Technologic Papers

Wireless (un)Protected Setup

מאת: דר' אריק פרידמן

"Everything should be made as simple as possible, but not one bit simpler."

Albert Einstein

מבוא

בדצמבר 2011, חוקר אבטחת מידע אוסטרי בשם סטפן ויבוק (או כפי שהוא מגדיר את עצמו בחשבון Google+ שלו, "infosec enthusiast, doing stuff") פירסם בבלוג שלו פוסט בנוגע לנקודת תורפה בפרוטוקול WPS (Wi-Fi Protected Setup), פרוטוקול שנועד לאפשר הגדרה והתקנה של רשתות אלחוטיות בצורה פשוטה ובטוחה. הפרוטוקול מיועד לנתבים עבור שוק הבתים והעסקים הקטנים, קהל יעד המורכב ברובו מאנשים לא טכניים. התהליך הרגיל הדרוש להתקנת רשת אלחוטית בטוחה, הכולל בחירת סיסמה ארוכה וקשה לניחוש והזנתה בכל פעם שמחברים התקן חדש לרשת, מהווה מכשול להגדרה תקינה של הרשת בידי אנשים לא טכניים. בעקבות זאת מתכנני פרוטוקול WPS שאפו להפוך את תהליך החיבור לפשוט עד כדי לחיצת כפתור, או הקלדה של מספר קצר, ללא צורך בהתפשרות בחוזק ההגנה על הרשת או חוזק הסיסמה.

בפועל, כפי שסטפן ויבוק הראה, הפרוטוקול עצמו מהווה נקודת תורפה המאפשרת לגורמים לא מורשים להשיג את סיסמת הרשת האלחוטית בזמן קצר יחסית (סדר גודל של מספר שעות עד יום) ולהתחבר אליה כמשתמשים לגיטימיים.

פרצת האבטחה הזאת היא דוגמה מעניינת להרבה מההיבטים המעשיים הקשורים לאבטחת רשתות אלחוטיות (ואבטחת מידע באופן כללי). יום לאחר פרסום נקודת התורפה, חברה בשם Tactical Network Solutions פרסמה פוסט משלה, והודתה שהיא ידעה על היכולת הזאת והשתמשה בה כבר כמעט שנה (למרות שלא היה פירוט מהו בדיוק אופי השימוש), ואף פרסמה כקוד פתוח כלי בשם Reaver שמנצל את נקודת התורפה כדי להשיג את סיסמת הכניסה לרשת מאובטחת. לעולם לא נדע בודאות אם גורמים נוספים גילו את הבעייה בפרוטוקול (ככל הנראה כן) וכמה זמן כבר ידעו עליה. מצב זה מאפיין בעייה כללית באבטחת מידע - אף פעם לא ניתן לדעת בבטחון שמנגנוני האבטחה בשימוש אכן משיגים את מטרתם במלואה. גם אם בקהילה הרחבה מנגנון מסויים נחשב לבטוח, תמיד קיים סיכוי שהאקרים (מטעם עצמם, מטעם גופים פרטיים או מטעם גופים ממשלתיים) מצאו דרך לשבור אותו, והם שומרים על כך בסוד כדי שיוכלו להמשיך ולנצל את הפירצה כל עוד היא קיימת.

הבעייה בפרוטוקול גם ממחישה את הלחצים והאילוצים הפועלים בתכנון מנגנוני אבטחה - מול הצורך ברמת אבטחה גבוהה, יש גם לחצים שיווקים למצוא פתרון שיהפוך את הטכנולוגיה לקלה לשימוש (לפעמים על חשבון אבטחה), ולחצים כלכליים שדוחפים לפשרות בתכנון המוצר ובמימוש הפרוטוקול כדי להוזיל עלויות. כל אלה מחריפים את הבעייה.

רוב הנתבים החדשים המשווקים כיום למגזר הפרטי תומכים ב-WPS, ויצרני הנתבים עדיין מנסים להבין את ההשלכות של הבעיה שהתגלתה בפרוטוקול, וכיצד להתמודד איתה. סביר להניח שיקח פרק זמן משמעותי למצוא פתרון - גם הפתרון הבסיסי של יצור גרסאות קושחה (firmware) חדשות לנתבים, בהן הפרוטוקול יהיה מנוטרל, לא באמת יפתור את הבעייה, מאחר ולשם כך דרוש שבעלי הנתבים (כאמור, ברובם אנשים לא טכניים שספק אם אפילו יגלו על קיום הבעיה) ינקטו בפעולה כדי להתקין את הקושחה.

Wi-Fi Alliance

מימוש פרוטוקול תקשורת לקישור בין מוצרים שונים הוא מלאכה מורכבת. תקשורת אלחוטית מתקיימת בין מגוון רחב של התקנים, ביניהם נתבים, מחשבים אישיים וטלפונים סלולריים, והפרוטוקול ממומש על-ידי מספר רב של יצרנים. כדי להבטיח שהמוצרים השונים יעבדו אחד עם השני בצורה חלקה (interoperability) יש צורך בסטנדרטים מוסכמים שכולם יפעלו לפיהם. לצורך זה הוקם ב-1999 גוף ללא מטרת רווח בשם Wi-Fi Alliance. במרץ 2000 האיגוד השיק את תוכנית Wi-Fi Certified, במסגרתה נבדקת התאמתם של מוצרים לסטנדרט וניתן להם תו תקן. תו זה מאשר שהמוצר נבחן תחת תנאים שונים ומול מגוון של התקנים אחרים כדי לוודא שהוא מתאים לעבודה עם מוצרים מאושרים אחרים. מוצרים שעברו את המבחן רשאים להשתמש בלוגו Wi-Fi Certified. ה-Wi-Fi Alliance מאגד כיום למעלה מ-400 יצרנים, וכמעט כל היצרנים הגדולים של נתבים אלחוטיים מוודאים כי המוצרים שלהם עומדים בסטנדרט.

הגדרת רשת בטוחה באופן פשוט

סטנדרט האבטחה המקובל כיום לאבטחת רשתות אלחוטיות הוא פרוטוקול WPA2 (Wi-Fi Protected Access), המניח סיסמה (או passphrase) אותה מגדירים בשתי נקודות הקצה - הנתב וההתקן המתחבר אליו. עם זאת, כדי לוודא רמת אבטחה גבוהה, הסיסמה צריכה להיות ארוכה וקשה לניחוש, דבר שמסרבל את תהליך הגדרת הרשת. ב-2007 האיגוד הוסיף לסטנדרט מנגנון חדש בשם Wi-Fi Protected Setup, או בקיצור WPS (בשלבים מוקדמים יותר המנגנון כונה Wi-Fi Simple Config, או WSC). המטרה של המנגנון היא לאפשר הקמת רשת המאובטחת באמצעות WPA2 בצורה פשוטה ככל האפשר. במקום להסתמך על הגורם האנושי לצורך הגדרת WPA2 ולצורך יצירת והגדרת הסיסמה, שני ההתקנים האלחוטיים מחליפים ביניהם

את הגדרות אבטחת הרשת, כולל הסיסמה. הדרישה בתקן היא שבברירת המחדל המנגנון של WPS יהיה פעיל, כדי שיהיה ניתן להפעיל אותו מיד כאשר מוציאים את המוצר מהקופסה. הפרוטוקול מומש גם במסגרת Windows 7, שם הוא מכונה בשם Windows Connect Now.

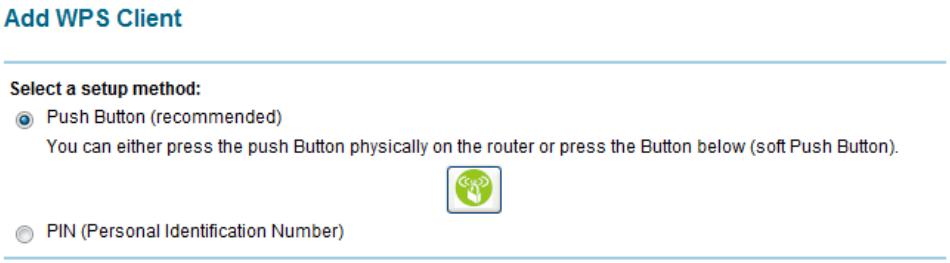
על-פי האתר של Wi-Fi Alliance, למעלה מ-200 מוצרים קיבלו את תו התקן מאז השקתו. סביר להניח שלפשטות החיבור שמציע פרוטוקול זה יש חלק לא מבוטל בהטמעת WPA2 (לפרטים נוספים, דן קמינסקי פרסם בבלוג שלו סטטיסטיקות מעניינות על התרחבות התפוצה של WPA2 בשנים האחרונות, על-סמך נתונים מ-Wigle). יש לציין שמוצרי אפל אינם תומכים בפרוטוקול, אלא משתמשים בטכנולוגית AirPort הקניינית של אפל, ולפיכך אינם חשופים להתקפה שתתואר בהמשך.

מה בעצם עושה WPS?

כאמור, WPS נועד לעקוף את הצורך להשתמש בסיסמאות ארוכות ומסורבלות לצורך חיבור הרשת (אם כי "מאחורי הקלעים" הסיסמאות האלה ממשיכות לפעול). לפיכך, הבעייה הבסיסית אותה WPS צריך לפתור היא האימות ההדדי בין שני התקנים על גבי רשת אלחוטית, אותו מספקת סיסמת הרשת. אך כעת נדרש מנגנון חלופי (ופשוט יותר) שיאפשר לשני הצדדים לאמת אחד את השני ולייצר ערוץ מאובטח, שמעליו יוכלו להעביר ביניהם את הגדרות הרשת, כולל הסיסמה עבור WPA2.

המנגנון מציע מספר דרכים לבסס קשר בין נקודת גישה אלחוטית (למשל נתב) לבין התקן (כגון מחשב אישי, מדפסת או טלפון).

1. PBC (Push Button Connect) - המשתמש לוחץ על כפתור (פיזי או וירטואלי) גם בנתב וגם בהתקן האלחוטי שרוצים לחבר לנתב. הלחיצה תפעיל את המנגנון לחלון זמן של שתי דקות. אם במסגרת שתי דקות אלה ההתקן זיהה התקן אחר שהפעיל את המנגנון, הם יתחברו אחד לשני באופן אוטומטי (אם יש יותר משני התקנים פעילים בו-זמנית, לא יתבצע חיבור). בגישה זו אין באמת אימות הדדי של הצדדים, אלא מסתמכים על חלון הזמן הקצר שבו שני ההתקנים מנסים להתחבר בו-זמנית כאינדיקציה לכך שאלה ההתקנים "הנכונים" שאותם המשתמש רצה לחבר.

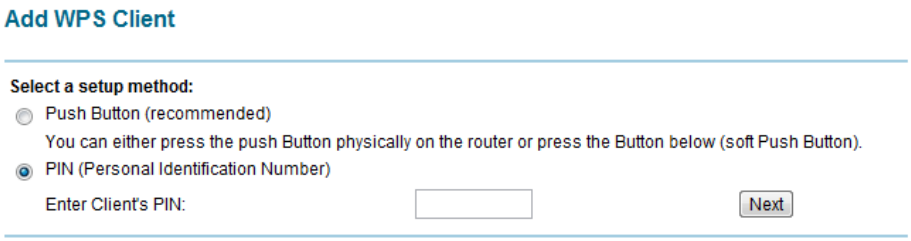


[צילום מסך מממשק ה-web של נתב CG3000 של Netgear. הממשק מציג כפתור וירטואלי המפעיל את פרוטוקול WPS במוד PBC.]

2. Out-of-band Configuration - בגישה זו משתמשים בערוץ תקשורת נפרד, כגון כונן USB או ממשקי NFC (Near Field Communication) כדי להעביר הודעות בין ההתקנים, ובפרט מידע שהם יכולים להסתמך עליו לצורך אימות הדדי על גבי הרשת האלחוטית.

3. In-band Configuration - בגישה זו שני הצדדים משתמשים בתווך האלחוטי כדי לוודא ששניהם מסכימים על סוד משותף קצר (מספר בן 4 או 8 ספרות). גישה זו מסתמכת על גורם אנושי שיקרא את המספר מאחד ההתקנים, ויקליד אותו בשני. יש שתי גרסאות של גישה זו, תלוי מי הגורם הקובע את קונפיגורציית הרשת ו"מנהל" את התהליך:

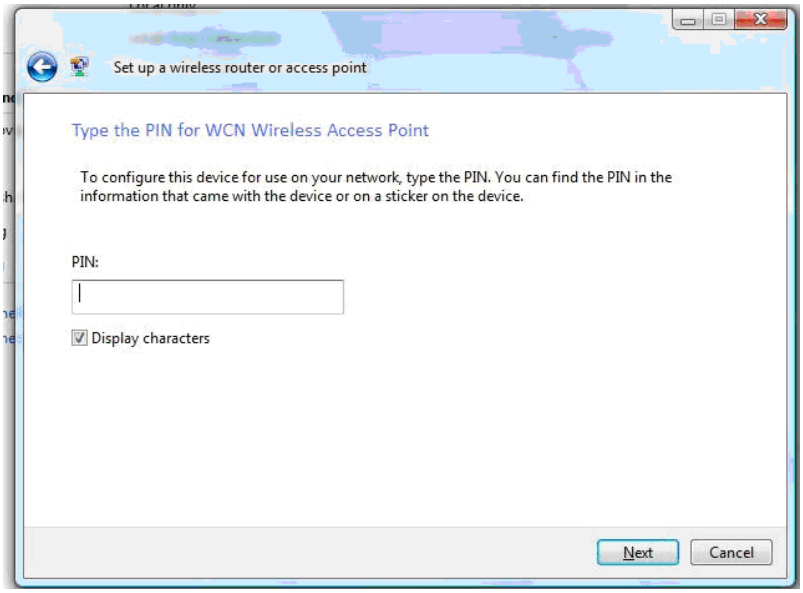
   a. Internal Register - במוד זה התהליך מנוהל על-ידי הנתב האלחוטי. כדי להפעיל את התהליך, המשתמש צריך להתחבר לממשק ה-Web-י של הנתב (דבר שדורש אימות של המשתמש כבעל הרשאות לשינוי הגדרות רשת). בממשק זה הוא יכול להזין מספר סידורי (PIN) המזהה את ההתקן שהוא רוצה לחבר. למשל, מדפסת עם לוחית בקרה יכולה להגריל מספר בן ארבע ספרות ולהציג אותו על המסך. המשתמש יזין את מספר זה בממשק הנתב, ומכאן הנתב והמדפסת יוכלו להתחבר זה לזה באופן אוטומטי.



[צילום מסך מממשק ה-web של נתב CG3000 של Netgear. הממשק מאפשר להזין מספר אותו המשתמש קורא מההתקן שהוא רוצה לחבר לרשת, כדי להפעיל את פרוטוקול WPS במוד Internal Registrar.]

   b. External Register - במוד זה התהליך מנוהל על-ידי התקן הלקוח. לדוגמה, מערכת ההפעלה Windows 7 תומכת בתהליך זה. היא מאפשרת למשתמש להזין מספר המזהה את הנתב האלחוטי. למשל, זה יכול להיות מספר קבוע בן שמונה ספרות המודפס על תווית על-גבי הנתב. נתבים יקרים יותר יכולים לכלול גם מסך שעל פניו יציגו PIN (Personal Identification Number) המוגרל לצורך החיבור, ובמקרה כזה מותר להם להשתמש בארבע ספרות.

[תווית המודבקת על-גבי הנתב האלחוטי, ועליה מודפס PIN עבור WPS. התמונה נלקחה מהמאמר של סטפן ויבוק.]



[הגדרת נתב ב-Windows 7 על-ידי הזנת ה-PIN המודבק עליו. התמונה נלקחה מ-Windows Connect Now Spec, שם מתואר התהליך המלא.]

כדי לחדד את ההבדל בין הדרכים, נדמיין האקר משועמם שרוצה לגלוש ברשת האלחוטית המוגנת של השכן או של עסק מקומי, ממגרש החניה הקרוב. אם הרשת הותקנה בצורה נכונה (מבחינת הגדרות אבטחה), היא משתמשת בפרוטוקול חזק כגון WPA2, המוגן באמצעות סיסמה חזקה שתהפוך התקפה ישירה ללא מעשית. המנגנון של WPS עשוי לספק דרך לעקוף את תהליך החיבור הסטנדרטי, שבו מספקים סיסמה עבור WPA2. כדי להפעיל את דרך החיבור הראשונה של WPS, ההאקר צריך גישה פיזית לנתב כדי ללחוץ על הכפתור שמפעיל את ה-PBC (Push Button Connect) ומאפשר לו להתחבר, דבר שאינו מעשי בתרחיש המתואר.

כדי להפעיל את דרך החיבור השנייה, Out-of-band configuration, ההאקר צריך גישה לערוץ הנפרד בו מתקיימת התקשורת (למשל להשיג את כונן ה-USB שבו השתמשו להעברת מידע אימות), דבר שגם הוא אינו מעשי בתרחיש זה. כדי להפעיל את דרך החיבור 3א', In-band Configuration with Internal Registrar, ההאקר צריך להיות מסוגל להתחבר לממשק ה-web של הנתב ולהזדהות כמנהל הרשת, וגם זו לא דרך מעשית. כדי להפעיל את דרך החיבור 3ב', In-band Configuration with External Registrar, כל מה שההאקר צריך לדעת זה את ה-PIN של הנתב. לפיכך, שיטה זו חשופה להתקפת Brute Force, בה ההאקר ינסה לנחש את המספר פעמים רבות. מנקודה זו והלאה נתמקד בדרך זו, ונבחן את החשיפה שלה להתקפה כזו.

בטיחות השימוש ב-PIN לאימות

הסטנדרט של WPS מציג שתי חלופות לשימוש ב-PIN. החלופה הראשונה מניחה שלנתב האלחוטי יש מסך המאפשר להציג PIN דינמי. בכל פעם שתהיה הרצה של פרוטוקול WPS, הנתב ייצר PIN חדש ויציג אותו על המסך. במקרה זה, הסטנדרט דורש מספר בן ארבע או שמונה ספרות. עם זאת, כיוון שמסך המציג PIN דינמי ייקר את עלויות הנתב, יצרני נתבים מעדיפים לרוב את החלופה השנייה, שנועדה למקרים בהם אין לנתב דרך "לתקשר" PIN דינמי. במקרים אלה היצרן נדרש לספק מספר אקראי בן 8 ספרות, שלא יהיה תלוי באף תכונה של המכשיר (כלומר, ה-PIN לא יהיה תלוי בדגם המכשיר, או בכתובת ה-MAC שלו).

הבחירה בסטנדרט הייתה בכוונה במספר קצר יחסית, כדי להפוך את התהליך לקל ופשוט יותר. יתר על כן, מתוך 8 הספרות, האחרונה משמשת כספרת Checksum התלויה בספרות האחרות, במטרה לזהות הקלדה שגויה של המספר עוד לפני שמריצים את הפרוטוקול ולאפשר למשתמש לתקן את השגיאה (במקרה של 4 ספרות אין שימוש ב-Checksum). מכאן שבפועל התוקף צריך לנחש במקרה זה 7 ספרות, סך הכל 10,000,000 ערכים אפשריים, שווה ערך בקירוב לניחוש מפתח סימטרי בן 23 סיביות. עבור מחשב זהו מרחב חיפוש מאוד קטן, ולכן כאשר נתב משתמש ב-PIN סטטי לזיהוי, הסטנדרט של WPS מחייב אותו גם לעקוב אחרי נסיונות התחברות כושלים: אם זוהו שלושה נסיונות התחברות כושלים בטווח של 60 שניות, הנתב צריך להינעל למשך 60 שניות.

בחישוב פשטני, בקצב של שלושה נסיונות חיבור בדקה, יקח 6.34 שנים לבדוק את כל הערכים האפשריים, ובממוצע יקח 3.17 שנים להגיע לניחוש נכון. זוהי אמנם לא הגנה מושלמת, אך הגנה סבירה בהחלט לתרחיש המדובר, כיוון שפרק הזמן הנדרש הינו ארוך מספיק כדי להפוך את ההתקפה ללא

כדאית. חשוב לציין שהעיכוב שנוצר על-ידי נעילת הנתב עוזר רק במקרה של התקפה מקוונת (online), בה התוקף מבצע פרוטוקול "חי" מול הנתב. כפי שיתואר בהמשך, הפרוטוקול תוכנן מראש לסקל התקפות offline, בהן התוקף מבצע הרצה כושלת של הפרוטוקול מול הנתב, "מקליט" אותה ואחר-כך משתמש בהקלטה כדי לבדוק איזה ערכי PIN "מסתדרים" עם הנתונים שהנתב שלח. במידה והתקפה כזו הייתה אפשרית, מרחב חיפוש של 23 סיביות לא היה מספק להגנה והיה ניתן "לשבור" את הפרוטוקול בקלות.

איך עובד WPS?

פרוטוקול WPS כולל 8 הודעות, והמורכבות שלו נובעת מכך שהוא מנסה להתגונן מפני התקפות שונות ולספק תכונות אבטחה רבות, ביניהן הגנה מפני שידורים חוזרים (Replay Attacks), אימות הדדי, יצירת ערוץ תקשורת מאובטח, ומניעת התקפות offline. בהמשך מופיע תיאור של המרכיבים המרכזיים בפרוטוקול, כדי להבהיר מה נקודת התורפה אותה ניצל סטפן ויבוק בהתקפה שלו, אולם אין כאן מטרה לבצע הצגה ממצה או ניתוח מפורט של הפרוטוקול, ולפרטים נוספים הקורא המעוניין יכול לעיין במקורות המוזכרים בסוף המאמר.

כדי להבין כיצד WPS עובד (ובהמשך, למה הוא לא עובד), נסתכל תחילה על הבעייה היסודית שהוא מנסה לפתור והאתגרים שהיא מציבה. ברמה הבסיסית, שני צדדים צריכים לנצל ערוץ תקשורת לא מאובטח כדי לאמת באופן הדדי שהם מסכימים על סוד קצר משותף (מספר בן 8 ספרות), ולייצר ערוץ תקשורת מאובטח שבו יוכלו להעביר את פרטי הקונפיגורציה עבור הרשת האלחוטית. לצורך יצירת ערוץ תקשורת מאובטח WPS מסתמך על פרוטוקול Diffie-Helman להסכמה על מפתחות. הסטנדרט כולל הגדרה של מספר ראשוני גדול p ובסיס g, אותה מכירים כל ההתקנים התומכים ב-WPS. צד אחד מגריל מפתח פרטי a ושולח לצד השני מפתח פומבי $g^a \bmod p$. הצד השני מגריל מפתח פרטי b ושולח בחזרה מפתח פומבי $g^b \bmod p$. מרגע זה, שני הצדדים יכולים להשתמש במפתח המשותף:

$$g^{ab} \bmod p = (g^a)^b \bmod p = (g^b)^a \bmod p$$

כדי לייצר ערוץ תקשורת מאובטח, בעוד כל גורם שמאזין לתעבורה לא יוכל לייצר את המפתח הסודי המשותף מתוך המפתחות הפומביים שהוחלפו. פרוטוקול Diffie-Helman, עם זאת, חשוף להתקפות Man In The Middle, כלומר גורם הנמצא בין שני הצדדים יכול להחליף את המפתחות העוברים מצד לצד. כאן נכנס השימוש בסוד הקצר המשותף, ה-PIN בן 8 הספרות - סוד זה אינו מספק בפני עצמו כחומר גלם למפתחות שיגנו על הערוץ המאובטח, אך ה"ערבול" שלו ביחד עם מפתחות Diffie-Helman שהוחלפו מאפשר לוודא שהחלפת המפתחות נעשתה בצורה תקינה.

כפי שצויין לעיל, הרצה של הפרוטוקול צריכה להימנע מלהסגיר "רמזים" בהם תוקף יוכל להשתמש להתקפות offline, מאחר ומספר בן 8 ספרות אינו מספיק ארוך כדי לעמוד בפני התקפה כזו. לצורך זה, הפרוטוקול מסתמך על העקרון של הוכחות באפס-ידע (ראו סקירה רחבה יותר של הנושא בכתבה מגליון 9), והתהליך נעשה בשלבים - תחילה כל צד "מתחייב" לערך מסויים, ורק אחרי ששני הצדדים מתחייבים, הם מגלים את הנתונים ששימשו ליצירת ההתחייבויות. יתר-על-כן, ה-PIN מחולק לשני חלקים, PIN1 ו-PIN2 וכל צד חושף מידע על PIN2 רק אחרי שבדק שהצד השני מכיר את PIN1.

להלן תיאור הפרוטוקול (הדיאגרמה נלקחה מהמאמר של סטפן ויבוק, ומבוססת על Windows Connect Now Spec):

M1	Enrollee → Registrar	N1 || Description || PK_E	Diffie-Hellman Key Exchange
M2	Enrollee ← Registrar	N1 || N2 || Description || PK_R || Authenticator	
M3	Enrollee → Registrar	N2 || E-Hash1 || E-Hash2 || Authenticator	
M4	Enrollee ← Registrar	N1 || R-Hash1 || R-Hash2 || $E_{KevWrapKev}$(R-S1) || Authenticator	proove posession of 1st half of PIN
M5	Enrollee → Registrar	N2 || $E_{KeyWrapKey}$(E-S1) || Authenticator	proove posession of 1st half of PIN
M6	Enrollee ← Registrar	N1 || $E_{KeyWrapKey}$(R-S2) || Authenticator	proove posession of 2nd half of PIN
M7	Enrollee → Registrar	N2 || $E_{KeyWrapKey}$(E-S2 ||ConfigData) || Authenticator	proove posession of 2nd half of PIN, send AP configuration
M8	Enrollee ← Registrar	N1 || $E_{KeyWrapKey}$(ConfigData) || Authenticator	set AP configuration

Enrollee = AP Registrar = Supplicant = Client/Attacker PK_E = Diffie-Hellman Public Key Enrollee PK_R = Diffie-Hellman Public Key Registrar Authkey and KeyWrapKey are derived from the Diffie-Hellman shared key. Authenticator = $HMAC_{Authkey}$(last message || current message) $E_{KeyWrapKey}$ = Stuff encrypted with KeyWrapKey (AES-CBC)	PSK1 = first 128 bits of $HMAC_{AuthKey}$(1st half of PIN) PSK2 = first 128 bits of $HMAC_{AuthKey}$(2nd half of PIN) E-S1 = 128 random bits E-S2 = 128 random bits E-Hash1 = $HMAC_{AuthKey}$(E-S1 || PSK1 || PK_E || PK_R) E-Hash2 = $HMAC_{AuthKey}$(E-S2 || PSK2 || PK_E || PK_R) R-S1 = 128 random bits R-S2 = 128 random bits R-Hash1 = $HMAC_{AuthKey}$(R-S1 || PSK1 || PK_E || PK_R) R-Hash2 = $HMAC_{AuthKey}$(R-S2 || PSK2 || PK_E || PK_R)

1	2	3	4	5	6	7	0
1st half of PIN				checksum 2nd half of PIN			

בתרחיש שאנו דנים בו (External Register), הנתב ממלא את תפקיד ה-Emrolee בפרוטוקול, וההתקן המתחבר לרשת (או ההאקר) ממלא את תפקיד ה-Register. בשתי ההודעות הראשונות, הצדדים מחליפים את מפתחות Diffie-Helman הפומביים, PKE של ה-Emrolee, ו-PKR של ה-Register. בהודעות M3-M7 מתבצע האימות ההדדי, תוך התבססות על ה-PIN כסוד משותף לאימות. בהודעות 7M ו-8M מועברים פרטי הקונפיגורציה של הרשת, אחרי שנעשה אימות הדדי והוקם ערוץ תקשורת מאובטח.

החלק של הפרוטוקול אותו מנצלת ההתקפה של סטפן ויבוק הוא האימות ההדדי בהודעות M3-M7. כדי לפשט את ההסבר, אני אנקוט בגישה דומה לזאת שנקט דן קמינסקי בפוסט בו תיאר את ההתקפה, ואציג תחילה גרסה בסיסית של הפרוטוקול הכוללת רק חילופי הודעות בסיסיים. לצורך זה, נחליף את Enrolee בשם Router, ונחליף את Registrar בשם Attacker (במקרה הכללי זה יהיה התקן המנסה להתחבר לנתב).

התוקף מנסה לגלות את ה-PIN באמצעות ניחושו. הפרוטוקול הבסיסי עובד כך:

1. Router -> Attacker: Hash(RandomString1 || PIN)
2. Attacker -> Router: Hash(RandomString2 || PIN), RandomString2
3. Router -> Attacker: RandomString1

בהודעה הראשונה, הנתב "מתחייב" למחרוזת RandomString1, באמצעות כריכתה עם ה-PIN עם פונקציית hash קריפטוגרפית. בשלב זה, מאחר והמחרוזת האקראית אותה הנתב יצר לא ידועה, לא ניתן ללמוד מתוך ה-hash דבר על ערכו של ה-PIN. בהודעה השנייה, התוקף נדרש להתחייב גם הוא, הפעם למחרוזת RandomString2, והוא מייד "פותח את ההתחייבות" באמצעות חשיפת המחרוזת. בשלב זה הנתב משתמש ב-RandomString2 כדי לייצר את ה-hash עם ה-PIN הידוע לו, והוא משווה אותו ל-Hash שקיבל בהודעה. במידה והם שווים, זה מראה ששולח ההודעה ידע את ה-PIN גם כן, ואז הנתב "פותח" את ההתחייבות שלו באמצעות שליחת RandomString1. במידה והם שונים, הנתב חייב להחזיר הודעת שגיאה ולא להמשיך את הרצת הפרוטוקול. אין לו שום ברירה אחרת - אם ינסה "לזייף" RandomString1 כאילו שה-PIN היה נכון, התוקף יגלה מיד את הזיוף, שכן חישוב hash עם ה-PIN שניחש ו-RandomString1 המזוייף לא יתאים ל-hash שקיבל מהנתב בהודעה הראשונה. לעומת זאת, אם הנתב ימשיך וישלח את RandomString1 האמיתי, הרי שהרגע נתן לתוקף חומר להתקפת offline - התוקף יוכל לבדוק את כל הערכים האפשריים עבור PIN, עד שימצא אחד שבשילוב עם RandomString1 נותן את אותו hash שהנתב שלח בהודעה הראשונה. מכאן ששליחת הודעת שגיאה היא הפתרון הקביל היחיד במצב זה.

כפי שצויין לעיל, עבור ההתקפה הנתונה, המימוש שתואר כרגע עשוי להיות פתרון מספק עבור PIN בן 7 ספרות, כיוון שיידרש לכך פרק זמן ארוך מספיק כדי להרתיע תוקף מביצוע ההתקפה. נסתכל כעת על גרסה מורכבת יותר של הפרוטוקול, שדומה יותר למתרחש בהודעות M3-M7 בפרוטוקול WPS, ובה במקום להסתכל על ה-PIN כמספר אחד, מחלקים אותו לשני חלקים, PIN1 ו-PIN2 להלן, שההתחייבויות על שניהם נשלחות יחד, אך נפתחות בנפרד:

M3:Router -> Attacker:hash(RandomString1 || PIN1), hash(RandomString3 || PIN2)

M4:Attacker-> Router:hash(RandomString2 || PIN1), hash(RandomString4 || PIN2), RandomString2

M5: Router -> Attacker:RandomString1

M6: Attacker -> Router:RandomString4

M7: Router -> Attacker:RandomString3

מחרוזות ה-hash באמצעותן הנתב מתחייב מכונות E-Hash1 ו-E-Hash2 בפרוטוקול המקורי, והמחרוזות האקראיות הן E-S1 ו--E-S2. באופן דומה, התוקף מתחייב באמצעות מחרוזות ה-hash R-Hash1 ו-R-Hash2, והמחרוזות האקראיות שהוא שולח הן R-S1 ו-R-S2. ההבדל הנוסף הוא שב-hash משולבים גם מפתחות Diffie-Helman הפומביים משתי ההודעות הראשונות, כך שחילופי ה-hash מספקים גם אימות הדדי על מפתחות אלה כדי לספק הגנה מפני Man in the Middle על המפתחות

העובדה שהתהליך מבוצע בשני שלבים, קודם עם חלקו הראשון של ה-PIN ואחר-כך עם חלקו השני, היא מה שמאפשר לתוקף לבצע התקפת Brute Force ולחשוף את ה-PIN בפרקי זמן קצרים בהרבה מהתכנון המקורי, וזו נקודת התורפה שסטפן ויבוק ניצל. בעקרון, התוקף מנסה לחשוף את כל אחד מהחצאים בנפרד. לחלק הראשון של ה-PIN ישנם 10,000 ערכים אפשריים. התוקף פשוט מנסה את כל אחד מהם באופן סדרתי, ושולח בהתאם את הודעה 4M. אם הנתב המותקף החזיר תשובה שלילית ו"הרג" את הפרוטוקול, התוקף פשוט מתחיל פרוטוקול חדש (אם צריך, אז לאחר המתנה קצרה במקרה שהנתב "ננעל" לאחר הנסיון הכושל) עם ה-PIN הבא בתור.

כאמור לעיל, אם ה-PIN שניחש אינו נכון, אין לנתב שום ברירה אחרת מלבד שליחת הודעת כשלון. ברגע שיש הרצת פרוטוקול שבה הנתב ממשיך עם הודעה 5M, התוקף יודע שהוא ניחש את החצי הראשון נכון, ומרגע זה הוא יכול להתמקד בחצי השני. מאחר והספרה האחרונה בחצי השני היא ספרת Checksum אותה ניתן לחשב על סמך שאר הספרות ב-PIN, בפועל ישנם רק 1,000 ערכים אפשריים אותם צריך לבדוק. ההפרדה בין שני חלקי ה-PIN הופכת בפועל מרחב ערכים הכולל 10,000,000 אפשרויות למרחב בן 11,000 אפשרויות בלבד, כאשר במקרה הממוצע צריך לסרוק 5,500 מהן עד למציאת ה-PIN הנכון. במקום תהליך התקפה שעשוי לארוך שנים, ההתקפה יכולה לחשוף את ה-PIN הנכון בטווח של מספר שעות עד יום. ברגע שה-PIN ידוע, ניתן להמשיך את הפרוטוקול ולקבל את הקונפיגורציה של הרשת, לרבות סיסמת WPA2. החלפת הסיסמה של הרשת האלחוטית לא תעזור, מכיוון שתמיד אפשר יהיה להשתמש שוב ב-PIN כדי לקבל את הסיסמה המעודכנת.

איך זה קרה?

מה בעצם התועלת בביצוע ההתחייבויות ופתיחת ההתחייבויות בנפרד על כל אחד מחלקי ה-PIN בחמש הודעות, במקום לעשות התחייבות ופתיחת התחייבות של כל ה-PIN בשלוש הודעות? איך חמק הפגם הזה מעיניהם של מתכנני הפרוטוקול?

ככל הנראה (זוהי ספקולציה) שורש הבעייה אינו בפרוטוקול עצמו, אלא באופן בו הוא יושם במסגרת WPS. בספרות האקדמית יש לא מעט מאמרים העוסקים באימות על סמך סוד קצר, אך בכולם נקודת ההנחה היא שהסוד הזה הוא חד-פעמי. חלוקת ה-PIN לשני חלקים נועדה להתמודד עם התקפות Man In The Middle בהן התוקף מתחזה לנתב (ולא להתקן המתחבר לנתב, כמו בהתקפה שתוארה לעיל). במקרה כזה, התוקף יכול לשלוח בהודעה 3M הודעות אקראיות כהתחייבות, ואז לקבל בהודעה 4M התחייבות ופתיחת התחייבות מה-Register על החלק הראשון של ה-PIN. מידע זה מאפשר לו לבצע חיפוש ממצה על טווח ערכים מצומצם, ולחשוף את הערך האמיתי של PIN1. אולם זה לא עוזר לו - חלוקת הסוד לשני חלקים למעשה "תוקעת" אותו. הוא אמנם כבר יודע את ערכו של החצי הראשון, אך הוא כבר נתן התחייבות בהודעה קודמת, וכעת הוא צריך להשקיע מאמץ גדול (ובפועל לא מעשי) כדי למצוא RandomString1 שיתאים בדיעבד ל-PIN1 הנכון. בעקבות זאת, התוקף לא יוכל לשלוח הודעה שתביא לגילוי החצי השני של ה-PIN.

בעולם שבו ה-PIN הוא חד-פעמי, בהרצה הבאה של הפרוטוקול התוקף לא יוכל להשתמש במידע שגילה כבר, מאחר ויהיה כבר PIN חדש. לעומת זאת, בעולם של WPS, בו אפשרי PIN סטטי, התוקף יכול להשתמש בשיטה זו כדי להתחזות לנתב יחסית בקלות וכך לגלות את ה-PIN. שיטה זו תאפשר את חשיפת ה-PIN במהירות ובקלות, אך ניתן להוציא אותה לפועל רק אם התמזל מזלו של התוקף והוא היה בסביבה בזמן שמשתמש לגיטימי ניסה לחבר התקן חדש לנתב. את ההתקפה של סטפן ויבוק, לעומת זאת, ניתן להוציא לפועל בכל זמן שהוא, מאחר ובאופן מעשי נתבים זמינים לחיבור בכל זמן שהוא.

גורם המפתח כאן הוא ששימוש ב-PIN דינמי דורש שלנתב האלחוטי יהיה מסך שבו הוא יכול להציג את הערך המעודכן למשתמש. הוספת מסך תייקר את עלות הנתב, ולכן מתכנני WPS איפשרו שימוש ב-PIN סטטי שיופיע על מדבקה על גבי הנתב. שיקולי העלות הביאו לפגיעה משמעותית בבטיחות הפרוטוקול.

עוד בעיות...

כאמור, יום אחרי פרסום נקודת התורפה, הודות לחברת Tactical Network Solutions כבר היה זמין ברשת קוד פתוח לכלי בשם Reaver היודע לנצל את נקודת התורפה. מספר ימים לאחר-מכן כבר הופיעה כתבה ב-Life Hacker עם הסבר מפורט איך להשתמש בכלי כדי לפרוץ לרשתות אלחוטיות. בינתיים הוקם ברשת דף העוקב אחר דגמי הנתבים השונים, האם הם חשופים לבעייה של WPS והאם הם מאפשרים למנוע אותה. תמונת המצב מראה כי מעבר לבעייה הבסיסית בפרוטוקול, חלק מיצרני הנתבים נקטו ב"קיצורי דרך" נוספים המחריפים את הבעייה. כמה מפגמי המימוש הנוספים שניתן למצוא:

1. ישנם נתבים בהם לא ממומשת נעילה לאחר מספר נסיונות כושלים, כך שמשך ההתקפה הנדרש עד חשיפת ה-PIN מתקצר משמעותית.
2. ישנם יצרני נתבים שלא טרחו לייצר מספרי PIN שונים לנתבים שונים. למשל, סדרה שלמה של נתבים משתמשת ב-PIN הקבוע 12345670, כך שלא נחוץ אפילו לנחש את ה-PIN אם ידוע דגם הנתב (או פשוט אפשר להתחיל את סדרת הניחושים במספרי ה-PIN הידועים).
3. בחלק מהנתבים אין שום אפשרות לנטרל את המנגנון של WPS, כך שגם למשתמש המודע להתקפה אין דרך לחסום אותה על הנתב שלו.
4. בחלק מהנתבים ישנה אפשרות בממשק המשתמש לנטרל את WPS, אך בפועל גם לאחר ניטרול WPS בממשק המשתמש הנתב נשאר חשוף להתקפה (כלומר, השינוי בממשק המשתמש לא באמת משפיע על פעולת הנתב).

דוגמאות אלה לא מתייחסות לנתבים שיוצרו על-ידי יצרנים איזוטריים, אלא לנתבים שייצרו חברות ידועות ומוכרות בתחום, ביניהן Belkin, Cisco, Netgear ו-Linksys.

בתחילת ינואר דן קמינסקי (חוקר אבטחת מידע ידוע) פרסם סקירה במהלך טיול בברלין, שם העריך כי כרבע מנקודת הגישה המאובטחות בהן נתקל היו חשופות להתקפה. הוא העריך, באופן שמרני, כי יש כ-4 מליון נקודת גישה החשופות להתקפה (וקרוב לודאי שיותר מזה), כך שבהחלט מדובר בתופעה רחבת היקף.

מה לעשות?

בהתאם לדגם הנתב שברשותם, לבעלי נתבים עשויות להיות מספר אפשרויות פעולה. הצעד הראשון אותו כל בעל נתב יכול לנקוט במידה והנתב שלו תומך ב-WPS, היא פשוט לנטרל את המנגנון. לרוע המזל, כפי שצויין קודם, האפשרות הזאת לא זמינה בכל הנתבים (ולפעמים גם באלה שכן היא לא עושה כלום). במקרה כזה, חלופה אחת היא לחרוק שיניים ולהמתין לעדכון קושחה מטעם היצרן שיאפשר את נטרול המנגנון. חלופה אפשרית אחרת, יותר מורכבת, היא להתקין על הנתב (במידה וניתן) קושחה כדוגמת Tomato או DD-WRT, שאינן תומכות ב-WPS ולכן אינן חשופות להתקפה. כל האפשרויות שלעיל מומלצות אך ורק למשתמשים מתקדמים (ועל אחריותם בלבד).

בטווח הארוך יותר, מהנדסי ה-Wi-Fi Alliance יצטרכו לבחון כיצד לתקן את הבעייה. מצד אחד הפשטות של WPS הינה מרכיב חשוב בהטמעת WPA2, ובלעדיו יתכן שמשתמשים רבים ירתעו מתהליך הגדרת WPA2 וישאירו את הרשת האלחוטית פתוחה ולא מאובטחת. מצד שני תיקון הפרוטוקול לא יהיה טריוויאלי, וכפי שהוסבר לעיל, "פתרונות קסם" כדוגמת תשובות כוזבות לתוקף אינם אפשריים כאן. יש לציין ששינוי הגדרות הנתב כך שיגביל את הגישה לרשת רק להתקנים בעלי כתובות MAC מסויימות, אינה אמצעי הגנה אפקטיבי במיוחד. זיוף כתובת MAC (spoofing) הוא קל, ולא יהווה מכשול אמיתי בפני תוקף המעוניין להשיג גישה לרשת.

לעומת זאת, קרוב לוודאי שעם הזמן יצוצו כלים פשוטים, צאצאים של Reaver, שיבטיחו להשיג גישה לרשתות אלחוטיות מוגנות בלחיצת כפתור, וללא שום רקע טכני. האקרים נטולי עכבות מוסריות יגלו שקל יותר להתחבר לאינטרנט בחינם מכל מקום, כשחלק לא מבוטל מהרשתות האלחוטיות הביתיות יהיו זמינות להם למרות היותן מאובטחות. ככל הנראה הבעייה הזאת תלווה את העוסקים בתחום במהלך החודשים והשנים הקרובות.

מקורות ומידע נוסף

1. Wi-Fi Protected Setup PIN Brute force vulnerability, blog post by Stefan Viehböck (published on 27.12.2011).
2. Cracking WiFi Protected Setup with Reaver, blog post on Tactical Network Solutions website (published on 28.12.2011).
3. Windows Connect Now-NET - A Windows Rally Specification, 8.12.2006.
4. How the WPS bug came to be, and how ugly it actually is, blog post by Dan Kaminsky, 26.1.2012.
5. Security Associations in Personal Networks - A Comparative Analysis, by Jani Suomalainen, Jukka Valkonen and N. Asokan, Nokia Research Center, 2007.
6. Security Now Podcast, Episode 335: WiFi Protected (In)Security, with Steve Gibson and Leo Laporte, recorded on 9.1.2012.
7. Security Now Podcast, Episode 337: WPS: A Troubled Protocol, with Steve Gibson and Leo Laporte, recorded on 25.1.2012.

על המחבר

ד"ר אריק פרידמן עובד כחוקר במכון המחקר NICTA בסידני, אוסטרליה. תחומי המחקר שלו מתמקדים בפרטיות ואבטחת מידע, ובעיקר שילובם במסגרת אלגוריתמים ללמידה ממוחשבת וכריית נתונים. אריק סיים את לימודי הדוקטורט בפקולטה למדעי המחשב בטכניון בשנת 2011, והוא מחזיק גם בתואר MBA מאוניברסיטת תל-אביב.