Digital Whisper

גליון 48, ינואר 2014

מערכת המגזין:

מייסדים:	אפיק קסטיאל, ניר אדר
מוביל הפרויקט:	אפיק קסטיאל
עורכים:	שילה ספרה מלר, ניר אדר, אפיק קסטיאל
כתבים:	דר' גבי נקבלי, אפיק קסטיאל (cp77fk4r), יוני יחזקאל, עו"ד יהונתן קלינגר, נדב איבגי וליאור גבעון.

פניות, תגובות, כתבות וכל הערה אחרת - נא לשלוח אל editor@digitalwhisper.co.il

דבר העורכים

ברוכים הבאים לגיליון ה-48 של DigitalWhisper! הגיליון הראשון של שנת 2014...

עם צאת הגיליון ה-48 אנו חוגגים שני אירועים, הראשון - פתיחת שנת 2014, שזה מגניב לכשעצמו, והשני - העובדה המתמטית שהמספר 48 מתחלק ב-12 ללא שארית. מה שאומר שסגרנו עוד שנת פעילות של המגזין, שנה רביעית ברציפות של פרסום גליונות. במובני זמן של קהילת ההאקינג המקומית, מדובר, ללא ספק, בפרק זמן מכובד ביותר ☺

אז ראשית, היינו רוצים לאחל לכל הקוראים שנה אזרחית טובה, ושנית, להגיד תודה רבה לכל מי שעזר לנו השנה, השקיע מזמנו וכתב מאמרים. תודה רבה ל:

לירן בנודיס, רועי (Hyp3rInj3cT10n), יובל נתיב, shackrack, סשה גולדשטיין, חיליק טמיר, שי חן, אמיר שגיא, ניר גלאון, דן פלד, בר חופש, יובל נתיב, ישראל חורז'בסקי (Sro), אמיתי דן (PopShark), ד"ר אריק פרידמן, מיתר קרן, יונתן גולדהירש, רון הרניק, יצחק דניאל (iTK98), שחר גייגר מאור, יוחאי (hrr) אטון, עו"ד לילך צאירי-כהנוב, שרון ברק, ד"ר גדי אלכסנדרוביץ', עו"ד יהונתן קלינגר, יובל סיני, דודו ברודה, משה פרבר, רועי חי, אריק יונאי, לאוניד יזרסקי, יורי סלובודיאניוק, מריוס אהרונוביץ', רן לוי, יניב מרקס, רונן שוסטין (Antartic), פלג הדר (P), תומר זית, ליאור בר-און, מור כלפון, דר' גבי נקבלי, יוני יחזקאל, נדב איבגי וליאור גבעון.

וכמובן, ברצוננו להודות במיוחד לכל מי שכתב מאמרים לגיליון הנוכחי, ובזכותו המגזין ממשיך להתפרסם: תודה רבה ל**דר' גבי נקבלי**, תודה רבה ל**יוני יחזקאל**, תודה רבה ל**עו"ד יהונתן קלינגר**, תודה רבה ל**נדב איבגי** ותודה רבה ל**ליאור גבעון**. וכמובן, תודה מיוחדת לעודכת שלנו - **שילה ספרה מלר**. בלעדיכם לא היינו כאן החודש.

קריאה מהנה!

ניר אדר ואפיק קסטיאל.

תוכן עניינים

Owning the Routing Table - OSPF Attacks

מאת דר' גבי נקבלי (עובד ברפאל ומרצה בטכניון)

תורגם מאנגלית ע"י אפיק קסטיאל (cp77fk4r)

תקציר

במאמר זה אסקור מספר מתקפות על אחד מפרוטוקולי הניתוב הנפוצים בעולם - הפרוטוקול OSPF (קיצור של Open Shortest Path First). במאמר זה אתאר בפירוט שלוש מתקפות חדשות שפרסמתי במשך השנים האחרונות בכנסי Black Hat האחרונים בלאס-וגאס. ההתקפות שעליהן נדבר, אינן מנצלות אופן מימוש ספציפי של הפרוטוקול בדגם נתבים ספציפי, אלא מנצלות כשלים בפונקציונליות הסטנדרטית של הפרוטוקול, כך שכל נתב אשר תומך בפרוטוקול זה - יהיה חשוף למתקפות אלו. בנוסף, כלל המתקפות שאציג הינן מתקפות "פנימיות" - זאת אומרת שאת כולן יש לבצע מתוך הרשת עצמה.

רב המתקפות הידועות כיום על פורטוקול ה-OSPF מתבססות על דיווח כוזב של ה- Link State Advertisement (רשימת הקשרים של נתב לשכניו) של נתב שבשליטת התוקף. למתקפות הנ"ל פוטנציאל נזק עצום אם תוקף מצליח להשתלט על נתב הנמצא בנקודה אסטרטגית ברשת. עם זאת, בשל תצורת ה-OSPF, במתקפות מסוג זה ניתן לזייף רק חלקים קטנים ברשת, מה שבדרך כלל מגביל את השפעתן.

מתקפות OSPF "חזקות" יותר הינן מתקפות אשר מאפשרות לתוקף לזייף לא רק את ה-LSA של הנתב שבשליטתו אלא גם LSA-ים של נתבים אחרים ברשת שאינם בשליטתו. עם זאת, ברב המקרים, מתקפות אלו יעירו מנגנון "Fight Back" המובנה בפרוטוקול, המאפשר לנתב קורבן אשר זיהה כי נשלח LSA כוזב בשמו לשלוח LSA נוסף עם הנתונים המתוקנים אשר יבטלו את השפעת ה-LSA המזויף, כך שמתקפות אלו ברב המקרים לא יצרו אפקט קבוע ברשת אלא זמני בלבד.

במאמר זה נתאר מספר מתקפות חדשות המנצלות חולשות "by design" בתקן של הפרוטוקול, המתקפות הנ"ל מאפשרות ליוזם אותן לזייף LSA-ים של נתבים שאינם נמצאים בשליטתו מצד אחד, ומצד שני גם להתחמק מאותו מנגנון "Fight Back". בעזרת שילוב שתי היכולות הנ"ל, אותן המתקפות מאפשרות לתוקף לשנות לאורך זמן את תמונת טפולוגיית הרשת כפי שנתבים אחרים ברשת רואים אותה ובכך להשפיע על טבלאות הניתוב של נתבים אלו. **על כן בעזרת המתקפות הנ"ל, תוקף יכול לבצע השתלטות מלאה על טבלאות הניתוב של כלל הנתבים ברשת ע"י השתלטות רק על אחד הנתבים ברשת.**

בעזרת מימוש המתקפות הנ"ל, תוקף יוכל לגרום ללולאות ניתוב ברשת, ניתוק איזורים שלמים ברשת או הארכת הניתוב ברשת על מנת לגרום לאפקט של Denial Of Services ברשת, או להנגיש איזורים ברשת או מקורות מידע שלאותו תוקף לא הייתה גישה אליהם מלכתחילה. המטרה העיקרית של מחקר זה הינה להראות כיצד לגורם בעל אופי זדוני וגישה לנתב בודד יש את היכולת להשפיע ולשנות את טופולוגיית הניתוב ברשת כולה באופן קבוע.

הקדמה

Open Shortest Path First הינו פרוטוקול הניתוב הנפוץ ביותר השייך למשפחת פרוטוקולי ה- Interior Gateway Routing. משפחת פרוטוקולים זו מאפשרת לנתבים בתוך מערכות אוטונומיות (Autonomous System, או בקיצור - AS) לבנות את טבלאות הניתוב שלהם ולעדכנן באופן דינאמי בעת זיהוי שינויים בטופולוגיית הרשת. נכון לכתיבת שורות אלו, פרוטוקול ה-OSPF ממומש ונמצא בשימוש ברב מערכות הניתוב האוטונומיות ברשת האינטרנט.

הסטנדרט עליו מבוסס הפרוטוקול נכתב ע"י ה-OSPF working groupב-IETF (Internet Engineering Task Force). כיום, ה-OSPF נמצא בשימוש בגרסאתו השניה (RFC2328), שעוצבה במיוחד עבור עבודה אל מול רשתות IPv4 (ולכן רק גרסה זו נמצאת בשימוש כיום), הגרסא הבאה של הפרוטוקול מעוצבת לעבודה עבור רשתות מבוססות IPv6, אך המכניזם הבסיסי של הפרוטוקול נשמר.

OSPF הינו פרוטוקול ניתוב מסוג "Link-State", כך שכל נתב ברשת מפרסם לכלל הנתבים האחרים ב-AS את הקשרים לשכנים הישירים שלו. כל נתב ברשת מסוגל לזהות את השכנים שלו באופן עצמאי על ידי שליחת הודעות "Hello" ברשת המקומית. פרסומי הניתוב מכונים "Link State Advertisements" (או בקיצור: LSA). אחד הפרטים החשובים ביותר באותם LSA, הם "עלות הקישור" לכל אחד מהשכנים, אותה עלות בדרך כלל נקבעת על פי המנהל של אותה הרשת. במקרה שנתב ברשת מקבל LSA מאחד השכנים שלה, היא תמשיך לפרסם אותה הלאה ברשת לשכנים הישירים שלה, כך שבסופו של דבר, כל נתב ונתב ברשת מסוגל להרכיב תמונה שלמה של כלל טופולוגיית הניתוב ברשת, ומכאן שכל נתב מסוגל לבצע חישוב בעזרת אלגוריתם Dijksatra ובכך לקבוע את העלות הנמוכה ביותר עבור כל נקודה ברשת ובפרט את ה-Next Hop עבור כל חבילה שהנתב קיבל.

בעבודה זו, אנו מציגים מתקפות עוצמתיות חדשות המנצלות את הארכיטקטורה של ה-OSPF במטרה לשפוך אור על חולשות אבטחה במכניזם של הפרוטוקול. כלל המתקפות שנציג במאמר מנצלות חולשות בארכיטקטורה ובסטנדרטיזציה של הגרסא השניה של הפרוטוקול (כפי שהיא ב-RFC2328), ומכאן שהבטחת הצלחת המתקפות אינה תלויה באופן המימוש של כל יצרן ויצרן. באופן קונספטואלי, כל רכיב התומך בפרוטוקול ב-OSPF עלול להיות חשוף למתקפות אלו.

מבדיקה שביצענו, המתקפות בוצעו בהצלחה אל מול הגרסא האחרונה של מערכת ההפעלה לנתבים של סיסקו iOS 15.0(1)M. המתקפות איפשרו לרכיב זדוני לחתור תחת הטופולוגייה הנוכחית של רכיב הרשת ולשנות כך את טבלאות הניתוב של כלל הרכיבים ובעזרת זה לשנות את כלל תהליך הניתוב ברשת.

כאשר לתוקף יש אפשורת לחבל ולחתור תחת טופולוגיית הרשת, באפשרותו לקבוע טבלת ניתוב עבור כל חבילת מידע, ולא משנה באיזה שכבת תעבודה היא מבצעת שימוש. שליטה בכלל טופולוגיית הניתוב ברשת מאפשרת לתוקף ליזום שתי סוגי מתקפות:

הראשונה היא היכולת לגרום למתקפת מניעת שירות (Denial Of Service) עבור חלק ספציפי ברשת (או הרשת כולה) ע"י שינוי טופולוגיית הרשת כך שחבילת המידע לא תגיע לעולם ליעדה, או תגיע באיחור רב, ניתן לעשות זאת ע"י מספר דרכים:

- **Link overload** - במידה והתוקף יחליט להעמיס נפח תעבורה הרחב בהרבה ממה שהלינק מסוגל לספק או לעמוד בו, הוא יגרום לכך שהלינק לא יוכל לספק שירות עבור מידע "אמיתי" אותו הוא אמור להעביר.
- **Long routes** - התוקף יוכל לגרום לכך שמסלול ניתוב של המידע ברשת יעבור דרך מספר רב של צמתים שאין בהם צורך אמיתי, ובכך גם לגרום לכך שהרשת תעבוד בצורה איטית וגם לכך שאותם משאבים יבזבזו את המשאבים שלהם עבור העברת אותו המידע.
- **Delivery failure** - התוקף יוכל לגרום לכך שמידע יאלץ לעבור דרך נתב ברשת שאינו מסוגל לבצע את העברה (מבחינת חוקי ניתוב וכו'), ובכך למנוע מהמידע להגיע ליעדו. או לחלופין - לגרום לאותו נתב במסלול הניתוב, לחשוב כי היא מנותקת מהרשת אליה היא מיועדת להעביר את המידע.
- **Routing loops** - במידה והתוקף ישנה את טופולוגיית הרשת כך שטבלאות ניתוב של מספר נתבים לא יהיו מסונכרנות באופן שיווצרו לולאות-ניתוב ברשת, כל מידע שיגיע אליהם יתקע באותן לולאות, האפקט כאן הוא בדיוק כמו בסעיף הקודם, רק שבמקרה הזה, יבוזבזו משאבי רשת באופן ניכר.
- **Churn** - תוקף יוכל לגרום לשינויים מאג'ורים בטופולוגיית הרשת באופן אינטנסיבי ורב ובכך להשפיע על יציבות הרשת ועל האמינות של מנגנוני בקרת העומס בה.

והשניה היא היכולת לבצע האזנה לכלל חבילות המידע ברשת, ובייחוד לחבילות מידע שללא אותן מתקפות לא יועדו לעבור דרכו. במקרה זה, תוקף יוכל להקליט ואף לשנות כל חבילת מידע העוברת ברשת ובכך ליזום מתקפות נוספות ברשת במטרה להשיג שליטה מלאה בשאר רכיבי הרשת.

בעבודה זו, אנו מניחים כי התוקף כבר נמצא בתוך הרשת והוא מסוגל לשלוח חבילות LSA לנתבים ברשת, וגם כי אותם נתבים יחשיבו את נתוני ה-LSA כאמיתיים ולכן יתייחסו אליהם. בדרך כלל, הנחה זו אינה נכונה אם התוקף ממוקם מחוץ ל-AS, מפני שכיום, רב ה-AS-ים מסננים החדרת חבילות OSPF מבחוץ. על כן עבודה זו יוצאת מנקודת הנחה כי מדובר בתוקף אשר כבר נמצא בתוך הרשת ובפרט, כי לתוקף קיימת גישה ללפחות רכיב אחד ברשת. תוקף יוכל להשיג את נקודת הבסיס הנ"ל בעזרת מספר דרכים, כגון יצירת קשר עם גורם מתוך הרשת ולשכנוע לבצע זאת, או על ידי תקיפת הרכיב בצורה מרוחקת, ע"י ניצול חולשה המאפשרת הרצת קוד באחד רכיבי הנתב, כגון חולשות שונות המאפשרות יכולת זו אשר פורסמו בעבר. לאחר השתלטות על רכיב בודד ברשת, התוקף יוכל לגרום לאותו רכיב לשלוח חבילות OSPF כרצונו אשר יתקבלו על ידי שאר הרכיבים ברשת כרלוונטים לחישוב.

בעבודה זו אנו מניחים מספר הנחות לגבי התוקף:

- **מיקום:** כאמור, אנו מניחים כי התוקף נמצא בתוך הרשת, ויש לו יכולת הרצת קוד על לפחות נתב לגיטימי אחד ברשת.
- **משאבים:** לתוקף קיימים משאבים, רוחב פס, וכמות זכרון עיבוד כמו שיש לכל נתב ממוצע ברשת, ובמיוחד, לתוקף אין יכולת לבצע חישובים מעבר למה שנתבים אחרים ברשת מסוגלים.
- **אחיזה בודדת:** לתוקף קיימת נקודת אחיזה בודדת ברשת, אין לתוקף את היכולת להמשיך להתפשט ברשת ולהגיע ליכולת הרצת קוד על נתבים וחוליות אחרות ברשת. מלבד הנתב אליו לתוקף יש גישה - שאר רכיבי הרשת נחשבים כ-"תמימים" ולא ניתן להשפיע עליהם בצורה לא טבעית.

בעבר פורסמו מספר עבודות אשר הציגו מתקפות שונות המנצלות את ארכיטקטורת ה-OSPF:

- **False self LSAs** - במתקפה זו, על התוקף לשלוח ברשת חבילות LSA בעלות מידע שקרי, כגון מידע אשר יגרום לנתבים אחרים לחשוב כי הנתב אשר נמצא בשליטתו מחובר לאיזורים ברשת אליהם הוא לא באמת מחובר, או מידע שקרי עבור עלות הקישור אל שכניו. עם זאת, מתקפה זו מאפשרת לתוקף לזייף רק את הקשרים הישירים של הנתב הנמצא בשליטתו.

- **False Hello** - במתקפה זו, התוקף שולח חבילות Hello עם מידע שקרי על מנת לגרום לשאר הנתבים ברשת המקומית לחשוב כי הם מזהים רכיבים חדשים ברשת, וכי רכיבים אשר אותם הם מכירים כבר - התנתקו. גם שימוש במתקפה זו מאפשר לתוקף להשפיע בצורה מינורית על הרשת, מפני שניתו להשפיע רק על נתבים ברשת המקומית.

- **False phantom LSA** - במתקפה זו, התוקף שולח חבילות LSA בשם נתב מדומה שאינו באמת קיים ברשת. אך הבעיה במתקפה זו, היא שלא תהיה השפעה על טבלאות הניתוב של שאר רכיבי הרשת, מפני שפרוטוקול ה-OSPF מצפה לקבל, עבור כל קישור ברשת, פרסומי LSA משני קצוותיו. אך מפני שאף נתב אמיתי אחר לא יפרסם קישור לנתב המדומה הפרוטקול לא יתייחס לקישור שפורסמו כביכול ע"י הנתב המדומה.

- **False peer LSA** - במתקפה זו, תוקף שולח פרסומי LSA בשם נתב (אמיתי) אחר שאינו נמצא בשליטתו. ע"י שימוש במתקפה זו, תוקף יוכל לזייף את כלל הקישורים ברשת ובכך להשפיע בצורה ניכרת על טבלאות ניתוב של שאר רכיבי הרשת. הבעיה העיקרית במתקפה זו היא שהשינויים אינם קבועים, מפני שאותם פרסומי LSA יגיעו גם אל הנתב אותו הוא מזייף, והוא בתורו ישלח חבילות LSA מתוקנות (כחלק ממנגנון ה-"Fight-Back") אל הרשת, ובסופו של דבר כלל הראוטרים יקבלו את אותן חבילות המידע - ויתקנו את השינויים שגרם התוקף.

בעבודה זו, אנו מציגים מתקפות חדשות, המנצלת חולשה בספציפיקציה של ה-OSPF המאפשרת לבצע False peer LSA תוך התחמקות ממנגנון ה-"Fight-Back", בכך המתקפות שתוצגנה, תאפשר לתוקף **לשנות בצורה קבועה את טבלאות הניתוב של שאר נתבי הרשת, ללא צורך להריץ עליהם קוד מרחוק**.

עבודות קרובות

כיום, יש קומץ קטן יחסית של עבודות המנתחות את מנגנוני האבטחה הקיימים ב-OSPF, ואלו הן:

Ref. [Wang97] - עבודה זו מציגה דפוס פעולה אשר בו הנתב הנשלט על יד התוקף מתחזה לנתב הנמצא בקצה הרשת (AS border router) ומפרסם חבילות LSA ליעדים שכביכול נמצאים מחוץ לרשת. דפוס פעולה זה מתאפשר מכיוון שב-OSPF אין לנתב דרך לדעת מה המיקום האמיתי של נתבים אחרים. תוקף יוכל להשתמש במתקפה זו על ידי שליחת עדכוני LSA חיצוניים (עם קישורים לכתובות IP של גוגל או פייסבוק דוגמא), ועדכונים אלו יכללו עלות קישור נמוכה מאוד, או שימוש בכתובת Subnet ארוכה יותר ובכך לנסות למשוך אליו תעבורה מנתבים אחרים ברשת. בעזרת מתקפה זו, תוקף יוכל למשוך אליו את המידע המועבר ברשת ולעשות בו כרצונו - ליצור Black-Holes ברשת, להאזין לתעבורה, או סתם לגרום לכך שהמידע יגיע בצורה איטית יותר ליעדו.

אחד החסרונות של מתקפה זו היא שהתוקף לא יוכל לגרום לשינויים בתעבורת הרשת הפנימית, מפני שבטופולוגיית OSPF, כאשר מידע נשלח מתוך הרשת אל תוך הרשת, הנתבים תמיד יעדיפו שימוש בלינקים בתוך הרשת מאשר לינקים מחוצה לה.

Ref. [Wu99] - מסמך זה, מתאר מספר מתקפות שבהן התוקף שולח חבילות LSA מפוברקות בשמו של נתב אחר הקיים ברשת. כלל המתקפות במסמך זה מעירות את מנגנון ה-"Fight-Back" על הנתב עליו התוקף מנסה לעדכן, ולכן מתקפות אלו אינן יכולות לבצע שינויים בטופולוגיית הרשת לאורך זמן, עובדה שתאלץ את התוקף לבצע את המתקפה שוב ושוב. מצד אחד, התוקף יוכל לנמף את המצב הנ"ל ולהפוך את תהליך הניתוב ברשת ללא יציב, אך מצד שני, על מנת לבצע זאת, על התוקף להשאר ברשת לאורך זמן ולהפעיל את המתקפה שוב ושוב, פעולה אשר יכולה לגרום לחשיפתו על ידי מנהל הרשת.

Ref. [Jones06] - מסמך זה מסכם את כל סוגי וקטורי התקיפה עבור OSPF, במסמך זה קיים אפילו פירוט אודות מתקפות OSPF חדשות. מתקפה אחת מבטלת את מנגנון ה-"Fight-Back" ע"י שליחה עיתית של פרסומי LSA כוזבים (חבילה אחת כל חמש שניות). שיטה זו מבטלת את אותו מנגנון הגנה על ידי ניצול העבודה כי נתב העומד בתקן OSPF מוגבל לשליחת חבילת LSA אחת בכל MinLSInterval (פרמטר שעל פי הפרוטוקול נקבע ל-5 שניות כברירת מחדל). בנוסף, התקן של OSPF מורה על הפעלת מנגנון ה-"Fight-Back" רק לאחר ניתוח חבילת ה-LSA המפובקת. מה שאומר שבמידה והנתב מקבל חבילת LSA פעם ב-5 שניות, הוא אינו יכול לשלוח חבילת LSA מתוקנת כחלק ממנגנון ה-"Fight-Back". בשל-כך מנגנון ה-"Fight-Back" מבוטל, התוקף יוכל לבצע שינויים קבועים ברשת, אך גם כאן, עלות המתקפה היא גדולה - על התוקף להמשיך לשלוח עדכוני LSA כוזבים בקצב מהיר.

מתקפה נוספת המוזכרת במסמך הנ"ל היא מתקפה אשר בה התוקף גורם לנתב אשר נמצא בשליטתו לשולח הודות "Hello" שקריות ברשת באופן כזה שיגרמו לשאר הנתבים ברשת לחדש עימו את הקשר. תהליך חידוש הקשר בין הנתבים לראוטר עליו יושב התוקף אורך כעשרות שניות. בשלב זה, החלק ברשת אותה מקשר הנתב מתפרסמת כ-Stub Network (רשת המחוברת לנתב יחיד), ושום חבילת מידע לא תשלח בשלב זה דרך הנתבים לאותה הרשת. תהליך זה יגרום לנתבים לבצע חישובי ניתוב מספר רב של פעמים ולהרכיב את טבלאות הניתוב שלהם כל פעם מחדש, ובכך להוציא את הרשת מכלל יציבות.

סוג נוסף של מתקפות המוצגות באותו מסמך, הוא מתקפות מניעת שירות (Denial Of Services), בסוג זה, התוקף גורם לנתב הנמצא בשליטתו להציף נתב אחר בצורה כזאת שתגזול ממנו את כל המשאבים. פעולה זו תגרום לנתב הנתקף להפסיק לתפקד בצורה תקינה ולצאת מכלל שימוש. במתקפה אחת המוצגת תת קטגוריה זו, התוקף שולח מספר רב של חבילות "Hello", מכתובות IP שונות אל עבר הקורבן, ובכך לגרום לנתב ליצור עוד ועוד רשומות בטבלת ה-Neighbors. על ידי הצפתו בנתונים אלו, התוקף יוכל להבטיח כי אותו נתב לא יוכל עוד לעדכן רשומות עבור נתבים אמיתיים אחרים המצטרפים לרשת. במתקפה אחרת תחת אותה הקטגוריה התוקף שולח מספר עצום של חבילות LSA אל עבר הקורבן, הנתב שמקבל אותן מחוייב לשמור אותן עד שהתוקף שלהן פג (שעה אחת), על ידי העמסת ה-LSDB (מסד הנתונים שתפקידו לשמור את נתוני ה-LSA שאותן מקבל הנתב) התוקף יוכל להבטיח כי אותו נתב לא יוכל להעבד נתוני LSA חדשים ולהתעדכן בשינויים המתבצעים בטופולוגיית הרשת.

מתקפות חדשות

כעת נתאר שלוש מתקפות חדשות המאפשרות לתוקף לשלוח עדכוני LSA לנתב אחר ברשת, אשר יגרמו לשינוים בטבלאות הניתוב שלו ללא הפעלת מנגנון ה-"Fight-Back", שתי המתקפות הראשונות שנציג, הוצגו לראשונה בכנס ההאקינג Black Hat USA 2011 בעזרתם של דימה גוניקמן ואלכס קירשון. המתקפה השלישית שתוצג בשורות הבאות, פורסמה לראשונה באותו כנס, בשנת 2013, ובעזרתם של איתן מנחם, אריאל וייזל ויובל אלוביץ'.

Disguised LSA

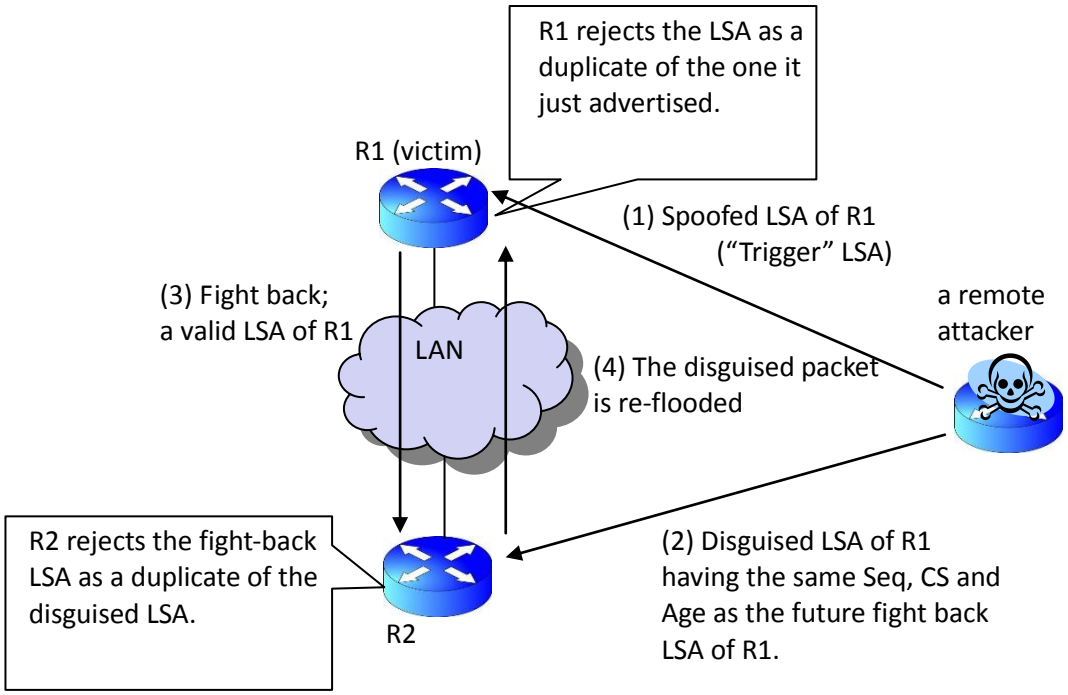
על פי RFC 2328, פרק 13.1, שני מופעים של חבילת LSA נחשבים זהים, אם מתקיימים שלושת הדברים הבאים:

- בשני המופעים של חבילת ה-LSA ה-Sequence Number זהה.
- השני המופעים של חבילת ה-LSA ה-checksum זהה.
- ערכו של שדה ה-Age בשני המופעים קרוב עד 15 דקות.

כאמור, לפי התקן, במידה ושלושת התנאים הנ"ל מתקיימים, שתי חבילות ה-LSA נחשבות אותה חבילה, והמצב כך גם אם התוכן המדווח בהן שונה. תוקף יכול לנצל עובדה זו על מנת לשלוח חבילת LSA עם אותם המזהים של חבילת LSA ואלידית (Sequence Number, Checksum ו-Age) אך עם תוכן כוזב, בשמו של נתב אחר. במקרה כזה, גם כאשר הנתב אשר בשמו נשלחה חבילת ה-LSA הכוזבת מקבל את אותה חבילה סוררת, הוא לא יפעיל את מנגנון ה-"Fight-Back", מפני ש(כביכול) מדובר באותה החבילה שהוא שלח, והוא יתייחס אליה כאל העתק של ה-LSA האמיתי שהוא פרסם.

עם זאת, ההתקפה הזאת כפשוטה לא תעבוד שכן גם נתבים אחרים ברשת יתעלמו מה-LSA הכוזב, מפני שהם גם ייתיחסו אליו כאל העתק של ה-LSA האמיתי שכבר קבילו בעבר. על מנת לסדר זאת, על התוקף להסוות את חבילת ה-LSA הכוזבת כך שתראה כחבילת ה-LSA האמיתית הבאה שמצופה מהקורבן לייצר. התוקף יגרום לקורבן לשלוח את ה- LSA האמיתי הבא בעזרת עירור מנגנון ה-"Fight –Back"

בתרשים בעמוד הבא, ניתן לראות את מהלך הדברים.



[תרשים 1 - הדגמת מהלך המתקפה]

1. בשלב הראשון, התוקף מתחיל בכך שהוא שולח ל-R1 הודעת LSA בשמו של R1 (נקרא לחבילת זו "Trigger") מהלך זה יעיר את מנגנון ה-"Fight-Back" של R1 ויגרום לו להגיב.

2. בזמן זה, התוקף שולח ל-R2, הודעת LSA שמקורה זוייף על מנת שתראה כאילו היא נשלחה מ-R1. החבילה הנ"ל נבנתה בצורה כזאת שהיא תראה כאותו מופע של חבילת ה-"Fight-Back" ש-R1 עתיד לשלוח (כל שעל התוקף לעשות הוא לייצר חבילה עם אותו Sequence Number, checksum ו-Age עם זמן משוער (בקירוב של עד 15 דקות) כמו שמצופה ממנגנון ה-"Fight-Back" לייצר. (בהמשך נראה כיצד ניתן לחזות את אותם ערכים). נקרא לחבילה זו "Disguised LSA".

3. כמצופה, כתגובה לפעולת התוקף בשלב הראשון, R1 שולח חבילת LSA מתוקנת לשאר הרשת, על מנת לתקן את הפרטים הכוזבים שנשלחו עם חבילת ה-LSA Trigger המזוייפת שנשלחה על ידי התוקף. כל זה מתרחש באופן אוטומטי ע"י מנגנון ה-"Fight-Back". החבילה הנ"ל מגיעה גם ל-R2, אך R2 בתורו, מניח כי הוא כבר קיבל העתק של חבילה זו (בשלב 2), ולכן הוא מתעלם ממנה לחלוטין. הוא לא מעדכן את ה-LSDB ולא מעביר את החבילה לשכניו.

4. R2 מעביר את חבילת ה-LSA המזוייפת שקיבל מהתוקף בשלב 2 לשאר הרשת, ובין היתר גם ל-R1, אך מפני שלחבילת ה-LSA המזוייפת יש את אותם הפרמטרים כמו לחבילת ה- LSA Fight Back ש-R1 שלח בעצמו, הוא יתעלם ממנה. לא יעביר אותה הלאה, ולא יעיר את מנגנון ה-"Fight-Back".

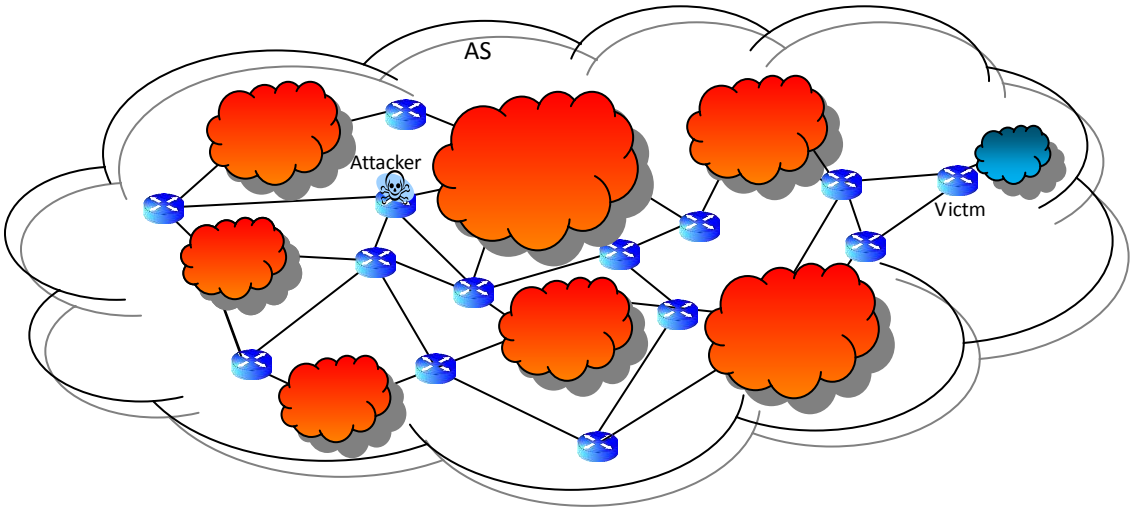
לאחר מהלך זה, ל-R1 ול-R2 יש שתי עותקים שונים של רשומות LSA אודות R1, ובשלב זה, השינוי הוא קבוע, המצב הנ"ל ישתנה, רק לאחר ש-R1 ישלח את העדכון הבא שלו לרשת (עניין של חצי שעה, אם נתחשב בערך ברירת המחדל של ה-LSA Interval).

נראה עתה מה הם ערכי שלושת השדות (Sequence Number, Checksum ו-Age) של חבילת ה-LSA שעל התוקף לשלוח ל-R2 (בשלב השני של המתקפה). קביעת הערכים של השדות Age ו-Sequence Number היא עבודה פשוטה, אם נקבע את הערך של השדה Age להיות 0, אזי הפרשי הערכים של שדה זה בין חבילתו של התוקף לבין חבילת ה-"Fight-Back" לא אמורים לעלות על 15 דקות. באשר ל- Sequence Number, מכיוון שידוע כי ה-LSA Fight Back תמיד נשלח עם ערך Seqeunce Number גדול באחד מערך זה ב-LSA המזויף שגרם לו, אזי ערכו של שדה ה-Sequence Number ב-LSA Disguised צריך להיות גדול ב-1 מערכו של שדה ה-Sqeucence Number של חבילת ה-Trigger. קביעת ערכו של ה-checksum קצת יותר טריקי, על ערך ה-checksum של ה-LSA Disguised להיות זהה לערך ה-checksum של ה- fight back LSA, אך התוכן שלהם בהכרח שונה. בכדי להשוות את ערכי ה-checksum בשני ה-LSA, ניתן להוסיף ל-Disguised LSA עוד link דמה (בנוסף לכל הלינקים המזויפים שהתוקף מעוניין לפרסם בשמו של הקורבן). נקבע את ערכו של לינק הדמה להיות כך שערך ה-checksum של כל ה-Disguised LSA יהיה זהה ל-checksum של ה-fight back LSA. ניתן לחשב את הערך הנדרש של לינק הדמה בקלות, מכיוון שה-checksum הינו פונקציה לינארית של ערך ה-LSA.

שימו לב לכך שבתרשים 1, על מנת שהמתקפה תצליח, על התוקף לדעת את מפתח ה-MD5 של הקישור בין הנתב שעליו יש לו גישה לבין הקורבן. דרך נוספת לממש את המתקפה היא ע"י שליחת חבילת ה-Trigger וחבילת ה-Disguised LSA (שבה נעשה שימוש בשלב 2)על הרשת המקומית במקום לשלוח רק לקורבן. בשלב זה, שתי החבילות יוצפו בכלל הרשת ויגיעו לשאר שכניו של הנתב, וגם אל הנתב אותו התוקף מזייף, וכמובן - עם קבלת ה-Trriger ישלח הנתב חבילת LSA מעודכנת (כחלק ממנגנון ה-"Fight-Back" שלו) לכלל השכנים. אך אם השכנים כבר הספיקו לקבל את חבילת ה-Disguised LSA מהתוקף, הם יתייחסו לחבילת ה-"Fight-Back" של הקורבן כאל העתק נוסף של החבילה - יתעלמו ממנה ולא יעבירו אותה לשאר הרשת.

במקרה זה ה-Disguised LSA נמצאת במירוץ כנגד מנגנון ה-"Fight-Back" של התוקף. החבילה הראשונה שתגיע לנתבים ברשת - תותקן, והשניה תאופיין כהעתק ותזכה להתעלמות מצד אותם הנתבים. מפני שה-Disguised LSA נשלחת עוד לפני שמנגנון ה-"Fight-Back" מופעל, יש לה יתרון יחסית משמעותי על חבילת ה-"Fight-Back" שתשלח על-ידי הקורבן והיא תגיע ראשונה לרוב הנתבים ברשת.

להלן שרטוט הממחיש כיצד הרשת תראה לאחר הפעלת מתקפה זו, האיזור האדום הינו אזור בו נמצאים הנתבים אשר התקינו את חבילת ה-LSA שנשלחה על ידי התוקף, ובאיזור הכחול נמצאים התנבים אשר התקינו את חבילת ה-LSA של מנגנון ה-"Fight-Back" של הקורבן:



[תרשים 2 - תצורת הרשת לאחר ביצוע המתקפה]

כמו שניתן לראות, המתקפה הנ"ל, הינה כלי יעיל לעריכת טבלת ה-LSA על נתב שאליו אין לתוקף גישה. בעזרת מתקפה זו, התוקף יוכל להגיע למצב אשר בו רב / כלל הנתבים ברשת התקנו את חבילת ה-LSA הכוזבת שאותה יצר. על מנת להשיג מטרה זו, על התוקף לחזור על שלבי המתקפה כך שבכל פעם עליו לבחור קורבן אחר.

Remote False Adjacency

מתקפה זו מנצלת חולשה המתועדת ב-RFC 2328, בפרק 10.8, פרק זה מתאר את התהליך בו משתמשים הנתבים ברשת על מנת לשלוח את תיאור מסד הנתונים שלהן בעת שלב ה- Adjacency Setup. שלב זה מתרחש כאשר שני שכנים מגלים אחד את השני ברשת המקומית ומעוניינים לסנכרן ביניהם את ה-LSA DB של שניהם. כל נתב מספר לנתב השני את רשימת ה-LSA שנמצאים ב-LSA DB שלו. בתהליך זה הנתב בעל ה-ID הגדול יותר נבחר להיות ה-Master והשני נבחר להיות ה-Slave. הנתב הראשון ("Master Router") מסוגל להשלים את השלב הנ"ל ללא שום הצורך לראות את ההודעות שנשלחו על ידי שכניו ("Slave Router") ב-LAN. ע"י התחשבות בעובדה זו, תוקף יוכל לבחור קורבן ולהקים איתו Adjacency, כל עוד אותו קורבן מוגדר כ-"Slave Router" בכל שלב הקמת ה-Adjacency. מפני שכאשר נתב מעוניין להיות "שכן" של נתב אחר ברשת הוא חייב שתהיה לא כתובת IP ב-Subnet של אותו נתב, על התוקף להקים יישות פקטיבית ("Phantom Router") ברשת הקורבן, וממנה לייצר את ה-Adjacency עם הקורבן.

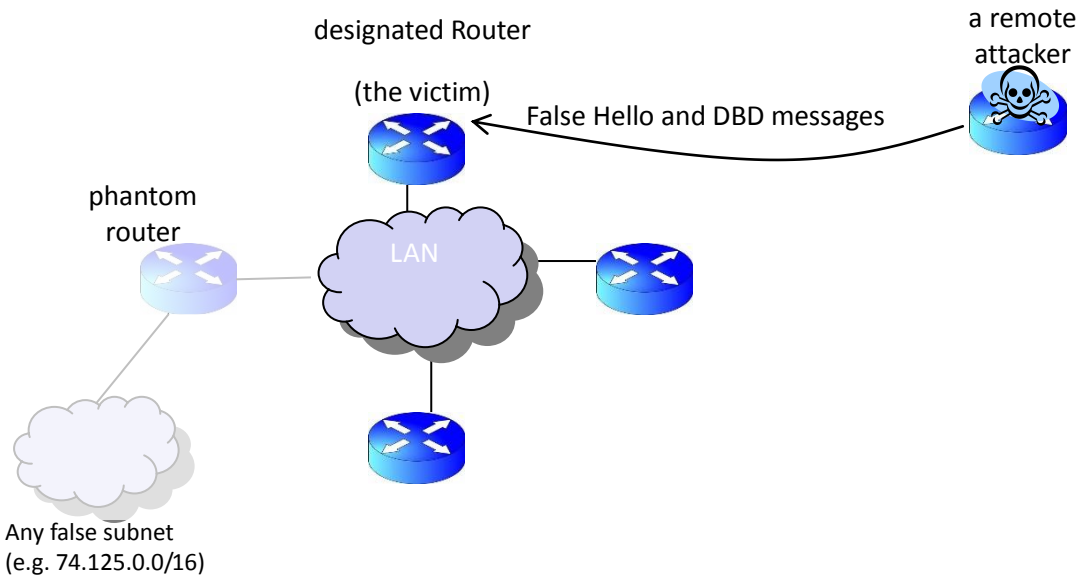
לאחר הרצת תהליך זה, התוקף מגיע למצב בו לקורבן קיימת Adjacency חוקי עם היישות הפקטיבית שהוא הקים, ובשלב זה, הקורבן יתחיל אף לפרסם קישור אליה ב-LSA שלו. פרסום ה-LSA אודות הלינק הפקטיבי הינה הנקודה המרכזית במתקפה זו, ואחת היתרונות שלה.

לאחר מכן, אם התוקף יפרסם LSA כוזב אודות הקישור בין היישות הפקטיבית ובין הקורבן - לקורבן עצמו, הקישוריות בין השניים תהפוך להיות קישוריות דו-כיוונית. ומכאן שכלל הנתבים ברשת, יקבלו את הקישוריות ויתחשו בה בעת חישוב טבלאות הניתוב שלהם.

המתקפה הנ"ל הינה המתקפה הראשונה בעולם אשר מאפשרת לתוקף להקים קישוריות דו-כיווניות בין נתב אמיתי לבין נתב פקטיבי באופן קבוע באופן כזה שהקישוריות הנ"ל תחשב בעת חישוב טבלאות הניתוב של שאר הנתבים ברשת. הגעה למצב זה, מאפשרת כעת לתוקף, לפרסם כל LSA בשמו של הנתב הפקטיבי ונתונים אלו יחושבו בעת חישוב טבלאות הניתוב של כלל הנתבים ברשת.

על מנת להשלים את המתקפה, על התוקף לדעת את פיסות המידע הבאות:

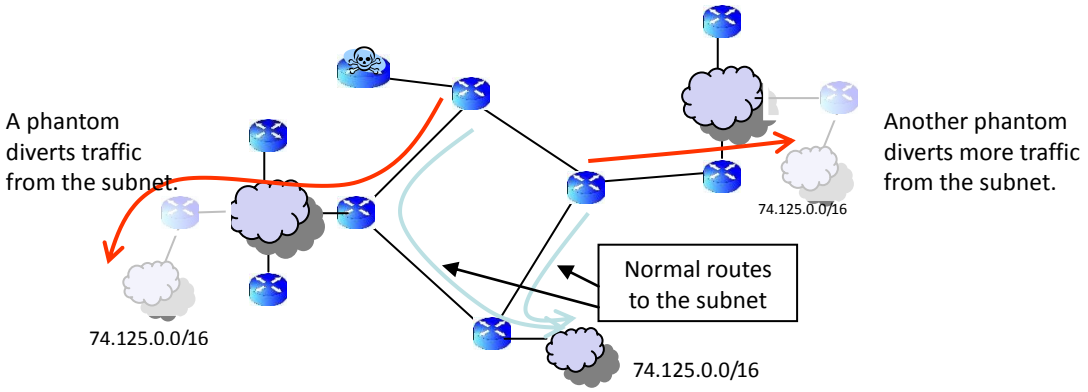
- מפתח ה-MD5 של הרשת המרוחקת (Remote LAN), ברב המקרים, המפתח הנ"ל הינו אחיד בין כלל הרשתות בתוך ב-AS.
- פרמטרי קונפיגורציות הרשת המרוחקת, כדוגמת HelloInterval, RouterDeadInterval, וכו'. וגם כאן, ברב המקרים, הפרמטרים הנ"ל הינם אחיים בין כלל הרשתות בתוך ה-AS.



[תרשים 3 - הדגמת מהלך המתקפה]

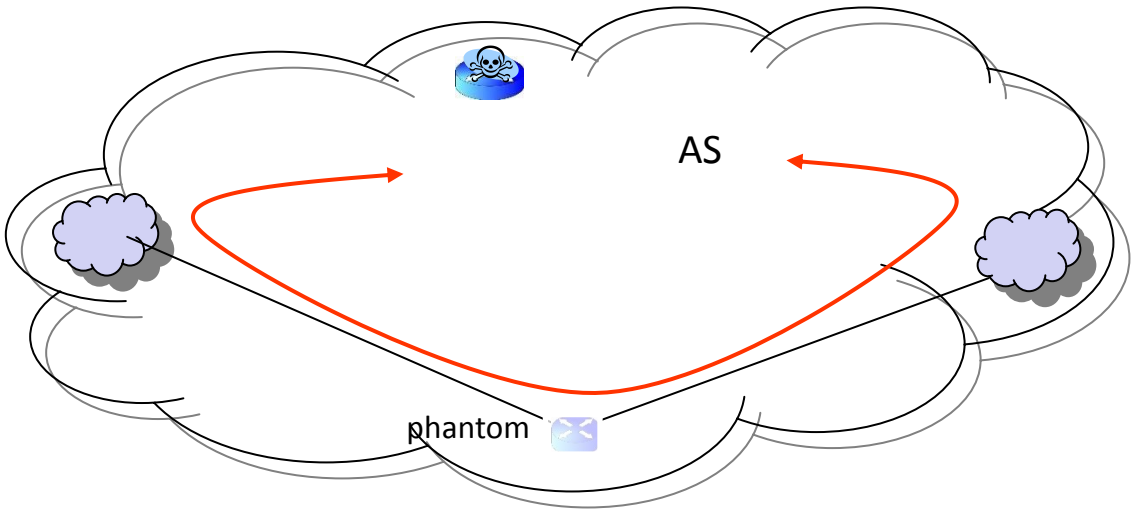
שימוש אפשרי במתקפה זו הוא לטובת יצירת Black-Hole לתעבורת המידע ברשת ספציפית, ע"י פרסום יישות פקטיבית שתוביל לאותו ה-Black-Hole. בהתחשב בעובדה שתוקף יכול ליצור נתב פקטיבי בכל מקום ברשת ניתן להבין כי תוקף יוכל ליצור Black-Hole לכל תעבודה ברשת ומכל תת-רשת ברשת.

להמחשת הרעיון, ניתן להסתכל בתרשים הבא:



[תרשים 4- יצירת Black-Holes ברשת וניתוב כלל המידע אליהם]

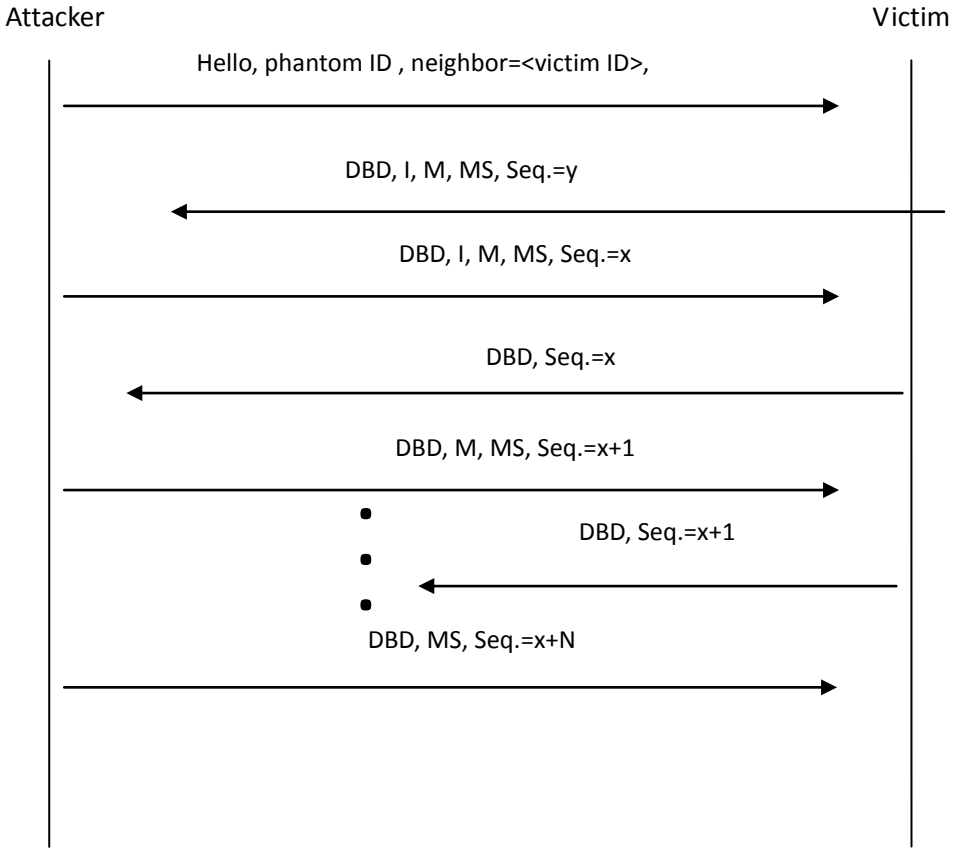
שימוש נוסף אפשרי במתקפה זו היא יצירת נתב פקטיבי ברשת במיקום "אסטרטגי" המאפשר לו להוסיף קיצור-דרך לכמות תעבורה גדולה ברשת, לדוגמא, ע"י הוספת נתב פקטיבי המגשר בין שתי רשתות רחוקות ב-AS, כפי שניתן לראות בתרשים הבא:



[תרשים 5 - קישור שתי רשתות מרוחקות ע"י נתב פקטיבי]

ניתן לבצע זאת על-ידי איתור שני נתבים הנמצאים בשתי רשתות שונות, ועל התוקף לחבר בינהם דרך נתב פקטיבי שהוא יוצר.

באיור הבא מתואר מהלך המתקפה (כל חבילות המידע אשר נשלחות על ידי התוקף צריכות להראות כאילו הן נשלחו מהכתובת של הנתב הפקטיבי ועליה להיות ב-Subnet של הקורבן):



[תרשים 6 - הדגמת מהלך המתקפה]

המתקפה מתחילה בכך שהתוקף שולח חבילת "Hello" לקורבן. מאחר שהחבילה כוללת ברשימת השכנים שלה אתה-ID של הקורבן, הקורבן נכנס למצב של 2-way (המצב אליו הוא נכנס כאשר הוא מבין כי קיימת קישוריות דו-כיווניות [bidirectional] עם שכנו), נניח כי הקורבן הינו designated router של הרשת המקומית שלו, לכן הקורבן ישאף מיד ליצור Adjacency עם הנתב הפקטיבי. על כן הקורבן נכנס למצב ExStart (מצב זה מציין כי החל תהליך יצירת adjacency). לאחר מכן, הקורבן שולח חבילת DBD (חבילה הכוללת את ה-DB description של הנתב) עם Sequence התחלתי אקראי (y). **חבילת המידע הנ"ל, כמו כלל החבילות אשר נשלחות אל עבר הנתב הפקטיבי ברשת הקורבן והתוקף לא רואה אותן**

כעת, התוקף שולח את ה-DBD הראשון, ה-DBD הראשון של התוקף (של הנתב הפקטיבי, למעשה), נשלח עם הביטים הבאים ששהם שווים ל-1:

- **I** - Initialize (מציין כי זו ההודעה הראשונה שנשלחת).
- **M** - More (מציין כי זו איננה ההודעה האחרונה שתישלח בתהליך).
- **MS** - Master (מציין כי הנתב הפיקטיבי מחשיב את עצמו כמסטר) .

ואת ה-Sequence Number הוא קובע לערך אקראי (x). בנוסף, החבילה בנויה כך שה-ID של הנתב הפקטיבי גדול מה-ID של הקורבן, וכך הקורבן נקבע להיות ה-Slave והנתב הפקטיבי נקבע להיות ה-Master. במצב זה הקורבן "מאמץ" את ה-Sequence Number של הנתב הפקטיבי (x), ושולח לו את החבילת ה-DBD הבאה רק לאחר שהוא מקבל את חבילת ה-DBD של הנתב הפקטיבי, כך למעשה שהתוקף לא חייב לראות את חבילות ה-DBD של הקורבן ולדעת איזה Seqence Number הוא בחר.

בזמן זה, התוקף ממשיך לשלוח את ה-DBD כך שבכל פעם הוא מעלה את הערך של ה- Sequence Number. בכדי לפשט את ההתקפה הודעות ה-DBD של הנתב הפקטיבי לא כוללות רשומות LSA. עם זאת, התוקף ממשיך לשלוח את חבילות ה-DBD ובכך מאפשר לקורבן להמשיך לשלוח את הודעות ה-DBD שלו עד אשר הוא ישלח את כל ה-LSA-ים הנמצאים ב-DB שלו. התוקף לא יכול לדעת מראש כמה הודעות DBD הקורבן יצטרך בכדי לשלוח את כל רשומות LSA קיימות ב-LSDB שלו, אך אין זאת בעיה, מפני ש-10 חבילות DBD זה ברוב המקרים מספיק.

לאחר שהתוקף (השולט על הנתב הפקטיבי) סיים לשלוח את כלל חבילות ה-DBD, אנו מניחים כי גם לקורבן נגמרו הרשומות אצלו במסד, הקורבן כעת מדלג על ה-Loading State (שבו כל צד מבקש מהשני את הפרטים המלאים של ה-LSA-ים שאין לו אך יש לשני), מכיוון שהנתב הפיקטיבי דיווח כי אין לו כלל LSA-ים ונכנס ל-Full State. **בשלב זה, לקורבן יש שכנות מלאה עם הנתב הפקטיבי**, ולאחר מכן - הוא ידווח עליה לכלל הרשת באמצעות שליחת חבילת LSA! המשימה בוצעה בהצלחה! ☺

להצלחת מתקפה זו דרושים מספר תנאים:

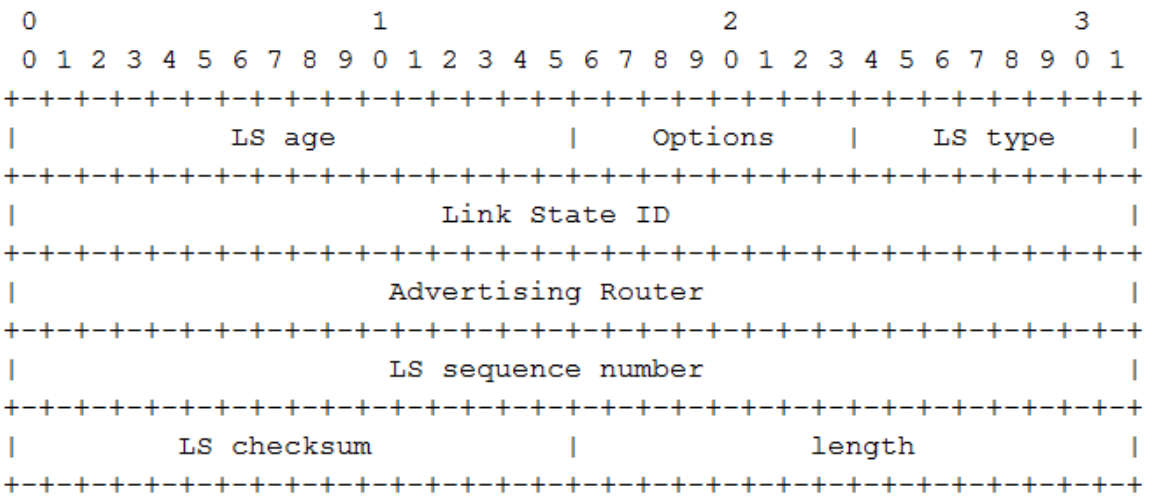
- לאחר יצירת השכנות, על התוקף לשמור עליה על-ידי שליחת הודעות Hello כל RouterDeadInterval שניות (כברירת מחדל, ערך זה שווה ל-40).

- לאחר יצירת השכנות, הקורבן שולח לנתב הפקטיבי חבילות LSA ומצפה לקבל ממנו Ack-ים בחזרה, על פי התקן, במידה והנתב הפקטיבי לא יחזיר Ack, על הקורבן להמשיך לשלוח לו חבילות LSA ללא סוף, אך בפועל, על נתבי Cisco, ראינו כי לאחר 125 שניות, הנתב מרים ידיים ומסיר את השכנות.

הסעיף האחרון אומר שבמידה ומדובר בנתבי Cisco, על התוקף לבצע את המתקפה כל פעם מחדש, לאחר 125 שניות. אך במידה וגם התוקף וגם הקורבן נמצאים באותה הרשת (area), התוקף, באופן עקרוני, מודע לכל חבילות ה-LSA שהקורבן שולח ולכן הוא גם יכול לזייף Ack-ים כתגובה, פעם ב-120 שניות. עם זאת, במהלך המחקר שלנו לא בדקנו אופציה זו.

Mismatched Fields Attacks

הכותרת של כל חבילת LSA בנוי באופן הבא:



- LS age - הזמן בשניות מאז יצירת החבילה.
- Options - תמיכה ביכולות נוספות.
- LS type - סוג חבילת ה-LSA (לדוגמא: Router, Network, Summary וכו'). בפרק הקרוב, נתמקד רק בחבילות מסוג Router.
- Link State ID - מזהה את החלק של טופולוגיית ה-AS המתואר בחבילת ה-LSA.
- Advertising Router - מזהה ה-Routerשל הנתב שיצר את החבילה.
- LS sequence number - המספר הסידורי של החבילה.
- LS checksum - הערך של Fletcher checksum עבור כלל תוכן החבילה.
- Length - הגודל, בבטים, של חבילת ה-LSA.

כעת, בואו נסתכל במבט קרוב על השדות מעל, ונבחן את הערכים שלהם כאשר מדובר ב-Router LSA:

- Link State ID - השדה הנ"ל מזהה את הנתב אשר הקישורים אליו רשומים בחבילת ה-LSA. הערך הנ"ל הינו מזהה של אותו נתב, כלומר ה-Router ID[1].
- Advertising Router - השדה הנ"ל מזהה את הנתב שפרסם במקור את חבילת ה-LSA הנ"ל. התקן של OSPF מכתיב כי רק הנתב עצמו יכול לייצר ולפרסם את ה-LSA של עצמו, על כן גם ערך שדה זה חייב להיות שווה ל-Router ID.

על כן, שני השדות "Link State ID" ו-"Advertising Router" חייבים לכלול את אותו הערך. עם זאת, התקן של OSPF לא מחייב לוודא זאת בעת קבלת חבילת ה-LSA. ומכאן שניתן לשלוח חבילה ובה יהיו ערכים שונים באותם השדות. בהמשך השורות נראה כיצד עובדה זו יכולה להיות מנוצלת על-ידי התוקף.
ע"פ פרק 13.4 בתקן של OSPF, נתב יפעיל את מנגנון ה-"Fight-Back" רק כאשר הוא מקבל חבילת LSA שאינה תקינה וגם ששדה ה-"Advertising Router" שווה ל-"Router ID" של הנתב עצמו. ציטוט מהמקור:

"The Advertising Router is equal to the router's own Router ID"

ומכאן, שנתב לא יגיב באמצעות הפעלת מנגנון ה-"Fight-Back" לגבי חבילת LSA זדונית, גם היא היא טוענת שהיא מפרסמת על אותו נתב, כל עוד הערך של שדה ה-"Advertising Router" שונה מה-"Router ID" שלו עצמו. המתקפה הולכת כך: נניח כי תוקף מעוניין לפרסם חבילת LSA בשם נתב אחר, נניח נתב Rv, עליו לפרסם LSA הכולל בכותרתו את הערכים הבאים:

- Link State ID = R_v ID.
- Advertising Router = any value other than the ID of router R_v.

במידה ותוקף ישלח חבילה כזאת, הוא מוגן על ידי התקן של OSPF. התקן מבטיח לו ששום מנגנון Fight-Back ושום חבילת LSA לא תוחזר בעקבות שליחת חבילה בסיגנון זה. מעבר לכך, כלל הנתבים ב-AS יזהו כי מדובר חבילת LSA תקינה ויתקינו אותה ב-LSA DB כאילו R_v שלח אותה. אך עם זאת, אמורה להיות בעיה קטנה עם התקן של OSPF בנוגע למתקפה זו. על פי פרק 12.1, כל חבילת LSA מזוהה באופן ייחודי על ידי צירוף שלושת הערכים של השדות:

- LS Type
- Advertising Router
- Link State ID

LSA האמיתי של הקורבן שונים (כי הם שונים בשדה ה-Advertsing Router). מכאן שהמזהה של ה-LSA המזויף והמזהה של ה-לכן, ה-LSA המזויף **לא אמור** להחליף את חבילת ה-LSA המקורית ששולח הקורבן.

[1] ה- Router ID הינו אחת מכתובות ה- IP של הנתב הנבחרת ע"י הנתב. כתובת זו משמשת כמזהה על הנתב ב-AS.

אבל מה, לפי פרק 16.1 של התקן, בעת חישוב טבלאות ה-LSA כאשר הנתב מאחזר LSA-ים השמורים ב-DB שלנו הנתב מבצע זאת כך:

"This is a lookup … based on the Vertex ID"

כאן, כאשר כתוב "Vertex ID", התקן מתייחס ל-"Link State ID". ולכן, כאשר נתב מחשב את טבלאות ה-LSA שלו, הוא שולף LSA-ים מתוך ה-DB שלנו תוך זיהויים על ידי שדה ה-"Link State ID" בלבד!

במקרה הנ"ל, התקן של OSPF אינו אחיד, מצד אחד, חבילת LSA מזוהה באופן ייחודי על פי שילוב של שלושה שדות (סעיף 12.1 בתקן), אך מצד שני, כאשר טבלאות הניתוב מחושבות, חבילת ה-LSA מזוהה על פי מזהה אחד (Link State ID) בלבד (סעיף 16.1 בתקן).

דו-הלשוניות במקרה הנ"ל מעלה את השאלה הבאה: איזו חבילת LSA תילקח בחשבון בעת בניית טבלאות הניתוב? חבילת ה-LSA המקורית או חבילת ה-LSA המפוברקת? שימו לב, כי לשני ה-LSA-ים הללו יש את אותו ערך בשדה ה-Link State ID – ה-Router ID של הקורבן. עם זאת, התקן של OSPF לא עונה על שאלה זו, ומכאן שהתשובה נמצאת במימוש. כל חברה והאלגוריתם שלה. במידה והחברה תממש את פרוטקול ה-OSPF כך ה- LASהאמיתי יישלף במהלך חישוב טבלאות הניתוב - הנתב שלה יהיה מוגן מפני מתקפה זו, אך אם היא המימוש ישלוף את ה-LSA המזוייף, המוצר שלה יהיה פגיע ועוד איך.

Evaluation of Cisco

כעת, נעבור לדבר על המערכות אשר מממשות את רב ה-OSPF בעולם: Cisco IOS. על פי [Infonetics12], סיסקו מחזיקה בכ-75% נתח שוק הנתבים בעולם. על מנת לבחון את המימוש של סיסקו עבור ה-OSPF, השתמשנו באמולטור GNS3 ובה השתמשנו ב-Image-ים של IOS. גרסאת ה-IOS שהצלחנו להשיג, הייתה 15(1)M[2]. והסקריפט שבו השתמשנו נכתב ב-Scapy, וקישור אליו מופיע בסוף המאמר. הערכה שלנו למימוש ה-OSPF של סיסקו היא **שמכשיריה פגיעים למתקפה זו**, ולהלן הממצאים:

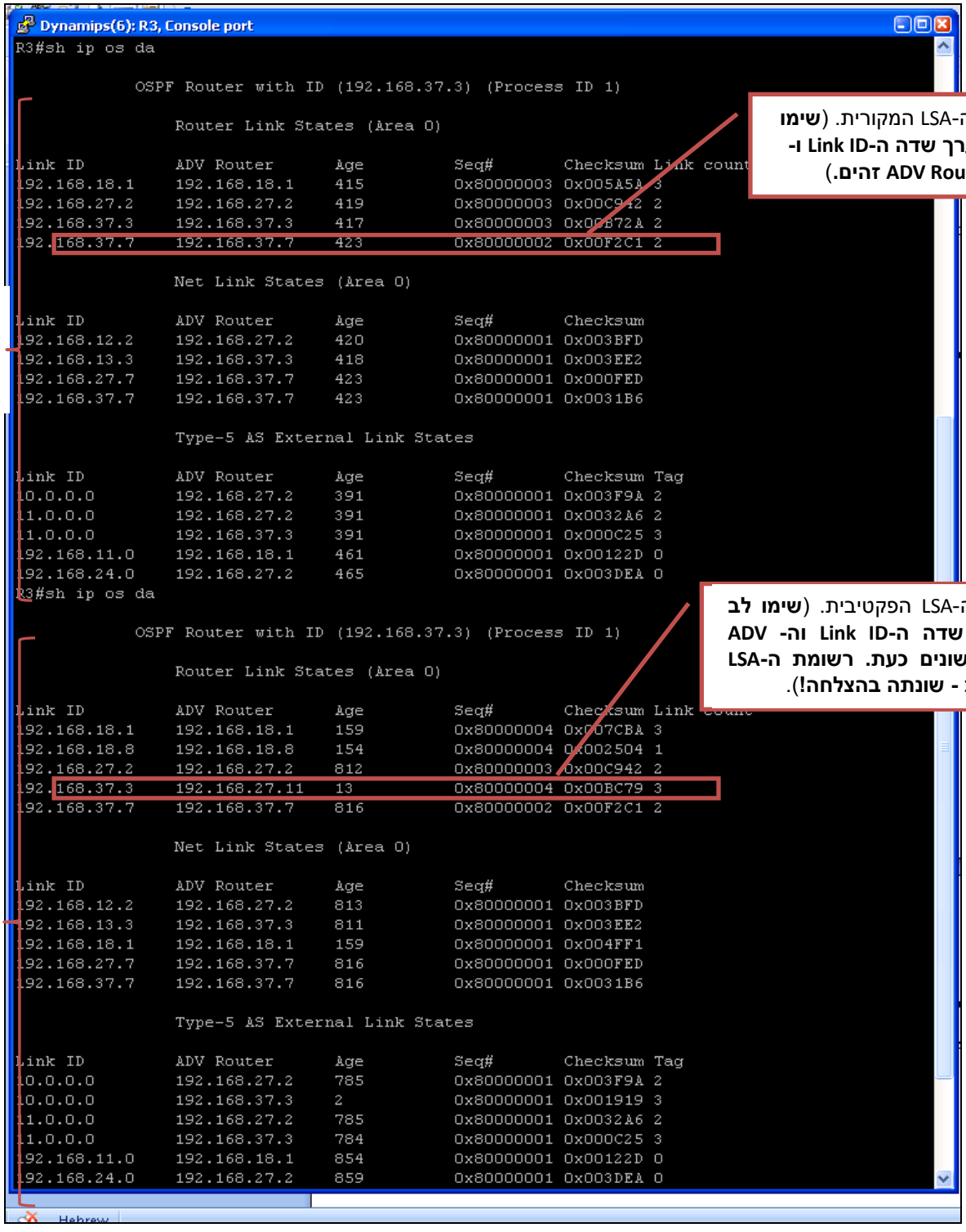
- **רשומת ה-LSA הפקטיבי מחליפה את רשומת ה-LSA המקורית:**

במידה וחבילת ה-LSA הפקטיבית מתפרסמת עם Sequcene number הגבוה מה-Sequence number של חבילת ה-LSA מקורית, חבילת ה-LSA הפקטיבית לא רק שתותקן ב-LSA DB של הנתב, אלא גם תחליף את רשומת ה-LSA המקורית. תרחיש זה קרה בכלל הנתבים ב-AS כולל ב-LSA DB של הקורבן עצמו, וכמובן, כלל הנתבים יתייחסו לרשומה זו בעת חישוב טבלאות הניתוב. בעמוד הבא, ניתן לראות תצלום מסך של ה-LSA DB של הקורבן.

[2] נכון לכתובת שורות אלו, מערכת ה-IOS האחרונה שפורסמה הינה 15(4)M2.

LSA DB **לפני** המתקפה

LSA DB **אחרי** המתקפה



[תרשים 7 - ה-LSA DB לפני ואחרי המתקפה]

- **טבלאות הניתוב של כלל הנתבים ברשת (מלבד הקורבן) הורעלו:**

לאחר הפעלת המתקפה, טבלאות הניתוב של כלל הנתבים ברשת (מלבד טבלת הניתוב של הקורבן) מסתמכות על רשומת ה-LSA המזוייפת.

- **טבלת הניתוב של הקורבן נמחקת:**

לאחר הפעלת המתקפה, ה-LSA DB של הקורבן אינו מכיל רשומת LSA הכוללת שדה Advertising Router השווה ל-ID של הקורבן. (שימו לב כי רשומת ה-LSA המקורית הוחלפה ברשומת LSA פקטיבית אשר לה ערך Advertising Router שונה!). במימוש של סיסקו ל-OSPF, מקרה זה מוביל מצב אשר בו תהליך חישוב טבלאות הניתוב לא מוצא אף נתב או רשת, כלל יעדי / מקורות הניתוב שהגיעו לנתב באמצעות OSPF נמחקו, מה שמשאיר את הנתב עם טבלת ניתוב ריקה. ומכאן, שבמידה ולא קינפגו לקורבן ניתוב דיפולטי סטטי ("static default route"), הוא יעדוף כל תקשורת IP שאינה מיועדת אליו או לרשת המקומית אליה הוא מקושר.

מחיקת טבלת הניתוב של הנתב הינה קבועה. וכל עוד התוקף לא יחליט לבצע "Undo" למתקפה (הסבר בפסקה הבאה), הנתב לא יוכל להשתקם באופן עצמאי. ועל מנהל הרשת לאתחל את תהליך ה-OSPF באופן יזום.

- **ביצוע Undo למתקפה:**

אם התוקף מעוניין, הוא יוכל בצורה קלה לבצע Undo למתקפה ע"י שליחת חבילת LSA פקטיבית נוספת, רק שהפעם ערכי השדות Link State ID ו-Advertising Router יהיו זהים ושווים לערך Router ID של הנתב. ערכו של שדה ה-Sequence Number בחבילת ה-LSA הפקטיבית הנ"ל להיות מעל ערכו של שדה ה-Sequence Number שנשלח ב-LSA המזוייף הקודם.

פעולה זו תפעיל את מנגנון ה-"Fight-Back" של הקורבן ותגרום לו לייצר חבילת LSA מקורית אשר תחליף את רשומת ה-LSA הפקטיבית בכלל הנתבים.

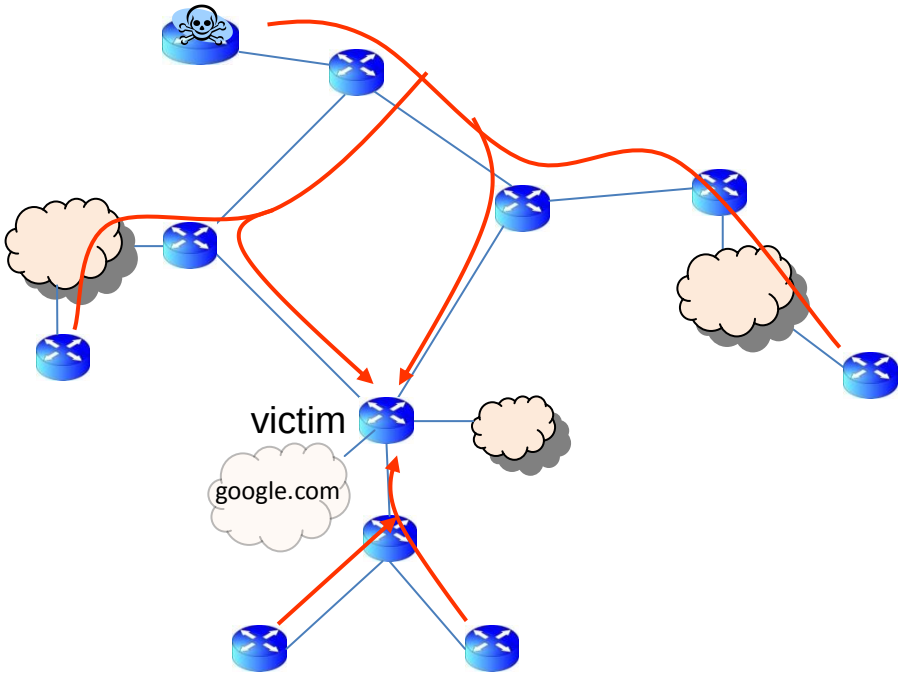
יישומים אפשריים למתקפה

בשורות הבאות נציג יישומים אפשריים למתקפה, אנו מציינים כי התוקף יכול להיות בכל מקום ב-AS על מנת שמתקפה זו תצליח.

- **Black Hole**:

ביישום הבא, התוקף מסוגל לנתק את כלל הנתבים ברשת ה-AS מרשת יעד מסויימת הממוקמת מחוץ לרשת ה-AS. התוקף מסוגל לממש זאת על ידי הפיכת אחד הנתבים ברשת ל-Black Hole עבור אותו היעד.

על מנת לבצע זאת, על התוקף לשלוח חבילת LSA פקטיבית אשר מודיעה כי נתב מסויים מחובר ישירות לרשת (נקרא לה: net-X) הנמצאת באמת מחוץ לתחומי ה-AS, לדוגמא, ה-IP של גוגל. מאחר שנתיב אשר נמצא בתוך ה-AS מקבל עדיפות על פני כל נתיב אשר יוצא מחוץ ל-AS, הנתיב המזוייף הנ"ל ייבחר ע"י כל הנתבים ב-AS. לכן כל חבילה המיועדת ל-net-X תנותב לנתב הקורבן. אך מאחר שטבלת הניתוב של הקורבן הנ"ל נמחקה - הוא לא יבצע שום העברה של מידע אל אותה הרשת, ומכאן שלא יהיה ניתן לגשת ל-net-X מאותו ה-AS, להלן תרשים:



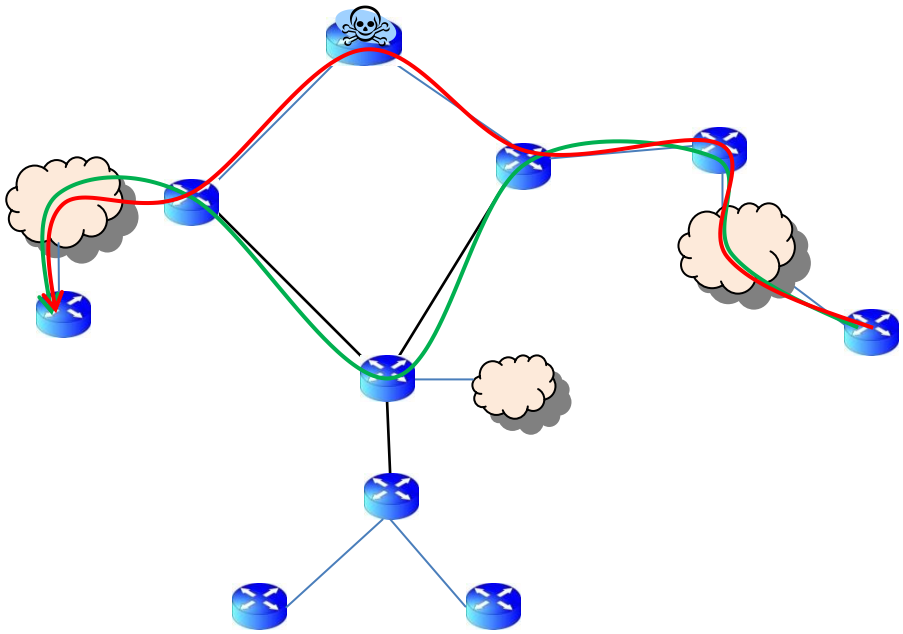
[תרשים 8 - ניתוק יעד ספציפי אשר נמצא מחוץ ל-AS]

- **שינוי ניתוב:**

במימוש זה, תוקף גורם לכך כי תעבורה ספציפית על פי בחירתו תעבור בנתיב שהוא יבחר בתוך ה-AS ולא תעבור בנתיב המקורי, על ידי מימוש זה תוקף יוכל לבצע מתקפת Man In The Middle לטובת האזנה לתקשורת ואף שינויה.

על מנת לממש מתקפה זו, על התוקף לפרסם חבילת LSA אשר מדווחת כי לקורבן (הנתב דרכו התעבורה אמורה לעבור במקור) לא קיימת קישוריות עם נתבים אחרים או עם רשתות נוספות, צעד זה יוריד את הקורבן לחלוטין מטבלאות הניתוב ברשת וכלל הנתבים ברשת יבצעו חישוב מחדש של טבלאות הניתוב שלהם (שיבו לב כי הנ"ל יקרה למרות שהנתבים השכנים של הקורבן ימשיכו לפרסם את הקישוריות אליו[3]).

התוצאה הסופית של מהלך זה היא כי כלל הרשת תבצע חישוב מחדש של ערוצי ניתוב על מנת להגיע לאותו היעד. בתרשים הבא, ניתן לראות את הקווים הירוקים שמהווים את ערוץ הניתוב לפני ניתוק הקורבן ואת הקווים האדומים אשר מהווים את ערוץ הניתוב לאחר המתקפה, במקרה זה, לתוקף כעת יש גישה יותר מידע ברשת מאשר שהיה לו לפני המתקפה:



[תרשים 9 - שינוי מסלול ניתוב אל עבר יעד ספציפי, הנתיב הירוק הינו מסלול הניתוב לפני המתקפה והנתיב האדום הינו מסלול הניתוב לאחריה.]

[3] זאת מפני שלפי התקן של OSPF "קישוריות" נתקלחת בחשבון רק ובמידה ומתפרסמים קישורים אליה משני צידי החיבור.

מסקנות

במסמך זה סקרנו את המתקפות הקיימות כיום בעולם ה-OSPF, בנוסף, הצגנו לפרטים שלוש מתקפות חדשות על הפרוטוקול המנצלות חולשות חדשות בסטנדרט של הפרוטוקול.

עד כה, הדעה הרווחת הייתה כי גם אם לתוקף קיימת גישה פנימית לרשת, וגם אם באמצעותו היכולת לשלוט על רכיב רשת בודד הוא עדיין אינו יכול לגרום לשינויים נרחבים ברשת ולשמרם לאורך זמן. עבודה זו, מציגה כי ההפך הוא הנכון, וכי גם בעזרת הנחות אלו בלבד, לתוקף יש אפשרות לגרום לנזק זה, **ובעזרת שליטה ברכיב בודד אחד, התוקף יכול לגרום לשינויים קובעים בכלל הנתבים ברשת**.

ניתן להוריד את הסקריפטים (Python) בהם השתמשו לטובת מימוש המתקפות במאמר זה, מהקישור הבא:

http://www.digitalwhisper.co.il/files/Zines/0x30/OSPF.rar

על המחבר

גבי הינו עמית במעבדת מחקר ברפאל וכן מרצה וחוקר נספח בפקולטה למדעי המחשב בטכניון.

קישורים לקריאה נוספת

- **[RFC2328]** J. Moy, "OSPF Version 2", IETF RFC 2328, April 1998.
- **[Wang97]** F. Wang et. al., "Secure routing protocols: theory and practice", Technical Report, North Carolina State University, May 1997.
- **[Wu99]** S. Wu et. al., "JiNao: Design and implementation of a scalable intrusion detection system" for the OSPF routing protocol", Journal of Computer Network and ISDN systems, 1999
- **[Jones06]** E. Jones et. Al… "OSPF Security Vulnerability analysis", IETF draft-ietf-rpsec-ospf-vuln-02, June 2006.
- **[BH11]** Gabi Nakibly, Alex Kirshon, Dima Gonikman "Owning the routing table new OSPF attacks", Black Hat USA 2011.
- **[BH12]** Gabi Nakibly, Eitan Menahem, Ariel Waizel, and Yuval Elovici "Owning the routing table - part II", Black Hat USA 2013.

שימוש בטוח בביטקוין מהסוף להתחלה

מאת יוני יחזקאל

הקדמה

ביטקוין, המטבע הדיגיטלי המבוזר שצבר כותרות בשנה האחרונה ובייחוד בחודש האחרון כאשר ערכו עבר את ה-$1000 ל"יחידה", הולך וצובר משתמשים. אחד היתרונות הגדולים של הביטקוין על פני מטבעות אחרים הוא הניהול המבוזר שלו ועל כן היכולת של כל אחד לנהל בעצמו את ארנק (חשבון) הביטקוין שלו ללא צורך בבנק או בכל צד ג' נוסף. קיימים כיום שירותים המציעים לנהל עבור המשתמשים את ארנק הביטקוין שלהם, אך מרביתם אינם מספקים אחריות במקרה של אובדן או גניבה, אפילו אם מדובר בתקלה או בפרצה אצל נותני השירות. לאחרונה נגנבו מארנקי אונליין כאלה ואחרים אלפי בטקויינים בשווי כמה מיליוני דולרים[4]. יכולת הניהול העצמית של הביטקוין, חוסר האחריות ובהרבה פעמים גם חוסר הכשירות של ספקי השירות הקיימים היא הסיבה שרבים ממחזיקי המטבע מעדיפים לנהל את הארנקים שלהם בעצמם ולא לסמוך על צד שלישי שינהל אותו.

בתחילת דרכו של המטבע, הרוב המוחלט של המשתמשים בו היו חובבי קריפטוגרפיה ומחשבים, אך היום מחזיקים במטבע מספר רב של אנשים ללא רקע טכני ויש צורך באמצעי אבטחה שיהיו חזקים מספיק כדי לשמור את הארנקים בטוחים אך מצד שני פשוטים כדי שגם המשתמש הממוצע יוכל להשתמש בהם.

במאמר זה נסקור את הדרכים בהן ניתן לאבטח ארנק ביטקוין מפני אובדן וגניבה. בסיומו נתייחס לעוד שני נושאים שחשובים גם הם בשימוש יומיומי ונרחב של ביטקוין - הגנה מפני רמאויות, ושמירה על פרטיות. במאמר נניח שרוב הקוראים מתמצאים באבטחת מידע ברמה זו או אחרת ועל כן לא נכנס לפרטים בנושאים בסיסיים כמו ניהול סיסמאות, זיהוי בשני שלבים, הצפנה, האשינג, ונושאים אחרים שכמובן חשובים מאוד כדי ליצור ארנק מאובטח אך אינם קשורים ספציפית לביטקוין. כמו כן, לא נדבר בכלל על פרוטוקול הביטקוין עצמו ובעיות אבטחה שעלולות להתקיים בו ונניח כי מערכת הביטקוין עצמה פועלת בצורה תקינה ובטוחה.

[4] http://www.wired.com/wiredenterprise/2013/11/inputs

כמה מילים על ארנק ביטקוין

ארנק ביטקוין מורכב מאוסף של כתובות ביטקוין וזוג מפתחות עבור כל כתובת מפתח פרטי ומפתח ציבורי. בקצרה נזכיר שכתובת הביטקוין היא בעצם ההאש של המפתח הציבורי. בעת ביצוע העסקה משמש המפתח הפרטי לחתום עסקאות שמעבירות כספים מהכתובת המקושרת למפתח הפרטי לכל כתובת אחרת. בשום שלב המפתח הפרטי אינו נשלח ברשת והמידע היחיד שיוצא בפועל מהמחשב של המשתמש ונשלח בין הצמתים המפעילים את רשת הביטקוין הוא תיעוד העסקה החתומה במפתח.

[כתובת ביטקוין שנוצרה בעזרת האתר bitaddress.org]

במידה והעסקה חתומה בצורה תקינה, ונשלחים הביטקוינים הנמצאים ברשותו של השולח, היא בסופו של דבר תרשם בבלוק ותכנס לעד לבלוקצ'יין[5] (מאגר המידע המשותף בין כל הצמתים ברשת הביטקוין). כאשר אנו מדברים על אבטחה של "חשבון" ביטקוין אנחנו בעצם מדברים על הדרך בה נוכל לשמור את המפתח הפרטי זמין לשימוש שלנו אך מוגן מגניבה או אובדן.

אבטחת ארנק

ארנק חומרה

[ארנק חומרה של חברת trezor]

ארנקי חומרה הם המילה האחרונה באבטחה ושימוש בביטקוין. ארנק חומרה שומר את המפתח הפרטי אצלו בדרך שאינה ניתנת לשליפה. הארנק מתחבר ב-USB או bluetooth למחשב או לפאלפון. כאשר המשתמש מעוניין לבצע עסקה פרטי העסקה נשלחים לארנק החומרה. הארנק חותם את העסקה ושולח חזרה את העסקה החתומה. בגלל שאין דרך להוציא את המפתח הפרטי מהארנק אך עדיין ניתן להשתמש בו בקלות באופן שוטף הוא נחשב לאחת הדרכים הנוחות והבטוחות להשתמש בביטקוין.

[5] https://en.bitcoin.it/wiki/Block_chain

חשוב מאוד שלארנק החומרה יהיה מסך בו הוא יכול להציג למשתמש את פרטי העסקה שהוא מאשר כך שלא יווצר מצב בו הפלאפון או המחשב מציג למשתמש עסקה אחת אבל בפועל המשתמש מאשר עסקה אחרת. כמו כן המסך יכול לשמש להצגת המפתח הפרטי בצורה בטוחה לשם גיבוי שלו על ידי המשתמש. ארנקי חומרה מתוחכמים מעט יותר יכולים גם להכיל מקלדת קטנה ולדרוש קוד כדי לבצע חתימה על עסקה דבר שמגן על השימוש בהם במקרה של גניבה פיזית ואפילו להכיל בתוכם רכיבי תקשורת GSM ו-Wifi המאפשר להם לבצע טראנסקציות עצמאית ללא צורך בחיבור למכשיר מתווך.

חסרונו העיקרי של ארנק חומרה הוא שעדיין מדובר ברכיב אלקטרוני שעלול להיכשל בשלב כזה או אחר ועל כן יש חשיבות מירבית בשמירה על גיבוי במקום בטוח. הדרך המומלצת לבצע גיבוי לארנק החומרה היא באמצעות ארנק נייר עליו נדבר בהמשך. לצערנו, מרבית ארנקי החומרה נמצאים עדיין בשלב האבטיפוס ואינם זמינים לרכישה ושימוש (אם כי גרסאות ראשונות של המכשירים הללו אמורים להיות זמינים בחודשים הקרובים). כך שנכון לעכשיו משתמשי הביטקוין צריכים לחשוב על פתרונות אחרים לשמירה על הארנקים שלהם.

ארנק נייר

שימוש בארנק נייר היא הדרך הקלה והבטוחה לשמור כתובת ביטקוין, אם כי היא עלולה להיות מעט מסורבלת. במקום לשמור את המפתחות בצורה דיגיטלית על המחשב, המפתח מודפס על דף נייר (ארנק זה מכונה גם הרבה פעמים אחסון אופליין או אחסון קר). לאחר תהליך יצירת המפתחות והכתובת המשתמש יכול להעביר את הכתובת לכל מי שהוא מעוניין ולהמשיך לקבל תשלומים. שירותים מסויימים כמו blockchain.info והגרסה הבאה של bitcoinQT מאפשרים למשתמשים לצפות ולקבל עידכונים על המאזן של כתובת מסויימת ללא צורך במפתח הפרטי שלה.

[כתובת נייר שנוצרה בעזרת האתר bitaddress.org]

מעבר ליתרון האבטחה בשמירה של מפתחות אופליין, ניירות הן צורת אחסון אמינה הרבה יותר מרכיבים אלקטרונים. כל עוד לא יהיה חשוף לנזקי אש או מים, המידע על דף נייר יכול להישמר בקלות לעשרות שנים הרבה אחרי שכל מצע אלקטרוני יסיים את חייו.

לעומת זאת, קיימים לארנקי נייר מספר גם חסרונות:

- **יצירת הכתובות חייבת להתבצע בסביבה סטרילית ובטוחה לחלוטין:**

יצירת סביבה סטרילית לחלוטין היא משימה לא קלה ויש שיגידו בלתי אפשרית. במחשב הממוצע מותקנות תוכנות זדוניות כאלה ואחרות. ובעוד שחוקר אבטחה אולי יכול לסמוך על סביבת העבודה שלו יותר מהמשתמש הממוצע, ניתן להניח שדווקא מי שמודע לסכנות האפשריות לא ירצה להשאיר יותר מידי פתח לסיכונים.

ההסכמה הגורפת היא שבשביל ליצור ארנק נייר מומלץ להשתמש במחשב ייעודי למטרה זו, מחשב שלא מחובר לרשת וגם לא יהיה מחובר אליה בעתיד. ישנן אף מדפסות ייעודיות כמו ה-Piper אשר יכולות להדפיס כתובות ביטקוין עצמאית ללא צורך בחיבור למחשב או לרשת. למרות המחירים האטרקטיבם של מחשבים בימינו עדיין למעטים ישנו מחשב ייעודי או מדפסת הזמינים אך ורק בשביל יצור כתובות ביטקוין. ולכן למי שאין ברשותו מחשב ייעודי ההמלצה היא להשתמש ב-liveCD של הפצת לינוקס כזו או אחרת כדי לייצר את הכתובות. בעת הייצור על המחשב להיות מנותק לחלוטין מהרשת ויש לכבות אותו מיידית לאחר הדפסת הכתובות. כמובן שגם כאן עלולים להיות סיכונים.

מדפסות רבות היום מחוברות בעצמן לרשת וחלקן אף שומרות העתקים של הדפים המודפסים באחסון פנימי. ועל כן חשוב שהמדפסת עצמה תהיה מדפסת "טיפשה" אשר אינה מחוברת לרשת בעת ההדפסה. כמו כן גירסת ה-LiveCD עצמה בה משתמשים עלולה להכיל מראש רכיב זדוני או שהביוס עצמו עלול להיות נגוע בנוזקה כזו או אחרת שיוכלו לשמור את המפתחות במחשב המשתמש ולשלוח אותן בעתיד כאשר הוא יחובר מחדש לרשת. נכון לעכשיו לא נראה שקיימות בשטח נוזקות עם היכולות האלה, אבל ניתן לשער שבעתיד עם העליה בשימוש ובערך של הביטקוין ייצאו גם מי שיכתבו וינסו להפיץ נוזקות כאלה.

[מדפסת פייפר וכתובת נייר שהודפסה בעזרתה]

- **ארנק נייר הוא חד פעמי:**

ניתן להפקיד מטבעות לאותו ארנק נייר מספר פעמים. אך כאשר אנו מעוניינים למשוך כספים מהחשבון עלינו להשתמש במפתח כדי לחתום את העסקה. שלב החתימה עצמו אינו דורש חיבור לרשת אך יצירת העסקה והשליחה שלה כן דורשים חיבור. לרוב המשתמשים יהיה קשה ומסורבל להפריד בין השלבים (למרות שישנן תוכנות ארנק כדוגמת Armory המאפשרות לייצר עסקאות על מחשב אחד ולחתום אותן על מחשב אחר), ולכן בתהליך זה המפתח עלול להיחשף ולא יהיה בטוח להשתמש בו שנית. הדרך הפשוטה ביותר להתגבר על החסרון הזה היא לייצר מראש מספר רב של כתובות בארנק נייר וכאשר משתמשים בכתובת אחת להעביר את היתרה שנשארה בה לכתובת הבאה (גם כאשר משתמשים בארנק חם אשר אינו אופליין שימוש נכון בביטקוין ממליץ לא להשתמש באותה כתובת יותר מפעם אחת ותמיד להעביר את היתרה לכתובת חדשה).

- **ארנק נייר חשוף לגניבה ופגיעה פיזית:**

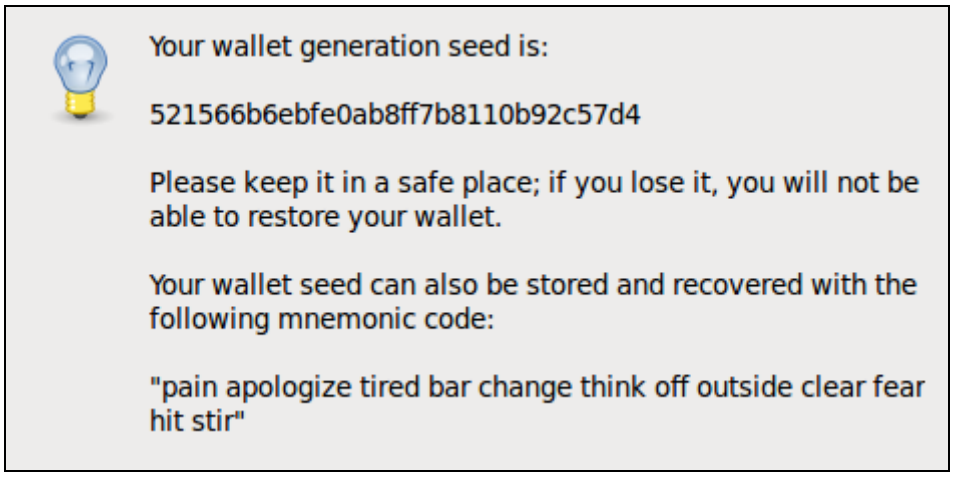
חשוב לזכור שלמרות רמת האבטחה הגבוהה שיש לנו בשימוש בארנק נייר מפני פריצות דיגטליות. במידה ומישהו יפרוץ לנו פיזית לבית או יעשה לנו חיפוש עם צו בדירה, הוא יוכל לקחת מאיתנו את המפתחות הפרטיים. כמובן שניתן לשמור את דפי הנייר עצמם בכספת בבית או בבנק אבל בדומה למחשב נקי האופציה הזו לא זמינה בפני רוב המשתמשים. למרות שהסיכוי לפריצה פיזית קטן משמעותית מהסיכוי לפריצה למחשב עדיין נרצה להיות בטוחים גם ממנה. ועל כן ניתן לשמור את המפתח הפרטי על הנייר בצורה מוצפנת עם סיסמא (אותה כמובן חשוב לזכור!). אבל גם הסיסמא ואולי אפילו הכספת לא יעזרו לנו אם מחר השכנים שלנו ישכחו את התנור דולק בלילה וכל הבניין יעלה באש. לכן מומלץ כמובן להדפיס שני העתקים ועותק אחד לשמור אצל חברים או קרובי משפחה. מי שרוצה להשקיע קצת יותר יכול לפצל את המפתחות למספר גורמים עם השיטה שפיתח עדי שמיר המאפשרת לקחת חתיכת מידע ולפצל אותה למספר גורמים כך שהמידע גם מוגן מגניבה אך עדיין זמין לשיחזור אם אחד החלקים אובד[6]. לא נפרט על הטכניקה כאן אבל מי שלא מכיר אותה מוזמן לקרוא עליה עוד בקישורים.

למרות החסרונות והסרבול הנדרש ביצירת ארנק נייר, זו השיטה המומלצת לשמירת גיבוי לכתובות ביטקוין. לא משנה באיזו שיטה נוספת נשתמש בכדי לייצר את הארנק שלנו תמיד מומלץ בנוסף לגבות אותו גם בצורה פיזית. 95% מהמקרים של אובדן ביטקוין היא על ידי משתמשים שאבדו את הסיסמה לארנק שלהם ולא היה להם גיבוי על נייר.

[6] http://en.wikipedia.org/wiki/Shamir's_Secret_Sharing

- **ארנק מוח \ ארנק דטרמיניסטי:**

שימוש בארנק מוח היא שיטה מאוד קלה ובטוחה, אך היא סומכת לחלוטין על זכרונו של המשתמש. ארנק מוח הוא בסופו של דבר סיסמא, רק שבמקום להשתמש בסיסמא על מנת לפתוח מידע מוצפן או לגשת לשירות מסויים, הסיסמא מתוגרמת ישירות למפתח פרטי באמצעות האשינג עם SHA256. ארנק דטרמיניסטי בדומה לארנק מוח זקוק רק לסיסמא בכדי לשחזר את הארנק אבל בשיטה זו מחושב מהסיסמה seed היכול לשמש ליצירה של מספר אינסופי של כתובות במקום כתובת אחת. החסרון העיקרי בשיטות אלה הוא שעל המשתמש לבחור סיסמא חזקה מאוד שתהיה עמידה לברוטפורס ברמות הגבוהות ביותר. מומלץ שהסיסמא תהיה מורכבת מעשרות תווים ולא תכיל משפטים מוכרים. תוכנת Electrum מאפשרות לקבל גיבוי של ה-seed כסיסמא המורכבת מ-12 מילים ובכך מייעלת את תהליך בחירת הסיסמא.



[גיבוי דטרמיניסטי לארנק של תוכנת Electrum]

בדומה לארנק נייר יש צורך בסביבה סטרלית בעת היצירה של הכתובת הראשונית. ובמידה ורוצים למשוך את הביטקוין יש צורך לעשות זאת גם כן בסביבה סטרלית או להתייחס לכתובת כחד פעמית. אם מדובר בארנק דטרמיניסטי נעדיף כמובן לחשוף רק את המפתח הפרטי של כתובת יחידה ולא את הסיסמא כולה במידה והדבר אפשרי.

חסרון נוסף של ארנק מוח הוא שבמידה ובעל הארנק שוכח את הסיסמא, או חס וחלילה נפגע שכלית ואולי אף נפטר בצורה פתאומית ולא נשמר גיבוי בשום צורה אחרת הביטקוין שלו אבדו לנצח. מומלץ להשתמש בארנק מוח רק על ידי מי שמבין את הסיכונים בצורה מלאה ויודע לבחור סיסמאות חזקות בצורה מתאימה. גם אז מומלץ תמיד לגבות את הארנק בצורה נוספת. הסיבה היחידה אולי להשתמש בארנק מוח בלבד ללא גיבוי היא כאשר אנו רוצים להגן על עצמנו באופן מוחלט מפני החרמה של הביטקוין על ידי רשויות החוק. אבל כל עוד אנחנו פועלים במסגרת החוק קשה להאמין שיש סיבה שלא לשמור גיבוי נוסף.

- **ארנק תוכנה \ ארנק אונליין:**

ארנקים אלה מכונים גם ארנקים חמים. ארנקים אלה מאוד קלים ליצירה ושימוש יומיומי. ההבדל העיקרי בין ארנק תוכנה המותקן במחשב לבין ארנק אונליין הוא שבארנק תוכנה כל המידע נשמר על מחשב המשתמש ואילו בארנק אונליין המידע נשמר בשרתים של החברה שמספקת את השירות. במידה וניתן לסמוך על החברה המספקת את הארנק רמת האבטחה של ארנק אונליין יכולה להיות דומה ולפעמים אפילו גבוהה יותר מרמת האבטחה של ארנק מקומי.

כדי להשתמש בארנק תוכנה כל שצריך הוא להתקין את אחת מתוכנות הארנק הזמינות על המחשב. בעת ההפעלה התוכנה תייצר מספר כתובות לשימוש מיידי. מכיוון שכל המפתחות זמינים ישירות לתוכנה, כל מה שהמשתמש צריך לעשות הוא להשתמש בתוכנה להעברת כספים או לקבל ממנה כתובת אליה יכולים לשלוח לו כספים. ניהול המפתחות הפרטיים ויצירה של כתובות חדשות נעשה בצורה כמעט שקופה לחלוטין מבחינת המשתמש.

למרות שניתן ואף מומלץ להצפין את המפתחות ולגבות אותם הדבר לא קורה כברירת מחדל ועל המשתמש להיות מודע לאפשרויות האלה. יותר מזה, מכיוון שארנקי תוכנה יוצרים כתובות חדשות בעת השימוש בהן (בעיקר בעת שליחה של כספים) מומלץ לגבות את הארנק מחדש לאחר ביצוע פעולות.

ארנקי אונליין קלים אף יותר לשימוש מארנקי תוכנה מכיוון שהמשתמש אינו צריך אפילו להתקין תוכנה על המחשב. ברוב המקרים ארנקי אונליין יקחו על עצמם הן את משימת הגיבוי והן את משימת האבטחה של הארנק ויחסכו מהמשתמש לבצע את הפעולות האלה בצורה ידנית.

על ארנקי אונליין קשה לדבר כמכלול שלם מכיוון שישנו מספר רב של שירותי אונליין המציעים שמירה של הביטקוין עבור המשתמשים וכל יום צצים שירותים חדשים. לצערנו ישנם שירותים רבים אשר אינם מבינים באבטחת מידע ויותר מזה עלולים בעצמם לגנוב את המטבעות המופקדים בהם ולכן חשוב מאוד להשתמש בשירות מוכר ואמין ולהבין כיצד הוא עובד. מכאן והלאה נדבר רק על שירותי צד שלישי שנחשבים לאמינים. ישנם שני סוגים עיקריים של ארנקי אונליין הראשון כדוגמת coinbase מנהל בצורה מוחלטת את הביטקוין של המשתמשים בו. השירות כלל לא שומר את הביטקויינס של כל לקוח בנפרד וכל המטבעות נשמרים בחשבונות בשליטת החברה. לטענת מפעילי האתר בפועל 90% מהמטבעות שלהם נשמרים בכלל באחסון קר על ארנקי נייר השמורים בכספות במספר מוקדים. פורץ שיפרוץ לשרתי coinbase ויגנוב משם את 10%הביטקוינים שכן שמורים בארנקים חמים, לא יגנוב בפועל ביטקוינים מהמשתמשים אלא מהחברה עצמה. מעניין יהיה לראות כיצד תגיב החברה לכך וכמה מנזקי הפירצה יגולגולו חזרה למשתמשים. מה שברור הוא שבמקרים קודמים בהם נפרצו חשבונות ספציפיים של משתמשים ונגנבו מהם מטבעות החברה לא פיצתה את המשתמשים לאחר

מכן. כך שעל המשתמשים לדאוג לאבטח את החשבונות שלהם בצורה הולמת (סיסמא חזקה, זיהוי בשני שלבים) וכן את המחשבים בהם הם משתמשים לגשת לשירות. במידה ושירות coinbase יסגר למשתמשים לא תהיה שום יכולת לגשת למטבעות שהופקדו שם.

הסוג השני של ארנק אונליין הוא ארנק היברידי, כדוגמת blockchain.info, המשתמש בשרתיו אך ורק כגיבוי מוצפן למפתחות הפרטיים. גם אם שרתי החברה נפרצים כל המפתחות מוצפנים ולא ניתנים לשחזור ללא סיסמת המשתמש. כל תהליך ההצפנה והפתיחה של הארנק מתבצע בדפדפן והסיסמה מעולם לא נשמרת או אפילו נשלחת לשרתי החברה. אם השירות נסגר ומפסיק לפעול המשתמש עדיין יכול להשתמש בגיבוי שנשמר אצלו מקומית כדי למשוך את הביטקוין מהארנק שלו ללא צורך בשיתוף פעולה מצד מפעילי השירות. בהנחה שהמשתמש בחר בסיסמא חזקה (ומומלץ גם להשתמש בזיהוי 2 שלבים) הוא יכול להיות בטוח שגם אם שרתי החברה יפרצו או רשויות החוק יבקשו גישה למטבעות שלו המטבעות שלו עדיין יהיו בטוחים כי לא ניתן להשתמש במידע השמור על שרתי החברה ללא הסיסמה. מכיוון שאם שרתי החברה יפרצו ביכולתו של הפורץ לשנות את קוד האתר כך שישלח את הסיסמא לשרת ולשמור אותה שם מומלץ להשתמש בתוסף לדפדפן המריץ העתק של האתר בצורה מקומית ולא סומך על קוד המגיע מהשרת כלל. משתמש אשר ישתמש בכל אמצעי האבטחה העומדים לרשותו עם ארנק מסוג זה יוכל להגיע לרמת אבטחה הכמעט זהה לזו של ארנק תוכנה מקומי. בנוסף יזכה לגיבויים אוטומטים וכן להצפנה הכרחית כך שלמשתמש הפשוט ארנק אונליין מסוג זה יהיה מוגן יותר מארנק היושב מקומית במחשב ולא עובר גיבוי והצפנה, אך בסופו של דבר גם ארנקים אלה בטוחים רק כל עוד המחשב בו משתמשים בהם בטוח.

- **ארנק פלאפון:**

ארנק פלאפון הוא הדרך הזמינה ביותר להשתמש בביטקוין. במדינות רבות בעולם ואפילו בישראל ישנם עסקים מקומיים המקבלים ביטקוין. הדרך הנוחה אם לא היחידה היום לשלם בעסקים אלה היא באמצעות תוכנות ארנק המותקנות על הפלאפון. בדומה למחשב גם לפלאפון ישנן תוכנות ארנק המותקנות על המכשיר ותוכנות הזמינות אונליין. רמת האבטחה של ארנקים אלה דומה לארנקים המשתמשים בהם במחשב כאשר ההבדל העיקרי הוא שסביבת הריצה בפלאפונים באופן כללי נחשבת להרבה פחות בטוחה מסביבת הריצה במחשב. כמו כן, ישנה סכנה הרבה יותר גדולה שהפלאפון יגנב יאבד או יתקלקל ועל כן חשוב מאוד לשמור על גיבוי לארנק. באופן כללי אם אנחנו מסתכלים על ארנק נייר כמקביל לכספת בבנק אנחנו צריכים להסתכל על ארנק בפלאפון כמקביל לארנק אותו אנחנו סוחבים בכיס כל הזמן ולדעת שקיים סיכוי שהוא יגנב או יאבד..

מהסקירה שעשינו על מגוון השיטות לשמור על כתובת ביטקוין ניתן להבין שאין שיטה אחת שניתן להשתמש בה. ארנקי נייר אולי מאוד בטוחים מפני אובדן או גניבה אבל קשים לשימוש יומיומי. ואם נרצה להשתמש בביטקוין שלנו לשלם על בירה בפאב השכונתי כנראה שנהיה חייבים להסתובב עם תוכנת ארנק שתהיה מותקנת בטלפון. ועל כן הדרך הבטוחה והשימושית ביותר להשתמש בביטקוין היא בשילוב של מספר ארנקים:

- ארנקי נייר לחסכונות ביטקוין האמורים להשמר לזמן רב וכן כגיבוי לכל שאר הארנקים שלנו.
- ארנקי תוכנה \ ארנקי אונליין לסכומים קטנים לקניות אונליין ושימוש יומיומי.
- ארנק בפלאפון לסכומים קטנים עוד יותר לשימוש יומיומי מיידי.

כניסה של ארנקי חומרה לשוק ושימוש נרחב בהם יוכל להקל על האבטחה והשימוש היומיומי בביטקוין אבל גם אותם כנראה שנרצה לגבות על ארנק נייר.

מניעת רמאויות

ביטקוין הוא מטבע מבוזר השואף להיות דומה למזומן דיגטלי. לאחר ביצוע העברה לא ניתן לבטל את העברה במקרה של רמאות או אי-קבלת המוצר. לכן יש לשים לב - כאשר מבצעים רכישה בביטקוין חשוב לדעת שניתן לסמוך על המוכר לספק את הסחורה. אתרי מסחר רבים המשתמשים כמתווכים בין המוכר לקונה מספקים שירותי ESCROW בהם הכספים נשמרים אצל האתר המתווך עד שהלקוח מאשר את קבלת המוצר. הבעיה בשירותים אלה היא שבמידה והגורם המתווך נסגר או מחליט להעלם עם הכסף שהופקד אצלו ללקוחות ולעסקים אין דרך לקבל את הביטקוינים שלהם בחזרה.

פתרון לבעייתיות בשימוש ב-ESCROW הוא שימוש בכתובת מרובות חתימה. כתובות ביטקוין אלה מאפשרות למשוך מהן כספים אך ורק באישור החתום על ידי מספר מפתחות פרטיים (ניתן להגדיר סף של m מ-n). מוכר וקונה יכולים לייצר כתובת כזו במשותף עם צד ג' ששניהם סומכים עליו. הקונה יעביר אליה את סכום הרכישה. לאחר קבלת המוצר, יאשרו הן המוכר והן הקונה העברה המעבירה את הכסף מהכתובת המשותפת לכתובת שברשות המוכר בלבד. במידה ולא תהיה הסכמה יוכל כל אחד מהצדדים לפנות לגורם המתווך הנבחר ולבקש ממנו לחתום איתם על העברה של הכסף. הגורם המתווך אינו יכול למשוך את הכספים בעצמו ללא שיתוף פעולה של אחד הצדדים ואם שני הצדדים פועלים בצורה הוגנת ומסכימים בניהם הם אינם זקוקים כלל לשיתוף פעולה מצד הגורם המתווך.

למרות שכתובות מרובות חתימה מוגדרות ונתמכות על ידי פרוטוקול הביטקוין אין כיום תוכנות ארנק נפוצות המאפשרות ליצור ולהשתמש בכתובות כאלה אך ניתן לשער שאם תהיה לכך דרישה רבה מצד המשתמשים ארנקים רבים יתחילו לממש את החלק הזה של הפרוטוקול.

פרטיות

למרות ההצהרות הרבות בתקשורת על השימוש בביטקוין בידי ארגוני פשע והאנונימיות בשימוש בו הדבר רחוק מלהיות נכון. חשוב לזכור כי הבלוקצ'יין ובו כל עסקאות הביטקוין זמין לעיון על ידי כל מי שמעוניין בכך. אומנם אין קישור בין כתובת מסויימת לבעלי הכתובת אבל במידה וישנה דרך לקשר בין הכתובת לבעל החשבון (נגיד במידה והוא פרסם אותה באתר האינטרנט שלו) ניתן לגלות את כל ההעברות שהגיעו לכתובת זו ונשלחו ממנה. יותר מזאת אם התבצעה העברה מכתובת זו הכוללת שימוש בכספים מכתובות נוספות (כפי שמתאפשר בפרוטוקול ומתבצע אוטומטית על ידי מרבית תוכנות הארנק), ניתן לשער כמעט בוודאות כי כל הכתובות שייכות לאותה יישות ואף נעשו נסיונות לקשר בין כלל הכתובות בבלוק ציין ליישויות המחזיקות אותן[7]. ארגוני פשע מנוסים בהעלמת כספים ושימוש באנשי קש גם בכסף שאינו דיגיטלי אך לאדם הפשוט יהיה הרבה יותר קשה לקבל מראש כספים בביטקוין מבלי לחשוף את הזהות שלו.

כדי לשפר מעט יותר את האנונימיות של ביטקוין מוצעים היום מספר שירותי "מכבסה" המשמשים להלבנה של ביטקוין. המשתמש שולח את המטבעות שלו לכתובת של השירות ומשם הם מתערבבים לעשרות כתובות ביטקוין עם מטבעות של לקוחות אחרים. מעשרות כתובות אחרות נשלחים מטבעות בסכום דומה לכתובת חדשה של המשתמש. ככה כל אחד מקבל כסף של מישהו אחר ורק נותני השירות יכולים לדעת למי שייך הכסף.

הבעיה גם פה היא בהסתמכות על צד שלישי שיכול גם לגנוב את הכסף וגם לחשוף את זהות המשתמשים או לפחות את הכתובות המקוריות של המשתמשים בשירות. כיום מפותח פרוטוקול zerocoin הפועל על גבי רשת הביטקוין ויאפשר לצמתי ביטקוין המעוניינים בכך להפעיל שרותי מכבסה מבוזרים שיהיו אנונומיים לחלוטין.

בנוסף חשוב לדעת כי למרות שכתובת ה-IP של המשתמש לא נרשמת בבלוקצ'יין הצמתים המקבלים אליהם את ביצוע העסקה יודעים מי הצומת ממנה הם קיבלו אותה. ממשלה או גוף גדול יכול להפעיל מספר צמתי ביטקוין ברחבי העולם ובכך לזהות כמעט במדיוק מה כתובת ה-IP של המחשב שיצר את

[7] http://eprint.iacr.org/2012/584.pdf

העסקה. לכן מי שרוצה לשמור על אנונימיות גבוהה מומלץ שישתמש ב-TOR או ב-VPN אנונימי בכדי לבצע את הטראנסקציות.

על כותב המאמר

יוני יחזקאל - מהנדס תוכנה המתמחה בפיתוח פרונט אנד וטכנלוגיות ווב. חובב אבטחת מידע, קוד פתוח וביטקוין. תוכלו למצוא עוד מידע וכלים כמו LiveCD המיועד לייצור ארנקי נייר ושימוש ביטקוין בבלוג שלי ב:

https://blog.non.co.il

תודות

תודה למני רוזנפלד מאיגוד הביטקוין הישראלי על העזרה בכתיבת המאמר.

לקריאה נוספת

- https://en.bitcoin.it/wiki/Securing_your_wallet
- http://bitcoin.org/en/secure-your-wallet
- https://en.bitcoin.it/wiki/Hardware_wallet
- https://www.bitaddress.org
- http://bitcoin.org/en/protect-your-privacy
- http://fieryspinningsword.com/2013/12/01/how-to-create-a-reasonably-secure-bitcoin-paper-wallet/
- http://bit.ly/JuEe2Q - (כיצד לייצר ארנק נייר - מאת גיל אסייג)
- http://passguardian.com

פתרון בעיית התשלומים ב-Bitcoin

מאת יהונתן קלינגר, נדב איבגי וליאור גבעון

מבוא וגילוי נאות

ביטקוין הוא אחד מני כמה מטבעות מבוזרים המופצים באמצעות רשתות עמית לעמית (Peer to Peer) ומבוססים על פרוטוקול של קוד פתוח. בהיבט הפיננסי, ביטקוין מציב אתגרים משמעותיים על הדרך בה אנו תופשים כיום את המוסדות הפיננסיים; אולם, לביטקוין נקשרו מספר עסקאות בעייתיות הקשורות לפעילות פלילית כגון סמים או הלבנת הון, מה שמרחיק ממנו את המוסדות הפיננסיים לאחרונה.

אחת הסיבות שמונעות את אימוץ פרוטוקול הביטקוין בקהילת הגולשים ברשת, מעבר לאוריינות טכנולוגית, היא חוסר הפיכות העסקאות שבו. מצד אחד, מדובר על דרך שבה לא ניתן לעקוץ מוכרים בצורה של הכחשת עסקה. מצד שני, קיומו של מנגנון מסגנון של "הכחשת עסקה" מייצר עלויות עסקה ועמלות אשר מייקרות את מחיר המוצר.

על ידי יצירת מנגנון בוררות, כשם שמוצג על ידי Bitrated באמצעות מערכת ה-MultiSig, ניתן להשיג חסכון בעמלות ועלויות עסקה, ולמצוא מערך שיוכל לטפל בעסקאות Bitcoin אף בצורה שתוכל להכניס ודאות עסקית.

במאמר זה נדבר על היתרונות של מעבר לטכנולוגית MultiSig בכלל, ועל הצורך האבטחתי בכך. לצורך העניין ננית כי אין צורך להציג את ביטקוין כמטבע מבוזר, את היתרונות שלו ואת היקף השימוש בו, ולכן נעסוק בכך בצורה קצרה במיוחד.

לצורך העניין, ובקליפת אגוז, ביטקוין הוא מטבע מבוזר, שאינו תלוי באדם אחד או בבנק אחד, ומבוסס על אמון החברה בכוחו של המטבע. המטבע עצמו אינו הילך חוקי ואינו מוגדר כמטבע, אלא כרכוש לצרכי סחר חליפין. ככזה, ישנם לא מעט גורמים שמקבלים אותו כאמצעי תשלום לגיטימי למדי, כמו שירותי אחסון וכדומה. יש גם חברות המציעות המרה של ביטקוין לדולרים מוחשיים או מטבעות אחרים, ואפילו שמועה על חברות שמציעות כרטיסי אשראי מבוססי ביטקוין. היתרון בביטקוין הוא שהוא לוקח את הטוב משני העולמות: את האנונימיות והשליטה של מזומן, יחד עם המיידיות והאפשרות לקיים עסקאות מרחוק בכרטיסי האשראי.

היתרון המשמעותי של ביטקוין הוא הניתוק שלו מכלכלה אחת מרכזית והפיכתו למטבע של הרשת. מאז הקמתו של ביטקוין, אגב, הוקמו עשרות שונות של מטבעות מבוזרים, כשלכל אחד מהם יש יתרונות וחסרונות אחרים, ובאים לטפל בבעיות כאלו או אחרות בפרוטוקול. אולם, נכון להיום באף אחד ממטבעות אלו אין מסחר משמעותי, בניגוד לביטקוין אשר ניתן לרכוש באמצעותו בשלל שירותים אפילו בישראל.

גילוי נאות: מערכת Bitrated מופעלת על ידי נדב איבגי וליאור גבעון, יהונתן קלינגר חבר בועד המייעץ של המיזם ונותן לו ייעוץ משפטי.

בעיית התשלומים

ביטקוין החל עם מאמרו של סטושי נקמוטו (שהוא כנראה שם בדוי) שמדבר על בעיית התשלומים והיכולת לבצע ניהול של ספרי החשבונות בצורה של עמית לעמית (Peer to Peer) כך שבכל רגע נתון כל אחד מכל החברים ברשת יחזיק עותק של יומן העסקאות הכללי. בצורה כזו, אם לאליס יש מטבע דיגיטלי והיא העבירה אותו לבוב, כל ספרי הניהול הדיגיטליים המנוהלים ביחד יכתבו זאת בספר, וכעת כאשר אדם ישאל "מי מחזיק את המטבע" התשובה תהיה "בוב".

בעיה זו, שנפתרה במאמרו של נקמוטו על ידי יצירה של שרשרת בלוקים שמכילים את כלל העסקאות (Blockchain) שזמינה לכל הציבור לצפיה (כאן), מאפשרת עסקאות בלתי הפיכות; מרגע שהעסקה עברה, אין יכולת להחזיר את הכספים. כך, לדוגמא, כאשר ישנו שוד מקוון שמאפשר גניבה של מיליוני דולרים אזי המחזיקים בכספים יכולים לראות את השוד בשידור חי, אך לא לבטל את העסקאות האלו.

בעיה זו, של תשלומים, קיימת גם כאשר משלמים במזומן, אולם מטרת מאמר זה היא לדון בבעיות המשפטיות/טכנולוגיות הקיימות ולהסבירן, כך שיהיה ניתן להבין את מטרת השירות ב-Bitrated ושירותים דומים.

בעיית התשלומים בעולם האמיתי וכרטיסי האשראי

בתחילת הרשת, סוגיית כרטיסי האשראי והשימוש בהם היו מחסום משמעותי מביצוע מסחר אלקטרוני. עד לאמצע שנות התשעים של המאה הקודמת חברות האשראי סרבו כמעט להשתתף במשחק הדיגיטלי בטענה כי עסקאות מקוונות מסוכנות יותר. עסקאות מסוג Chrageback, בהן אדם מתכחש לעסקה שבוצעה, בטענה כי לא הוא ביצע אותה, או כי הוא לא קיבל מוצר, מסוכנות יותר בתחום האינטרנט: כל עוד אין זהות דיגיטלית חזקה, חברות האשראי אינן יכולות להוכיח כי האדם שביצע את העסקה הוא אכן בעל כרטיס האשראי (בהתחשב בקלות בה ניתן להשיג מספרי כרטיסי אשראי). כך, לדוגמא, באתרים למבוגרים יש אחוז רב יותר של הכחשות עסקה, כאשר אנשים טוענים שלא הם היו מי שצרך את השירותים (הרבה פעמים לאחר שמשפחתם נחשפת לאותו החשבון). סוגיית הכחשת העסקה יוצרת מצב

בו בעל עסק צריך להכין מראש עודף שמיועד למקרים כאלו. העודף העסקי מתחשב בכך שחלק משמעותי מעסקאותיו מוכחשות (בין אם מדובר באתרים למבוגרים או בכלל). כאשר, ברוב המקרים מדיניות חברת האשראי היא לזכות את בעל הכרטיס ולהעניש את בית העסק, ובמקרים חריגים לספוג את עלויות העסקה המבוטלת. התוצר המשמעותי במקרה כזה הוא כפול: (1) לקוחות ועסקים טובים משלמים יותר, למרות שהם לא מבצעים הונאה ו-(2) לקוחות אינם מפנימים את הסיכונים, כיוון שהם יודעי שבכל מקרה יזוכו על ידי חברות האשראי ולכן נכנסים לעסקאות מסוכנות יותר.

מודל אמון בביצוע רכישות

חלק מזירות המסחר ברשת, כדוגמאת eBay, הפעילו מערכת של דירוג סוחרים ועסקים. בצורה כזו, צדדים לעסקה יכולים לדעת כמה עסקאות ביצע האדם בעבר, האם קיבל עליהן דירוג חיובי מהצדדים, האם המוצרים שמכר הגיעו בזמן, האם התשלומים שביצע הוכחשו וכדומה. לשיטה זו יתרונות משמעותיים כאשר צדדים רוצים להכנס לעסקה: היא מאפשרת לתגמל עסקים הוגנים אשר יש להם מוניטין לשמר, ומזהירה אנשים לבל ישתמשו במוניטין לרעה, שכן כל ירידה מינורית מ-100% ל-99.8% דירוג חיובי עשויה להשפיע על מכירותיו של העסק. הבעיה העיקרית במערכת מסוג כזה היא שהיא נתונה למניפולציה בקלות יחסית: מצד אחד, ניתן למכור הרבה מאוד מוצרים לחשבונות פיקטיביים בשמך כדי לקבל מוניטין טוב, או אפילו מוצרים מוחשיים בצורה מוזרה מעט, ומצד שני, ניתן למוטט עסק על ידי כתיבת ביקורות שליליות למרות שלא מגיעות לו כאלו על ידי מתחרים. לכן, האמון אמנם עוזר, אך אינו הדרך היחידה והמובטחת.

מעשי עוקץ והונאה של רוכשים

ביטקוין, בניגוד לכרטיסי אשראי ומערכות אחרות, מכיל עסקאות שאינן הפיכות. המשמעות היא שרוכש אשר ביצע רכישה אינו יכול לקבל את כספו בחזרה. אם כן, איזה סוג של מעשי עוקץ על ידי רוכשים קיימים? הסוג הראשון של מעשי עוקץ הינו כזה אשר משלם בכספים שלא שלהם. לדוגמא, על ידי פריצה לחשבונות קיימים ושימוש בכספים המצויים שם. במצב כזה, כאשר כתובת התשלום של בית העסק מזוהה, מגיע הנעקץ לבית העסק ומבקש את כספו בחזרה, שכן לא הוא ביצע את העסקה. הבעיה? במקרים רבים פעולה זו מבוצעת לאחר שסחורה כבר נשלחה, או שניתנו שירותים עבור אותו התשלום.

מעשי עוקץ והונאה של מוכרים

מעשה העוקץ השני הוא דווקא על ידי מוכרים או ספקי שירותים. הם מקימים אתר אינטרנט מכירתי, המציע מוצרים בתשלום מיידי בביטקוין. לאחר התשלום, הם מתחייבים למשלוח, אשר לעולם לא יבוצע. שיטה אחרת היא הקמת שירותים פיננסיים מבוססי מטבעות קריפטוגרפיים והעלמות עם הכסף. לדוגמא, בשוד הדוגיקוין האחרון (מטבע וירטואלי אלטרנטיבי שצובר פופולריות), נפרץ אתר אינטרנט שסיפק שירותי ארנק וירטואלי. אולם, יש הטוענים כי בכלל לא מדובר על פריצה, אלא על הונאה של מפעיל השירות.

מודלים מקובלים בעולם

אז לצורך העניין, הנה נסכם כיצד אפשר להתמודד עם בעיות התשלום ואי התשלום, וכיצד הדבר נעשה עד כה. הרשימה, ברור, אינה ממצה, אבל היא מכסה את רוב הפתרונות המקובלים שהגיעו. כאמור, המטרה היא לדבר על עסקאות צרכניות קטנות, ולא על רכישות עסקיות (כגון, נניח, הקניה של מניות בחברה, או רכישת נדל"ן). כאשר, במקרים כאלו יש סיכונים מובנים אשר מטופלים בדרך הכלל על ידי ביטוחים רבים ואחריות אישית על נושאי משרה בעסקה.

כרטיסי אשראי

הכחשת עסקה. בתחום כרטיסי אשראי נפוצה השיטה של "הכחשת עסקה". צורה זו של טיפול בהונאות עובדת כך: מבוצעת העסקה, כסף יוצא מחשבון הבנק של הלקוח, ועובר לעסק. כאשר הלקוח שם לב כי לא הוא ביצע את העסקה, הוא יוצר קשר בצורה אקטיבית עם חברת האשראי, ומעביר הצהרה כי לא ביצע את העסקה. לאחר העברת הטופס, חברת האשראי עורכת בירור ומשיבה לו את הכסף, והרבה פעמים אף מענישה את בית העסק על כך.

זירות אלקטרוניות: בוררות

זירות אלקטרונית, כדוגמאת eBay, מפעילות הליך בוררות חובה על הצדדים לעסקה, בה הזירה האלקטרונית מהווה את הבורר. לכך יש יתרון משמעותי של עלויות; אולם, פעמים רבות הזירה נדרשת להחזיק בעצמה חלק מהכסף ולגבות עמלות עבור השימוש בזירה. בהתחשב בכך שחברות כמו eBay מפעילות שירותים שנועדו להגן על הלקוח במקרים בהם העסק לא מספק לו את הסחורה, ולהשיב את כספו, הן צריכות לגבות עמלות אשר יצדיקו את השבת הכסף (פעמים רבות על חשבונן). כלומר, הזירות הופכות לצד מעורב בעסקה.

עסקאות מזומן: התעלמות

עסקאות מזומן, ככאלו, יוצרות בעיה משמעותית: אם מדובר על עסקת המזומן הקלאסית בה אדם רוכש מוצר בשוק פשפשים, אין לו את היכולת לזהות את המוכר, אין לו ודאות שהמוכר כלל יהיה שם לאחר זמן מסוים לספק אחריות, ואין לו את האפשרות לקבל את כספו בחזרה בשום צורה שהיא ללא רצונו הטוב של המוכר (או בית משפט).

חוק הגנת הצרכן וחוסר הרלוונטיות שלו

חוק הגנת הצרכן הישראלי מקנה לצרכנים ולקוחות הגנה משמעותית על פי החוק: הוא מאפשר ביטול עסקאות מכר מרחוק (עסקאות טלפוניות או אינטרנט), מחייב אחריות לשירותי מסוימים ועוד. אלא, שיש בו בעיה רצינית: החוק חל בישראל, ודורש פניה לבית משפט כאשר מפרים אותו. כאשר עסקה אלקטרונית מבוצעת בין שני קצוות תבל, קשה מאוד לאכוף את החוק על סוחר בסין, ועוד יותר קשה לקבל את הכסף בחזרה כאשר העלות של משלוח כתב התביעה בדואר רשום לסין גבוהה יותר מאשר עלות העסקה.

פרוטוקול MultiSig

למרות שהפרוטוקול עצמו קיים במערכת ביטקוין, מערכת MultiSig לא זכתה להרבה הכרה או לכניסה לתוכנות ארנק רשמיות של ביטקוין בצורה פשוטה ונוחה. פרוטוקול MultiSig מתבסס על העקרון הבא: אם P מתוך N אנשים לעסקה יאשרו אותה, אז הכסף יעבור מגורם א' לגורם ב'. אם לא, אז הכסף ישאר בעסקה. הדבר דומה מאוד לשיטת ה-Secret Sharing בה ניתן לאחסן מידע שיהיה זמין גם רק אם P מתוך N אנשים יהיו חיים. הפרטוקול עצמו מוגדר כעסקה יחסית אקזוטית, שלא נדרשת לכל אדם. אולם, Bitrated מיישמת את שיטת ה-MultiSig לצורך ביצוע עסקאות.

שירותי Escrow ושירותי Trust

עד היום, רוב שירותי התשלום התבססו על מערכות של Escrow (נאמנות). בצורה כזו, הצדדים לעסקה נתנו לצד שלישי (נאמן) להחזיק עבורם את הכסף, וכאשר הם אישרו לנאמן להעביר את כספי העסקה, הוא יעשה זאת. שירותי נאמנות קיימים ברחבי העולם ומטופלים לא אחת על ידי עורכי דין. פעמים רבות מטרת השירותים היא להבטיח קיומו של תנאי (נניח, העברת בעלות על קרקע). היתרון המשמעותי בשירותי נאמנות הם האמון והביטוח של הנאמן. נאמנים בונים את עסקיהם על שמם הטוב ועל יכולתם להחזיק בנאמנות נכסים וכספים, ולכן כל תביעה או טענה כנגדם תפגע קשות באמון זה ובשמם הטוב. לכן, בשים לב לשירותים כאלו, הלקוח אשר מסתמך על נאמן יודע כי ברוב המקרים, הנאמן והמוניטין הרב שיש לו לא יפעלו כנגדו.

החסרונות בשירותי Escrow או Trust

החסרון הראשון בשירותי נאמנות הוא כי הנכס נמצא בבעלות מוחלטת של הנאמן. במצב כזה, פגיעה בנאמנות בסגנון פרשת אתי אלון, בה אדם מדווח ללקוחות כי הוא מחזיק בסכום מסוים, כאשר בפועל הוא לקח את הכספים לעצמו, היא אפשרית בצורה טכנולוגית, הגם שיש לה סנקציות משפטיות. הבעיה השניה היא יצירת אמון של הנאמן; נאמן חדש אשר אין לו מוניטין, לא יוכל לקבל אמון ציבורי אלא על ידי סיכון משמעותי של נכסיו האישיים וכניסה לעסקאות מפוקפקות יותר, אשר נאמנים רגילים לא יקחו. מצב זה מייצר גם עלויות עסקה משמעותיות: הנאמן חייב להחזיק את הנכס בנאמנות, חייב לגבות עליו עמלה, וחייב לבטח את עצמו בגין מעילת פנים או רשלנות כדי להמנע מנזקים.

מערכת Bitrated

מערכת Bitrated ושימוש בפרוטוקול MultiSig מבטלת את הסיכונים הקיימים לנאמן מכמה סיבות. המערכת עובדת כך: אליס ובוב מתקשרים בעסקה; הם קובעים כי צ'ארלי יהיה הבורר בעסקה בכל מקרה של מחלוקת בינהם, ומקבלים את הסכמתו של צ'ארלי. אליס מעבירה כספים לחשבון MultiSig של 2

מתוך 3; וכאשר בוב מספק לה את השירות, שניהם בהסכמה יכולים לשחרר את הכספים לטובת בוב. במקרה בו אליס לא תהיה מוכנה לשחרר את הכספים, הרי שבוב יוכל לפנות לצ'ארלי ולבקש את שחרורם. אז, ורק אז, מעורבותו של צ'ארלי בעניין תפתח. כלומר, לצ'ארלי אין עלויות עסקה עד שנוצר סכסוך בין הצדדים. בהתחשב בכך שברוב המקרים אין סכסוך כזה, הרי שקל יותר לנהל את המערכת.

כעת, אם ישנו סכסוך, צ'ארלי יכול להוות בורר בין הצדדים (ולא נאמן) ולחקור את האמת ואת הנסיבות, לשאול מה הסיבות לאי הרצון ולנסות להביא את הצדדים לידי פתרון מוסכם. אם הוא לא מצליח, אז הוא יכול (בהסכמה של לפחות אחד מהצדדים) להעביר את הכספים לגורם שהוא חושב שזכאי לקבלם.

אם נוצר מצב אחר, בו בין אליס ובוב יש מחלוקת, אך לאחר בירור של צ'ארלי הם מאמינים שהוא לא בורר טוב מספיק, אזי הם אפילו יכולים להעביר את הכסף לכתובת MultiSig אחרת, אשר תנוהל על ידי בורר אחר.

השגת אמון באמצעות קוד פתוח

חלק משמעותי מהדרישה במצב כזה היא שהאמון יהיה לכל חלקי המערכת. אם הכספים מוחזקים על ידי פלטפורמה אשר גם יכולה לגשת לכספים, לדוגמא, אז יש צורך באמון באותה הפלטפורמה. מה המשמעות במצב כזה? במצב כזה, אם ישנה מעילה בפנים (כמו במקרה שנחשד בעניין Dogewallet, נניח), אזי כל מנגנוני האמון נשברים בנקודה אחת. לכן, חשוב במיוחד כי למי שמפעיל את המערכת לא תוכל להיות גישה לכספים, ולא יהיה מסוגל להעלים את הכספים במקרים שבהם ירצה לעשות זאת. לצורך כך, יש צורך במערכת שתחולל לכל משתמש את המפתחות הפרטיים שלו, אך לא תשמור אותם, וגם יש צורך במערכת שתאפשר לכל משתמש לוודא כי לאף גורם אחר אין יכולת להגיע לכספיו. דבר זה יכול להתקיים רק כאשר המערכת כתובה בצורה פתוחה, שבה קוד המקור של המערכת זמין לכל אדם, וכאשר לאותו אדם יש יכולת לוודא (בסופו של דבר) כי המערכת עליה הוא עובד היא המערכת אשר הוא בחן את קוד המקור שלה. כלומר, **על מנת לוודא כי אין הונאה, למשתמש צריכה להיות היכולת לוודא את זהות הקבצים שמטפלים בעסקה, ולוודא שאין גורמים אחרים מעורבים באמצע**.

שמירת מידע בדפדפן בלבד

נושא נוסף שיש לטפל בו הוא העדר שמירה של היסטוריה של עסקאות או מפתחות פרטיים בצד השרת. הסיבה לכך היא ששמירה כזו תתאפיין, בסופו של דבר, בכך שיהיה גורם מרכזי שיוכל לשלוט בכספים (ראה את הסעיף הקודם) או שיוכל לתעד את היסטוריית העסקאות עצמן, ולהשפיע עליהן על ידי שינוי מערך הזיהוי במערכת בעתיד. כלומר, גם אם כרגע המערכת לא שומרת דבר, אין כל הבטחה כי שינוי עתידי במערכת, כאשר חלק מהמידע נשמר בצד השרת, לא יאפשר שינויים כאלו. לכן, התנאי השני ההכרחי לצורך השגת האמון הוא שמירה בצד הדפדפן בלבד.

עסקאות ללא תיווך

תיווך, וזירות לכשעצמן, יוצרות שני אפקטים: הראשון, חיובי במיוחד, הוא קיומה של תחרות ועקב כך השפעה על המחיר לטובת הצרכן על ידי שימוש בשיטות כגון מכירות פומביות או העמדה של מספר מוצרים אחד לצד השני. האפקט השני, והבעייתי יותר, הוא יצירה של עלויות עסקה במקביל: כפי שהסברנו, הזירה האלקטרונית יוצרת עלויות בצורה של עמלות עבור שימוש בה, עמלות עבור תיווך וטיפול במחלוקות, ועמלות עבור הכחשות עסקה לא לגיטימיות. כל זה יחדיו מייקר את ביצוע העסקאות מעבר למחיר הממשי שישנו. כלומר, ניתן עוד להוזיל את המחירים לסוחרים הגונים, אשר אין להם הכחשות רבות, כאשר הם יודעים שאין להם צורך במערכת בוררות ברוב המקרים. לכן, התנאי השלישי הוא שהשימוש במערכת יגרום להורדת מחירים.

מערכת ניקוד

השלב הבא, והדרך לקדם את הנושא היא על ידי יצירה של מערכת ניקוד; כלומר, כל צד במערכת: מוכר, קונה ובורר, יקבל ניקוד על סמך היסטוריית העבודה שלו. הניקוד יאפשר מצד אחד לדעת מיהם הגורמים עם האמון הרב יותר לצורך ביצוע עסקאות, ומצד שני, יאפשר גם לתגמל אנשים אמינים במיוחד, או לגבות פרמיה מאלו שאינם כאלה. יתר על כן, הניקוד יאפשר גם לטפל בסוגיית המחיר שהבורר יגבה (כלומר, שבוררים אמינים יותר יגבו מחיר גבוה יותר עבור שירותיהם) וגם לטפל בסוגיית הקצאת הסיכונים (לדוגמא, מי משלם על הבוררות). מערכת הניקוד תאפשר ביקורות חיוביות ושליליות, וצריכה להיות נלווית לתוך פרוטוקול ה-MultiSig עצמו כדי לטפל בבעיות של החלפת זירות מרובה או יצירה של חשבונות רבים באותה הזירה.

סיכום

השימוש במערכת MultiSig יכול להוזיל את עלויות העסקה של עסקאות אלקטרוניות, הוא יכול לאפשר אמון רב יותר בעסקאות, הוא גם יכול לאפשר קיומן של עסקאות שלא היו יכולות להתקיים בלי מערכת כזו (כגון העברה בטאבו של בתים). המערכת יכולה לחסוך עלויות רבות המשולמות כיום עבור שירותי נאמנות, או עמלות שמשולמות לזירות מסחר. לצורך אימוץ רחב יותר של השיטה, אולם, יש לפתח את התוכנות הרשמיות כך שיכללו את המערכת כברירת מחדל, ולאפשר נגישות רחבה יותר לבוררים.

Android Fragment Injection

מאת רועי חי

הקדמה

במאמר זה אציג פגיעות חדשה שקבוצת המחקר שלי גילתה באנדרואיד. ליתר דיוק הפגיעות היא ב-Android Framework, והיא משפיעה על כל אפליקציה אשר מכילה exported Activity שיורש מ-PreferenceActivity. מכיוון שהשימוש ב-Activity זה הוא שכיח למדי, לא הופתענו לגלות מספר רב של אפליקציות פגיעות (Android Settings, Dropbox, Gmail, Evernote וכד').

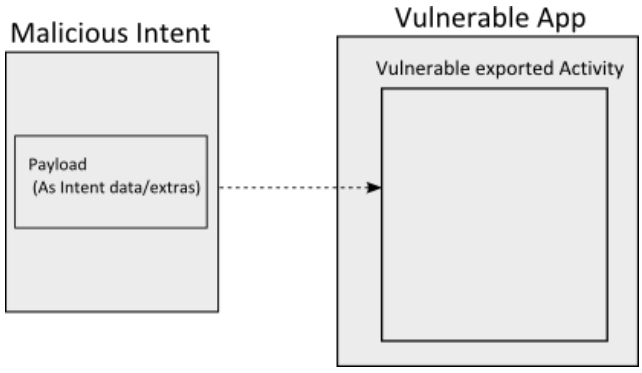
נתחיל מהקדמה קצרה על Android ועל מנגנוני ה-Security בו, נצלול לפגיעות עצמה, נתאר תקיפה אפשרית, ונקנח בתיקון.

קצת על אפליקציות Android

אפליקציות באנדרואיד בנויות ממספר סוגי רכיבים. העיקרי ואולי החשוב מהם הוא ה-Activity. כל Activity מספק מסך UI למשתמש, למשל מסך ה-Bookmarks של הדפדפן. אחת התכונות המרכזיות ב-Android היא שאפליקציה אחת יכולה להריץ (חלק) מה-Activities של אפליקציות אחרות. תכונה זו מאפשרת Feature reuse. למשל, הדפדפן מריץ את Google Play ברגע שהמשתמש גולש ל-Play Store. המימוש של מנגנון זה צריך לקחת בחשבון את העובדה שאפליקציות ב-Android רצות בסביבה מבוקרת, Sandbox. הסיבה לכך היא שהנחת היסוד, בשונה מ-PC, היא שקיים סיכוי גבוה שירוץ Malware על המכשיר, כך שמנגנון ה-Sandbox נועד למנוע מאפליקציה אחת לגשת למידע רגיש של המערכת או של אפליקציה אחרת. אפליקציות מוגבלות ע"י מספר מנגנונים. למשל, כל אפליקציה רצה עם User ID משלה, כך שקבצים אינם נגישים באופן דיפולטיבי ע"י אפליקציה אחרת. בנוסף לכך, Activities שאינם מוגדרים כ-exported (באופן מפורש או לאו) בקובץ ה-AndroidManifest שמסופק עם האפליקציה אינם יכולים להיות מורצים ע"י אפליקציה חיצונית.

Intents

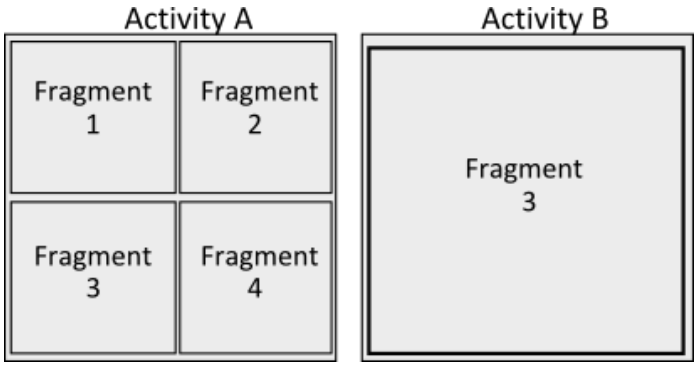
על-מנת להריץ Activity האפליקציה מייצרת אובייקט מסוג Intent ושולחת אותו ל-API המתאים (למשל Context.startActivity). אובייקט ה-Intent יכול לציין הן את Activity היעד והן את ה-Payload. האחרון נכלל במספר שדות באובייקט ה-Intent. בשדה ה-Data וכן ב-Extras שהוא שדה מסוג Bundle, מימוש אנדרואידי ל-Map. ברגע ש-Activity מוגדר כ-exported נוצר חור ב-Sandbox, כך שאפליקציה אחת יכולה לספק מידע זדוני לאפליקציה אחרת. אם המידע לא נצרך בזהירות (ע"י validation או sanitization) עלולה להיווצר בעיית Security. למשל, אם קיים פעפוע של המידע לשאילתת SQL, ללא בדיקת תקינות, האפליקציה תהיה פגיעה ל-SQL Injection. תרשים 1 מדגים את סכמת התקיפה.



[תרשים 1]

Fragments

אנדרואיד מספקת רמה נוספת של גרנולריות ב-UI: Fragments, אשר מהווים בעצם תתי Activities. הרעיון הוא שלהבדיל מ-Activity אשר מספק Feature reuse בכל המערכת, Fragments מספקים Feature reuse בתוך האפליקציה עצמה. כל מופע של Fragment משויך עם מופע יחיד של Activity מארח. הוא יכול לגשת אליו ולכן גם ל-Intent object שהריץ אותו. תרשים 2 מראה את היחס בין Activities ל-Fragments.



[תרשים 2]

מהו ה-PreferenceActivity?

Activity זה מסופק עם ה-Android Framework כך שכל אפליקציה יכולה להשתמש בו (לרשת ממנו). בעזרתו ניתן לייצר מסך הגדרות די בקלות, ולכן אפליקציות רבות עושות בו שימוש, כגון Settings (אפליקצית המערכת), Gmail, Dropbox ועוד. ההגדרות קשורות ל-PreferenceFragments. ה-PreferenceActivity קובע איזה Fragment להריץ ע"י Intent Extra שמסופק לו. הרצת ה-Fragment מבוצעת באופן דינמי ע"י Java Reflection, בתוך הפונקציה הסטטית Fragment.instantiate.

הקוד הבא מכיל את שרשרת הקריאות מתוך PreferenceActivity, מרגע יצירת ה-Activity (onCreate) עד לקריאה ל-Fragment.instantiate:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

   String initialFragment =
getIntent().getStringExtra(EXTRA_SHOW_FRAGMENT);
   Bundle initialArguments =
getIntent().getBundleExtra(EXTRA_SHOW_FRAGMENT_ARGUMENTS);

   if (savedInstanceState != null) {

   } else {
       if (initialFragment != null && mSinglePane) {
           // If we are just showing a fragment, we want to run in
           // new fragment mode, but don't need to compute and show
           // the headers.
           switchToHeader(initialFragment, initialArguments);

        } else {

           if (mHeaders.size() > 0) {
               if (!mSinglePane) {
                   if (initialFragment == null) {

                   } else {
                      switchToHeader(initialFragment, initialArguments);
                   }
               }
           }
       }

public void switchToHeader(String fragmentName, Bundle args) {
   setSelectedHeader(null);
   switchToHeaderInner(fragmentName, args, 0);
}

private void switchToHeaderInner(String fragmentName, Bundle args, int
direction) {
   getFragmentManager().popBackStack(BACK_STACK_PREFS,
```

```
FragmentManager.POP_BACK_STACK_INCLUSIVE);
   Fragment f = Fragment.instantiate(this, fragmentName, args);
   FragmentTransaction transaction =
getFragmentManager().beginTransaction();
   transaction.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE);
   transaction.replace(com.android.internal.R.id.prefs, f);
   transaction.commitAllowingStateLoss();
}
```

והמימוש של Fragment.instantiate באנדרואיד 4.3 הוא:

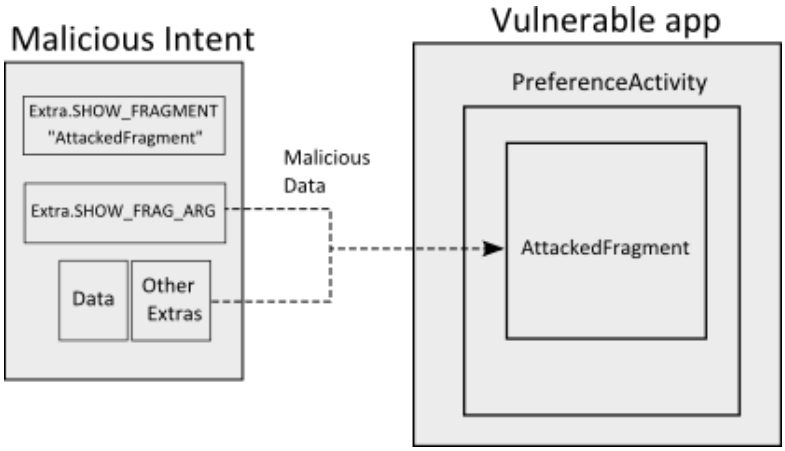
```
public static Fragment instantiate(Context context, String fname, Bundle
args) {
    try {
        Class<?> clazz = sClassMap.get(fname);
        if (clazz == null) {
  // Class not found in the cache, see if it's real, and try to add it
            clazz = context.getClassLoader().loadClass(fname);
            sClassMap.put(fname, clazz);
        }
        Fragment f = (Fragment)clazz.newInstance();
        if (args != null) {
            args.setClassLoader(f.getClass().getClassLoader());
            f.mArguments = args;
        }
        return f;
    }
...
}
```

הפגיעות

כל אפליקציה המכילה exported Activity שיורש מ-PreferneceActivity יכולה להיות מותקפת ע"י אפליקציה זדונית, עקב אי-בדיקת תקינות הקלט במנגנון הרצת ה-Fragments.

אפליקציה זדונית יכולה להריץ את ה-PreferenceActivtiy ולהחליט איזה Fragment הוא יריץ (ע"י שימוש ב-"android:show_fragment" Intent extra). במאמר המלא תיארנו שתי דרכי תקיפה אפשריות, אחת מהן היא למצוא Fragment באפליקציה הנתקפת שמשויך ל-non-exported Activity. בדרך זו ה-Fragment בעצם נחשף ע"י האפליקציה הזדונית, שיכולה לספק לו גם מידע. מכיוון שה-Fragment רץ בדרך כלל בתוך Activity שהוא non-exported, קיים סיכוי סביר שהוא יסמוך על הקלט, דבר העלול לעורר בעיית Security.

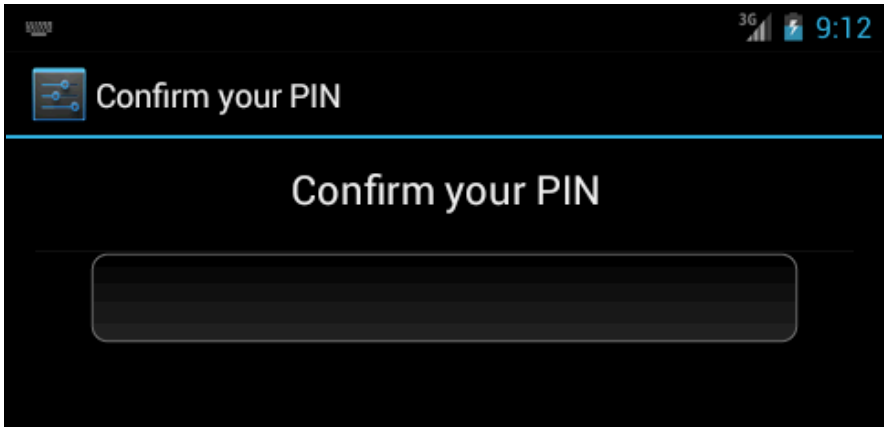
למעשה הפגיעות מעניקה יכולת לאפליקציה זדונית לשתול Fragment מסוים שחי בעולם אוטופי בו הקלט תמיד בטוח, בתוך עולם מסוכן בו לאפליקציה זדונית יש יכולת להשפיע על הקלט. איור 3 מדגים את התקיפה.



[תרשים 3]

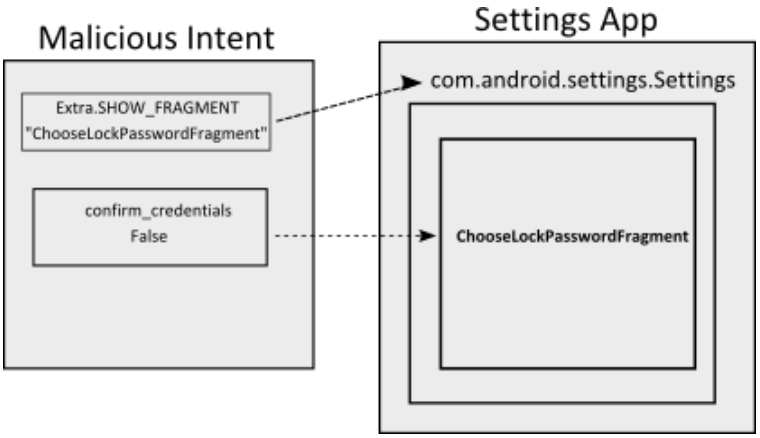
תקיפה לדוגמא: אפליקציית ה-Settings

בחרנו להתמקד באפליקציה זו (אשר פגיעה מכיוון שהיא מקיימת את התנאים המוזכרים בסעיף הקודם) כי היא מצויה בכל מכשיר אנדרואיד, ויש לה הרשאות רבות. למשל, היא יכולה לשנות את הסיסמא (או ה-PIN) של מסך הנעילה. פונקציה זו ממומשת תחת ChooseLockPassword$ChooseLockPasswordFragment. כאשר טוענים Fragment זה הוא מבקש מהמשתמש להכניס את הסיסמא הנוכחית, אלא אם מסופק ל-Activity המארח (ChooseLockPassword) פרמטר בשם "confirm_credentials" (תחת Intent extra). אם ערכו של פרמטר זה הוא false, אז ה-Fragment לא מבקש להכניס את הסיסמא (או ה-PIN). תרשים 4 מכיל צילום מסך של ה-Fragment כאשר לא מסופק הפרמטר.

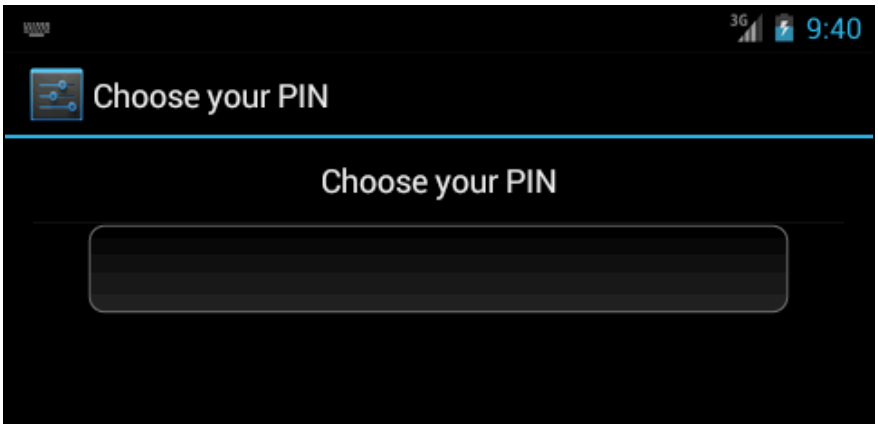


[תרשים 4]

בעזרת הפגיעות אפליקציה זדונית יכולה לטעון את ה-Fragment הנ"ל לתוך exported Activity (Settings), ולספק לו מידע ע"י Intent extras. כך היא יכולה לשלוט ב-"confirm_credentials" ולהגדיר אותו כ-false! המתקפה על ה-Settings מתוארת בתרשים 5, ותוצאתה בתרשים 6.



[תרשים 5]



[תרשים 6]

למעשה בעזרת הפגיעות אפליקציה זדונית יכולה לעקוף כל מגבלה שמוגדרת על הסיסמאות (למשל אורך מינימלי). משתמש פוטנציאלי של התקיפה על Settings הוא תוקף פיזי, למשל גנב (שמעוניין לשנות את הסיסמא) או עובד ארגון (שמעוניין לעקוף את מגבלות ה-Device Administration).

התיקון

גוגל תיקנה את הפגיעות ב-Kit Kat ולמעשה היא העבירה את האחריות למפתח. כעת, עליו לדרוס את הפונקציה PreferenceActivity.isValidFragment אשר מקבלת כפרמטר את שם ה-Fragment ומחזירה true או false האם הפרמטר בטוח לטעינה ולהיפך.

המימוש הדיפולטיבי של פונקציה זו בודק את ה-target SDK version של האפליקציה, אם הוא Kit Kat ומעלה, המימוש זורק exception (ולכן אפליקציות קורסות), אחרת הוא מחזיר true. הקוד הבא מראה כיצד נעשה שימוש ב-isValidFragment בגרסה החדשה של PreferenceActivity תחת Kit Kat.

```
private void switchToHeaderInner(String fragmentName, Bundle args, int
direction) {
        getFragmentManager().popBackStack(BACK_STACK_PREFS,
                FragmentManager.POP_BACK_STACK_INCLUSIVE);
        if (!isValidFragment(fragmentName)) {
            throw new IllegalArgumentException("Invalid fragment for
this activity: "
                    + fragmentName);
        }
        Fragment f = Fragment.instantiate(this, fragmentName, args);
        FragmentTransaction transaction =
getFragmentManager().beginTransaction();

transaction.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE);
        transaction.replace(com.android.internal.R.id.prefs, f);
        transaction.commitAllowingStateLoss();
}
```

לסיכום

לסיכום, אנו ממליצים להתייחס בחשדנות לכל נתון שמגיע מהמשתמש, אשר מחלחל לפונקציות רגישות, כדוגמת Fragment.instantiate. המקרה שתיארנו במאמר זה הוא מעניין במיוחד מכיוון שהקוד הוא של ה-Framework עובדה ההופכת את הפגיעות לחמורה בכמה סדרי גודל, מכיוון שכל אפליקציה אשר משתמשת בקוד זה יורשת את החולשה.

דברי סיום

בזאת אנחנו סוגרים את הגליון ה-48 של Digital Whisper. אנו מאוד מקווים כי נהנתם מהגליון והכי חשוב- למדתם ממנו. כמו בגליונות הקודמים, גם הפעם הושקעו הרבה מחשבה, יצירתיות, עבודה קשה ושעות שינה אבודות כדי להביא לכם את הגליון.

אנחנו מחפשים כתבים, מאיירים, עורכים ואנשים המעוניינים לעזור ולתרום לגליונות הבאים. אם אתם רוצים לעזור לנו ולהשתתף במגזין Digital Whisper - צרו קשר!

ניתן לשלוח כתבות וכל פניה אחרת דרך עמוד "צור קשר" באתר שלנו, או לשלוח אותן לדואר האלקטרוני שלנו, בכתובת editor@digitalwhisper.co.il.

על מנת לקרוא גליונות נוספים, ליצור עימנו קשר ולהצטרף לקהילה שלנו, אנא בקרו באתר המגזין:

www.DigitalWhisper.co.il

"Talkin' bout a revolution sounds like a whisper"

הגליון הבא ייצא ביום האחרון של חודש ינואר.

אפיק קסטיאל,

ניר אדר,

31.12.2013