מתקפות ipv6

ראשית נציין שכל מה שנכתב כאן לא מתיימר להציג את כל הידוע,

מטרתנו היא לסקור מתקפות ישנות וחדשות על הפרוטוקול, בצורה ציורית וקלה להבנה.

אנו מתנצלים אם המובא כאן מקטין את מורכבותו של הנושא, אין לנו כוונה לסכם אותו.

מסמך זה מבוסס בעיקרו על המאמר הבא:

https://www.usenix.org/system/files/conference/woot14/woot14-ullrich.pdf

על המאמר הנפלא של איגוד האינטרנט הישראלי:

https://www.isoc.org.il/wp-content/uploads/2018/06/ipv6.pdf

ועל רעיונות אישיים שעלו לנו במהלך הפרויקט.

הקדמה

בשנות ה 90 ההבנה שכתובות Ipv4 הן משאב מוגבל הגיעה, ושיש להיערך למציאות בה לא יהיו יותר כתובות Ipv4 פנויות להקצאה. מציאות זאת התרחשה אגב, בשנת 2014 כאשר IANA הקצתה לכל הארגונים תחתיה את יתרת כתובות ה Ipv4, ובזאת תמו ההקצאות ברמה העולמית (על פי המאמר הראשון זה קרה בשנת 2011). כמובן שישנם פה ושם כתובות ipv4 שלא נמצאות בשימוש, אבל לא ברמה משנת מציאות.

מסיבה זו, שנחזתה שנים ספורות קודם לכן, נולד Ipv6 בשנת 1998.

פרוטוקול זה נבדל מקודמו בעיקר בכמות הכתובות העצומה (2^128) בה הוא תומך. על פי ההארכות המדעיות הקיימות כיום, לא צפוי מחסור. על מנת להבין את המספר דמיינו את עצמכם מחלקים לכל גרגר חול על פני כדור הארץ כתובת Ipv6, ועדיין תישארו עם עודף כתובות.

מטבע הדברים, בפרוטוקול החדש (שצבר תאוצה בשנת 2010) פותחו שינויים שפותרים בעיות מ ipv4, אך גם ישנם בעיות שנותרו לא פתורות, וכמובן – ישנם בעיות אבטחה חדשות. נאמר כי אין זה אומר שהוא פחות טוב מקודמו – להפך.

על פי רוב הארכות כיום הוא מאובטח יותר, ואף מהיר יותר. (אם כי גם על זה יש מחלוקות)

לפני סקירת מתקפות ועל מנת להבין אותם בצורה מיטבית, נכיר מעט את השפעתו של ipv6 על עולם הרשת.

ארבעת השינויים העיקריים ש ipv6 מספק

- טווח כתובות המציג מספר בלתי נתפס העומד על כ 340 טריליון טריליון טריליון
- הוספת משפחה חדשה של כתובות בשם anycast, נוסף על unicast ו- multicast
- שינויים בשדות ההדר של הפרוטוקול והפשטה שלו לפחות שדות חובה (העברת חלק ל extension)
- פרגמנטציה הוגבלה לקצוות הקשר, על מנת להוריד משימות ועומסים מהנתבים שבדרך.
- היה שינוי חמישי בו שימוש ב IPsec היה חובה, אבל הוא הפך לאופציה עם התפתחות פרוטוקולים כדוגמת SSH ו SSL. https://he.wikipedia.org/wiki/IPSec

שינויים שבאו בעקבותיו

- פרוטוקול ICMP המוכר מהודעות שגיאה ואבחון התרחב ל ICMPv6
- אין יותר ARP, אותו מחליפות הודעות ICMPv6 כחלק מ NDP (חדש)
- NDP מביא איתו גם גילוי ראוטרים, כך שהוא כמעט מייתר את השימוש ב DHCP, אם כי זה האחרון כן קיים.
- כאשר לא נעבוד עם DHCP שהוא statefull, נעבוד דינאמית עם SLAAC – מנגנון stateless לקבלת כתובת כשמצטרפים לרשת. הרעיון הוא שהמחשב מג'נרט לעצמו את הכתובת. נעשה גם כאן שימוש ב NDP למען מטרה זו.

דו קיום

פרוטוקול ipv6 הוא כיום האידיאל, ועדיין הוא לא מאומץ על ידי רוב העולם כפרוטוקול המרכזי של שכבת הרשת. הסיבות לכך הן בעיקר טכניות, כגון החלפת ציוד ישן הכרוך בכסף רב, עדכוני גרסאות, והגורם האנושי – שמעדיף לעבוד עם המוכר והנוח.

לכן אימוץ השימוש בipv6 אינו מהיר, אם כי בשנים האחרונות הוא צובר תאוצה (ב 2018 למשל הוציא משרד התקשורת הממשלתי הישראלי הודעה ברורה שעל כל ספקיות התקשורת לעבור לתמיכה בפרוטוקול)

ועדיין, עד ש IPv4 ייחשב כנחלת העבר, יש לחיות עם שניהם ביחד.

קיימות מספר שיטות העוזרות לפרוטוקולים לתפקד בהרמוניה:

- tunnels:
  - 6to4
  - IPv6 rapid deployment
  - 6over4
  - ISATAP
  - Teredo

- translation:
  - כלומר המרה ממש של הדרים מ Ipv4 ל ipv6 ולהיפך. מזכיר קצת את NAT

השיטה המועדפת כיום היא שיטת הטאנלים.

בעיות אבטחה

- ההדרים המורחבים (extention headers)

- **בעבר** הייתה קיימת הרחבה בשם routing header type 0 שמאפשרת לצרף רשימת כתובות שהפקטה צריכה לבקר בהם בדרך ליעד. אבל, על ידי שינוי כתובות ניתן לגרום לפקטה להסתובב במעגלים ברשת, מה שעלול להוביל לDOS. כיוון שהתוספת הזאת הייתה יותר מזיקה ממועילה, היא **הוצאה מכלל שימוש לחלוטין**

- על מנת להקל על ראוטרים, היחידים שיש להם רשות להתעסק עם extension headers הם קצוות השיחה. היוצא מן הכלל הוא ההרחבה hop-by-hop שבה יש שדה בשם router alert option שמיועד לעדכון כלשהו בעתיד. הבעיה עם זה הייתה כאשר הרבה פקטות נשלחו, וזה כבר גרם לירידת איכות הביצועים של הראוטרים

- **בעבר** לא היה פורמט אחיד להרחבות, מה שגרם לכך שבחלק מההרחבות לא כל רכיבי הרשת יודעים להבין. **כיום** כבר יש סטנדרט להרחבות.

- פרגמנטציה

אחד האיומים הידועים שפרגמנטציה מאפשרת, היא פירוק של פקטה זדונית לפקטות קטנות יותר (פרגמנטים) וככה ניתן לעקוף בדיקות אבטחה, למשל של firewalls. למרות זאת, ipv6 לא אוסר פרגמנטציה במפורש