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

בחלקו על המאמר הנפלא של איגוד האינטרנט הישראלי:

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

על רעיונות אישיים שעלו לנו במהלך הפרויקט, ומידע שמצאנו בדרך ונמצא בביבליוגרפיה.

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 (האחרון הוא שדרוג של broadcast)
- שינויים בשדות ההדר של הפרוטוקול והפשטה שלו לפחות שדות חובה (העברת חלק ל extention)
- פרגמנטציה הוגבלה לקצוות הקשר, על מנת להוריד משימות ועומסים ובעיות אבטחה 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 https://www.gov.il/BlobFolder/rfp/16102018/he/Hearing%20IPv6%2016.10.18.pdf)

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

להלן מפת העולם המתעדכנת בלייב אודות מצב אימוץ תעבורה מעל Ipv6
https://www.akamai.com/internet-station/cyber-attacks/state-of-the-internet-report/ipv6-adoption-visualization

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 extention 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 שאת חלק 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 אסר בראשית דרכו** פרגמנטציה במפורש.

- ישנה דרך מוכרת ב ipv4 להימנע מהבעיה הנ"ל, והיא על ידי הפלה (drop) של פקטות עם גודל מסויים ב offset שלא עומס בסטנדרט. הבעיה ב Ipv6 היא שאין שום הגבלה על אותו offset, ולכן לא ניתן לחסום את איום הפרגמנטציה באותה דרך.
  וכך **היה ניתן** לחלק חבילות בצורה חכמה ולגרום לפקטה שמכילה פוגען (למשל) להגיע ליעד.

- אמנם firewalls נפוצים בכל מקרה אוספים את הפרגמנטים ומחברים מחדש על מנת לבצע את הבדיקות שלהם, או במילים אחרות – ipv6 **היה פגיע** מבחינת פרגמנטציה לאותם איומים כמו קודמו.

- **בסופו של דבר**, כיוון שעם פרגמנטציה צריך להתייחס בחשדנות יתרה על פקטות שחסרות במידע (כגון דגלים ופורטים שחשובים לבדיקות הבטיחות) והעסק עלול להסתבך אבטחתית, **כיום** הפרוטוקול נחשב כ atomic fragments, כלומר כל פקטה מורכבת מפרגמנט אחד בלבד.

- שדות החובה של הפרוטוקול

- ההדר flow label עשוי להוות בעיה כיוון שהוא יכול להאט ביצועי ראוטרים (בדומה ל router alert שהוזכר קודם), ובנוסף תוקפים עשוים לקבל מידע על איכות שידור של משתמשים על ידי שימוש באותו flow label

- NDP

ישנן השלכות רבות לשימוש ב NDP עקב ההנחות הנאיביות שלו שמסתמכות על אמון ברכיבים ברשת המקומית. כאשר תוקף יצליח לקבל גישה לרשת מקומית כלשהו – זה עלול לגרום לכמה וכמה בעיות אבטחה.

ובדומה ל ARP ובעיות הזיוף הידועות עליו, גם כעת תוקפים יכולים לזייף כתובות ip ו mac, ועוד.

- כך למשל תוקף יכול למנוע כניסתן של מחשבים אחרים לרשת על ידי מענה פיקטיבי להודעות DAD. דרך נפוצה להימנע מתקיפה זו הוא לצאת מנקודת הנחה שהכתובת שמחשב מג'נרט לעצמו היא ייחודית בכל מקרה. פתרון זה נקרא optimistic duplication address detection

- כל מחשב, ובפרט תוקף יכול להכריז על עצמו כראוטר על ידי שליחת RA

- תוקף יכול לזייף הודעות RA של ראוטרים אחרים ובכך למשל לגרום ל lifetime שלו להיות 0 וכך לגרום לשאר המחשבים ברשת שמקבלים את ההודעה "לשכוח" מזכרון המטמון את קיומו של הראוטר. באותה דרך אפשר לזייף prefix בשם הראוטר, ואפשר גם לגרום להצפת הודעות RA עם prefixes רבים שיגרמו לכל הצמתים להתעדכן כל פעם מחדש וליצור עקב כך dos attack.

- בעיות אלו לא נפתרות לגמרי עם DHCP, כי תוקף יכול לגרום לצמתים ברשת לנטוש את DHCP. לכן כאמצעי נגד קיים סנן מטעם הראוטר שמסנן מהרשת הודעות לא לגיטימיות של תוקפים. הבעיה עם זה, הוא שלפעמים נשלחות הודעות NA לגיטימיות בטעות, ולכן צריך להיזהר מאוד עם הסינון, כדי לא לפגוע בשירות.

- הודעות redirect של NDP יכולות לשמש תוקפים כדי להסיט פקטות ממסלולן (למסלול זדוני למשל)

- כחלק ממתקפות זיופי פקטות, ישנה מתקפה בשם "מתקפת דרדסים" הפועלת בצורה הבאה: תוקף שולח בקשה כלשהי ליעד multicast, כאשר מקור הבקשה זוייף להיות כתובת של קרבן כלשהו. ככה בקשה אחת של תוקף גורמת להודעות תגובה רבות שמכוונות כולן כלפי קרבן – ועשויות לגרום לו ל DOS. ניתן לדמות מתקפה זו כילד שנעמד בכיתה ומכריז בקול שילד אחר ביקש מכות. כל הכיתה תלך לילד השני ותתן לו מה שביקש. דרך טובה לבצע מתקפה זו היא על ידי זיוף פקטת ICMPv6 (פינג6)
אמנם הודעות multicast לרוב לא נענות בתגובות, אבל ישנן מימושים שבהן הן כן.

- כל הבעיות הנל עם NDP לא היו צפויות, כיוון שבזמנים בהם IPSec היה חלק אינטגרלי ב ipv6 לא היו הבעיות האלו. שצצו לאחר הפיכתו של IPSec לאופציולי.

- לכן פותח SeND (secure neighbor discovey), אבל התגלה כמסיבי מידי, (עקב ההצפנה) מה שעשוי לפתוח עוד אפשרויות ל DOS attack.

- ניתן לסכם כעת שהדרך הבטוחה ביותר למנוע את כל הבעיות האלו היא למנוע מלכתחילה מהתוקף להיכנס לרשת המקומית, באמצעות הגנה בשכבה הפיזית או בשכבת הקו.

- נסיים סעיף זה ונאמר כי הצלחנו לממש מתקפת MITM ב ipv6 באמצעות ניצול תכונותיו של NDP. https://github.com/SimchaTeich/IPv6/tree/main/MITMv6

- MLD

פרוטוקול זה אחראי לגילוי ושמירת מידע על צמתים ברשת המקומית (link) שמאזינים לכתובות multicast. ראוטרים משתמשים בו על מנת לגלות את אותם צמתים ולשמור את המידע, וגם לגלות את כתובות ה multicast בהן הם מאזינים, במטרה להעביר להם הודעות multicast. לקריאה נוספת: https://www.ibm.com/docs/en/zos/2.2.0?topic=protocol-multicast-listener-discovery

- מידי פעם שרתים מפרסמים הודעות MLD לגילוי צמתי nulticast, והצמתים מחזירים לו תגובה. תוקף עלול לעצור את העברת התעבורה ל multicast על ידי זיוף של הודעת MLD Done, וזה יחזיק עד שהראוטר יפרסם את בקשת ה MLD הבאה. ראוטרים כאלה נקראים quary routers.

- התוקף יכול להתחזות ל quary router על ידי שימוש בכתובת 1:: שהיא הכתובת השימושית הקצרה ביותר שקיימת – וזאת כי בחירת quary router היא על ידי בחירת זה עם הכתובת הכי קצרה מביניהם. לאחר התחזות מוצלחת, התוקף מפסיק את שליחת הודעות ה MLD ובכך גורם ל MLD Dos. דרך נפוצה להתגונן מפני זה היא שכאשר ה quary router הקודם רואה שאין יותר בקשות MLD כלליות, הוא לוקח את התפקיד חזרה. בגלל זה, התוקף יכול לשלוח את ההודעות הכלליות רק לראוטרים (ff02::1), מה שישביע את רצונם בזמן ששאר הצמתים ברשת הלוקאלית מתמודדים עם הפגיעה בשירות – וחזרנו ל DOS.

- דרך להתמודד עם האיום הוא להקצות את הכתובת :: לראוטר הרלוונטי.

- Tunneling

- **בעבר** לעשות טאנל של חבילה 'רעה' על גבי ipv6 היה קל, כי רוב ה firewalls נתנו לכל חבילת ipv6 לעבור. **כיום** זה כבר לא ככה, אבל נוצרו בעיות אחרות עקב טכנולוגיות התרגום של Ipv4 & ipv6

-