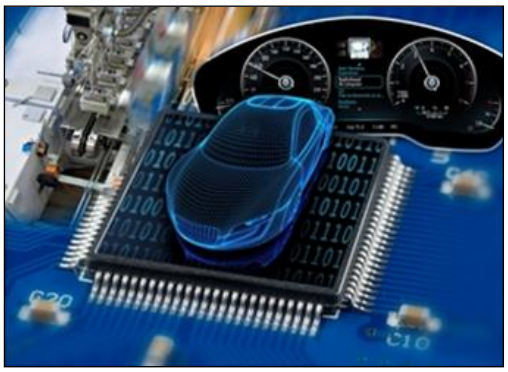
אבטחה משובצת - חלק ב'

מאת לירן בנודיס

הקדמה


[מקור: firmenpresse.de]

מערכות משובצות תופסות חלק גדול יותר ויותר בחיי היום-יום שלנו, בין אם אנו מבחינים בכך או לא, מקונסולות משחק ועד למערכות בקרת טילים ומספרן של מערכות אלו גדל מידי יום.

כיום, אבטחת מידע בצורה כזו או אחרת היא דרישת בסיס במערכות משובצות רבות כמו מכשירי כף-יד (PDAs), אוזניות אלחוטיות, כרטיסים חכמים, נתבים, חומות-אש (firewall) וכו'. ההתקדמויות הטכנולוגיות שאפשרו את הפיתוח של מוצרים אלו הובילו גם להתקדמות מקבילה בתחכום של ההתקפות על מערכות אלו.

במאמר הקודם בנושא סקרנו כמה הבדלים בין מערכות משובצות למחשבים שאנו מכירים, וראינו שבכל הקשור לאבטחה של מערכות משובצות נעשה שימוש בסט כלים שונה, טכניקות שונות ומטרות התקיפה גם הן שונות. במאמר זה נציג מהן דרישות האבטחה עבור מערכות משובצות, נראה במה הן שונות מדרישות האבטחה בתוכנות מחשב וכיצד הן משפיעות על עיצוב המוצר, נסקור ונציג את סוגי התוקפים, המשאבים שברשותם ומטרותיהם. לבסוף נצלול קצת יותר לפרטים, נציג טכניקות אבטחה שנכשלו ונבין למה וכמה טכניקות בהן משתמשים במערכות היום לאבטחה של מערכות משובצות.

דרישות האבטחה

לרוב אמצעי האבטחה הנמצאים במכשירים המוכרים לנו יש מטרה אחת, להגן על מידע. המידע דורש הגנה לא רק בעת שליחתו בערוץ לא מאובטח אלא גם בעת טיפול במידע במערכות הנמצאות אצל משתמשי קצה. חולשה במערכת הקצה כמו גישה קלה למפתחות הסודיים המשמשים להצפין או לפענח מידע רגיש עלולה להפיל את כל מנגנוני ההגנה.

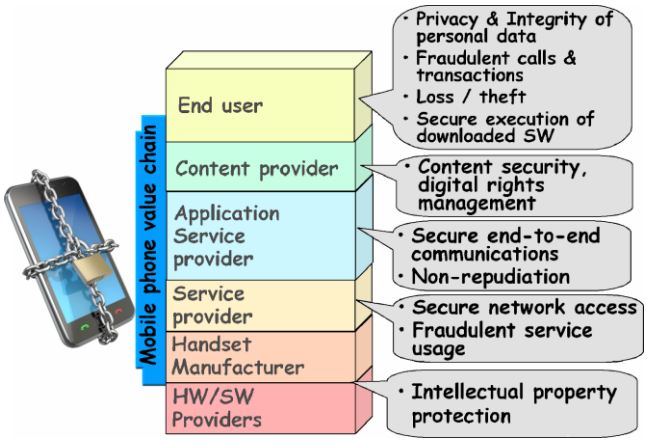
שליחה של מידע רגיש בצורה מאובטחת מעל רשתות לא מאובטחות עושה שימוש בהצפנות וזהו נושא די מסובך בעצמו, אך הטיפול במידע בתוך המערכת עצמה הנמצאת אצל משתמש הקצה דורשת טיפול זהיר הרבה יותר מכיוון שלרוב מנסים להגן על המידע מפני המשתמש עצמו.

מי ומה?

חשוב להבין שקיימים הרבה גופים המעורבים בייצור, בהפצה ובשימוש של מערכות משובצות. הדרישות של כל גוף מהגופים המעורבים עלולות להיות שונות. כך לדוגמה, אם נשקול את דרישות האבטחה עבור פלאפון חדש המסוגל לבצע שיחות, העברת מידע ומולטימדיה. דרישות האבטחה יהיו שונות בין יצרנית ששבב שלה נמצא בתוך הפלאפון החדש (למשל המאיץ הגראפי), יצרנית הפלאפון, ספקית התוכן ומשתמש הקצה.

[מקור: nvidia.com]

דרישות האבטחה של משתמש הקצה יהיו קשורים כנראה לאבטחת המידע הפרטי שלו הנמצא על הפלאפון והמידע שהמשתמש מעביר דרך הפלאפון לגורמים אחרים. דרישות האבטחה של ספקית התוכן יכולות להיות קשורות לאבטחת המידע ומניעת העתקה של המידע המגיע ממנה לפלאפון. לעומת זאת, יצרנית הפלאפון עלולה להיות מוטרדת מכך שמישהו ינסה להעתיק את או להחליף את הקושחה הפלאפון.



[דרישות האבטחה מפלאפון סטנדרטי]

עבור כל אחד מהמקרים קבוצת התוקפים משתנה גם כן. לדוגמה, ספקית התוכן לא יכולה לסמוך על המשתמש שלא יעתיק את התוכן ולכן מתייחסת אליו כמשתמש זדוני. כאשר שתי ישויות מעבירות מידע רגיש מעל רשת לא מאובטחת, הם צריכים לוודא כי קיימות פונקציות האבטחה הבאות:

- **סודיות המידע** - מגן על המידע מפני האזנה של גורם לא רצוי
- **שלמות המידע** - מוודא שהמידע לא השתנה באופן לא לגיטימי
- **אימות עמית** - מוודא שהמידע מועבר לגופים הרצויים ולא למתחזים

איפה?

העובדה שמערכות משובצות, לעיתים קרובות, נמצאות פיסית אצל צד הנחשב עויין מבחינת אחד הגופים האחראי לייצור המערכת, יוצרת מצב שבו יש לממש שיטות להעברה בטוחה של מידע מהמכשיר החוצה ומחוץ למכשיר אליו. בנוסף לכך, למנוע ניסיונות גישה לא מאושרת מהמכשיר עצמו. נסווג את דרישות האבטחה אם כן לשניים:

- **דרישות אבטחה למעבר מידע**:

המידע ברשתות ציבוריות עובר דרך מספר של נקודות ביניים הנחשבות לא בטוחות. לכן המידע הרגיש שמעבירה המערכת דרך הרשתות הציבוריות, צריך להיות מבולבל בצורה כזו שיהיה לא מועיל עבור כל ישות אשר לא מורשית לגשת למידע. את זו ניתן להשיג בעזרת מנגנונים קריפטוגרפים כמו הצפנות סימטריות ואסימטריות, הסכם מפתחות, חתימות דיגיטליות, ואישורים דיגיטליים. מכיוון שנושאים אלו זהים בין מערכות משובצות לבין מערכות מחשב רגילות אנו לא נרחיב עליהם.

[מקור: blackmereconsulting.com]

- **דרישות אבטחה בתוך המכשיר**:

כל הצפנה דורשת מפתח כלשהו, בין אם זה מפתח פרטי וציבורי או מפתח סימטרי. בטיחות המידע העובר בשימוש בהצפנות אלו תלוי בבטיחות המפתחות הללו. הבעיה הנוצרת במערכות משובצות היא שמפתחות אלו לעיתים רבות מאוחסנים על המערכת עצמה!

רמת האבטחה של מידע בתוך המכשיר משתנה על פי הטבע של המידע עליו מנסים להגן. דרישות אבטחה של מידע המאוחסן בתוך המכשיר שכיחות יותר בקרב מכשירים המאחסנים או מעבירים מידע המוגן בזכויות יוצרים כמו סרטים או תמונות מאשר מכשירים המאחסנים מידע פרטי של המשתמש. זה בעיקר מכיוון שבמכשיר המחזיק מידע פרטי של המשתמש עצמו נוטל באחריות לגבי מי מקבל גישה פיסית למכשירו. בהנחה שמישהו הצליח לחלץ את המפתחות, הוא יקבל גישה רק לקבצים הפרטיים של אותו משתמש ואותן המפתחות לא יכלו לשמש אותו במכשירים אחרים מכיוון שמפתחותיהם שונים. לעומת זאת, אם נקח בחשבון כי משתמש יכול לגשת למפתחות של מכשיר

המשמש להזרמת מדיה, הוא יכול להוריד סרטים, לפענח את התשדורת וליצור אינסוף עותקים של השידור המפוענח.

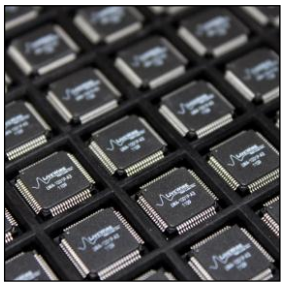
לא עוד פיצ'ר

עד כה דיברנו על אבטחה של מערכות משובצות בצורה די אבסטרקטית ,אך יש לזכור כי מערכות משובצות רבות מוגבלות על ידי הסביבה בה הן עובדות והמשאבים שברשותם. עבור מערכות שכאלו יש מספר גורמים המעבירים את אבטחת המידע מלהיות עוד פונקציה או פיצ'ר במכשיר לגורם משמעותי בתכנון המוצר, לדוגמה:

- כוח העיבוד של מערכות משובצות מוכרע בקלות על ידי הדרישות של אלגוריתמי ההצפנה, על היצרנים להחליט האם להשתמש בכוח עיבוד הנמצא במערכת ולהפחית מן הביצועים, להוסיף כוח עיבוד למערכת ובכך לייקר את המוצר, או (חס וחלילה) לוותר על אבטחת המערכת.

- מערכות ניידות לעיתים קרובות מוגבלות במקום אחסון, סוללה ויכולות חישוב. הגבלות אלו רק מחמירות כאשר דורשים מהמכשיר להיות גם מאובטח.

- מגבלות החישוב של מערכות משובצות והרצון ליצור אותם בעלות נמוכה יוצר פיתוי להשתמש ב-Application-specific integrated circuit (ASICs) בכדי לבצע הצפנות מהירות. אך אלו מגבילות את גמישות המערכת. עולם האבטחה מתקדם מהר מאוד, פרוטוקולים ושיטות הצפנה חדשות צצות וחולשות מתגלות במנגנונים קיימים, הארכיטקטורה של מערכות משובצות צריכה להיות גמישה מספיק בכדי להתמודד עם התפתחות זו.



[מגש של שבבי ASIC]

נציג כעת רשימה (חלקית?) של התוקפים הפוטנציאלים:

- **חובבנים** - בעלי ידע מוגבל, לעיתים קרובות מנסים לנצל חולשות מוכרות ולרוב אין בבעלותם כלים מתוחכמים.
- **מומחי אבטחה** - בעלי התמחות טכנית רחבה ובעלי מכשור וכלים מתקדמים.
- **ארגונים ממומנים** - ברשותם מומחי אבטחה בעלי מימון גדול. אשר מסוגלים לבצע ניתוחים רחבים למערכות, לבצע התקפות מתוחכמות, כשברשותם הכלים המתקדמים ביותר.

משאבים	האקר חובב	האקר מומחה	ארגוני פשיעה	ממשלות
זמן	מוגבל	בינוני	גדולה	גדולה
תקציב	קטן מ-1000$	10k$-100k$	גדול מ-100k$	לא ידוע
יצירתיות	משתנה	גדולה	משתנה	משתנה
סיכוי להתגלות	גבוה	גבוה	נמוך	נמוך
מטרה	אתגר	פרסום	כסף	משתנה
מפרסמים הישגים?	כן	כן	משתנה	לא

מטרות התקיפה

קיימות מספר רב של מטרות, אך ניתן לחלק את "הסטנדריות" לכותרות הבאות:

- **העתקה** - הנדוס לאחור של מוצר מסוים ויצירת מוצר דומה עד זהה.
- **גניבת שירות** - קבלת שירות שעולה כסף בחינם (כמו משחקים ל-XBox).
- **התחזות וקבלת הרשאות** - זיוף זהות בכדי לקבל הרשאות למערכת.
- **Privilege Escalation** - קבלת שליטה נוספת על המערכת או פתיחת אפשרויות נוספות של המערכת.

התקפות ומגננות

נסיון העבר מלמד שהאקרים קוראים תיגר על החוזק התיאורטי של אלגוריתמים קריפטוגרפים לעיתים רחוקות, ובמקום זאת הם מחפשים ומנצלים חולשות בתוכנה ובחומרה של המימוש. בחלק זה נראה שאם אבטחת המוצר לא נלקחה בחשבון בכל שלבי התכנון, יהיה ניתן למצוא ולנצל חולשות וכך לעקוף את אבטחת המוצר.

התקפות תוכנה

תוכנה היא חלק מרכזי במחשבים (ובמערכות משובצות) ומקור חיוני לוויטמינים, נוגדי חמצון ופרצות אבטחה. כיום, תוכנות הולכות ונהיות גדולות יותר, נכתבות בשפות גבוהות ובשימוש בספריות וכל זה יוצר קוד מאוד מסובך שקשה לבדוק והסיכוי שימצאו בו חולשות גדול. התקפות תוכנה כנגד קרנל של מערכות הפעלה, כמו אלו המבוצעות על ידי RootKits, עלולות לפגוע גם במערכות משובצות. לקרנל יש גישה מלאה לכל המערכת והוא יכול לתקשר עם כל רכיב בה. זה אומר שתוקף שהשתלט על הקרנל יכול לקרוא ולכתוב לזיכרון של ה-BIOS. בכל מכשיר יש כמה מגה ביטים של זיכרונות פלאש (Flash ROM), זיכרונות אלה כמעט אף פעם לא מנוצלים לחלוטין ובדרך כלל יש בהם מספיק מקום לאחסן Back Doors, וירוסים ועוד.

עבור תוקף, היכולת להחדיר זדונה לאזור שכזה בזיכרון מפתה. שכן קשה לנתר אותם, הם חסינים להפעלות מחדש והתקנות מחדש של המערכת, והם לרוב בלתי נראים לתוכנה הרצה על המערכת. בכדי להגיע לזיכרון חומרה כזה צריך לרוץ ברמת דרייבר. וירוס חומרה יכול לגרום למערכת לקבל מידע כוזב מהחומרה (נשמע מוכר?) או לגרום למערכת להתעלם מאירועים מסוימים ולא להעבירם לתוכנה.

וירוסים שכאלו נמצאים "בטבע" כבר הרבה זמן, למרות שווירוס ה-CIH (צ'רנוביל) זכה לפרסום גדול על ידי המדיה - וירוסי תוכנה שכותבים את עצמם ל-BIOS היו קיימים הרבה לפניו. היום, כאשר כמעט בלתי אפשרי למצוא מערכת משובצת שאינה משתמש בזיכרון EEPROM, וירוסים מסוג זה מסוכנים מתמיד.

הנדוס לאחור

בכדי למצוא פרצות במערכת (שלא בגישת ה-Black Box) יש צורך להבין איך היא פועלת. לשם כך מבצעים הנדוס לאחור. כאשר מדובר על תוכנה עושים זאת לרוב בעזרת IDA, OllyDbg וכלים דומים. אך כאשר מגיעים לחומרה, ישנן שיטות רבות ומגוונות.

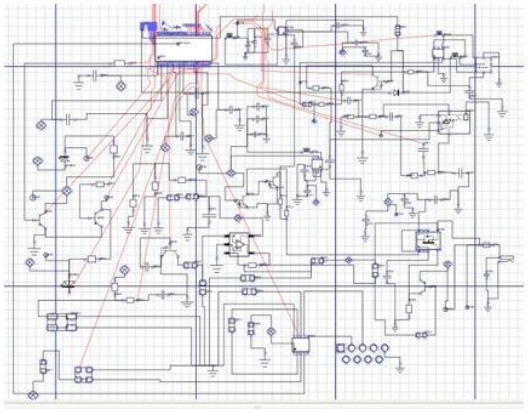
אריזת המוצר

עבור מוצרים שונים אריזת המוצר צריכה לשרת צרכים שונים. למשל עבור מערכת כמו ה-XBox, יש צורך לדעת שהאריזה נפתחה לאחר מעשה, כך כשמשתמש ששיחק עם האריזה יבוא ויבקש להחליף את המוצר הוא יענה בסירוב. בשביל זה קיימות מדבקות אחריות. עבור מערכות צבאיות לעומת זאת, אולי נראה משהו בסגנון של השמדה עצמית בעת פתיחה.

[מקור: freewebs.com]

הבנת ה-PCB

בכדי להבין כיצד מערכת מסוימת בנויה, איזה רכיבים מתקשרים ומה תפקידו בכוח של כל רכיב ורכיב יש להבין את ה-PCB, והדרך הטובה ביותר היא לצייר תרשים.

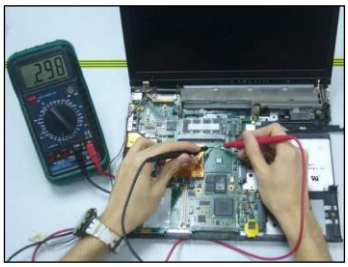


בכדי לצייר תרשים צריך לבצע שני דברים: לזהות את הרכיבים ולאתר כיצד הם מחוברים ביחד.
מעגלים פשוטים ניתן לשרטט יחסית בקלות:

- בדרך כלל, על הרכיבים רשום שם היצרן והדגם וחיפוש קצר בגוגל יניב לנו מפרט מלא של אותו רכיב.
- את החיבורים אפשר לזהות באופן ברור מהסתכלות על ה-PCB או תוך שימוש בתוכנות גראפיות.

בקישור הבא ניתן לראות מדריך מלא כיצד עושים זאת בעזרת מצלמה ותוכנה גראפית:
http://www.instructables.com/id/How-to-reverse-engineer-a-schematic-from-a-circuit

אך לא כל המעגלים פשוטים, והיום קשה מאוד למצוא מעגלים שכאלו. מכיוון שהיום מדפיסים את המעגלים במספר שכבות בכדי לחסוך במקום ולייצר כרטיסים קטנים יותר. אבל זה עדיין לא אומר שאי אפשר לשרטט את ה-PCB.


[מקור: images01.olx.in]

ישנם כמה שיטות לזהות את חיבורים ב-PCB הבנוי בשכבות:

- להפריד פיסית בין השכבות במעבדה
- לעבור עבור כל שני חיבורים, ובעזרת מולטי מטר לראות האם הם מחוברים
- לצלם בעזרת X-Ray ושיטות שונות את ה-PCB (כמו בקישור הבא)

אמנם לוח רב שכבות הוא רק התקדמות טבעית ולא פעולה מכוונת להקשות על אלו המנסים להנדס לאחור את המערכת. אך לפעמים היצרן רוצה לעשות לנו חיים קשים בכוונה ואנו עלולים למצוא רכיב ב-PCB שלא נצליח לזהות בקלות. זה קורה מכמה סיבות:

- על הרכיב לא רשום דגם או יצרן
- הרכיב יוצר על ידי היצרן עצמו
- הרכיב יוצר במיוחד עבור המערכת
- היצרן הסיר את הכתוב בכוונה
- הרכיב מכוסה באפוקסי (במקרה הזה: http://www.youtube.com/watch?v=kTPXKA66baQ)

גם במקרה הזה זהו לא סוף העולם, אבל זה בהחלט עושה לנו חיים קשים.

הבנת ה-SoC

עבור מצבים כמו שהצגנו, בהם אנו לא יכולים לזהות רכיב בקלות (לא רשום עליו הדגם), יש צורך להשתמש בשיטות מתוחכמות ולעיתים בציוד יקר. הרכיבים המעניינים יותר הם רכיבי SoC (System On Chip), מכיוון שהם מסובכים בהרבה משאר הרכיבים שאנו עלולים למצוא ב-PCB.

נתחיל משיטות פשוטות לזהות רכיב כללי:

- ראשית נסתכל על הרכיב עצמו:
  - כמה רגליים יש לו? - ידוע למשל שלמעבדים יש המון רגלים מכל הכיוונים.
  - מה צורתו? - מעבדים נוטים להיות ריבועים בעוד שזיכרונות הם בדרך כלל מלבניים.
  - יש לו מאפיינים ייחודיים כלשהם?
- נסתכל לאיזה רכיבים אחרים הוא מחובר. אם אנחנו מזהים את אלו, אולי נצליח להסיק מסקנות לגבי הרכיב עצמו. הגיוני שזיכרון יהיה מחובר למעבד ושדוגם יהיה מחובר לאיזשהו קלט אנלוגי.
- ניתן להסניף את ה-Bus המוביל אל הצ'יפ ולנסות להסיק מסקנות. האם מאתרים רגל שהיא שעון? מה התדירות? האם ניתן להסיק מסקנות מהמידע שעובר? אולי לפענח אותו אפילו.
- נעשה Fuzzing ונראה מה הפלטים עבור קלטים שונים.

אם אחרי כל אלו לא הבנו מהו הרכיב (בשלב זה נניח שהוא SoC), או שאנו רוצים להבינו יותר טוב מתקדמים לשיטות המסובכות יותר.

- שימוש באמצעים כימים ומכניים בכדי לחשוף את הרכיב .
- לסרוק את הרכיב בשיטות שונות .

בדרך כלל בתוך SoC ניתן למצוא חלקים מוכרים. מהתבוננות בחלקים אלו ובעזרת אוסף המידע שניתן להשיג בשיטות השונות ניתן לנסות ולהבין את הרכיב, ממש כמו פאזל.

בשיטות אלו ונוספות חברת ChipWorks הנדסה לאחור את רכיב ה-A6 באייפון החדש (להרחבה)

מתקפות

בנוסף למתקפות תוכנה קיימות מתקפות פיסיות ומתקפות Side-Channel המנצלות את מימוש המערכת או את התכונות המאפיינות אותה. מתקפות פיסיות ומתקפות Side-channel בדרך כלל מסווגות כמתקפות חודרניות (invasive) ומתקפות לא חודרניות (non-invasive):

מתקפות חודרניות כוללות השגת גישה למערכת, מחקר שלה בנוסף לשינוי והתערבות במערכת ומימושה. מכיוון שמתקפות מסוג זה כנגד מעגלים מודפסים דורשות ציוד יקר, הן בדרך כלל קשות יותר לביצוע ולשחזור. דוגמאות למתקפות כאלו הן מחקר מקיף של מיקרו מערכות והנדוס לאחור כפי שהצגנו בחלק הקודם.

מתקפות לא חודרניות, כפי שהשם מרמז, הן מתקפות שלרוב לא דורשות את פתיחת המוצר. למרות שפיתוח וביצוע של מתקפות כאלו דורש יצירתיות והשקעת זמן, הן נוטות להיות זולות וניתנות לשחזור יחסית בקלות.

מתקפות פיסיות

עבור מערכות הנמצאות על PCB, ניתן לבצע מתקפה פיסית על ידי האזנה לתקשורת בין רכיבים שונים, עבור מערכות SoC נוצר צורך בהאזנה בעזרת כלים מתקדמים ויקרים (מאמר בנושא: Low-Cost Chip Microprobing). השלב הראשון במתקפות כאלו הוא פתיחת המוצר וחשיפת הרכיב בעזרת חומצה. לאחר מכן יש למצוא את החיבורים אותם רוצים להסניף, חיבורים מעניינים הם פיוזי וחיבורים בהם עובר תוכן (Data), את אלו מאתרים בעזרת הנדוס לאחור או ניסוי וטעייה. לאחר מכן נותר רק להסניף את המידע.

בשיטה זו ניתן להוציא יחסית בקלות מידע המאוחסן בזיכרון הפנימי של ה-SoC, כמו גם את מרחב הזיכרון ומגבלות המעבד. ניתן גם לחלץ את הפקודות אותן מריץ המעבד ואת רמות ה-Cache השונות שלו. מתקפות כאלו נחשבות קשות לביצוע עקב הציוד היקר שנדרש בכדי לבצען. למרות זאת, ניתן לבצען פעם אחת בכדי לתכנן מתקפות Side-channel עם המידע שנאסף.

מתקפות תזמון

פעמים רבות אפילו אם המערכת מחשבת תוצאה נכונה, זה לא מבטיח הגנה. בשנת 1996 פול קוצ'ר הציג כיצד ניתן לקבוע את ערכם של מפתחות בעזרת מדידות של שינויים קטנים בזמן שלוקח למערכת לבצע חישובים קריפטוגרפים.

בכדי להבין את המתקפה ניתן לחשוב על חישוב כלשהו המתחיל בקלט קבוע וכולל מספר צעדים, כאשר כל צעד עושה שימוש בביט רנדומאלי אחד ולוקח זמן לא ידוע (ומשתנה). עבור קלט מסוים שתי מקרים אפשריים עבור הצעד הראשון וכמות הזמן שהצעד ייקח תלוי בביט שניתן כקלט.

בביצוע המתקפה התוקף נותן למערכת סדרת קלטים ומודד את הזמן שלוקח למערכת לעבד כל קלט. לאחר מכן התוקף מחשב את הקורלציה בין הזמנים שנמדדו ובין הזמן המשוערך בהינתן שהביט שנעשה

בו שימוש בצעד הראשון הוא 0, את אותן הקורלציות התוקף מחשב גם עבור המקרה שבו הביט שנעשה בו שימוש בצעד הראשון הוא 1. המדידות שבהן הביט הוא זה שהמערכת באמת משתמשת בו (הביט זהה לביט שבמפתח) צריכות לתת את הקורלציות הגבוהות ביותר. לאחר מכן התוקף חוזר על התהליך עבור ביטים נוספים.

מה שמעניין במתקפה זו היא שה"פתרונות" שנראים כביכול ברורים לא עוזרים. כך למשל נסיון לייחס ערכים מוגדרים (לא רציפים) לסכום הזמן הכולל של כל החישובים (למשל, כל החישובים ייקחו זמן שהוא כפולה של 10ms) או נסיון להוספה של גורם רנדומאלי לחישובים יוסיפו רעש למדידות, אך כמו שכל סטטיסטיקאי טוב יודע, על רעש מסוג זה ניתן להתגבר בהינתן מספיק מדידות. כמובן שאם כל החישובים ייקחו אותו זמן בדיוק ההתקפה תהיה חסרת תועלת (אך כשחושבים איך לממש זאת, זה ממש לא טריוויאלי).

אנליזת צריכת כוח

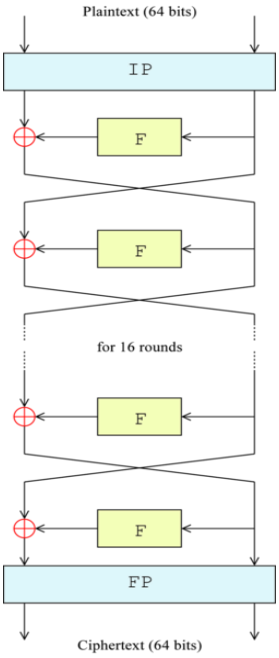
מתקפות זמון הן לא הדרך היחידה בה מכשיר "מדליף" מידע. לדוגמה, הזרם הנצרך על ידי רכיבי חומרה משתנה בהתאם לחישובים אותו הוא מבצע. רוב רכיבי הקריפטוגרפיה ממומשים בעזרת שערים לוגים, אלו מורכבים בעזרת טרנזיסטורים. ברוב המעגלים המודפסים רכיבים אלו יוצרים את צריכת החשמל העיקרית. יש להבין שרכיבים אלקטרונים מתנהגים כמו מכונת מצבים, צריכת החשמל עולה כאשר עוברים בין מצבים הרבה פעמים או כאשר עוברים בין מצבים בשערים עם קיבולת גדולה יותר. ישנם שני סוגים עיקריים של תקיפות מסוג אנליזות צריכת כוח, Simple Power Analysis (SPA) ו- Differential Power Analysis (DPA).

מתקפות SPA מסתמכות על הבחנה כי עבור מערכות מסוימות, ניתן להשתמש בפרופיל צריכת הכוח עבור חישובים קריפטוגרפים על מנת לזהות את המפתח בו נעשה שימוש. לדוגמה, ניתן להשתמש ב-SPA על מנת למצוא הבדלים בין צריכות הכוח בעת הפעולות הכפל והשורש הנעשות בעת חישובי מודולו באלגוריתם ה-RSA, ובכך לשבור את האלגוריתם. במקרים רבים נעשה שימוש ב-SPA בכדי לפשט מתקפות Brute Force. כבר הראו בעבר שמספר המפתחות האפשריים באלגוריתם DES על מעבד 8-ביט עם 7 בית של מידע יכול לרדת מ-2^{56} ל-2^{40} בעזרת שימוש ב-SPA.

מתקפות DPA עושות שימוש במידע סטטיסטי על מנת לזהות את המפתח ממידע מסובך ורועש המתקבל ממדידות צריכת החשמל. נציג התקפה על אלגוריתם ה-DES:

האלגוריתם בנוי מ-16 סיבובים, DES מבצע שמונה מעברי ב-S-Box, כל S-Box לוקח קלט של שישה ביטים מן המפתח ומבצע פעולת XOR עם שישה ביטים של הרגיסטר R (הימני) ופולטת ארבעה ביטים. הפלט מסודר מחדש ואז מתבצע XOR עם הרגיסטר L (השמאלי). ולבסוף מחליפים בין התוצאות ב-R ו-L.

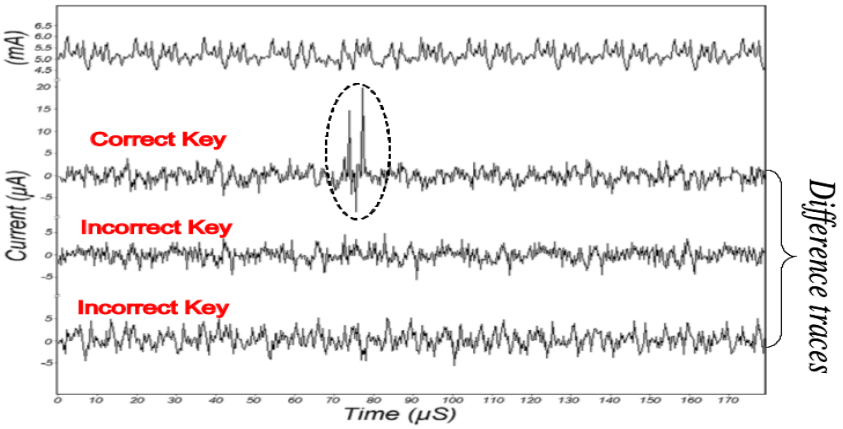
נגדיר פונקציה $D(C,b,K_s)$ ככזו שמחשבת את הערך של ביט $0 \le b \le 32$ של ה-DES בתחילת הסיבוב ה-16 עבור טקסט מוצפן C. כאשר המפתח בעל 6 התווים הנכנס ל-S-Box המקביל לביט b מיוצג על ידי $0 \le K_s \le 2^6$. נשים לב שאם K_s אינו נכון, $D(C,b,K_s)$ ייתן תוצאה נכונה עבור b בהסתברות של חצי עבור כל טקסט מוצפן C.



[מקור: wikimedia.org]

בכדי לממש מתקפת DPA התוקף דוגם את צריכת הכוח שדורש הרכיב עבור החישובים m פעמים ויוצר וקטור דגימות $T_{1..m}[1..k]$ כאשר בכל וקטור k דגימות. בנוסף התוקף שומר את הטקסט המוצפן $C_{1..m}$.

ניתוח DPA עושה שימוש במדידות של צריכת הכוח על מנת לקבוע האם הניחוש של K_s הוא נכון. התוקף מחשב סדרת שוני בת k דגימות $\Delta_D[1..k]$ על ידי מציאה של ההפרש בין הממוצע של כל המדידות עבורן $D(C,b,K_s)$ הוא 1 והממוצע של כל המדידות עבורן $D(C,b,K_s)$ הוא 0.



אם K_s שגוי, הביט שחושב באמצעות D יסטה מהערך האמתי עבור חצי מהטקסטים המוצפנים C_i. לכן הבחירה של הפונקציה $D(C_i,b,K_s)$ תהיה ללא כל קורלציה לחישובים האמיתיים שנעשים על ידי הרכיב. אם נעשה שימוש בפונקציה רנדומאלית בכדי לחלק את הווקטורים לשני קבוצות, אז ההפרש בין הממוצעים יתקרב ל-0 ככל שמספר הדגימות (k) יתקרב לאינסוף. כך נזהה שהניחוש שגוי. לעומת זאת, אם הניחוש נכון אז הערך שייתן $D(C_i,b,K_s)$ יהיה שווה לערך האמיתי של הביט b בהסתברות 1. וכך צעד אחר צעד ניתן לגלות את המפתח.

מתקפות אנליזת צריכת חשמל מהוות איום גדול מכיוון שכמעט ולא קיימים מוצרים המוגנים מפניה. המתקפה זולה, קלה למימוש ולא חודרנית, מה שמקשה על זיהוי התקיפה.

סיכום

נושא האבטחה במערכות משובצות הוא נושא חם מאוד שבו מתבצעים מחקרים רבים. חברות רבות מתעסקות בנושא בין אם בניסיונות לאבטח מוצרים ובין אם להנדס אותם לאחור. במאמר זה ראינו מגוון רחב של התקפות ושיטות הנדוס לאחור של מערכות משובצות. מערכות אלו תופסות חלק נרחב בחיי היום-יום שלנו וככל שהזמן יעבור חלק זה יעשה ילך ויתרחב.

בשונה מאבטחת תוכנה בה באופן תיאורטי אפשר ליצור תוכנה ללא פרצות, בכל הנוגע לאבטחה של מערכות משובצות הנמצאות בידי משתמש זדוני זהו רק חלום רטוב. מערכת משובצת נחשבת מאובטחת אם המידע שנשיג כאשר נפרוץ אותה לא שווה את הכסף שהתהליך יעלה. אבסטרקציה לא תמיד קיימת במערכות משובצות ולכן בהתעסקות איתם לעיתים נדרש ידע בנבחי הקרנל, מערכות הפעלה, באלקטרוניקה, ועוד.

ראינו שבתכנון מערכת שכזו יש לאפיין את הדרישות של השותפים השונים ליצירת המערכת מבעוד מועד. ישנם הרבה פרמטרים ואספקטים שיש לקחת בחשבון, ולהסתכל על חלקי המערכת בנפרד וכמכלול. כמובן שהחומר שהצגנו במאמר זה הוא רק קצה הקרחון ויש עוד הרבה לכתוב בנושא...

מקורות

- Security as a New Dimension in Embedded System Design
- Security needs in embedded systems
- Low Cost Chip Microprobing
- Differential Power Analysis
- Introduction to Embedded Security - Black Hat
- BIOS Protection Guidelines
- Timing Attacks on Implementations of Di e-Hellman, RSA, DSS, and Other Systems
- Low Cost Attacks on Tamper Resistant Devices