} = m$$
$$E(D(M)) = E(m^d \pmod{n}) = m^{ed} \pmod{n} = m$$

בקרת כניסה – אימות זהותו של המשתמש ובדיקה האם המשתמש מורשה להשתמש או לגשת אל המשאבים אליהם הוא מנסה לגשת.

פרוטוקולי Challenge Response

בכל נסיון כניסה של משתמש למערכת, המערכת שולחת לו challenge חדש. המשתמש מחזיר response ובכך מוכיח שהוא יודע את הססמא.
מימוש נכון מוגן מפני התקפת שידור חוזר.

EKE - פרוטוקול Challenge-response עם הגנה מפני התקפת מילון (ועל ידי כך מאפשר שימוש בססמאות חלשות), אימות דו כיווני ומניעת חטיפת קשר. ההגנה מפני התקפת מילון תעשה על ידי הוצאת מפתח הצפנה w מתוך הססמא החלשה. מפתח זה ישמש אותנו לצורך הצפנת מפתח a DH שישלח לצד השני. הצד השני, היודע את הססמא גם הוא, יצפין בעזרת w את ה-DH שלו, וכן ישלח challenge על סמך המפתח המשותף k. הצד הראשון יחזיר תשובה לאתגר, ויראה בכך שגם ברשותו המפתח המשותף.

SRP – שדרוג של EKE שבא להוסיף חסינות מפני חשיפת קובץ הססמאות.
בפרוטוקול זה השרת לא שומר את הססמא אלא תוצאה של חישוב עליה + salt שיישלחו מאוחר יותר אל המשתמש. התוצאה: הססמא לא נשמרת בשום מקום.

SSL – פרוטוקול אבטחה שהומצא על ידי Netscape. נמצא במודל השכבות שכבה אחת מתחת לשכבת האפליקציה, ולכן אפליקציות יכולות להשתמש בו בצורה שקופה. מורכב מ-4 שכבות משנה.
לכל אפליקציה כגון http מוגדר ערוץ נוסף עבור הגרסה המשתמשת ב-SSL, למשל https משתמש בערוץ 443.
האלגוריתם מספק אימות זהות השרת, אימות הדדי של השרת והלקוח, מהימנות הגעת המידע (מסתמך על TCP), שלמות ואימות ההודעות, סודיות ההודעות ודחיסה.
הפרוטוקול מתחיל ב-Handshake בו השרת והלקוח מאמתים את הזהות וגוזרים את המפתחות בהם ישתמשו בהמשך.
לאחר פרוטוקול זה מתחיל השידור באופן מוצפן עם המפתחות שנקבעו.

IPsec

פרוטוקול אבטחה העובד ברמת ה-IP. ממומש כשכבה נוספת אשר ממוקמת מעל שכבת ה-IP.

מספק: סודיות, אימות ושלמות המידע, אימות השולח, הגנה כנגד replay attacks.

שני מצבי עבודה עם הפרוטוקול:

- Transport mode – עבור הגנה מקצה לקצה
- Tunnel Mode עבור הגנה מרשת המקור אל רשת היעד.

IPsec משתמש באחד משני הפרוטוקולים: ESP או AH. ESP מבצע הצפה ו/או אימות השולח ובדיקת שלמות המידע. AH מבצע אימות השולח ובדיקת שלמות המידע.

מבני נתונים של האלגוריתם: SAD –מכיל רשומות עבור המידע הדרוש לצורך הגנה על התקשורת, SPD – מבנה נתונים בו מוגדרת המדיניות כיצד יש לנהוג בחבילות הנשלחות והמתקבלות.

בנוסף IPsec משתמש בפרוטוקול IKE לשם הסכמה על מפתחות.

כל רשומה ב-SAD נקראת SA.

IKE - פרוטוקול להסכמה על מפתחות, ממומש כאפליקציה העובדת מעל UDP בפורט 500.

תפקידו העיקרי - בניית SA משותף עבור IPsec.

ל-IKE שתי אופני פעולה - Main Mode המורכב מ-6 הודעות המועברות, או Aggressive Mode הכולל 3 הודעות אולם פחות בטוח.

שלבי IKE במצב Main Mode: (כל שלב 2 הודעות)

1. הסכמה על האלגוריתמים של ה-SA שיווצר.
2. החלפת מפתחות Diffie Hellman.
3. החלפת הזהויות ואימותן.

Local Exploit - פרצה שניתנת לניצול על ידי משתמש לגיטימי של המערכת.

Remote Exploit - פריצה הניתנת לניצול על ידי אדם שאינו במערכת.

TPM - ראשי תיבות של Trust Platform Module - מערכת אמינה לניהול דרישות אמון ואבטחה.

תכונות נדרשות: מינימליות, חסינות להתקפות פיסיות ולהתקפות תוכנה, מהירה.

TPM בודק שהמצב של המערכת הוא כמו שהוא אמור להיות - למשל - לפני העלאת ה-BIOS, בודק שה-BIOS מתאים לציפיות. לפני העלאת מערכת ההפעלה - בודק שקבציה לא שונו וכו'.

ישויות במודל המחשוב האמין:

ישויות מקומיות: מנהל המערכת (owner) השולט ב-TPM. TPM Entity – הישות שהיא ה-TPM. משתמש (user) המשתמש בפלטפורמה.

ישויות גלובליות:

- Certification Authority – הישות שמעידה על זהותם של תתי המערכות.
- Validation Entity – ישות המציינת שניתן לתת אמון בחלק מהמערכת, ומציגה ערכים שצריכים לעבור בדיקה לפני שהפלטפורמה מקבלתאמון. יתכנו מספר VE לאותה פלטפורמה. ה-VE מציין **מתי** החלק יכול להחשב אמין.
- Conformance Entity (CE) – ישות המעידה כי המערכת **תוכננה** בהתאם לעקרונות המחשוב האמין.
- Platform Entity (PE) – ישות המעידה כי **עותק מסוים** (של ההתקן או המודול) נבנה בהתאם לתכנון המאושר.