

פרויקט סיום -קורס מתקדם במערכות מחשב

Access Control

מאת: טל קליגמן , יובל טל
במסגרת קורס מתקדם במערכות מחשב סתיו 2017 אוניברסיטת תל אביב

3	הקדמה
3	הצורך
3	מבוא - תיאור בסיסי של הפרויקט
3	סקירת מערכות בקרת גישה קיימות
3	כרטיס בעל פס מגנטי
3	צ'יפ הזדהות מבוסס passive RFID:
4	טכנולוגיית NFC
4	המנגנון הביומטרי
4	פלטפורמה וסביבת העבודה שלנו
4	סיבות לבחירת הפרויקט
4	צורך יומימי
4	סקרנות בהשקה לעולם תוכן מוכר
5	התמקדות בתוכן ייחודי לקורס
6	תיאור
6	אפיון המערכת
6	עיצוב הקוד
6	master
6	hashtable ו DB
7	LOG ציקלי
7	BUTTON
7	BLE
7	אחרים
7	parser
7	led
9	ניתוח הפרויקט
9	מענה איומים אבטחתי
9	אדם פנימי בעל הרשאת יבצע ניסיון של פתיחה לאדם אחר באמצעות ההרשאות שלו
9	אויב בעל יכולת הצצה לזיכרון הרכיב ינסה לקרוא את פרטי ההזדהות
9	שיקולים ודילמות
9	מבנה הנתונים עם מבנה זיכרון ייעודי
10	flow התוכנית
1	

10	מענה שלם לאיומים האבטחתיים
10	בחירת BLE - צורת התקשורת
11	עיצוב קוד כללי
11	יתרונות מרכזיים
11	בקרת גישה ללא צורך ברכיב פיזי
11	מודולריות וקלות לתחזוק
11	אתגרים
11	דל משאבי זיכרון
11	אמינות - אבטחה
12	תכנות בסביבה לא מוכרת
12	כיווני המשך מרכזיים
12	צופן - הזדהות קריפטולוגית
12	חיבור לרכיבים פיזיים
12	חיבור לרשת רכיבים
13	נספחים

הקדמה

הצורך

מערכת בקרת גישה פשוטה ונוחה למתחמים שאיננה דורשת מהאדם ציוד שאינו מעבר לציוד היומיומי.

מבוא - תיאור בסיסי של הפרויקט

בפרויקט זה בנינו מערכת בקרת גישה מבוססת BLE באמצעות המכשיר הנייד. המערכת מנטרת ומאפשרת הזדהות באמצעות המכשיר הנייד בפני רכיב פשוט לצורך מעקב ואישור כניסה למתחמים. זאת בדומה למערכות בקרת גישה מבוססות טכנולוגיות NFC וכרטיס ייעודי. כחלק מהיותה מערכת חכמה, היא שומרת את פרטי ההזדהות ומבצעת מעקב כך שאדם לא יוכל 'לרמות' אותה ולהזדהות כאילו הוא נכנס בלי שיצא. עם התאמה לצרכים הספציפיים, המערכת שלנו יכולה להחליף את מערכות בקרת הכניסה הקיימות ולחסוך כרטיס ייעודי.

סקירת מערכות בקרת גישה קיימות

מערכות בקרת גישה (access control systems) הן מערכות שמטרתן לשלוט ולבקר על כניסה למתחמים (בניין, מסדרון, וכו'). אדם המעוניין להיכנס, נדרש להתאמת על מול המערכת, וזו האחרונה תאפשר/תמנע ממנו להיכנס למתחם, ואולי אף תתעד את הפעולה למטרות ניטור ותפעול. האתגר המרכזי במגננונים כאלה הוא אימות הזהות של מבקש הכניסה. מרב הגישות דוגלות ברכיב פיזי שנדרש להציג בעת המעבר. כיום קיימים לכך מספר פתרונות מרכזיים:

- כרטיס בעל פס מגנטי
מכיל שכבה פיזית ייחודית (הפס המגנטי) אשר, באמצעות קורא מתאים, ניתן לקרוא את הנתונים המוטמעים על גביו. הדבר דורש ייצור והפצת כרטיסים לכלל מורשי הגישה, והתקנת קורא מגנטי בכניסה למתחם.
- צ'יפ הזדהות מבוסס passive RFID:
בדומה לכרטיס מגנטי נדרשת הפצה לכל מורשה גישה וקורא ייעודי, אך הטכנולוגיה מתקדמת יותר. על הרכיב ישנם אנטנה ומעגל חשמלי המופעל באמצעות אנרגיה (שדה מגנטי) אשר מקורו בקורא, וכך מתקשר איתו. הדבר דורש סמיכות של לכל היותר 10cm מהקורא. צ'יפים אלו זעירים ובשנים האחרונות אף זולים, אך מחיר הקוראים גבוה מאוד.
- טכנולוגית NFC
(או active RFID) באמצעות מכשיר נייד: דורש אימות באמצעות מכשיר נייד אל מול מערכת צד בית. נדרשת קירבה פיזית של מספר סנטימטרים מהמכשיר לקורא.

- המנגנון הביומטרי

מנגנון המאפשר זיהוי באמצעות טביעת אצבע. המנגנון נחשב מאובטח ואף לא דורש רכיב חיצוני, אך עשויות להיות הפרעות שונות (כמו לכלוך, כפפות), ובמקומות בעלי תחלופת כח אדם גבוהה התפעול השוטף עשוי להיות מסורבל.

פלטפורמה וסביבת העבודה שלנו

בפרויקט עבדנו על CC1350 Wireless MCU LaunchPad של חברת Texas Instruments, הכולל מעבד ARM Cortex-M3 (ב-48MHz), ו-RF Transceiver המסוגל לשדר ב-2.4GHz (ואף ב-sub-1GHz) ותומך ב-BLE. כמו כן ישנם מגוון פיצ'רים ומודולים פריפראלים נוספים, לדוגמה, UART, I2C, Ultra-Low-Power Sensor Controller, ואף כלים נוחים יותר למפתח כמו נורות led וכפתורים. על הרכיב רצה מערכת ההפעלה TI-RTOS, וסביבת הפיתוח בה עבדנו בפרויקט זה היא Code Composer Studios.

סיבות לבחירת הפרויקט

- צורך יומיומי

אנחנו נמצאים מידי יום באזורים בעלי מערכות בקרת גישה בעייתיות. הפרויקט נוד לענות על צורך הקיים בחיי היומיום שלא נענה עם הפתרונות הקיימים כי מרבית הדרכים הקיימות כיום דורשות כרטיס/תג נוסף שעל האדם להסתובב איתו באופן קבוע. תקשורת BLE, באמצעות המכשיר הסלולרי, חוסכת את האביזר המיותר הנ"ל מכיוון שהסלולרי בכל מקרה זמין אצל מרבית האנשים ביומיום. בנוסף על כך, חלק מהמערכות הקיימות דורשות עלות ומורכבות גבוהה בהקמת תשתית וכרטיסי הזדהות, בניגוד למערכת שלנו המבוססת סלולרי ו-MCU זול המאפשרים תהליך הטמעה פשוט עם הוצאות נמוכות.

- סקרנות בהשקה לעולם תוכן מוכר

במסגרת עבודתנו אנו עוסקים בעולם תוכן אבטחתי, בחירת נושא זה הייתה הזדמנות לראות כיצד משתלב עולם התוכן היומיומי שלנו בעולם אחר - עולם ה-IOT שנלמד בקורס. השילוב בא לידי ביטוי במהות המערכת - צורך אבטחתי והשילוב של מנגנון הזדהות(שמהותי בעולם האבטחה). תהליך זה סקרן אותנו ועורר בנו עניין מקצועי רב.

- התמקדות בתוכן ייחודי לקורס

מרבית הקוד שנכתב הוא קוד שהאתגרים בכתיבתו קשורים ישירות לנושאי הקורס - הכרת תקשורת BLE על רכיב embedded, בניית מבני נתונים ברכיב דל משאבים ועוד. פרויקט זה הכוין את מאמצינו ללמידה ולא רק ליישום של ידע קיים. ניצול זה של הזמן אפשר לנו הזדמנות להתפתח ולהתקדם ביכולות המקצועיות שלנו.

תיאור

אפיון המערכת

ככתוב לעיל, המערכת מיועדת לשמש כדרך הזדהות עבור אדם המעוניין להיכנס או לצאת ממתחם מבוקר בכניסה על ידי הרשאות. השימוש בה לצורך יציאה או כניסה נעשה בשני שלבים:
הזדהות (authentication):

1. הזדהות מבוססת שם משתמש וסיסמא על גבי תקשורת BLE עם הטלפון הסלולרי.
2. לחיצה על כפתור הכניסה בצד הרלוונטי

אימות כניסה (authorization):

1. ווידוא שם משתמש וסיסמא.
2. ווידוא צד על בסיס לוגים אבטחתיים מפעילות קודמת במערכת.
3. תיעוד הכניסה להמשך.

הפרויקט מהווה הדגמה ואת ליבתה של מערכת זו ובהתאם לא ממשש אפליקציה בסלולרי וכן לא פותח דלת ממשית אלא משתמש בשתי נורות החיווי לכיוון הפתיחה.

עיצוב הקוד

• master

קוד המרכז את הלוגיקה הטהורה ומשתמש בכל יתר המודולים. נקרא כתוצאה מקבלת ערך מהמשתמש בBLE ובתחילת ריצת task המרכזי לצורך אתחולים.

• DB hashtable

בחרנו לממש DB המתחזק את פרטי המשתמשים במערכת, ומולו נבדקים הרשאות הכניסה. את מימוש בנינו באופן שמסד הנתונים הינו מכלול המיחצן API כלפי משתמשי ה-DB, יחד עם גישה מהירה למידע עצמו באמצעות hashtable. זאת כחלק מהוכחת יכולת הטמעת DB על מגוון חלקיו השונים ברכיב embedded דל זיכרון. ה-hashtable דרש מבנה נתונים משל עצמו בעל מבנה מורכב של רשימה של רשימות מקושרות. הדבר אמנם תפס מרחב זיכרון משמעותי, אך היה חיוני להכניסו לשם שלמות ה-DB. לאורך תהליך הפיתוח נאלצנו לייעל את אופן שמירת המידע, כך שינוצל המינימום ההכרחי בלבד. כך לדוגמא, איננו שומרים string בזיכרון כמה פעמים (נניח ב-log, db) אלא משיגים פוינטר לאותו אובייקט, כיוון שהדבר (כאשר קורה ב-scale רחב) יביא ל-consumption של הזיכרון.

• LOG ציקלי

אנו שומרים תיעוד (לוג) של המעברים במערכת. מפאת שיקולי זיכרון, נאלצנו להגביל את אורך הלוג. אמנם, כדי לנהל את הלוג בצורה חסכונית ביותר אנו מקצים את המשאבים

באופן ציקלי, כך שלוג חדש שיירשם (כתוצאה מכניסת משתמש) יקבל את ההקצאה של הלוג האחרון (והיה אם הלוג מלא).

• BUTTON

מודול זה מיחצן שימוש נוח בכפתורים שנמצאים על גבי הרכיב. הוא מספק לקוד שמשמש בו יכולת קליטת לחיצה על אחד הכפתורים יחד עם timeout לבחירתו. הוא עושה זאת על ידי הגדרת interrupt callback ושימוש ב semaphore לצורך timeout. כאשר באופן מדויק יותר, הפונקציה המרכזית מגדירה callback ואז מחכה על semaphore עם timeout. כך, אם הכפתור נלחץ אז callback ישחרר בתורו את semaphore והפונקציה המרכזית תדע שהייתה לחיצה.

• BLE

לצורך התקשורת עם המכשיר הסלולרי, בחרנו להשתמש בטכנולוגיית BLE הקיימת בלוח שלנו. עשינו זאת באמצעות שינויים באפליקצייה לדוגמא simple_ble_peripheral. השילוב נעשה במספר פונקציות ובראשן:

- SimpleBLEPeripheral_init - אתחול הפרופיל כרצוננו
- simpleProfile_WriteAttrCB - קליטת הערך מהמתמש, והשמה של null-terminator
- SimpleBLEPeripheral_processCharValueChangeEvent - קריאה מתוך task המרכזי ללוגיקת הזדהות ב master.

תהליך השילוב באפליקציה לדוגמא כלל שלב מחקרי של המשך הכרה לעומק(בנוסף לנעשה עד כה בקורס) של הפרוטוקול על שלביו השונים לצורך התאמה מיטבית לפרויקט.

• אחרים

בנוסף למודולים שתוארו לעיל בפרויקט יש קטעי קוד נוספים שאחראיים על חלקים אחרים.

• parser

אחראי על פרסור הקלט מהשתמש. בחרנו להפרידו מיתר הקוד מכיוון שבגרסא מתקדמת יותר יכול לקבל קלטים מורכבים מהשתמש ולשם כך חשובה המודולריות של חלק זה.

• led

אחראי על אתחול, קבלת ערך והשמת ערכים בLEDים.

תהליך הפיתוח

1. סקירת טכנולוגיות קיימות והתמודדותן עם הבעיה.
2. איפיון ראשוני של הפתרון.
3. עיצוב הפתרון ברמת ממשק תוכנתי בין מודולים:

- a. בלי , button - ללא הבנת פער איפיוני חדש.
- b. בלי hashtable.
- 4. תחילת תהליך פיתוח במתודולוגית מכוונת ספרינטים עקב הקושי בDEBUG ורצון באמינות גבוהה בין הוספת תכולות ועבודה בין מפתחים מרוחקים פיזית.
- 5. זיהוי הפער האיפיוני - תהליך עיצוב והתאמה מחדש.
- 6. שכתוב הקוד והוספת תמיכה במנגנון.
- 7. פתרון באגים אחרונים.

ניתוח הפרויקט

מענה איומים אבטחתי

המערכת, בהיותה מיועדת למטרות אבטחה, נדרשת לענות ולהתמודד עם איומים שונים. במהלך המחקר והפיתוח עלו אתגרים שהיה עלינו לתת עליהם מענה אבטחתי הולם:

- אדם פנימי בעל הרשאת יבצע ניסיון של פתיחה לאדם אחר באמצעות ההרשאות שלו המערכת נועדה לאפשר כניסה של האנשים הרשומים בלבד. אך אדם בעל הרשאות (קרי שם משתמש וסיסמא) יכול לפתוח את השער עבור אדם אחר באופן בלתי לגיטימי. לשם כך אנו מבצעים כל העת מעקב אחר הסטטוס הנוכחי של כל משתמש (נכנס/יצא) ובמידה והוא מנסה לבצע אותה פעולה פעמיים - הוא לא יקבל הרשאות.
 - אויב בעל יכולת הצצה לזיכרון הרכיב ינסה לקרוא את פרטי ההזדהות במערכת שמורים נתונים אישיים של המשתמשים (בעיקר סיסמאות, שלעיתים משמשות אף לשימושים אחרים). האיום כאן הוא אויב שמצליח למצוא חולשה המאפשרת לו לקרוא את הזיכרון ולגנוב פרטים אישיים. דוגמא לחולשה היא מקרה בו הקוד קורא מספר bytes לפי input מהמשתמש, ואז מחזיר לו את המידע חזרה. אדם שירצה לנצל את זה עשוי לבקש מספר גדול מאוד של בתים, עד שהוא יקבל bytes אקראיים מהזיכרון. אם קצת השקעה, אפשר להשיג את פרטי המשתמשים. זו רק דוגמא. יכולות להיות עוד חולשות רבות. אנו נרצה שהמידע האישי של המשתמשים יהיה מוגן על אף סיכון זה.
- הפתרון בו נקטנו הוא שמירת hash של הסיסמאות ולא את הסיסמאות עצמן. זאת בגלל ההנחה שבלתי אפשרי (כמעט) לשחזר את הסיסמא מה-hash, אבל אפשר כמובן לוודא סיסמא בהינתן ה-hash שלה.

שיקולים ודילמות

- מבנה הנתונים עם מבנה זיכרון ייעודי מבני הנתונים השונים נבחרו להתאים בדיוק לצרכים שלנו, ולוודא שלא ייווצר מצב שעקב מחסור בזיכרון המערכת לא תעבוד (לדוג' - לוג חדש לא יוכל להירשם במערכת). למעשה אנו רוצים לוודא שהגודל של ה-DB ייתמך בכל סיטואציה (כמו שב-DB אמיתי נוודא שיש נפח אחסון המוקצה ל-DB). הדרך בה התמודדנו עם העניין היא הקצאת מכלול המשאבים מראש, וניצול יעיל ככל הניתן שלהם. בנקודה זו בחרנו להקצות מראש מקום לאובייקטי ה-hash-table, וטבלה נפרדת (הטבלה המאונדקסת לפי ה-hash) למצביעים אליהם.

- flow התוכנית

במהלך הפיתוח נתקלנו בדילמה בעניין ה flow הסטדרטי של המערכת: האם נרצה משתמש קודם יזדהה ב BLE או קודם ילחץ על כפתור. בחרנו שהשתמש קודם יזין שם משתמש וסומא ורק אח"כ ילחץ על הכפתור. הסיבה לכך היא התועלת האבטחיתית שב flow זה. שכן, אחרת אדם בלי הרשאות כלל יכול בקלות ללחוץ הרבה פעמים על הכפתור ובכך 'לתקוע את המערכת' (denial of service) בגלל שהיא תחכה עד TIMEOUT להזנת פרטי הזדהות BLE. נשים לב, שבבחירתנו רק משתמש עם הרשאות יכול להגיע לסיטואציה שבה המערכת מחכה רק עבור קלט של אדם ספציפי. ראוי לציין שלכאורה במצב זה המערכת איננה חסכונית כלל בחשמל (ה BLE כל הזמן דולק) וזה בעייתי במערכות מסוג זה. אבל, המערכת שלנו צפויה להיות נייחת בתוך בניין ולכן סביר שיהיה לה מקור מתח קבוע זמין.

- מענה שלם לאיומים האבטחתיים

ניצבה בפנינו דילמה כאשר הבנו את התרחיש הבא - אדם הרוצה להכניס אדם אחר בהכרח יוכל להכניס אותו על אף כל קושי שנערים בפניו. לדוגמא, במבנה הנוכחי של המערכת משתמש יתחבר למערכת למטרת יציאה (לחיצה על כפתור היציאה), ואז יתחבר שוב ויכניס את האדם האחר. בשיטה זו עוקפים את האכיפה של "אי אפשר להיכנס/לצאת פעמיים ברצף". מצד שני, אם בודקים את העניין, נראה שלא ניתן להגן מפני זה - כי אין כל דרך לוודא בתוך wireless (כלומר ble) האם המשתמש/מכשיר נמצא בפנים או בחוץ. אז תמיד הוא יוכל לבצע את המשחק של "כאילו לצאת" ואז לחזור. עם זאת, תרחיש זה מתקבל על הדעת בעיקר מפני שבשאר המערכות הקיימות כיום בשוק (המשתמשות בטכנולוגיות אחרות, אף פיזיות) אין התמודדות עם התרחיש הזה כלל. על אף זאת, תמיד נרצה להקשות על האויב ככל הניתן - לכן אנו מאלצים אותו לעשות קודם כאילו הוא יוצא (פיזית ללחוץ על כפתור היציאה) ורק אז הוא יוכל להכניס את האדם השני.

- בחירת BLE - צורת התקשורת

בחרנו להשתמש דווקא בטכנולוגיית ה BLE כי רצינו טכנולוגיית RF שהטלפון הסלולרי יכול להתממשק איתה בקלות. רצינו דווקא טכנולוגיית RF כי רצינו שכל אדם ללא ציוד נוסף מעבר ליומיומי, להשתמש במערכת שלנו. יש לציין שתתכנה טכנולוגיות RF אחרות שגם יתאימו ובחרנו דווקא זו עקב התמיכה קיימת והרחבה בצד הלקוח בפרוטוקול.

- עיצוב קוד כללי

התלבטנו האם לעצב את הקוד בצורה טורית - מודלים הקוראים להבא בתור, או בצורת כוכב - master מרכז הכל. כפי שתואר קודם לכן, בחרנו באופציה השנייה. זאת מכיוון שצורה זו לדעתנו מודולרית יותר (כל מודול עובד אך ורק מול master) ויותר מתאים למתודולוגית

הפיתוח שרצינו. עיצוב זה יותר מתאים למתודולוגיה מכיוון שאפשר לנו לדמות בקלות פעולה מנוונת של מודולים מסוימים לצורכי בדיקה של האחרים. בנוסף על כך, מבנה זה איפשר לנו לרכז את הלוגיקה בצורה טהורה ולחסוך טעויות.

יתרונות מרכזיים

- בקרת גישה ללא צורך ברכיב פיזי
מערכת בקרת הגישה מורכבת מתשתית מינימלית המכילה את ה-CC1350 LaunchPad, דבר המשפיע על נוחות הטמעה ועלויות נמוכות. צד הלקוח של המערכת הינו המכשיר הסולרי בלבד. עובדה זו חוסכת את הסרבול שבתחזוק והפצת כרטיסי/תגי גישה וכדו', ע"י שימוש באביזר שבכל מקרה נמצא ברשות המשתמש.
- מודולריות וקלות לתחזוק
מבנה התוכנה בנוי בשכבה מעל התקשורת ובאופן מודולרי המאפשרים התממשקות ל-API של החלקים השונים (DB, BLE וכו'). הדבר מאפשר הוספת יכולות ושדרוגים נוספים מבלי הצורך להכיר לדוגמא לעומק את העבודה עם ה-BLE מעל TI-RTOS.

אתגרים

- דל משאבי זיכרון
הרכיב שהשתמשנו בו בעל זיכרון flash מאוד מוגבל. עקב כך נדרשנו לבנות מנגנון הקצאת זיכרון משלנו ולבצע התאמות במבני הנתונים שדורשים זיכרון רב. עניין זה לידי ביטוי בעיקר ב-DB ו-LOG. [להרחבה בנושא זה - ראה 'עיצוב קוד'].
- אמינות - אבטחה
המערכת שלנו היא מערכת שפותרת בעיה אבטחתית - בקרת גישה. על אף היכרותנו עם התחום, לא התנסינו בשילובו בעולם ה-IOT והכה LOW-LEVEL. זה הציב בפנינו איזמים שאיננו רגילים אליהם כמו הדילמה 'flow התוכנית' וכן דילמות נוספות שהוצגו. עקב כך נדרשנו לבצע כמה סבבי ופיתוח וכן למפות את האיזמים והציפיות האבטחתיות.
- תכנות בסביבה לא מוכרת
סביבת הפיתוח, הפלטפורמה, מערכת ההפעלה והפרוטוקול BLE היו חדשים יחסית בעבורנו ודרשו שלב מחקרי והיכרות. על אף מספר התרגילים שהיו בקורס, תחילת הפיתוח בפרויקט דרשה התרגלות והכרה לעומק יותר של מסמכי התיעוד, הספריות, הפורמים הרלוונטים באינטרנט(בהם עוד אנשים נתקלו בבעיות דומות) ועקרונות ועיצוב התשתית שעליה בנינו את הפרויקט. באתגר זה בלט החוסר בספריות סטנדרטיות רבות והמורכבות שבפעולות שהן פשוטות בסביבות הרגילות לנו. בנוסף, חלקנו לא מפתחים בשפת C ביומיום, וחזרה על עקרונותיה נדרשה גם כן.

כיווני המשך מרכזיים

- צופן - הזדהות קריפטולוגית

כיום, ההזדהות שקיימת במערכת מבוססת על שם משתמש וססמא. אבל, המערכת מממשת תשתית תקשורת ובסיס נתונים נוחים מאוד שניתן להרחיב בקלות לתמיכה בצופן. כלומר, מודולריות הקוד מאפשרת כבר היום התאמת פונקצית הפרסור בלבד לצורך שליחת מורכבות יותר על בסיס הודעות BLE. בנוסף, ניתן לקיים גם פרוטוקולים על בסיס תקשורת זו.

- חיבור לרכיבים פיזיים

כיום, חיווי הפתיחה של המערכת הם הפעלת הledים. בשלב הטמעה, ניתן לחבר מנועים רלוונטים שייפתחו את הדלת באמת.

- חיבור לרשת רכיבים

כיום, המערכת כולה קיימת בתוך רכיב יחיד. יישום מתקדם יותר שלה, יכול לכלול רשת של רכיבים כאלו שמתשרים לצורך בקרת כניסה ממספר כניסות לאותו המתחם.

נספחים

1. קוד המקור של הפרויקט.

2. סרטון הדגמת שימוש.

<https://drive.google.com/open?id=1YHf9C6WVm9fLgrSfCXEnCEalCsNKPtP5>

3. קישור לאתר הקורס:

<https://sivantoledoacademic.wordpress.com/teaching/advanced-computer-systems-fall-2017/>