Digital Whisper Technologic Papers

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

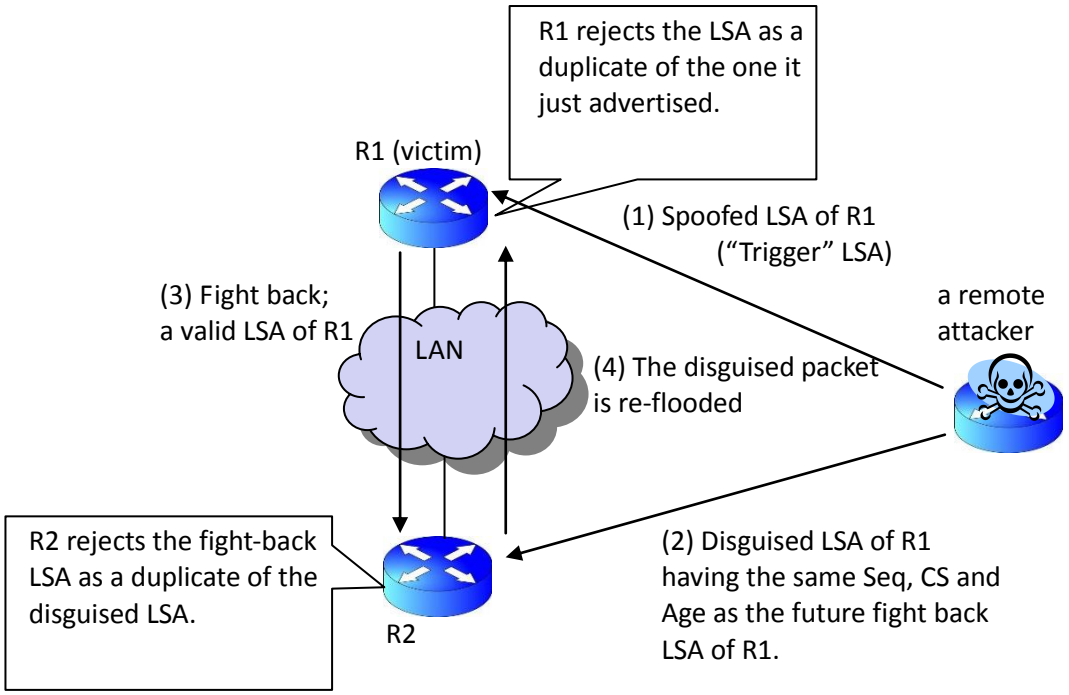
על פי RFC 2328, פרק 13.1, שני מופעים של חבילת LSA נחשבים זהים, אם מתקיימים שלושת הדברים הבאים:

- 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".

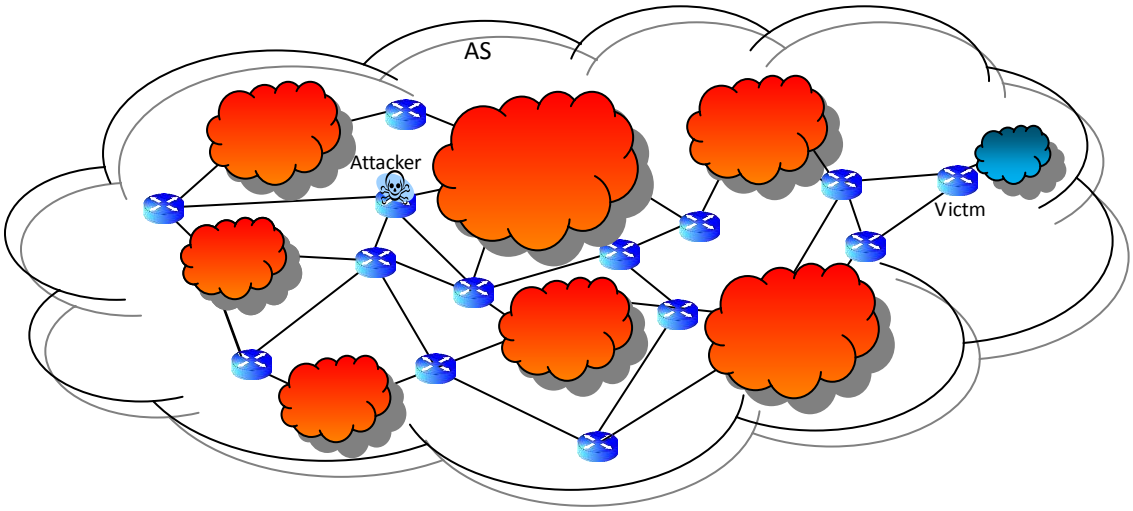
לאחר מהלך זה, ל-R1 ול-R2 יש שתי עותקים שונים של רשומות LSA אודות R1, ובשלב זה, השינוי הוא קבוע, המצב הנ"ל ישתנה, רק לאחר ש-R1 ישלח את העדכון הבא שלו לרשת (עניין של חצי שעה, אם נתחשב בערך ברירת המחדל של ה-LSA Interval).

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-Fight Back LSA תמיד נשלח עם ערך Seqeunce Number גדול באחד מערך זה ב-LSA המזויף שגרם לו, אזי ערכו של שדה ה-Sequence Number ב-Disguised LSA 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-Disguised LSA 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.

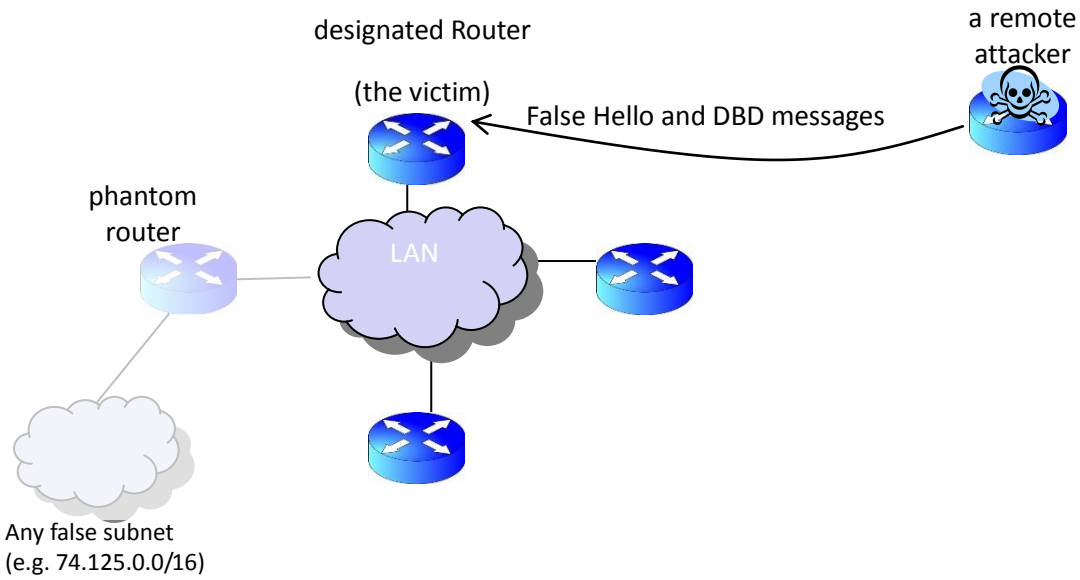
לאחר הרצת תהליך זה, התוקף מגיע למצב בו לקורבן קיימת Adjacency חוקי עם היישות הפקטיבית שהוא הקים, ובשלב זה, הקורבן יתחיל אף לפרסם קישור אליה ב-LSA שלו. פרסום ה-LSA אודות הלינק הפקטיבי הינה הנקודה המרכזית במתקפה זו, ואחת היתרונות שלה.

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-כיוונית 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:

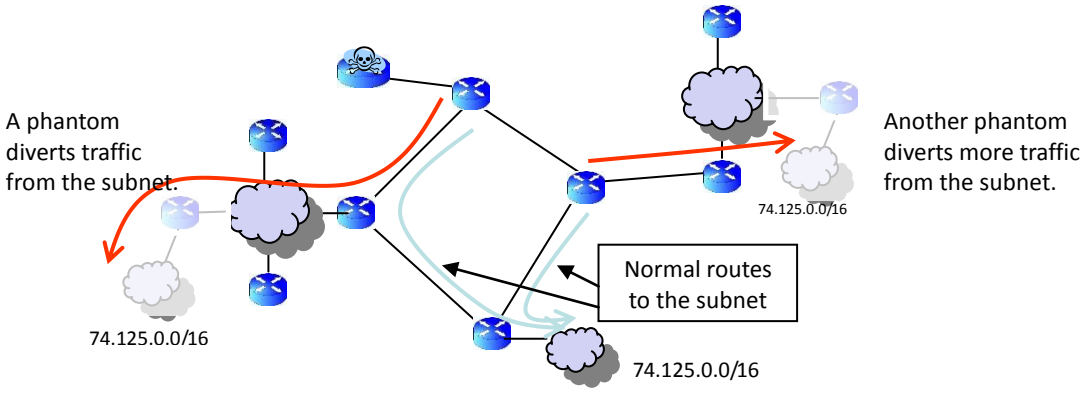
- מפתח ה-MD5 של הרשת המרוחקת (Remote LAN), ברב המקרים, המפתח הנ"ל הינו אחיד בין כלל הרשתות בתוך ב-AS.
- פרמטרי קונפיגורציות הרשת המרוחקת, כדוגמת HelloInterval, RouterDeadInterval, וכו'. וגם כאן, ברב המקרים, הפרמטרים הנ"ל הינם אחיים בין כלל הרשתות בתוך ה-AS.



[תרשים 3 - הדגמת מהלך המתקפה]

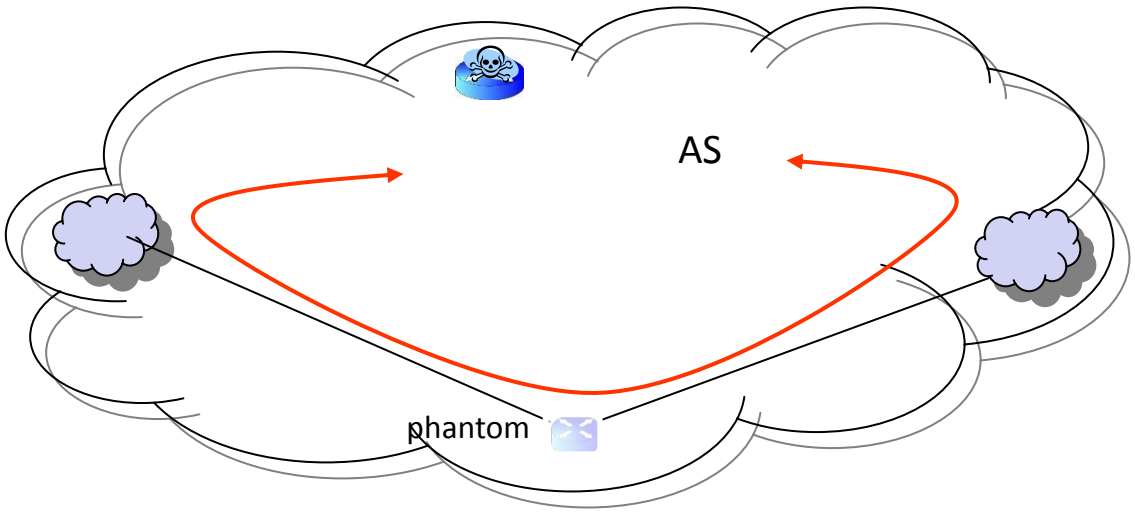
שימוש אפשרי במתקפה זו הוא לטובת יצירת Black-Hole לתעבורת המידע ברשת ספציפית, ע"י פרסום יישות פקטיבית שתוביל לאותו ה-Black-Hole. בהתחשב בעובדה שתוקף יכול ליצור נתב פקטיבי בכל מקום ברשת ניתן להבין כי תוקף יוכל ליצור Black-Hole לכל תעבודה ברשת ומכל תת-רשת ברשת.

להמחשת הרעיון, ניתן להסתכל בתרשים הבא:



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

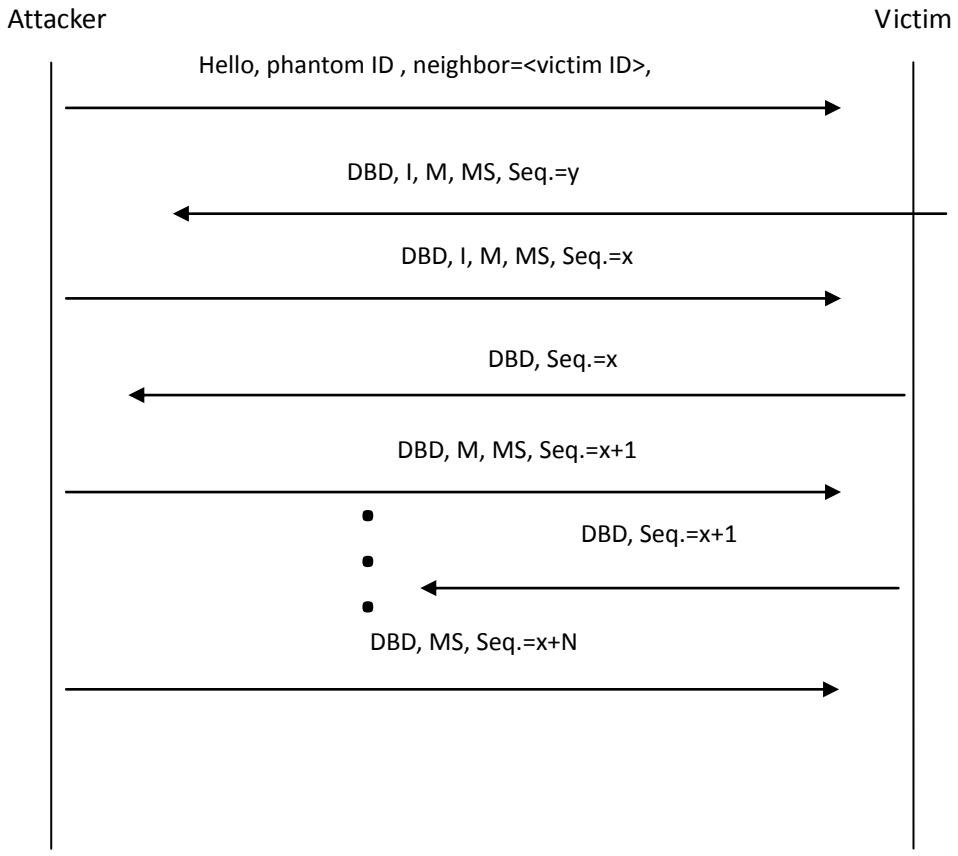
שימוש נוסף אפשרי במתקפה זו היא יצירת נתב פקטיבי ברשת במיקום "אסטרטגי" המאפשר לו להוסיף קיצור-דרך לכמות תעבורה גדולה ברשת, לדוגמא, ע"י הוספת נתב פקטיבי המגשר בין שתי רשתות רחוקות ב-AS, כפי שניתן לראות בתרשים הבא:



[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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LS age             |    Options    |    LS type    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Link State ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     LS sequence number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         LS checksum           |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- 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.

Dynamips(6): R3, Console port

```
R3#sh ip os da

            OSPF Router with ID (192.168.37.3) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link co
192.168.18.1    192.168.18.1    415         0x80000003 0x005A5A 3
192.168.27.2    192.168.27.2    419         0x80000003 0x00C942 2
192.168.37.3    192.168.37.3    417         0x80000003 0x00B72A 2
192.168.37.7    192.168.37.7    423         0x80000002 0x00F2C1 2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.12.2    192.168.27.2    420         0x80000001 0x003BFD
192.168.13.3    192.168.37.3    418         0x80000001 0x003EE2
192.168.27.7    192.168.37.7    423         0x80000001 0x000FED
192.168.37.7    192.168.37.7    423         0x80000001 0x0031B6

                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
10.0.0.0        192.168.27.2    391         0x80000001 0x003F9A 2
11.0.0.0        192.168.27.2    391         0x80000001 0x0032A6 2
11.0.0.0        192.168.37.3    391         0x80000001 0x000C25 3
192.168.11.0    192.168.18.1    461         0x80000001 0x00122D 0
192.168.24.0    192.168.27.2    465         0x80000001 0x003DEA 0
R3#sh ip os da

            OSPF Router with ID (192.168.37.3) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
192.168.18.1    192.168.18.1    159         0x80000004 0x007CBA 3
192.168.18.8    192.168.18.8    154         0x80000004 0x002504 1
192.168.27.2    192.168.27.2    812         0x80000003 0x00C942 2
192.168.37.3    192.168.27.11   13          0x80000004 0x00BC79 3
192.168.37.7    192.168.37.7    816         0x80000002 0x00F2C1 2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.12.2    192.168.27.2    813         0x80000001 0x003BFD
192.168.13.3    192.168.37.3    811         0x80000001 0x003EE2
192.168.18.1    192.168.18.1    159         0x80000001 0x004FF1
192.168.27.7    192.168.37.7    816         0x80000001 0x000FED
192.168.37.7    192.168.37.7    816         0x80000001 0x0031B6

                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
10.0.0.0        192.168.27.2    785         0x80000001 0x003F9A 2
10.0.0.0        192.168.37.3    2           0x80000001 0x001919 3
11.0.0.0        192.168.27.2    785         0x80000001 0x0032A6 2
11.0.0.0        192.168.37.3    784         0x80000001 0x000C25 3
192.168.11.0    192.168.18.1    854         0x80000001 0x00122D 0
192.168.24.0    192.168.27.2    859         0x80000001 0x003DEA 0
```

LSA DB **לפני** המתקפה

LSA DB **אחרי** המתקפה

רשומת ה-LSA המקורית. (**שימו לב כי ערך שדה ה-Link ID ו-ADV Router זהים.**)

רשומת ה-LSA הפקטיבית. (**שימו לב כי ערך שדה ה-Link ID וה-ADV Router שונים כעת. רשומת ה-LSA המקורית - שונתה בהצלחה!**).

[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.

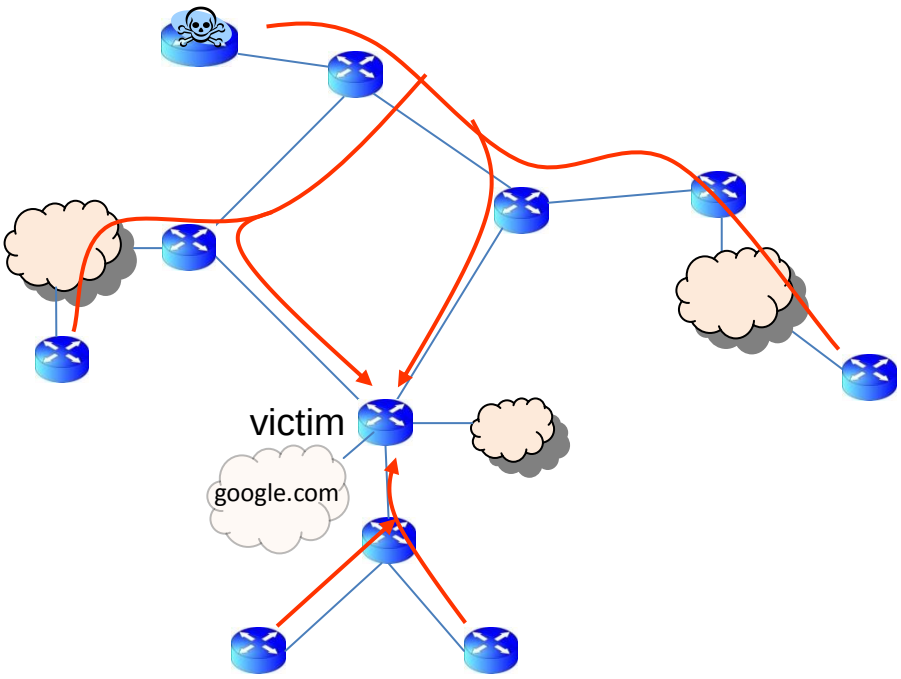
יישומים אפשריים למתקפה

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:



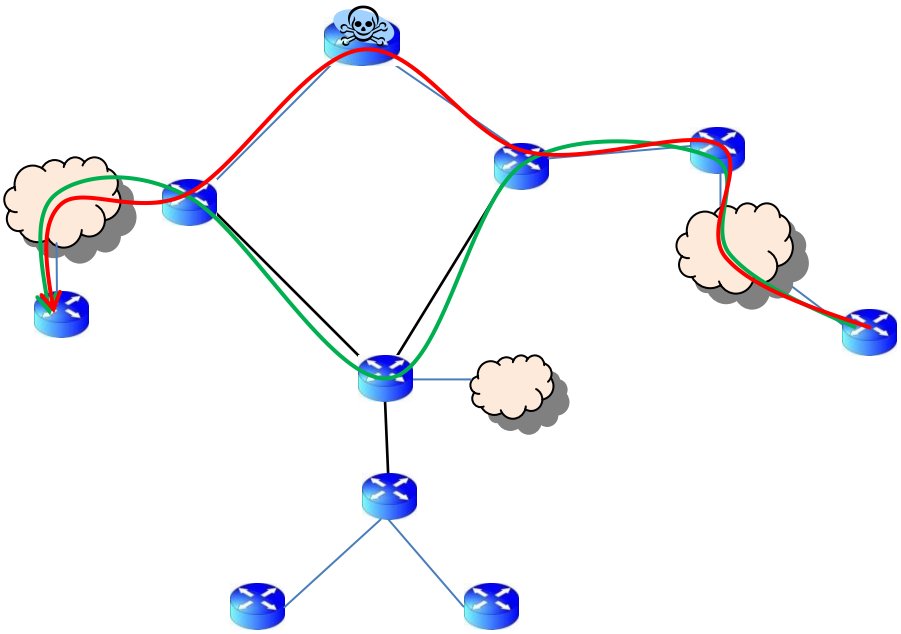
[תרשים 8 - ניתוק יעד ספציפי אשר נמצא מחוץ ל-AS]

- **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.
- **[BH13]** Gabi Nakibly, Eitan Menahem, Ariel Waizel, and Yuval Elovici "Owning the routing table - part II", Black Hat USA 2013.