
WPA2: Key Reinstallation AttaCK

מאת אפיק קסטיאל (cp77fk4r) ויובל (tsif) נתיב

הקדמה

ב-16 לאוקטובר, פרסמו החוקרים Frank Piessens ו-Mathy Vanhoef מקבוצת המחקר "DistriNet" מידע טכני אודות מתקפה על הפרוטוקול WPA2 שאותה כינו:

"Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2"

או בקיצור: "KRACK".

עד כה, המימושים השונים של WPA2 נחשבו כדרך הבטוחה ביותר לאגן על רשתות Wireless, הן כמשתמשים ביתיים והן כארגונים, וגם אם נמצאו ופורסמו מתקפות שונות על הפרוטוקול הן דרשו כוח חישוב לא קטן אשר הפך אותן לכמעט לא פרקטיות. הפרסום הנ"ל עורר (ונכון לכתיבת שורות אלו - עדיין מעורר) בהלה לא קטנה בתחום, שכן המתקפה הנ"ל אינה מצריכה מהתקף יכולות עיבוד גבוהות וגם - נמצא שהיא אפקטיבית לכמעט כלל המימושים השונים של פרוטוקול זה אצל ה-Vendor-ים השונים.

הרבה כבר נכתב על המחקר ועל ההשלכות שלו החל מחוקרים שהגיעו למסקנה שזהו יום הדין ועד לחוקרים שטענו שההשלכות של המחקר והבעיות האלה הינן שוליות ועוד רעש תקשורתי.

במאמר זה ננסה להביא את המידע הטכני הרלוונטי אודות המתקפה, ממה היא נובעת, מה הן באמת השפעותיה וכיצד ניתן להתגונן.

אך לפני כן, איך אפשר בלי קצת רקע?

הגנה על רשתות אלחוטיות

רבות נכתב (גם במסגרת המגזין) על הצורך באבטחת רשתות Wireless, ולכן לא נפרט על כך יותר מדי, אך הנקודה החשובה ביותר שיש להבין בעניין היא שההבדל המהותי בין רשתות קוויות לרשתות אל-חוטיות הוא שברשתות אל-חוטיות נקודת הממסר אינה יודעת מה המיקום המדויק של הלקוח (ויותר מכך - מיקום הלקוח יכול להשתנות בכל רגע נתון) ונכון להיום, אין לה דרך לשלוח ללקוח הספציפי את המידע באופן אישי. על מנת להתמודד עם בעיה זו, המידע מופץ באוויר לכלל הרשת. נשמע מסוכן? בהחלט, אף בר-דעת לא היה מעז לשלוח את פרטי ההזדהות שלו לחשבון הבנק אם הוא היה יודע שכל מי שמחובר



לרשת יכול באופן הפשוט ביותר לצפות במידע. ובדיוק כך חשבו גם החבר'ה מ-IEEE, ולכן בעת הנדסת התקן 802.11 הם הכניסו שכבת הצפנה אופציונלית בשם WEP (קיצור של Wired Equivalent Privacy).

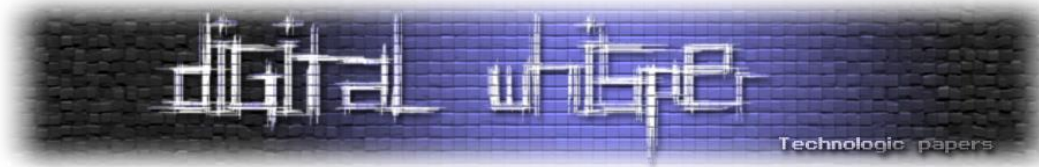
פרוטוקול זה היה נפוץ מאוד, הן בגרסאות WEP 64bit (לשימוש במפתח של 40bit ו-IV של 24bit) והן בגרסאות WEP 128bit (לשימוש במפתח של 104bit ו-IV של 24bit), ובתחילת עידן הרשתות האלחוטיות נחשב ל-"מספיק בטוח". הרעיון הוא לספק שכבת הצפנה שעל גביה תעבוד הרשת, וכך מי שאין לו את המפתח, אומנם יכול להסניף את התדר, אך לא יכול לפענח ולהבין את התוכן.

מי מהקוראים אשר מכיר קצת היסטוריה, יודע שלא עברו הרבה בימים באוויר לפני שהאקרים הראו כי הפרוטוקול הנ"ל הוא לא חומה מספיק גבוהה ול-WEP מספר חסרונות וכשלים בסיסיים. כשלים כגון החולשות בהנדסה של WEP כדוגמת העובדה שכלל הרשת מוצפנת באותו המפתח, זאת אומרת שברגע שיש לי את סיסמאת הרשת אני יכול להאזין לכלל התעבורה (כך שגלישה ברשת של בית הקפה השכונתי המוגנת ב-WEP עם סיסמה אינה בטוחה, כי המידע שלי יהיה זמין וגלוי לכל לקוח אחר). או חולשות במימוש של WEP, כדוגמת שליחת ה-IV באופן גלוי או שימוש ב-IV חלש מאוד (24bit נותן לנו 16,777,215 אופציות שונות ל-IV, מה שמגדיל משמעותית את הסיכוי לשימוש חוזר באותו ה-IV) וכך להחלשת מנגנון ההצפנה המבוסס RC4.

באותו הזמן נכתבו מספר רב של סקריפטים וכלים שאפשרו גם למי שלא מבין כלל בקריפטוגרפיה ליזום מתקפות אלו ולפרוץ לרשתות WEP, כגון הכלים AirCrack, Kismet, AirSnort. קהילות ההאקינג אכלו להמליץ על כרטיסי Wireless חיצוניים אשר מומלצים לשימוש בכלים אלו והחלו להימכר Kit-ים יעודיים לפריצה לרשתות אלחוטיות.

במקביל לגילויים אלו, עלה משמעותית השימוש ברשתות Wireless, הן במשק הבייתי והן במשק העסקי, יותר ויותר משתמשים ביתיים התקינו נתבים אלחוטיים, יותר ויותר חברות החלו לפרוש רשתות אלחוטיות במקום ה-LAN החביב והמוכר, כך שכיום כבר לא ניתן לקנות מחשב נייד עם יציאת RJ45 בכלל...

מסיבות אלו ונוספות, נוספו שיפורים רשמיים יותר ורשמיים פחות לפרוטוקול זה, כגון WEP2, WEPPlus, Dynamic WEP, כל מימוש מתמודד אחרת עם הבעיות שעלו במימוש המקורי (לדוגמא: אחד השינויים שהביא איתו WEP2 היה הגדלת המפתח וה-IV ל-128bit וכך להקטין את הסיכוי לשימוש חוזר באותו IV), אך בשל הכשלים הנוספים שהיו בפרוטוקול, נראה שלא היה מנוס אלא לכתוב שכבת הגנה חדשה.



קצת על WPA ו-WPA2

פרוטוקולי תקשורת אלחוטיים הפכו להיות שכיחים בנוף היום יומי של כולנו. בשנת 2004 IEEE הודיעו על התקן 802.11i ובמסגרתו על שחרור פרוטוקול WPA2 אשר היווה שידרוג לפרוטוקול WPA ששחרר מעט לפני כן (2003). במאמר זה אנחנו נתייחס בעיקר לגרסה הנפוצה שרובנו מכירים הידועה בשם WPA2-Personal, פרוטוקול זה הוא הנפוץ מבין השניים ומשמעותו הוא ביסוס על PSK - Pre Shared Key. הגרסה הנוספת של WPA2 מגיעה בתצורת השימוש לארגונים (WPA2-Enterprise) והינו תהליך אימות מבוסס שרתי RADIUS. ההבדל העיקרי בין השניים הוא שבשני תהליך האימות לא מתבצע מקומית בנתב אלא על ידי שרת נוסף.

כאשר אנו מתארים את הביטוי WPA2-PSK אנו בעצם מתארים את העובדה שתהליך האימות מבוסס על מפתח משותף. חשוב להבין זאת כי אין אנו מזכירים את אופן ההצפנה עצמה. כאשר נעבור לדון בשיטת ההצפנה עצמה אנחנו עוברים לבחירה המגוונת בין CCMP ל-TKIP. שתי הטכנולוגיות הללו נחשבות מאובטחות מאוד בעבר וגם היום. נתחיל מסקירה של הפרוטוקולים הללו.

TKIP - Temporal Key Integrity Protocol

הפרוטוקול הנ"ל נחשב למיושן יחסית והוצג על ידי IEEE ביחד עם פרוטוקול WPA הראשון. מטרתו העיקרית של הארגון בהצגת הפרוטוקול היה למנוע את רוב הבעיות שהתרחשו בפרוטוקול ה-WEP שנסקר כאן קודם. לכן הוצגו כמה שינויים עיקריים. פרוטוקול ההצפנה נשאר כשהיה, RC4 שהינו מסוג Stream. יחד עם זאת, הוכנס תהליך ערבול מפתחות (Key Mixing) יחד עם IV טרם אתחול מנגנון ה-RC4. בנוסף הוכנס לתוך החבילות על מנת לוודא שאין הזרקה של חבילות מחוץ לסדר. במידה ורכיב יקבל הודעה מחוץ לסדר החבילות המכשיר יבצע drop. שינוי אחרון חביב; לכל חבילה מצורף MIC (קיצור Message Integrity Check) באורך 64bit. כל הדברים הללו התאחדו סביב הרעיון העיקרי של מניעת שימוש באותו מפתח להצפנת החבילות. צפנים סימטריים רגילים למגוון של התקפות אשר מבוססות על שימוש חוזר/קרוב של מפתח כאשר הידועה ביותר היא ההתקפה Attack Known Plain Text בה התוקף יכול להניח קיום של מידע מסוים בחבילה (TCP Headers, HTTP Headers, וכו') ומשם לנסות "לפתור" מפתח מתאים וממנו לגזור את שאר המפתחות ל-session.

CCMP - CCM mode Protocol

גם כאן מדובר בפרוטוקול שעיקר עיצובו נועד למנוע את הבעיות החמורות אשר התגלו בפרוטוקול ה-WEP. ההבדל העיקרי והמשמעותי ביותר מבין שאר חבילות ההגנה המצויות בתקנים אלחוטיים הוא הבסיס על חבילת הצפנה סימטרית מסוג בלוק ולא מסוג Stream. חשוב לציין שגם CCMP וגם TKIP הינם פרוטוקולים המיועדים לאפיין ו"לסדר" את כל הסוגיות הקשורות לאימות, הצפנה, החלפת מפתחות ובקרת גישה.

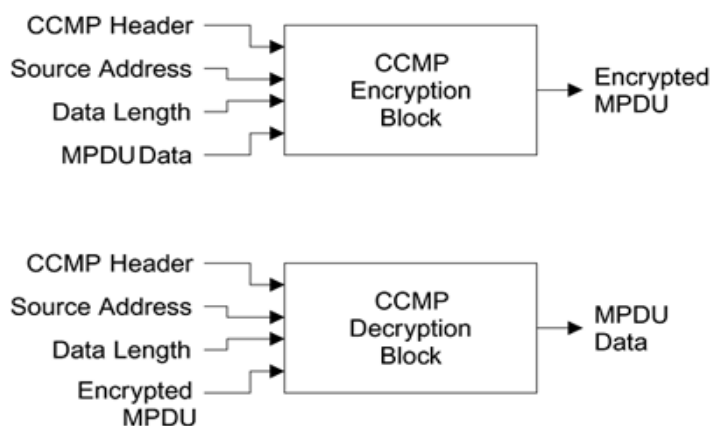
למרות ש-CCMP נשמע פרוטוקול זר, הינו פרוטוקול המיועד להגדיר את כל תהליך ההצפנה והאימות כאשר ההצפנה מבוססת על AES המפורסם והידוע.

תהליך ההצפנה מורכב מהשלבים הבאים:

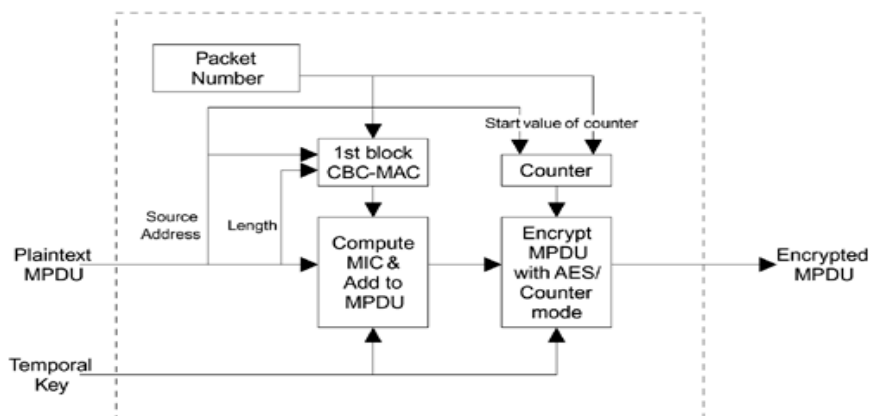
1. ההתקן מקבל הודעה לשליחה (MPDU). ההודעה הזאת כוללת את ה-Headers המתאימים. כתובת ה-MAC מועתקת ונשמרת לשלב מאוחר יותר.
2. מ-Headers של הודעת ה-MPDU מחושב MIC באורך 8 ביטים ונוצר Header ריק של הודעת CCMP. ה-MIC מחושב על ה-Header יחד עם Nonce על מנת למנוע שידור חוזר.
3. ה-MIC מצורף ל-Data של ההודעה.
4. ה-MIC וה-Data עוברים הצפנה ולאחר מכן מצרפים את ה-Header של ה-CCMP.
5. כתובת ה-MAC המקורית מצורפת ל-Headers החדשים שלא מוצפנים יחד עם המידע המוצפן. ההודעה משודרת.

יש לציין שה-Header של ה-CCMP אינו מוצפן באף שלב מכיוון שעל הלקוח להיות מסוגל לפענח ולהבין את ההודעה (ובכלל לדעת שעליו לקרוא את ההודעה). ל-Header של ה-CCMP יש שתי מטרות עיקריות:

1. למנוע שידור מחדש על ידי צירוף של Packer Number הידוע גם כ-PN (באורך של 48-ביט).
2. במקרה שבו מדובר בהודעה קבוצתית ישנו דגל שיאמר ללקוח בעזרת איזה מפתח עליו לפענח את ההודעה.



תהליך ההצפנה עצמו:





לא נפרט עוד על פרוטוקולים אלו במאמר זה, מפני שזה אינו סקופ המאמר, אך חשוב לזכור שפרוטוקולים ממשפחה זו רגישים מאוד לשימוש הצפנה בעזרת אותו המפתח, וכאשר נעשה שימוש חוזר באותו המפתח - די בקלות יהיה ניתן לפענח את ההודעה המקורית (למתעניינים: קראו על המתקפה "Crib Dragging").

לחיצת ידיים 4 שלבית

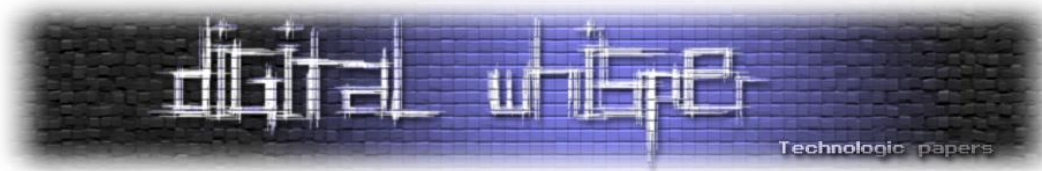
תהליך ה-4Way Handshake הינו השלב הראשון בעת התחברות לרשת המוגנת בתקן 802.11i. והמטרה שלו הוא לאפשר הן לצד המזדהה (לדוגמא - עמדת הקצה) והן לצד המזהה (לדוגמא - הנתב) לאמת כי הצד השני מחזיק ב-Pre-Shared Key או ב-Pairwise Master Key מבלי באמת להציג אותו. הנתב מעוניין לזהות את המשתמש על מנת להוכיח כי הוא אינו משתמש זדוני ועמדת הקצה מעוניינת לזהות את הנתב על מנת להוכיח כי מדובר ברשת אותנטית ולא ב-Evil Twin שתוקף הקים לטובת גניבת סיסמת ההזדהות לרשת.

מלבד הסיבה הזו, הבנת הליך זה חשוב מאוד - מפני שבו נמצאה החולשה בה עושה שימוש המתקפה KRack. אך לפני שנצלול לעומק העניין, בואו נבהר מספר מושגים, הבנת המושגים הנ"ל רלוונטית להבנת המשך העניין, אך טוב לזכור כי חלקם לקוחים מתחום הקריפטוגרפיה ולא מתחום רשתות ה-Wireless באופן ספציפי.

- **Nonce** - מספר אקראי שתפקידו להגביר את אפקט הראנדומיזציה של פעולת ההצפנה, ליצור ייחודיות למופע הספציפי של החבילה, למנוע מתקפות כמו Replay Attack וכו', חשוב מאוד לעשות שימוש יחיד במספר הזה ולא לחזור עליו, חזרה עליו שוב ושוב עלולה להקל על התוקף בעת ניסיון שבירת הסיסמה. במאמר זה נשמור על ההגדרות ההגדרה שמצויות בשאר החומרים הכתובים ונדגיש כי למרות ש-Nonce הינו ביטוי כללי למספר האקראי הנ"ל, פעמים רבות נתייחס אל S-Nonce כאל Nonce שמקורו מהלקוח המבקש להתחבר לרשת ו-A-Nonce שמקורו מרכיב התקשורת המנהל.

- **PSK** - קיצור של Pre-Shared Key, סוד שנקבע ע"י שני הצדדים מבעוד מועד, על מנת לבצע את ההזדהות כל צד ירצה לאמת כי הצד השני אכן מחזיק בה אך מבלי לחשוף את התוכן שלה. הסוג הנ"ל הוא אינו המפתח לרשת, אך בהחלט משתמשים בו בהתליך חישוב המפתחות (כך למשל ניתן להשתמש בסיסמה אחת לרשת אך במפתח הצפנה שונה עבור כל עמדת קצה - מה שימנע מרכיבים אחרים ברשת לפענח את התקשורת כולה).

- **PMK** - קיצור של Pairwise Master Key, יהיה בשימוש במידה וברשת נעשה שימוש ברשת הזדהות חיצוני. בעת שימוש ב-WPA2-Personal אין שימוש ברשת שכזה וה-PSK משמש בתור PMK, אך בעת השימוש ב-WPA2-Enterprise נעשה השימוש ברשת כזה ובעת הליך ההזדהות מתבצע שימוש גם ב-PMK. לטובת ייצור מפתח שכזה משתמשים בדרך כלל בתשתית EAP



(קיצור של Extensible Authentication Protocol) המאפשרת הזדהות ע"ב יחידת זיהוי חיצונית (לדוגמא Active Directory או שרת RADUIS) - המפתח הנ"ל הוא אחד המפתחות החשובים בהליך האימות. בשום שלב לא נרצה לשדר אותו. ה-PMK נוצר ע"י הפעלת PBKDF2 באופן הבא:

$$PMK = PBKDF2(\text{Hash_Function}, PSK, SSID, \text{Num_of_Hash_Iterations}, PMK_Size_In_Bits)$$

לדוגמא, שימוש נפוץ:

$$PMK = PBKDF2(HMAC-SHA1, PSK, SSID, 4096, 256)$$

לטובת העמקה, תוכלו לשחק באופן אינטרקטיבי עם העניין באתר הבא:

https://asecuritysite.com/encryption/ssid_hm

- **PTK** - קיצור של Pairwise Transient Key, מפתח זה הוא חיבור של ה-PMK, שני ערכי Nonce שמיוצרים אחד ע"י הנתב (ANonce) והשני ע"י עמדת הקצה (SNonce), וכן, כתובות ה-MAC של הנתב ושל עמדת הקצה, על המחרוזת שנוצרת מחיבור כלל המחרוזות הנ"ל (בסדר הזה) מפעילים Pseudo Random Function לטובת יצירת HASH שאיתו נוכל להשתמש:

$$PTK = PRF(PMK + AP_Nonce + WS_Nonce + EP_MAC + EP_MAC)$$

שימו לב שרוב הנתונים שיוצרים את ה-PTK ידועים לכלל, הסוד היחיד שמרכיב אותו הוא ה-PMK, וזה בדיוק תפקידו של ה-PTK, להוות נגזרת של ה-PMK בכל פעם שנרצה לבצע שימוש המבטיח ידיעה של ה-PMK מבלי באמת להשתמש ב-PMK. אנו נשאף להשתמש רק פעם אחת בכל PTK שנוצר. כל שימוש באותו PTK מעבר לפעם הבודדת - מסכן את כל בטיחות הערוץ, וזאת מכיוון שאז, בפועל, נעשה שימוש ב-Nonce-ים שנוצרו יותר מפעם אחת.

- **GTK** - קיצור של Groupwise Temporal Key, נועד לשימוש במקרים בהם יש צורך לשלוח הודעות Multicast ו-Broadcast ברשת (כאמור, ב-WPA2 יש מפתח שיחה ייחודי בין הנתב לכל עמדת קצה), המפתח הנ"ל מתקבל מהנתב בסוף הליך ההזדהות. את ה-GTK הנתב גוזר מתוך מפתח אחר בשם **GMK** (קיצור של Groupwise Master Key), המפתח ממנו גוזרים את ה-GTK, היחס בינו לבין ה-GTK דומה ליחס בין ה-PMK לבין ה-PTK.

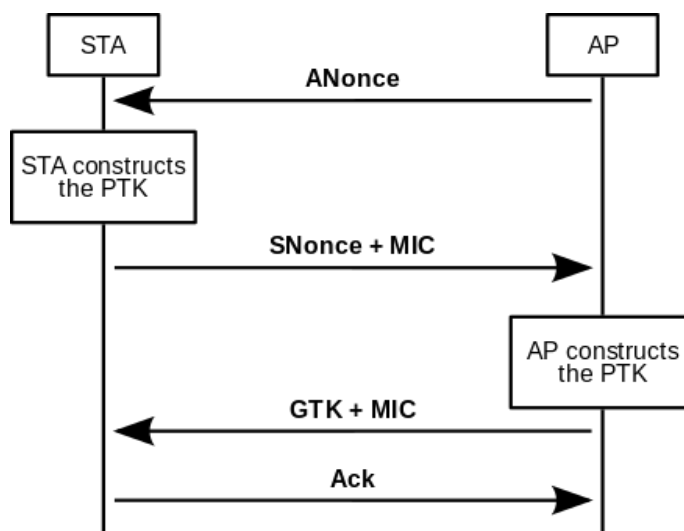
- **MAC** - קיצור של Message Authentication Code, הינו קונספט לפונקציות אימות מסרים עם צד מאמת שאיתו חלקנו מפתח מראש. השימוש בהן עובד באופן הבא: הן מקבלות מחרוזת (מסר אקראי) ומפתח, התוצר שלהן יהיה Authenticator (או "Tag") - מחרוזת שאותה ניתן לשלוח ביחד עם המסר המקורי לצד המאמת. הצד המאמת יוכל לקחת את המסר, להפעיל עליו את אותה הפונקציה עם מפתח שנמצא בידיו, ובמידה והתוצאה יוצאת זהה - הוא יודע שהצד המתאמת מחזיק במפתח גם הוא. המושג **MIC** (קיצור של Message Integrity Code) זהה ברובו המוחלט למושג MAC ומשתמשים בו (ברב המקרים) כדי שלא ייווצר בלבול בין עם המושג Media Access Control מעולם רשתות התקשורת. עם זאת, חשוב להוסיף כי במקרים שבהם

מדברים על MIC ולא כדי למנוע את הבלבול, מתכוונים לשימוש בפונקציות גיבוב ללא מפתח חיצוני.

בעזרת שימוש בפונקציות MAC אלה ניתן לאמת שני דברים:

- ראשית - את אותנטיות השולח, רק שולח המחזיק במפתח יוכל לשלוח מסקר אקראי ואת הצופן שלו עם המפתח הנכון.
- שנית - את אותנטיות המידע שהתקבל. הצד המאמת יוכל לדעת ששום גורם זדוני (שאינו מחזיק את המפתח) לא ערך את המידע שהגיע לאחר שיצאה מהשולח.

אז לאחר כל הכיף הזה, בואו נראה איך התהליך נראה ממבט על:



[מקור: https://en.wikipedia.org/wiki/IEEE_802.11i-2004]

השלבים הם:

בשלב הראשון, לא קורה יותר מדי, הנתב מחולל מספר אקראי (בתרשים: ANonce) ושולח אותו ללקוח. השלב הזה גם ידוע בשם Association.

- **בשלב השני**, על הלקוח לייצר את ה-PTK (תזכורת: Pairwise Transient Key), ולאחר שקיבל את ה-Nonce שחולל הנתב - יש בידינו את כלל המידע הדרוש:
 - את ה-PMK או ה-PSK הוא יודע לייצר לבדו / או באמצעות שרת האותנטיקציה שהוגדר לרשת
 - את ה-Nonce של הנתב הוא הרגע קיבל כך שעליו רק לחולל מספר אקראי משלו (בתרשים: SNonce)
 - את כתובות ה-MAC של הנתב הוא יודע להוציא מהחבילה שקיבל ואת כתובת ה-MAC שלו עצמו הוא יודע.
- לאחר יצירת ה-PTK הלקוח שולח לנתב TAG (בתרשים: MIC) שנוצר ע"י שימוש ב-PTK. בנוסף ל-Nonce שבו השתמש (בתרשים: SNonce)



- **בשלב השלישי**, הנתב מקבל את את ה-SNonce מהלקוח ומייצר באמצעותו את ה-PTK. מבצע אימות של ה-MIC (ברגע שהוא קיבל מהלקוח את ה-SNonce, הוא יכול לחולל את ה-PTK, להפעיל על החלק הגלוי של ה-MIC את פונקציית ה-MAC ולאמת שאכן הוא מקבל את מה שציפה לו כפי שקיבל מעמדת הקצה). בשלב זה הנתב מחולל MIC משל עצמו (ובמידת הצורך גם GTK מתוך ה-GMK) ושולח אותם לעמדת הקצה.

- **בשלב הרביעי**, עמדת הקצה מבצעת עימות ל-MIC שהתקבל מהשרת. ובמידה ושלב זה עובר בהצלחה - עמדת הקצה שולחת Ack.

לאחר שעמדת הקצה התקינה את ה-PTK, היא תגזור ממנו שלושה מפתחות חדשים: ה-KCK (קיצור של Key Confirmation Key), מפתח בשם KEK (קיצור של Key Encryption Key) ואת ה-TK (קיצור של Temporal Key). בשני המפתחות הראשונים היא תעשה שימוש לטובת הגנה על תהליכי ה-Handshake וב-TK היא תעשה שימוש לטובת איתחול וקטור הצפנת המידע בעזרת השימוש באלגוריתם ההצפנה שנקבע לרשת (כגון TKIP או CCMP עליהם הוסבר בראשית המאמר).

בכל פעם שעמדת הקצה תעשה שימוש באחד מפרוטוקולי ההצפנה, הללו, יעשה שימוש ב-TK שנגזר עם Counter עולה, וכך יובטח כי לא יעשה שימוש באותו וקטור איתחול, מה שיאפשר המשך עבודה בטוחה עם אותו מנגנון הצפנה. כל עוד מדובר ב-PTK חדש שלא נעשה בו שימוש, וכל עוד נעשה שימוש ב-Counter - אין מה לדאוג, הפרוטוקול והרשת בטוחים.

קצת פרקטיקה

מסניפים ביטים

אחרי שדיברנו לא מעט באויר (תרתי משמע), בואו נראה איך זה מתבצע בפועל. נכתב רבות על איך להסניף ב-Monitor Mode תחת Linux אבל על איך לבצע זאת ב-Windows כמעט ולא. ולכן נבחר לעשות זאת תחת מערכת הפעלה זו. אך לפני כן - מה זה אומר Monitor Mode?

לא מעט מתבלבלים בין Monitor Mode לבין Promiscuous Mode למרות שאין כל כך קשר בין השניים. ב-Promiscuous Mode אנו נבקש מכרטיס הרשת להעלות למערכת ההפעלה חבילות מידע שאינן מיועדות אליה (לדוגמה - חבילות שכתובת ה-MAC שלהן לא מיועדות אלינו) על מנת שנוכל לראות את תוכן, זהו מצב שניתן להשתמש בו הן בכרטיסי רשת קווים והן בכרטיסים רשת אל-חוטיים. **Monitor Mode** הוא מצב ייחודי לכרטיסי רשת אל-חוטיים, ובו אנו מורים לכרטיס הרשת להעביר לנו חבילות מידע גם כאשר הוא אינו מחובר לאף רשת. מצב זה הינו מצב אחד מתוך שבעה מצבים שונים שבהם ניתן להפעיל כרטיסי רשת אל-חוטיים:

- Master
- Managed

- Ad hoc
- Mesh
- Repeater
- Promiscuous
- Monitor mode

שאר המצבים מעניינים מאוד (לדוגמא, מצב Master מאפשר להפוך את כרטיס הרשת ל-Access Point, ו-Repeater מאפשר לנו לפרסם רשת קיימת), אך הם מעבר לסקופ המאמר ולכן לא נפרט עליהם עוד.

בואו נתחיל. לטובת ביצוע ההסנפה, נשתמש בתוכנה "Network Monitor" של חברת Microsoft (מפתיע, אה?) הגרסא האחרונה שלה הינה 3.4 וניתן להוריד אותה מהקישור הבא:

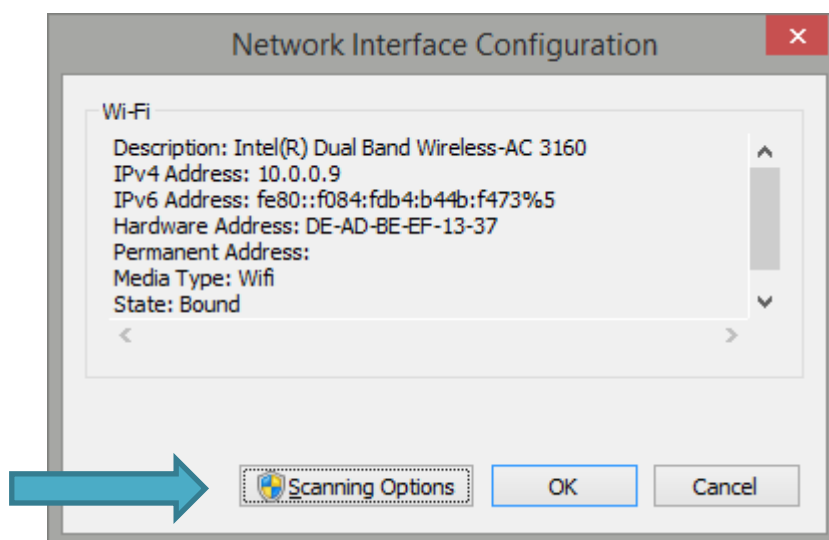
<https://www.microsoft.com/en-us/download/details.aspx?id=4865>

הפעילו את התוכנה. בצד שמאל למטה אמורים להופיע לכם כרטיסי הרשת שהתוכנה זיהתה, משהו כזה:

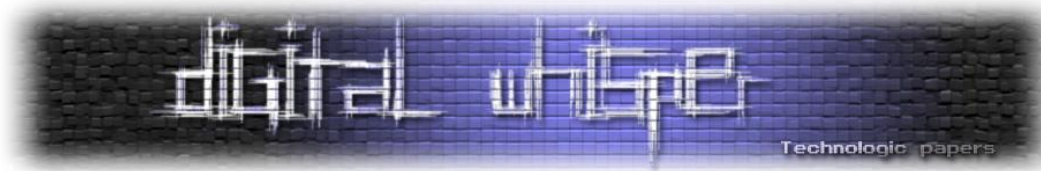
Friendly Name	Description	IPv4 ...	IPv6 Address	Hardware Address	P..	Media...	State
<input type="checkbox"/> Ethernet	Realtek PCIe GBE Family Controller	None	fe80::d0e:76ed:8910:1ab2%3	08-00-27-08-00-27		Ethernet	Bound
<input type="checkbox"/> isatap.{180B3C0C-AC12-4E4A-9B39-ADFB24D6CCE}	Microsoft ISATAP Adapter #3	None	fe80::5efe:10.0.0.4%9	00-00-00-00-00-00		Tunnel	Bound
<input type="checkbox"/> isatap.{6CCEB789-265F-416D-9EF6-A40AD181494D}	Microsoft ISATAP Adapter #2	None	fe80::5efe:192.168.25.1%46	00-00-00-00-00-00		Tunnel	Bound
<input type="checkbox"/> isatap.{A035A402-74AF-45FF-8DD9-83C21CC232F6}	Microsoft ISATAP Adapter	None	None	00-00-00-00-00-00		Tunnel	Bound
<input type="checkbox"/> Local Area Connection* 3	Microsoft Wi-Fi Direct Virtual Adapter	None	fe80::680f:6411:ebe8:9392%6	00-00-00-00-00-00		Wifi	Bound
<input type="checkbox"/> Local Area Connection* 4	Microsoft Hosted Network Virtual Adapter	None	fe80::cc8c:8863:6de7:df4c%7	00-00-00-00-00-00		Wifi	Bound
<input checked="" type="checkbox"/> Wi-Fi	Intel(R) Dual Band Wireless-AC 3160	10.0.0.4	fe80::561:b5be:a040:5f3c%5	08-00-27-08-00-27		Wifi	Bound

[אם התוכנה לא מזהה את כרטיסי הרשת שלכם - בצעו Logoff/Logon למשתמש]

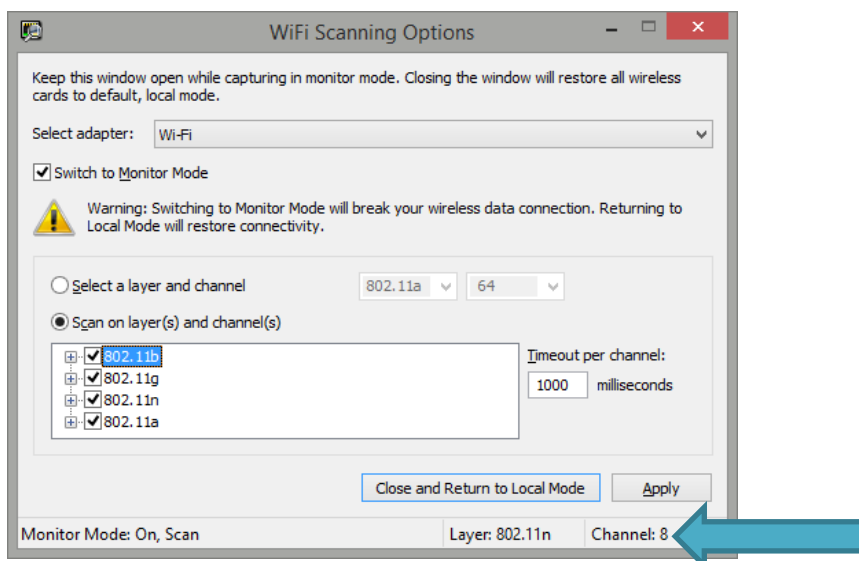
אתרו את כרטיס ה-Wireless שלכם, לחצו עליו פעמים ובחרו ב-"Scanning Options":



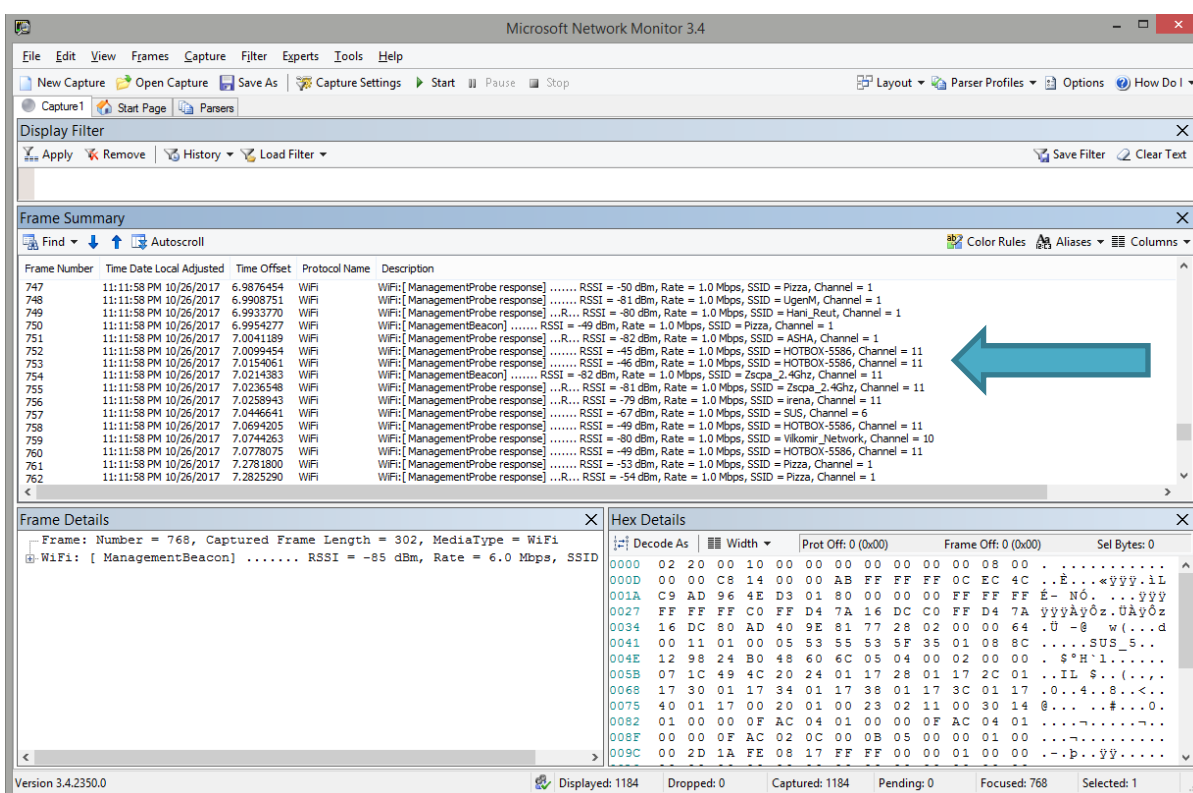
בתפריט שיפתח לכם, סמנו ב-"V" את האופציה "Switch to Monitor Mode" ובאופציה שתפתח לכם ביחרו ב-"(Scan in layer(s) and channels)", לעת עתה ביחרו בכלל התדרים ובכלל הערוצים. לחצו על Apply (שימו לב שברגע שתעברו ל-Monitor Mode אתם תתנתקו מהרשת אליה אתם מחוברים כעת).



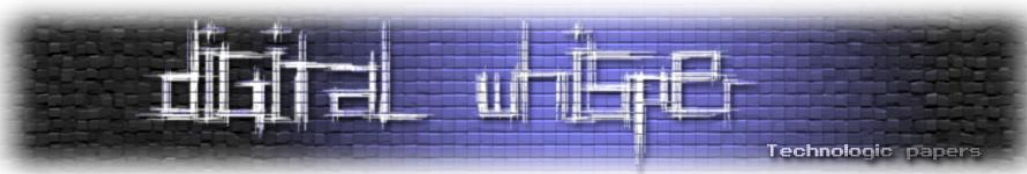
אתם אמורים לקבל חיווי חיובי בצד ימין למטה של החלון שמעידה על כך שהכרטיס מזהה פעילות בכל מני תדרים וערוצים:



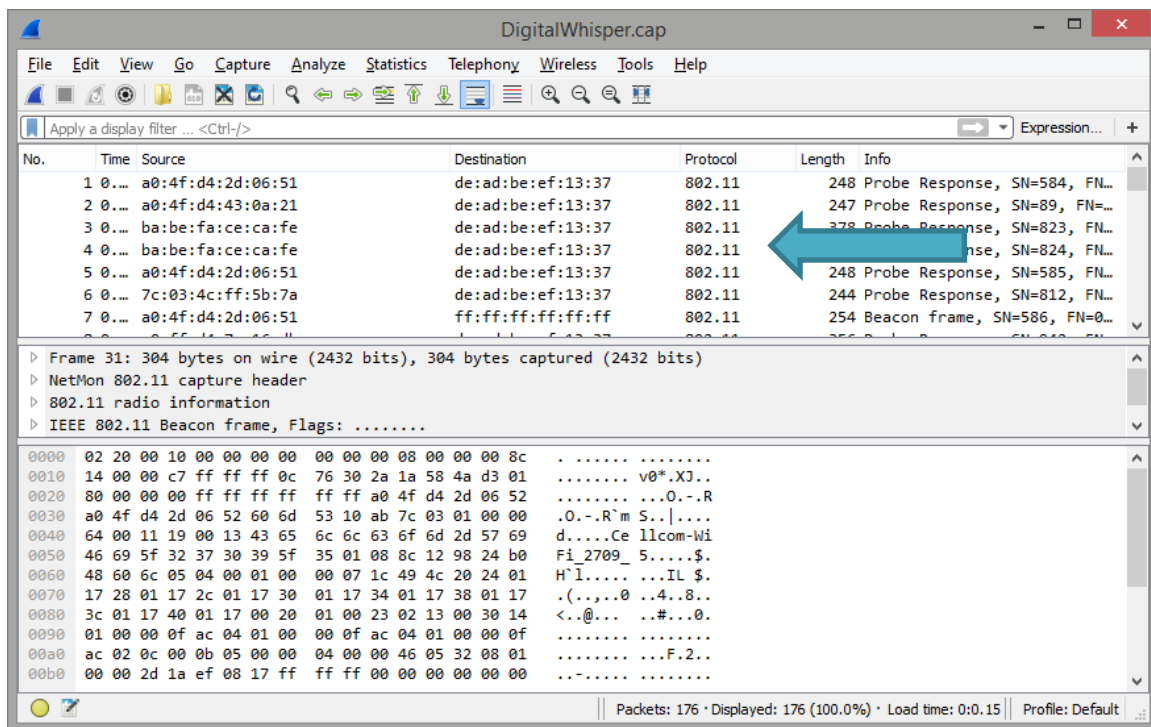
השאירו את החלון פתוח, מזערו אותו וחזרו לעמוד הראשוני של התוכנה. ביחרו ב-"New Capture" ושם ב-Start. כעת, בזמן שאתם מסניפים, נתקו וחברו עמדת קצה אחרת לרשת (לדוגמה - המכשיר הסלולארי שלכם). תחת Frame Summary אתם אמורים להתחיל לראות חבילות רצות:



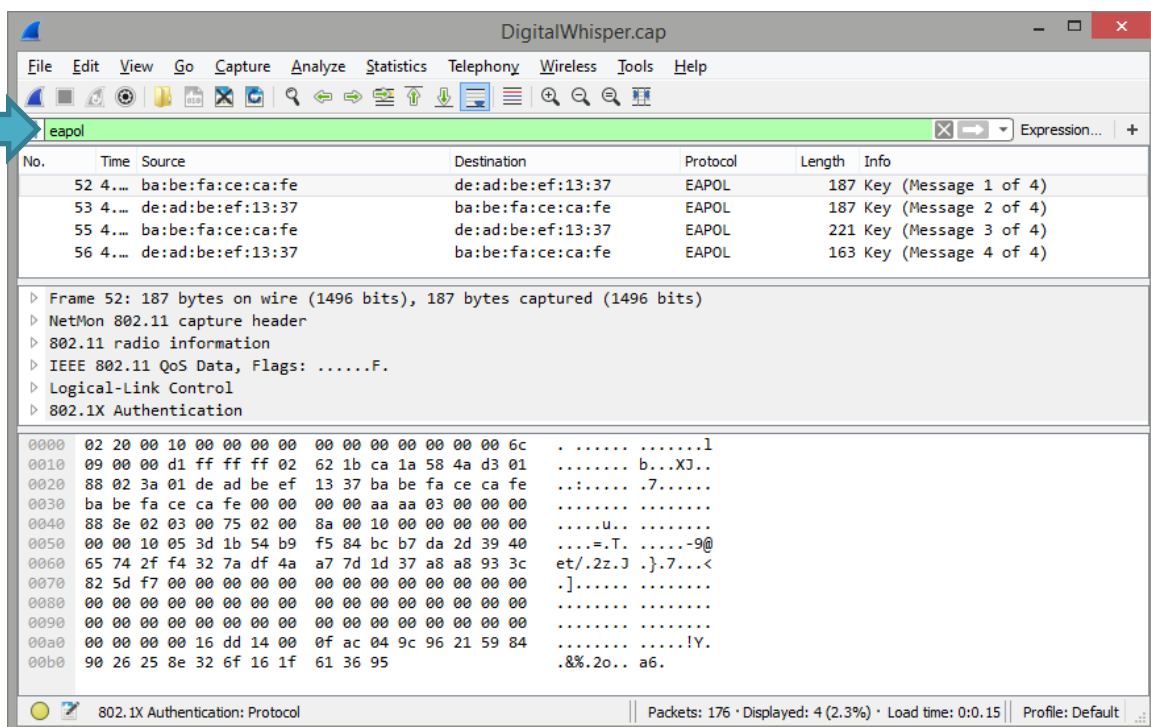
במידה ועשיתם הכל נכון, אתם אמורים לראות חבילות המזוהות כ-"WiFi" ולא חבילות העוברות על גבי IP (כגון HTTP, TCP, UDP וכו'). בגמר ההתחברות לחצו על Stop ושמרו את התוצאה לקובץ pcap.

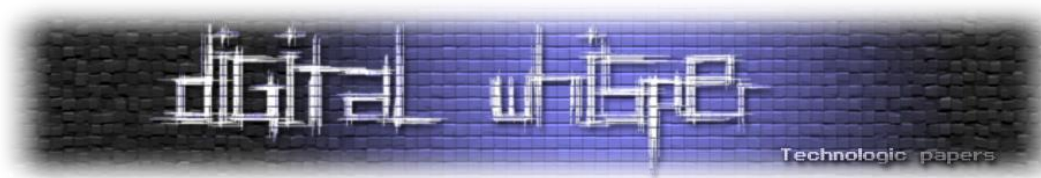


פתחו את ההסנפה עם Wireshark (אין מה לעשות, הכריש בהחלט נח יותר, בייחוד הממשק ה-Legacy...). אתם אמורים לראות חבילות המזוהות כ-802.11, כגון אלו:



על מנת לראות את תהליך ה-4Way Handshake, סננו על פי הפילטר: "eapol" (קיצור של Extensible Authentication Protocol Over LAN), רואים איך הכל מתחיל להתחבר? ☺





בואו נראה שאנו יודעים לזהות את ארבעת השלבים ב-PCAP, בשלב הראשון אנו אמורים לקבל Nonce מה-AP:

DigitalWhisper.cap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Expression...

No.	Time	Source	Destination	Protocol	Length	Info
52	4....	ba:be:fa:ce:ca:fe	de:ad:be:ef:13:37	EAPOL	187	Key (Message 1 of 4)
53	4....	de:ad:be:ef:13:37	ba:be:fa:ce:ca:fe	EAPOL	187	Key (Message 2 of 4)
55	4....	ba:be:fa:ce:ca:fe	de:ad:be:ef:13:37	EAPOL	221	Key (Message 3 of 4)
56	4....	de:ad:be:ef:13:37	ba:be:fa:ce:ca:fe	EAPOL	163	Key (Message 4 of 4)

Frame 52: 187 bytes on wire (1496 bits), 187 bytes captured (1496 bits)

- NetMon 802.11 capture header
- 802.11 radio information
- IEEE 802.11 QoS Data, Flags:F.
- Logical-Link Control
- 802.1X Authentication
 - Version: 802.1X-2004 (2)
 - Type: Key (3)
 - Length: 117
 - Key Descriptor Type: EAPOL RSN Key (2)
 - Key Information: 0x008a
 - Key Length: 16
 - Replay Counter: 16
 - WPA Key Nonce: 053d1b54b9f584bcb7da2d394065742ff4327adf4aa77d1d...
 - Key IV: 00000000000000000000000000000000
 - WPA Key RSC: 0000000000000000
 - WPA Key ID: 0000000000000000
 - WPA Key MIC: 00000000000000000000000000000000
 - WPA Key Data Length: 22
 - WPA Key Data: dd14000fac049c962159849026258e326f161f613695

Packets: 176 · Displayed: 4 (2.3%) · Load time: 0:0.15 | Profile: Default

בשלב השני, עמדת הקצה מחוללת את ה-GTK, ושולחת MIC ביחד עם ה-Nonce שלה:

DigitalWhisper.cap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Expression...

No.	Time	Source	Destination	Protocol	Length	Info
52	4....	ba:be:fa:ce:ca:fe	de:ad:be:ef:13:37	EAPOL	187	Key (Message 1 of 4)
53	4....	de:ad:be:ef:13:37	ba:be:fa:ce:ca:fe	EAPOL	187	Key (Message 2 of 4)
55	4....	ba:be:fa:ce:ca:fe	de:ad:be:ef:13:37	EAPOL	221	Key (Message 3 of 4)
56	4....	de:ad:be:ef:13:37	ba:be:fa:ce:ca:fe	EAPOL	163	Key (Message 4 of 4)

Frame 53: 187 bytes on wire (1496 bits), 187 bytes captured (1496 bits)

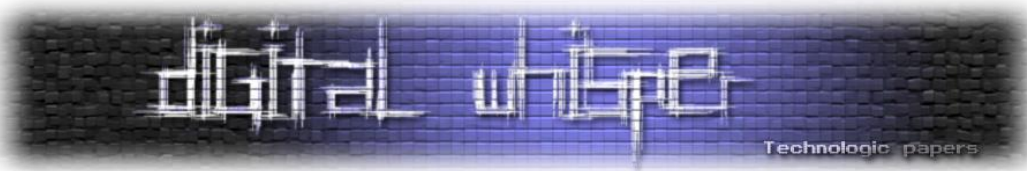
- NetMon 802.11 capture header
- 802.11 radio information
- IEEE 802.11 Data, Flags:T
- Logical-Link Control
- 802.1X Authentication
 - Version: 802.1X-2001 (1)
 - Type: Key (3)
 - Length: 119
 - Key Descriptor Type: EAPOL RSN Key (2)
 - Key Information: 0x010a
 - Key Length: 0
 - Replay Counter: 16
 - WPA Key Nonce: e848f5d77e49fc94f87e1b8901d5222ae746ed8ce690d760...
 - Key IV: 00000000000000000000000000000000
 - WPA Key RSC: 0000000000000000
 - WPA Key ID: 0000000000000000
 - WPA Key MIC: 5d8095e6e52b8e0aec04395b7d1fce0d
 - WPA Key Data Length: 24
 - WPA Key Data: 30160100000fac040100000fac040100000fac023c000000

Packets: 176 · Displayed: 4 (2.3%) · Load time: 0:0.15 | Profile: Default



ובשלב האחרון, שליחת ה-Ack המורה על כך שעמדת הקצה אישרה את ה-MIC של הנתב:

נראה שעד כאן - רמת ההבנה שלנו בסדר גמור. נראה שאפשר להתחיל לדבר על המתקפה!



הקלטה וניטור תקשורת בעזרת Mac

האופן ד"י מפתיע, MacOS מאפשרת לנו להכניס את המתאם הבנוי למצב מוניטור באותה קלות של מתאמים חיצוניים. כמובן שבעזרת כרטיס תקשורת נוסף לא יהיה צורך להתנתק מהרשת המוקמית על מנת לנטר. האפליקציה ב-MacOS שיודעת לעשות את זה נקראת airport ונמצאת כאן:

```
/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport
```

כדי לחסוך לנו זמן נייצר קישור לתוכנה:

```
sudo ln -s /System/Library/.../Versions/Current/Resources/airport /usr/local/bin/airport
```

עכשיו כדי שנוכל להבין איזה ערוצים עמוסים יותר ולהתחיל לתפוס נתונים נסרוק את הרשתות באזור:

```
sudo airport en0 -s

hostname:~ username$ sudo airport en0 -s
SSID BSSID RSSI CHANNEL HT CC SECURITY (auth/unicast/group)
AIS SMART Login 2c:5d:93:96:0f:6c -90 136,-1 Y US WPA2 (802.1x/AES/AES)
AIS SUPER WiFi 2c:5d:93:56:0f:6c -88 136,-1 Y US NONE
true home5G 4e8 94:09:37:99:f4:ec -88 60 Y US WPA2 (PSK/TKIP,AES/TKIP)
true_home2G_de8 a0:72:2c:95:1d:e8 -86 11 Y -- WPA2 (PSK/TKIP,AES/TKIP)
SPICYDISC 82:2a:a8:8b:50:b6 -78 157,+1 Y -- WPA2 (PSK/AES/AES)
Kronus5 94:10:3e:17:31:a8 -69 36 Y -- WPA2 (PSK/AES/AES)
CMG ec:c8:82:fb:02:b0 -91 11 N TH WPA (802.1x/TKIP/TKIP)
HUAWEI BEETHOVEN 8919 b0:89:00:2e:fa:3a -69 6 Y TH WPA2 (PSK/AES/AES)
SPICYDISC 80:2a:a8:8a:50:b6 -63 6 Y -- WPA2 (PSK/AES/AES)
CMG-Guest 64:d8:14:ef:24:e3 -64 6 Y TH NONE
CMG 64:d8:14:ef:24:e0 -64 6 N TH WPA (802.1x/TKIP/TKIP)
Kronusquest 96:10:3e:17:31:a8 -54 2 Y -- NONE
Kronus2 94:10:3e:17:31:a7 -54 2 Y -- WPA2 (PSK/AES/AES)

hostname:~ username$
```

מכאן ניתן להתחיל לכתוב לקובץ PCAP בעזרת:

```
hostname:~ username$ sudo airport en0 sniff 1
Password:
Capturing 802.11 frames on en0.
^C
Session saved to /tmp/airportSniffuwlOyq.cap.
```

התוצאה:

The screenshot shows the Wireshark network protocol analyzer interface. The top toolbar includes filters and display options. The main pane is divided into three sections: packet list, packet details, and packet bytes. The packet list shows 18 captured packets, with the first packet being a Probe Response from Ubiquiti_c1:94: to SamsungE_03:4c: on channel 802.11. The packet details pane for the selected packet shows the IEEE 802.11 frame structure, including the Radiotap header, radio information, and the IEEE 802.11 Probe Response frame details.

WPA2: Key Reinstallation AttaCK

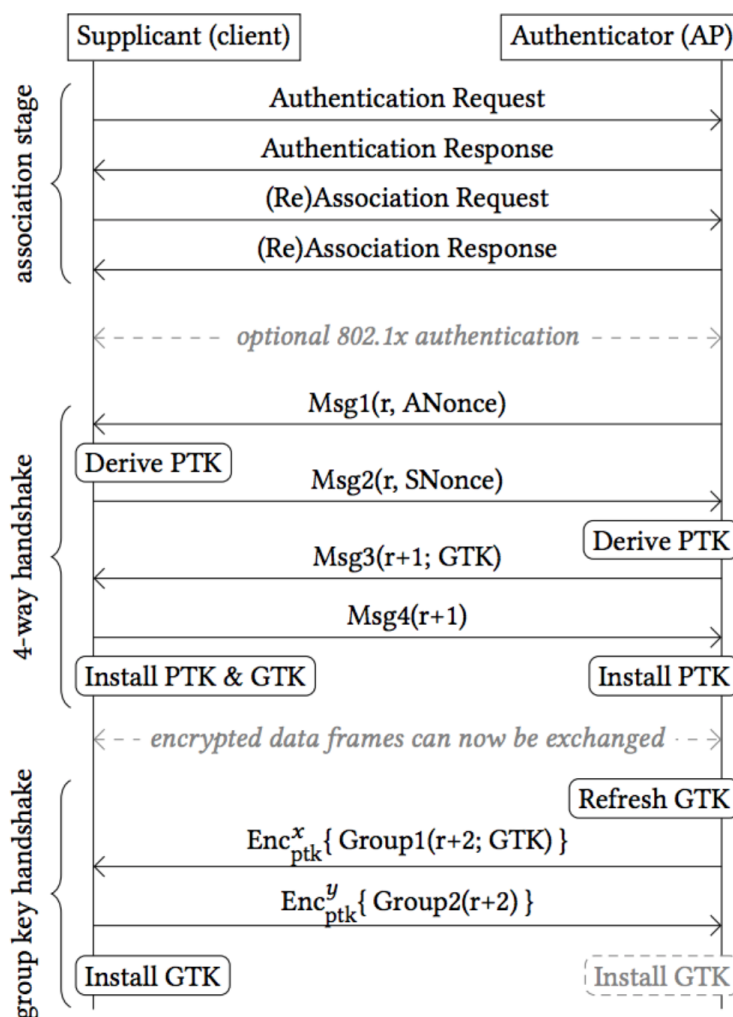
www.DigitalWhisper.co.il

תקיפת ה-4Way Handshake

כעת, משיש דרשותינו את ההבנה הבסיסית כיצד הליך ה-Handshake ב-802.11i עובד. נוכל להבין איפה טמונה הבעיה שאותה מנצלת המתקפה KRACK.

אז עד כה ראינו שבעת ההתחברות לרשת, על עמדת הקצה והנתב להחליט ולהחליף מפתחות ביניהם. ראינו את התרשים של תהליך ה-4Way Handshake ואת השלבים. בואו נסתכל קצת יותר לעומק על השלבים בכדי שנוכל לזהות היכן מסתתר הכשל.

להלן תרשים של אותו תהליך מוכר, רק מפורט יותר - תתי השלבים כלולים גם הם, בפרק הקרוב נתייחס רק לחלק הראשון של התרשים:



[מקור: <https://papers.mathyvanhoef.com/ccs2017.pdf>]

לפי התרשים, ניתן לראות כי רק בגמר השלב הרביעי של תהליך ה-Handshake (החלק השני בתרשים), שני הצדדים מתקינים את ה-PTK. גוזרים את המפתחות ובעזרתם מבצעים את השיחה. מה הבעיה כאן? כרגע הכל בסדר גמור, מדובר ב-PTK חדש, אשר יצרו אותו מ-Nonce שזה עתה הגרילו ושלא נעשה בהם שימוש עד כה ולכן הכל בסדר.

אך... בגלל שמדובר ברשת Wireless, ובגלל שהפרוטוקול נועד לתמוך גם במצבים בהם הקליטה לא חלקה, מהנדסי הפרוטוקול תכננו את מכונת המצבים שלו כך שתהיה רובוסטית ותדע להתמודד גם עם מצבי קליטה קשים. ובייחוד בשביל מצבים כאלה - יש לנו את השלב הרביעי, השלב בו עמדת הקצה שולחת Ack לנתב בכדי להגיד "קיבלתי את ה-MIC שלך, ומבחינתי אתה אכן מי שאתה טוען שאתה, אני מתקינה את ה-PTK".

הנתב, או רכיב ה-AP, יודע שכל עוד הוא לא קיבל את ה-Ack הוא לא יכול להניח שעמדת הקצה אכן קיבלה ואימתה את ה-MIC שלו, ולכן הוא לא יכול להניח שהם יכולים לדבר בעזרת אותו PTK. ולכן, במידה והנתב לא קיבל את ה-Ack של השלב הרביעי הוא צריך לבצע Retransmit של שלב 3. הוא יכול לשלוח את חבילה 3 מספר פעמים והחבילה תגיע ליעדה אחרי נניח... 5 פעמים שהיא נשלחה, וברגע שעמדת הקצה קיבלה אותה, היא תתקין את ה-PTK ותשלח Ack לנתב בכדי להורות על "אור ירוק" לשידור עם ה-PTK הנ"ל.

עד כאן הכל עדיין בסדר גמור, נראה שאפילו יותר מבסדר גמור - אכן מדובר בפרוטוקול שנועד להתמודד עם מקרים של קליטה בעייתית.

אז איפה הדברים מתחילים להסתבך? בדיוק באיזור האפור הזה, שבו הנתב משדר את שלב 3 אך לא מקבל את חבילת האישור של שלב 4. או יותר מזה - בשלב בו עמדת הקצה מקבלת את השידור של שלב 3 למרות שהיא כבר קיבלה אותו, שלחה את שלב 4, התקינה את ה-PTK ואפילו החלה לעשות בו שימוש(!)...

בסיפור שלנו, צמד החוקרים הסתכל בדיוק על המקרה הספציפי הזה וגילה כי בלא מעט מימושים שונים, עמדות קצה שונות, התקינו את ה-PTK ברגע שהם קיבלו את החבילה של שלב 3 בלי קשר למה היה המצב שלהם. ובמקרה כזה, אותם החוקרים יכלו לגרום להם לבצע התקנה ושימוש חוזר באותו ה-PTK, ובנוסף לכך - בכל התקנה של PTK חדש מתבצע איתחול של ה-Counter המועבר לוקטור האיתחול של פרוטוקול ההצפנה. מה שנוגד את כל מה שלמדנו עד עכשיו: אסור לבצע שימוש חוזר באותו PTK! מפני שאז מתבצע שימוש חוזר של אותם מספרי ה-Nonce! (וכאמור, התקנה של מפתח PTK מאפסת את ה-Counter המתגלגל!)

בשלב הזה, כבר תלוי מה הפרוטוקול שבו נעשה השימוש בהצפנה, אך אותם חוקרים הראו שמכאן כבר ניתן לבצע דברים מאוד נוראיים, כגון שליחה חוזרת של חבילות שנשלחו בעבר, זיוף חבילות מכל צד של השיחה ואף פענוח מלא של כלל תווך התקשורת.

נקודה נוספת מעניינת היא שאותם החוקרים ראו כי יש מספר יצרניות שלא עומדות בתקן הפרוטוקול - ולכן אינן חשופות למתקפה. כדוגמת מערכות ההפעלה Windows של Microsoft ו-iOS של Apple. שזה נתון די מדהים בעצמו.

כאן באופן תאורתי נגמר ההסבר על המתקפה, אך בפועל - קיימים עוד מספר מכשולים שעלינו להתגבר עליהם במידה ונרצה להוציא לפועל את המתקפה. שני מכשולים שצמד החוקרים זיהו בדרך לפרקטיקה הם:

- על מנת לבצע את המתקפה, עלינו להיות במצב של MitM מלא בין עמדת הקצה לנתב, אך בפועל אנחנו לא יכולים לפרסם רשת Wireless עם שם זהה וכתובת MAC שונה (במטרה לקבל את החבילות ולשדרן לכל אחד מהצדדים בשם הצד השני), וזאת בגלל שבעת השלב השני של לחיצת היד עמדת הקצה והנתב משתמשים בכתובת ה-MAC אחד של השנייה כדי לייצר את ה-PTK. במידה ונפרסם רשת Wireless מתוך נתב עם כתובת MAC שונה - שלב זה בעת לחיצת היעד לא תעבוד.
- בעת המחקר, התברר שמרגע שחלק מעמדות הקצה התקינו את המפתחות אחרי שלב 4, הם לא הסכימו להתייחס יותר לחבילות שנשלחו על גבי תווך לא מוצפן, ובפועל יצא שהם התעלמו (שלא כמו בתקן) מאותן "חבילות שלב 3" שנשלחו שוב ושוב על-ידי החוקרים.

אז על מנת להוציא לפועל את המתקפה, אותם החוקרים נאלצו לנסות להתגבר על שני המכשולים הנ"ל. כחלק מאותו ניסיון התמודדות עם המכשולים הנ"ל, פותחו עוד מספר תתי-מתקפות שבפועל מאפשרות לבצע את ה-Reinstallation שתגרום לשימוש חוזר ב-PTK, במסמך המתעד את המחקר החוקרים צרפו טבלה שמציגה אילו עמדות קצה פגיעות לאיזה סוג של תת-מתקפה:

Implementation	Re. Msg3	Pt. EAPOL	Quick Pt.	Quick Ct.	4-way	Group
OS X 10.9.5	✓	✗	✗	✓	✓	✓
macOS Sierra 10.12	✓	✗	✗	✓	✓	✓
iOS 10.3.1 ^c	✗	N/A	N/A	N/A	✗	✓
wpa_supplicant v2.3	✓	✓	✓	✓	✓	✓
wpa_supplicant v2.4-5	✓	✓	✓	✓ ^a	✓ ^a	✓
wpa_supplicant v2.6	✓	✓	✓	✓ ^b	✓ ^b	✓
Android 6.0.1	✓	✗	✓	✓ ^a	✓ ^a	✓
OpenBSD 6.1 (rum)	✓	✗	✗	✗	✗	✓
OpenBSD 6.1 (iwn)	✓	✗	✗	✓	✓	✓
Windows 7 ^c	✗	N/A	N/A	N/A	✗	✓
Windows 10 ^c	✗	N/A	N/A	N/A	✗	✓
MediaTek	✓	✓	✓	✓	✓	✓

^a Due to a bug, an all-zero TK will be installed, see Section 6.3.

^b Only the group key is reinstalled in the 4-way handshake.

^c Certain tests are irrelevant (not applicable) because the implementation does not accept retransmissions of message 3.

[מקור: <https://papers.mathyvanhoef.com/ccs2017.pdf>]

- עמודה 2 בטבלה מציגה מי מעמדות קצה מתייחסת לחבילות שלב 3 שנשלחות יותר מפעם אחת.
- עמודה 3 בטבלה מציגה מי מעמדות הקצה מוכנות להתקין מפתח PTK שנשלח באופן לא מוצפן לאחר שהגיעו כבר לשלב 4 (והתקינו כבר מפתח PTK).

ביצוע המתקפה כנגד עמדות קצה אשר תומכות בקבלת חבילה מספר 3 שאין מוצפנות גם לאחר שהתקינו PTK היא המתקפה הפשוטה ביותר לביצוע. זאת מכיוון שמספיק לתוקף לחסום את השידור של חבילה מספר 4 שנשלחה מעמדת הקצה לנתב בכדי לגרום לה לצאת לפעולה. בנוסף, הוא תמיד יכול לבצע Deauthentication Attack על מנת לגרום לעמדת הקצה להתחיל את תהליך ה-Association מהתחלה.

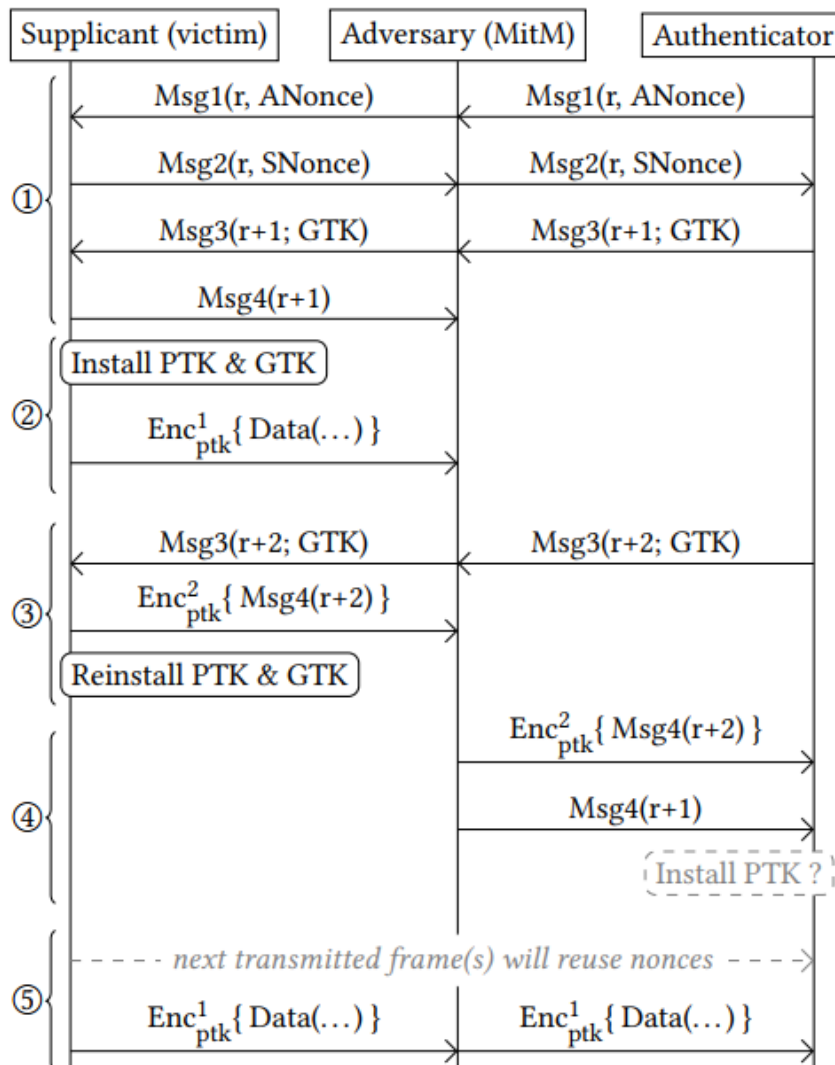
חסימת השידור של חבילה מספר 4 היא אפשרית, אך לא פשוטה כאשר לא נמצאים בעמדת MiTM. על מנת לעשות זאת באופן מלא, מבצעים Channel-based MiTM Attack, והיא הולכת באופן הבא:

1. ראשית, עלינו להשיג את ה-SSID של ה-Wireless שאליה הקורבן שלנו מחובר. מעבר לנתון זה, עלינו לברר תחת איזה ערוץ הרשת משודרת (הנתון הנ"ל משתנה בין מדינות לפי התקנים של משרדי התקשורת. אך באופן טכני קיימים 14 ערוצים שונים וכל ערוץ משדר בתווך תדרים אחר) (אחר) 2. לאחר מכן, עלינו לבצע Retransmit של רשת ה-Wireless הנתקפת על תדר אחר מהתדר המקורי ובמקביל לשדר רעש בעוצמה מספיק חזקה על הערוץ המקורי בו משודרת הרשת. 3. בשלב זה עמדת הקצה תזהה את הרעש בערוץ ותנסה להתחבר לרשת שאנו מפרסמים תחת הערוץ החדש. ברגע שזה קורה - נוכל להפסיק את הרעש בערוץ המקורי ועל גבי ערוץ זה למסר את התשדורת של עמדת הקצה. 4. בשלב זה חשוב שנשדר את הרשת בעוצמה חזקה יותר מהנתב שמשדר את הרשת המקורית ובכך נבטיח לשמור על עמדת הקצה מחוברת אלינו. 5. מי שמעוניין לקרוא קצת יותר על Channel-based MiTM Attack מוזמן לקרוא את המחקר על הנושא שכתבו אותו צמד חוקרים בשנת 2015 ופרסמו תחת הכותרת: "Advanced Wi-Fi Attacks Using Commodity Hardware".

<https://distrinet.cs.kuleuven.be/news/2015/AdvancedWiFiAttacksUsingCommodityHardware.pdf>

כעת, ברגע שעמדת הקצה מנסה להתחבר דרכנו אל הרשת המקורית, יהיה לנו קל מאוד לגרום לה לא לשדר את חבילה מספר 4 ובכך להגיע למצב שמצב אחד היא סיימה את שלב 3 - ולכן מבחינתה אין מניע מלהתקין את ה-PTK, ומצד שני הנתב לא מקבל את ה-Ack ולכן ינסה לשלוח שנית ושלישית את החבילה של שלב 3.

וכך המתקפה נראית בצורה סכמתית:



[מקור: <https://papers.mathyvanhoef.com/ccs2017.pdf>]

שימו לב שבגמר שלב 1 חבילה מספר 4 מגיעה לעמדה המבצעת MitM אך לא נשלחת לנתב המקורי, ולכן הנתב בשלב 3 לשלוח שנית את החבילה 3 (שבתורה כן מועברת דרך עמדת התוקף לעמדת הקצה), מה שגורם לעמדת הקצה לצאת לשידור בפעם השניה עם אותו PTK בשלב 5 (השידור נעשה עם אותו PTK שנשלח בשלב 2).

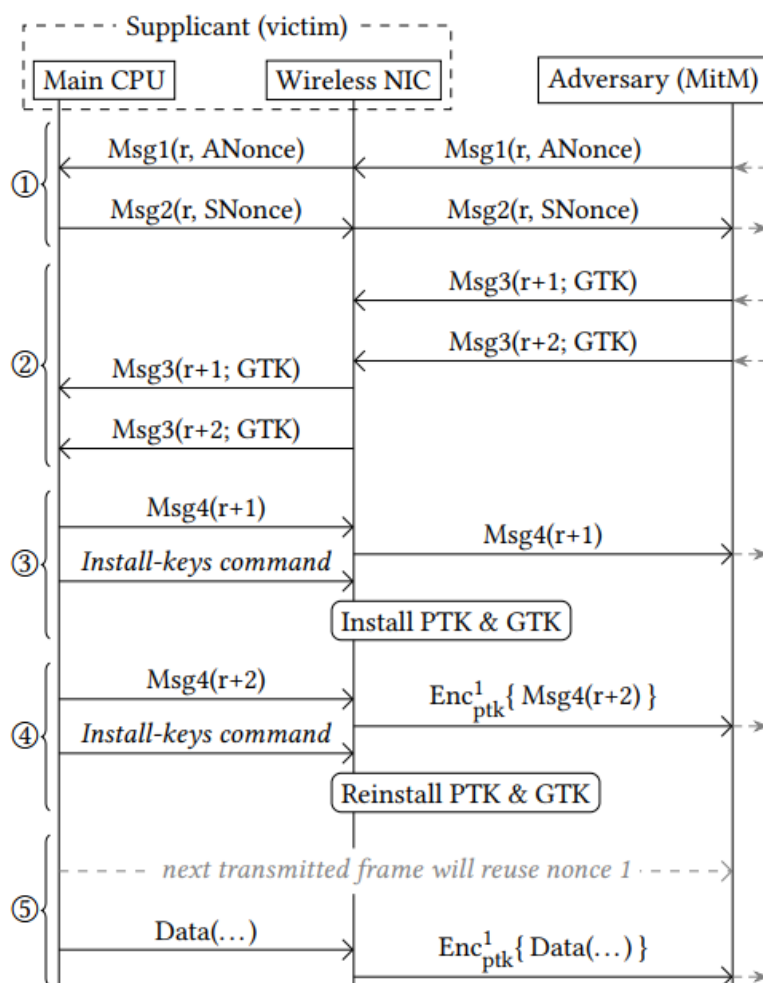
עד כאן, הכל עדיין פשוט יחסית, עמדת הקצה מוכנה לקבל חבילות שאינן מוצפנות גם לאחר שהתקינה את ה-PTK, אך מה בדבר תקיפת עמדות שלא יהיו מוכנות לקבל את חבילה מספר 3 אשר נשלחות באופן שאינו מוצפן מרגע התקנת ה-PTK?

במקרה כזה צמד החוקרים גילה כשל נוסף, מסוג Race Condition, בין כרטיס הרשת של עמדת הקצה ובין ה-CPU של מערכת ההפעלה שלה. על מנת לתקוף עמדות כאלה על התוקף לממש Channel-based MiTM בדיוק כמו במתקפה הוקדמת, רק שהפעם, לאחר שעמדת הקצה שולחת את חבילה מספר 2

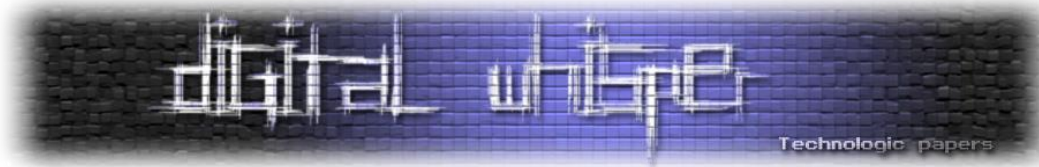
לנתב (עם ה-Nonce שהגרילה), והנתב שולח את חבילה מספר 3, העמדה התוקפת אינה מעבירה את החבילה לעמדת הקצה, ובמקום זאת מחכה שהנתב ישלח את חבילה מספר 3 בפעם נוספת (וזאת בגלל שהוא לא קיבל את ה-Ack משלב מספר 4).

ברגע שהעמדה התוקפת מזהה תשדורת שניה של חבילה מספר 3 מצב הנתב, היא מעבירה לעמדה הנתקפת את שתי החבילות במקביל, ובמקרה כזה, צמד החוקרים הבחין כי כרטיס הרשת יקבל את שתי החבילות, וזאת מפני שהחבילה השנייה הגיעה לפני שמערכת ההפעלה סיימה לנתח את חבילה 3 הראשונה והורתה להקין את מפתח ה-PTK, ומה שיקרה כעת זה שעמדת הקצה תשתמש במפתח הנ"ל בזמן שמערכת ההפעלה תקבל מכרטיס הרשת את חבילה 3 השנייה. מערכת ההפעלה מניחה שעם כרטיס הרשת העביר לה את החבילה היא מוצפנת בצורה מתאימה ולכן תתייחס אליה כאל רלוונטית ותורה להסיר את ה-PTK המקורי ולהתקינו מחדש. מה שיצור מצב שבו משתמשים באותו PTK מחדש, כך ששוב פעם יהיה שימוש באותם Nonce-ים סוררים.

וכך התקיפה נראית באופן סכמתי:



[מקור: <https://papers.mathyvanhoef.com/ccs2017.pdf>]



חשוב לציין שלמקפה זו יש הסתעפות נוספת (המתייחסת לזמן התקנת המפתחות ביחס לשליחת חבילת GTK-ה, בכוונה לא הרחבנו עליהן בשלב זה, בחלק הבא נבין את מנגנון ה-GTK וכיצד ניתן לנצלו לטובת תקיפת הרשת).

תקיפת ה-Group Key Handshake

עד כה דיברנו על החולשה הקיימת במנגנון ה-4Way Handshake, אך צמד החוקרים גילה כי מנגנון זה קיים גם במכניזם אשר אחראי לשליחת המפתחות לשיחות ב-Broadcast. אך לפני שנפרט על נושא זה, חשוב שנבין למה בכלל צריך את המנגנון הנ"ל וכיצד הוא פועל.

כמו שלמדנו, התקן 802.1x מביא איתו מספר הגנות על עמדות הקצה שבהן התקן הרגיל כשל. וכמו שראינו, אחת מאותן הגנות היא הפרדת השיחות בין כלל עמדות הקצה לבין הנתב בעזרת מפתחות שונים, כך שבפועל כל עמדת קצה מדברת עם הנתב בסט מפתחות שונה.

שכבת הגנה זו אכן הופכת את הסביבה שלנו למקום בטוח יותר, אך כדי שהרשת תתנהל בצורה תקינה עלינו לשדר מדי פעם חבילות שאינן Unicast אל מול הנתב, כדוגמת חבילות DHCP, חבילות ARP ו-NBNS, חבילות או פרוטוקולים שבמסגרתו נעשה שימוש ב-Broadcast/Multicast מעמדה אחת אל עבר כלל הרשת, עם סט הידע הקיים לנו לא נוכל לעשות זאת - כי הרי כל עמדת קצה מדברת עם מפתח שיחה שונה. כך שם אם אשלח חבילה אל עבר עמדה אחרת - היא לא תוכל להבין אותי, כי הרי את תהליך ה-4Way Handshake אני מבצע אך ורק מול הנתב.


וזאת אכן בעיה. וכדי לפתור אותה, הכניסו בתקן מנגנון נוסף שכולל צמד מפתחות חדש, מפתחות ה-Group.

כאשר עמדת קצה מסיימת את תהליך ה-4Way Handshake הנתב משדר אליה, ביחד עם חבילה מספר 3 מפתח בשם GTK (קיצור של Group Transient Key), מפתח אשר נגזר מה-GMK (קיצור של Group Master Key), שהוא מפתח קבוע על כל נתב, הנגזר מכתובת ה-MAC שלו בשילוב עם Nonce משתנה בשם GNonce.

עמדת הקצה משתמשת במפתח זה על מנת להצפין ולפענח חבילות ה-Broadcast/Multicast (בפועל היא גוזרת ממנו שני מפתחות חדשים ועושה שימוש בהם, אך לא נתייחס לכל במאמר זה).

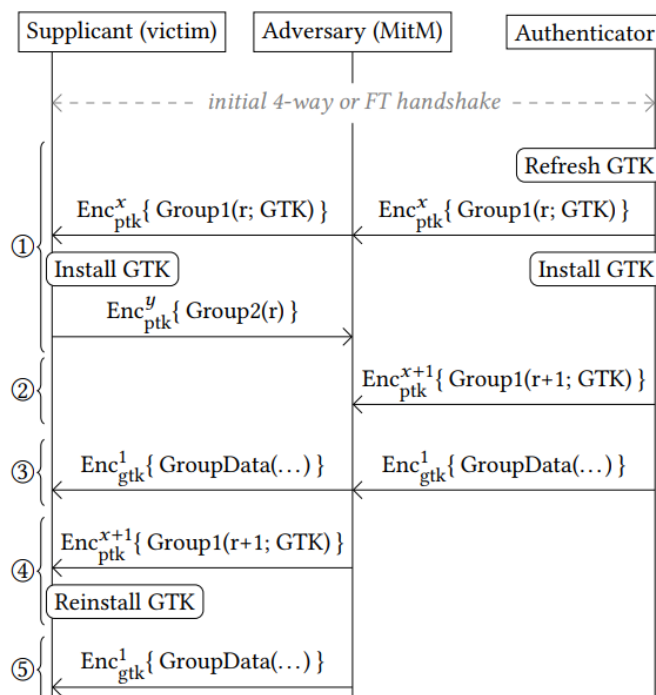
הנתב גוזר מפתח GTK חדש בכל פעם שעמדת קצה עוזבת את הרשת (על מנת להבטיח שאותה עמדה שעזבה לא תוכל לפענח חבילות כאשר היא אינה מחוברת לרשת) ומשדר אותה לכלל העמדות המחוברות ביחד עם MIC (המפתחות הנ"ל מוצפנים עם ה-PTK), כל עמדה מוודאת את ה-MIC ושולחת Ack לנתב לטובת מתן "אור ירוק" לשדר הודעות Broadcast/Multicast עם מפתח זה. במידה והנתב לא מקבל את

ה-Ack הוא יבצע שידורים נוספים של אותה החבילה (כל פעם עם Counter שונה המועבר לפרוטוקול ההצפנה, כך שלתפוס את ה-Retransmit הנ"ל לא פוגע באבטחת הפרוטוקול). לתהליך זה קוראים לפעמים 2Way Handshake.

צמד החוקרים גילה כי ניתן לשכפל את אותו הקונספט של המתקפה גם על תהליך זה, ולגרום למכשירי הקצה לבצע Reinstallation גם עם מפתחות ה-Group, וכאשר עמדות הקצה מתקינות מפתח שכבר הותקן בעבר - הן מאפסות את ה-Counter שמועבר למנגנון ההצפנה וכך למעשה עושות שימוש חוזר באותו המפתח. בינו .

על מנת לבצע את המתקפה בפועל, על החוקרים לחכות למקרה בו הנתב ישלח לעמדת קצה את הודעת ה-Group הראשונה (בניח, במקרה שעמדת קצה אחרת התנתקה), למנוע ממנה להגיע לעמדת הקצה, ולחכות שהנתב ישלח הודעה אחרת שכן תגיע לעמדת הקצה ותגרום לו לעדכן את ה-GTK שברשותו. לחכות שהוא ישדר חבילה ולאחר מכן - לשדר בחזרה את החבילה המקורית שמנעו מהנתב לשלוח אליו. קבלת חבילה זו תגרום לעמדת הקצה להתקין את אותו ה-GTK מחדש, וכך שוב ה-Counter יתאפס כאשר נעשה שימוש באותו GTK.

ואלו הם שלבי המתקפה:



[מקור: <https://papers.mathyvanhoef.com/ccs2017.pdf>]

בעת המימוש, גילו החוקרים כי הנתבים מתחלקים לשני סוגים: נתבים אשר מתקינים את מפתח ה-GTK החדשים לפני שהם מקבלים מכלל עמדות הקצה את ה-Ack על ה-GTK החדשים שנשלחו ונתבים אשר מחכים לקבלת כלל חבילות ה-Ack ורק לאחר מכן מתקינים את ה-GTK, עבור כל "משפחה" של נתבים פותחה תת-מתקפה שתדע להתמודד עם המקרה, אך לא נפרט על העניין יותר.

המקרה המוזר של ה-Android בשעת לילה

המקרה האחרון עליו נפרט במאמר זה הוא המקרה הבא. במהלך המחקר, התגלתה התנהגות חריגה בתפקודה של ספריית wpa_supplicant גרסאות 2.4-2.5 (ספריית ה-WPA בה נעשה שימוש ב-Android מגרסה 6 ומעלה). בעת שידור בקשה מספר 3 שוב לתחנות הללו, נראה שהמערכת בוחרת לאפס את הגדרות המפתח המיועד להצפנה התקשרות (TK) ומחליפה אותו באפסים. תופעה זו נגרמת בגלל שבאחת מגרסאות התקן, יצאה המלצה למחוק מהזיכרון כל מפתח שהתקבל יותר מפעם אחת, וזה בדיוק המצב כאן (:

המשמעות של איפוס המפתח היא שמערכות ההפעלה המשתמשות בגרסאות הללו של wpa_supplicant יבטלו הלכה למעשה את תפקודה של ההצפנה. לפי צמד החוקרים, 31% מהסמארטפונים בעולם פגיעים למתקפה זו.

אז... מה עושים?

על מנת להתגונן מפני מתקפות אלו יש להתקין את עדכוני התוכנה הרלוונטים של החברות השונות. כפי שהזכרנו, החולשה עצמה נמצאת במימוש של wpa_supplicant ולא בשכבות הגבוהות יותר ולכן אין כאן איזה הגדרה לשנות או לערוך. למזלנו ב-reddit כבר יש [megathread](#) עם רשימה מעודכנת של ספקים אשר עובדים על תיקון או שכבר הוציאו תיקון רלוונטי.

בנוסף, הינה עוד סיבה להיות חשדנים, אל תסמכו על אף רשת, אם אתם לא חייבים להתחבר לרשת האלחוטית זאת - אל תעשו זאת.

אם אתם יכולים - עשו שימוש ב-VPN.

סיכום ונקודות נוספות

לכל הדעות מדובר במתקפה חסרת תקדים על פרוטוקול אבטחה שעד כה נחשב בטוח מאוד. לא רק שהמתקפה הנ"ל אלגנטית מאוד (אין כאן דריסות זיכרון או בעיות במימוש תוכנתי, אלא ניצול של מספר בעיות Design בפרוטוקול), היא גם מאוד קלה לניצול והאפקט שלה משמעותי מאוד. וזה פחות או יותר הדרישות ממתקפה איכותית.

בנוסף לכל אשר צוין במאמר, חשוב לנו לציין כי לא הבאנו את כלל המידע אשר פורסם במסמך המחקר. במסמך עצמו קיימות מספר מתקפות נוספות שלא נגענו בהן והן חשובות ומעניינות לא פחות, כגון תקיפה של ה-Fast BSS Transition Handshake, אנו ממליצים בחום לכל הקוראים לקרוא את מסמך המחקר המקורי (מופיע כקישור ראשון במקורות) לטובת הבנה מלאה של הנושא ומתקפה זו בפרט.



מקורות וקישורים לקריאה נוספת

- <http://papers.mathyvanhoef.com/ccs2017.pdf>
- <https://www.krackattacks.com>
- <http://blog.erratasec.com/2017/10/some-notes-on-krack-attack.html>
- <https://www.reddit.com/r/KRaCK>
- <https://github.com/vanhoefm/krackattacks-test-ap-ft>
- https://asecuritysite.com/encryption/ssid_hm
- <https://www.ins1gn1a.com/understanding-wpa-psk-cracking>
- [Detailed documentation of CCMP](#)
- <https://distrinet.cs.kuleuven.be/news/2015/AdvancedWiFiAttacksUsingCommodityHardware.pdf>
- https://www.reddit.com/r/KRaCK/comments/77kz7x/vendor_patch_status_megathread/
- <https://en.wikipedia.org/wiki/KRACK>