פרוייקט סיום רשתות תקשורת

ת.ז:
315434902, 314830704, 318845690, 213104953

קישור לקוד בגיטהאב: https://github.com/e10withadot/NET_EXF

חלק 1

שאלה 1

A user reports that their file transfer is slow, and you need to analyze the transport layer to identify the potential reasons. What factors could contribute to the slow transfer, and how would you troubleshoot it?

כאשר הבעיה היא העברה איטית של קובץ צריך להתמקד בפרוטוקול התעבורה במספר גורמים שיכולים להוות בעיה:
גורם ראשון שיכול לגרום לאיטיות הוא אם חלון קבלה קטן מדי, בגלל שאין הרבה מקום בחלון יכולות להיות "באוויר" מעט חבילות וזה מאט את התקשורת ביניהם, הפתרון הוא ללכוד בעזרת wireshark את חבילת הSYN חלק מיצירת הקשר(handshake) בTCP header ולבדוק את גודל החלון ההתחלתי והאם יש תמיכה בwindow scaling בשני הצדדים, אם window scaling מופעל אפשר להגדיל את החלון כך שישלחו יורת נתונים לפני קבלת אישור (ACK).
גורם שני אובדן מנות (packet loss) ושידור חוזר (retransmission) , אובדן של פאקטות גורם לTCP להפעיל את מנגנון retransmissions ואם יש אובדן מרובה אז גם congestion control מה שגורם להאטת התעבורה. הפתרון יהיה לנתח בwireshark אם יש חבילות מסוג duplicate ack או TCP Retransmission או Fast Retransmission, אפשר גם לבדוק אם יש בעיות יציבות ברשת בשכבה הפיזית (כבלים וכדו'), עומס בנתיב הרשת, שגיאות בנתבים או מתגים.
גורם שלישי TCP Congestion Control מנגנון בקרת עומס, אם TCP מזהה עומס הוא יקטין את קצב שליחת הפאקטות על ידי צמצום גודל הcongestion window, אפשר לאתר על ידי בדיקה בwireshark את גודל congestion window , והפתרון להשתמש ב selective acks, מעקב אחרי גודל הcongestion window דרך wireshark כדי לזהות שינויים המתרחשים במהלך תקופות עומס.
גורם רביעי זה כמות הLatency באתר, הנגרמת על ידי זיהוי של עיכוב גדול בתעבורה. בודקים את זה עם חישוב RTT על ידי ping, על ידי ניתוח לחיצת יד משולשת, בודקים שינויים או בעיות במסלול על ידי traceroute, כדי לראות אם יש נקודה מסוימת שיש בה עיכובים גדולים פתרון לעשות אופטימיזציה לTCP Window Scaling, או שימוש בפרוטוקלים שמיועדים להאצת התעבורה כמו MPTCP.
גורם חמישי בעיות של אינטרנט איטי, איתור על ידי בדיקת רוחב הפס, פתרון על ידי חיבור לכבל אינטרנט אם לא מחובר או לשדרג תשתית אינטרנט.

שאלה 2

Analyze the effects of TCP's flow control mechanism on data transmission. How would it impact performance when the sender has significantly higher processing power than the receiver?

בקרת זרימה בTCP מונעת מצב בו שולח מהיר שולח יותר נתונים ממה שהמקבל מסוגל לעבד, זה נעשה באמצעות גודל חלון שמגדיר כמה נתונים אפשר לשלוח לפני שמקבלים אישור.
כאשר השולח משמעותית מהיר יותר מהמקבל זה יוביל לחלון קטן שיגרום לשולח להמתין הרבה עם העברת החבילות, מה שיוביל להאטה בתעבורה.יכול להוביל למצב של TCP Zero Window אם הזיכרון של המקבל מתמלא, הוא שולח הודעת **TCP Zero Window**, שמורה לשולח לעצור שליחה לחלוטין, השולח ימתין להודעת **TCP Window Update** כדי להמשיך, מה שעלול לגרום להשהיות משמעותיות. חלון קטן או עיכוב בשליחת ACK גורם לעיכובים בגלל הצורך להמתין לאישורים לפני שליחה. אם החלון קטן אפילו שהרשת מהירה ויש רוחב פס גדול עדיין התעבורה תהיה איטית מהאפשרי.

שאלה 3

Analyze the role of routing in a network where multiple paths exist between the source and destination. How does the path choice affect network performance, and what factors should be considered in routing decisions?

ברשת בה קיימים מספר מסלולים בין המקור ליעד, כשנרצה לבחור מסלול אופטימלי לניתוב, בחירת המסלול עשויה להשפיע באופן הבא: נרצה לבחור מסלול קצר ככל הניתן בעל מספר תחנות ביניים מינימלי משום שמסלול קצר יותר מפחית את זמן ההשעייה בדרך כלל. כמו כן נעדיף מסלול פחות עמוס, מכיוון שזמן ההמתנה לכל חבילה גדל ככל שהמסלול עמוס יותר, דבר שעשוי להוביל לשינויים בזמן ההגעה. בנוסף בדרך כלל נעדיף מסלול בעל רוחב פס גבוה יותר שיאפשר העברת נתונים מהירה יותר. נשים לב שעומס כבד על נתיב מסוים עשוי לגרום לצוואר בקבוק ולכן פרוטוקולי ניתוב חכמים מנתבים מחדש כדי למנוע האטות, לדוגמא אלגוריתם ECMP מחלק את התנועה בין מספר מסלולים במקביל ו-MULTIPLE ROUTING מבטיח שהנתונים ימשיכו לזרום גם במקרה של כשל במסלול אחד. כמו כן במידה וחלון הקבלה של המקבל קטן מדי, השולח חייב להמתין לפני שליחת מנות נוספות, מה שגורם לניצול לא אופטימלי של הרשת פרוטוקול TCP מאפשר להגדיל את החלון ולנצל בצורה טובה יותר את רוחב הפס.

כשנקבל החלטה על ניתוב נצטרך להתחשב בגורמים הבאים: לרוב נרצה לקחת מסלול קצר יותר ככל הניתן, ובעל רוחב פס גבוה ככל היותר אך לעיתים במסלול הקצר יהיו עומסים, לכן נרצה להתחשב גם בזה. בנוסף נעדיף מסלול יציב בו תנאי הרשת נשארים עיקביים לאורך זמן ואינם מושפעים משינויים כמו עומסים קיצוניים או תקלות ברשת. מסלול יציב יאפשר רמת אובדן חבילות נמוכה, רוחב פס יציב, מיעוט תקלות, ושמירה על סדר קבלת החבילות. בנוסף עבור תעבורה שאינה קריטית עבורה אין חשיבות לזמן הגעה מדויק (כמו קבלת מסרון או מייל) נוכל להסתפק במסלול פשוט וזול, אך עבור שירותים רגישים להשהייה (כמו שיחות וידאו) נעדיף מסלול יקר ויציב. כמו כן נרצה להתחשב בשיקולי אבטחה שונים בבחירת מסלול. נשתמש בפרוטוקול OSPF לחישוב המסלול הקצר ביותר, בפרוטוקול BGP עבור המסלול הטוב ביותר בין ארגונים שונים וב-QoS כשנרצה להתאים עבור כל תעבורה את היישום הרלוונטי עבורה.

שאלה 4

How does MPTCP (Multi-path TCP) improve network performance?

MPTCP מרחיב את פרוטוקול TCP הרגיל בכך שהוא מאפשר לחיבור יחיד לנצל בו זמנית מספר רשתות, בהמצאת יצירת מספר "תת חיבורים" לכל נתיב זמין, כך שבמקום לשלוח חבילות דרך נתיב אחד, MPTCP שולח דרך מספר נתיבים, עבור כל חבילה את הנתיב המהיר הפנוי ביותר, כך שבסוף החבילות מאוחדות מחדש אצל המקבל. הפרוטוקול משפר את ביצועי התעבורה ברשת באופן הבא: מאפשר מהירות גבוהה יותר, מכיוון שהוא משתמש במספר מקורות במקביל. עמיד לתקלות- אם אחד החיבורים נופל באופן פתאומי MPTCP מחליף אוטומטית לנתיב חלופי אחר. מאפשר לעבור באופן חלק בין רשתות, לדוגמא כאשר משתמש מחובר לWIFI ויוצא מהבית, MPTCP יחבר באופן אוטומתי את המתשמש לרשת סלולרית. בנוסף הפרוטוקול מנטר עומסים ברשת בזמן הפעלתו ובכך מאפשר באופן דינמי לפעול בצורה המהירה והאופטימלית ביותר כדי לשמור על רציפות בתעבורה.

שאלה 5

You are monitoring network traffic and notice high packet loss between two routers. Analyze the potential causes for packet loss at the Network and Transport Layers and recommend steps to resolve the issue.

סיבות אפשריות לאובדן החבילות:

בשכבת הרשת:

- עומס ברשת - יותר מדי חבילות מגיעות לנתב והוא לא מספיק לעבד אותן, אז חלק מהן נזרקות כי אין לו מספיק מקום בזיכרון. זה כמו תור ארוך בסופר – אם אין מספיק קופות פתוחות, אנשים מתחילים לוותר וללכת הביתה.
- יכול להיות שהנתב לא מוגדר נכון, והוא ישלח חבילות לכתובות הלא נכונות, ואז הן פשוט ילכו לאיבוד. זה יכול לקרות גם כי פרוטוקולים של הניתוב לא מעודכנים כמו שצריך.
- עומס ביציאה - אם הנתב שולח חבילות לאט יותר ממה שהוא מקבל, נוצר פקק של חבילות שמחכות בתור, ובסוף חלק מהן נמחקות כי אין מספיק מקום לאחסן אותן.

בשכבת התעבורה:

- בקרת עומס ב-TCP - TCP מתוכנן לזהות עומסים ולצמצם את קצב השליחה כדי למנוע עומס נוסף. זה טוב, אבל אם הוא מצמצם יותר מדי, זה יכול לגרום לאובדן חבילות נוספות.
- בעיות בשידור חוזר - כשחבילה הולכת לאיבוד, TCP שולח אותה שוב, אבל אם יש עומס גדול, זה עלול רק להחמיר את הבעיה כי הוא מציף את הרשת בניסיונות חוזרים לשלוח את אותה חבילה.

פתרונות:

בשכבת הרשת:

1. ניהול עומסים חכם עם שימוש בשיטה שמזהה עומס מראש וזורקת חבילות מסוימות לפני שהזיכרון של הנתב מתמלא לגמרי.
2. לבדוק שטבלאות הניתוב מעודכנות ושהפרוטוקולים פועלים כמו שצריך כדי למנוע הפניות שגויות.

3. הגדלת הקיבולת- אם הנתבים נתקעים כי הקישורים שלהם איטיים מדי, אפשר לשדרג את התשתית, להוסיף רוחב פס או קווים נוספים.

בשכבת התעבורה:

1. אפשר לשנות את גודל החלון של TCP כדי לשפר את הזרימה ולמנוע אובדן חבילות כשיש עומס.
2. במקום TCP במקרים מסוימים עדיף להשתמש ב-UDP, כי הוא לא מחכה לאישורי קבלה (ACK) ולכן נמנע מעומסים מיותרים.
3. שיפור ניהול העומסים ב-TCP עם מנגנונים שעוזרים לשלוח מחדש רק את החבילות שנאבדו, במקום להציף את הרשת בשידורים מיותרים.

חלק 2

מאמר 1:

FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition

התרומה המרכזית של המאמר-

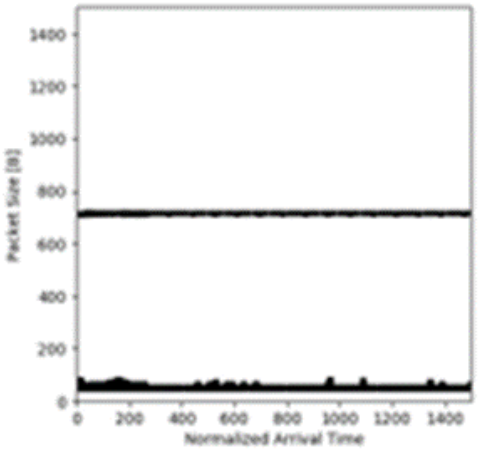
זיהוי סוג התעבורה באינטרנט משמעותי ופופולרי במיוחד בשנים האחרונות בשימוש של הנדסת תעבורה, אכיפת חוק וזיהוי תוכנות זדוניות. רוב הטכניקות הפועלות לזיהוי תעבורה מסתמכות על חילוץ תכונות סטטיסטיות מזרימת התעבורה, לאחר מכן ביצוע תהליך של סינון תכונות ולבסוף ביצוע באמצעות אלגוריתמים פשוטים. בשנים האחרונות, התקדמות בלמידה עמוקה הובילה להתקדמות רבה בתחומים רבים. רשתות נוירונים קונבולוציוניות (CNNs) הביאו במיוחד לפריצות דרך בתחום סיווג התמונה ובפריצת דרך ביישומים רפואים והנדסיים. השיטה flowpic פועלת באופן הבא: במקום להסתמך על תוכן החבילה, היא ממירה את נתוני הזרימה לתמונה ומסווגת אותם באמצעות רשת נורונית קונבולוצית (CNNs) עד עתה סיווג התעבורה התבצע באמצעות 3 קטגוריות עיקריות: שיטות מבוססות מטען (DPI) – מנתחות את תוכן החבילות, אך סובלות מבעיות פרטיות, דרישות חישוביות גבוהות, וחוסר יכולת להתמודד עם הצפנה. שיטות מבוססות יציאות – משתמשות בשדות כותרת החבילה TCP/UDP אך הפכו כיום לפחות יעילות בשל יציאות דינמיות המאפיינות רבות מהאפליקציות היום וכן שימוש ביציאות ברירת מחדל של שרתים רבים. שיטות מבוססות סטטיסטיקה ולמידת מכונה – מתבססות על חילוץ ידני של תכונות (כגון גודל חבילות וזמני הגעה) ולמידה מפוקחת לסיווג תעבורה. לעומתן folwpic מבצע למידת מכונה באמצעות מידע סטטיסטי אך במקום חילוץ ידני הוא ממיר את הנתונים לתמונה ומפעיל עליה CNN. החדשנות בשיטה זו היא שכך מתאפשר שיטה שאינה פולשנית כי לא נדרשת בדיקת עומק על תוכן החבילות, אלא רק מתבסס על מטא-נתונים כגון גודל החבילה וזמן הגעתה. בנוסף למידה באמצעות מכונה מאפשר דיוק וגמישות לעומת בדיקה ידנית שלעיתים קרובות מורכבת יותר וספיפית יותר. השיטה משיגה דיוק גבוה גם כאשר הנתונים מוצפנים, זהו ייתרון גדול משום שרוב השיטות אינן מצליחות להתמודד עם הצפנה בעוד flowpic מזהה תבניות גם ללא גישה תוכן החבילה.

אילו מאפייני תעבורה המאמר משתמש ואילו מהן הן חדשים?

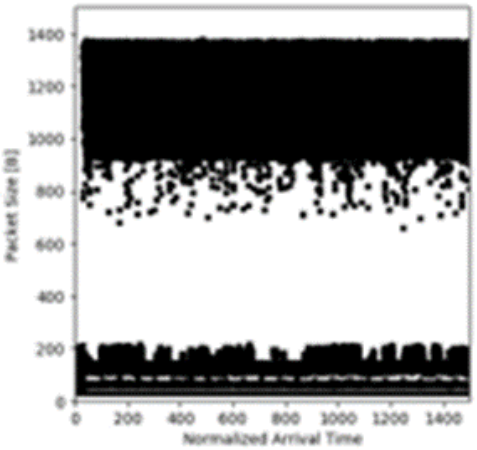
בניגוד לשיטות הנפוצות, flowpic אינה משתמשת במידע ישיר על תוכן החבילות אלא במטא-נתונים הכוללים – גודל החבילות- גודל כל חבילה ממופה לערך בתמונה לפי שכיחות הופעתו, זה מאפשר זיהוי אפליקציות שונות משום שלשירותים שונים יש דפוסי פעולה אופיניים של חבילות למשל, סטרימינג שולח חבילות גדולות בקצב קבוע, בעוד צ'אט VoIP שולח חבילות קטנות ותכופות. כמו כן flowpic מודדת מרווחי זמן בין הגעת החבילות באמצעות רישום של זמן הגעה לכל חבילה, שירותים כמו וידאו וצ'אט קוליים מייצרים זרם נתונים קבוע עם מרווחי זמן יציבים, בעוד שתעבורה אחרת (כמו דפדפן) יכולה להיות בעלת פיקים משתנים. בנוסף המאמר מתמקד רק בזרימה חד כיוונית, כלומר מנתח את כיוון התנועה רק מנקודת מבט אחת (client\server), מה שמאפשר מהירות מכיוון שאפשר לסווג כבר מהחבילות הראשונות שיוצאות ולא לחכות לתגובה מצד שני. כמו כן, כשמשתמים בפרוטוקולים מוצפנים ייתכן שהתשובות מגיעות ממקורות שונים ובתזמון לא יציב, גישה חד כיוונית שומרת על יציבות הסיווג ומונעת בלבול במקרין אלו. החידוש המרכזי של flowpic הוא בכך שהוא מתרגם את הנתונים לייצוג ויזואלי במקום להזין את הנתונים כמספרים ישירות לרשת נוירונים, מבצעים המרה של המידע לתמונה בה הציר האופקי מייצג את גודל החבילה, הציר האנכי את גודל החבילה, והבהירות של הפיקסלים את שכיחות החבילות, ולאחר מכן מבצעים ניתוח באמצעות רשת CNN.

מהם הממצאים המרכזיים, ואילו תובנות עולות מהם?

המאמר מראה כי השיטה מצליחה לסווג תעבורה ברמת דיוק גבוהה, בעיקר כשאין הצפנה ומספקת אלטרנטיבה לשיטות אחרות שבוצעו עד כה.



(a) Netflix



(b) Skype

התמונה ממחישה אבחון גרפי של flowpic , כל תמונה מציגה את התפלגות גודל החבילות לאורך זמן. וההתמקדות אינה בתוכן החבילות אלא במאפיינים כמותיים באופן הבא- הציר האופקי מסמן את זמן הגעת החבילות והציר האנכי את גודל החבילה, ניתן לראות בבירור את ההבדלים בין סקייפ לנטפליקס, בסקייפ החבילות מפוזרות במיקומים משתנים, כלומר הגדלים שלהם משתנים משמעותית, ניתן להסיק שזה נובע מכך שכאשר המשתמש מדבר נשלחות חבילות גדולות יותר וכאשר יש שתיקה נשלחות חבילות קטנות או בכלל לא, נובע מהעובדה ש-VoIP הוא שירות אינטראקטיבי, לכן כנראה משתמש בפרוטוקול כגון UDP שמעדיף מהירות על פני אמינות. לעומת זאת בנטפליקס החבילות נשלחות בגודל כמעט קבוע באופן רציף משום שנטפליקס שולחת וידאו באיכות קבועה עם מנגנון באפרינג, כנראה משתמש בפרוטוקול TCP שמתעדף אמינות ומספק מנגנון טיפול בשגיאות שמתאים לשימוש בנטפליקס.

Problem	FlowPic Acc. (%)	Best Previous Result	Remark
Non-VPN Traffic Categorization	85.0	84.0 % Pr., Gil *et al.* [15]	Different categories. [15] used unbalanced dataset
VPN Traffic Categorization	98.4	98.6 % Acc., Wang *et al.* [7]	[7] Classify raw packets data. Not including browsing category
Tor Traffic Categorization	67.8	84.3 % Pr., Gil *et al.* [15]	Different categories. [15] used unbalanced dataset
Non-VPN Class vs. All	97.0 (Average)	No previous results	
VPN Class vs. All	99.7 (Average)	No previous results	
Tor Class vs. All	85.7 (Average)	No previous results	
Encryption Techniques	88.4	99. % Acc., Wang *et al.* [7]	[7] Classify raw packets data, not including Tor category
Applications Identification	99.7	93.9 % Acc., Yamansavascilar *et al.* [10]	Different classes

הטבלה מספקת השוואה בין דיוק FlowPic לבין מחקרים קודמים, הטבלה מחולקת ל3 חלקים- 1. סייווג שיטת התעבורה, כלומר לנסות להתאים שיטה אחת נכונה מבין האפשרויות. ניתן לראות שבחלק זה flowpic מצליחה לזהות vpn בהצלחה יתרה, אך מתקשה עם tor ככל הנראה מכיוון שtor מתוכנן להסתיר מידע על התעבורה ומשנה את גודל החבילות כך שלא ייראו דפוסים קבועים.

2. שיטת one-vs-all נותנת תוצאה בינארית עבור כל דגימה האם התעבורה שייכת לשיטה מסוימת. גם כאן עבור vpn השיטה מציגה נתונים מדויקים, עבור tor עדיין מתקשה אך מציגה נתונים טובים יותר, One-vs-All מספק ביצועים טובים יותר מסיווג רגיל, כי הוא מתמקד במקרה אחד בכל פעם.

3. סיווג לפי שיטות ההצפנה וזיהוי יישומים- מתמקד בזיהוי הצפנה או האפליקציה שמייצרת את התעבורה ולא רק בסיווגה הכללי. ניתן לראות כי הזיהוי יישומים מדויק מאוד, אך זיהוי ההצפנה פחות מדויק, מכיוון שהמאפיינים שהשיטה מפענחת דומים מאוד בין סוגים שונים של הצפנות.

מאמר 2:

Early Traffic Classification With Encrypted ClientHello: A Multi-Country Study

מה התרומה העיקרית של המאמר?

המאמר מציג שיטה חדשה לזיהוי סוגי תנועה ברשת בצורה מהירה ומדויקת, גם כאשר חלק מהמידע הרגיל שמשמש לסיווג מוסתר על ידי הצפנה. הוא מציע אלגוריתם חדש בשם hRFTC, שמצליח לסווג זרמי תנועה על בסיס נתונים לא מוצפנים של פרוטוקול TLS יחד עם תכונות סטטיסטיות של זרם התנועה. הוא מציג איסוף מערך נתונים גדול ומגוון ממדינות שונות (אמריקה, אירופה ואסיה), עם הוכחה שאלגוריתמים ישנים שמסתמכים רק על נתוני TLS, לא מספיקים כאשר ה-SNI מוצפן עם Encrypted ClientHello – ECH ומוכיח שהאלגוריתם החדש משיג F-score של 94.6%, לעומת רק 38.4% של שיטות ישנות.

אילו מאפייני תעבורה המאמר משתמש בהם, ואילו מהם הם חדשים?

המאמר משתמש בשני סוגי מאפייני תעבורה עיקריים: מאפייני TLS לא מוצפנים – פרטי החיבור שנותרו גלויים בפרוטוקול TLS, כמו Cipher Suite, Key Share Group, סדר ההרחבות, ואורך הנתונים בהודעת ClientHello. ובמאפייני הזרם הסטטיסטיים החדשים – גודל החבילות (Packet Sizes – PS), זמני הגעה בין חבילות (Inter-Packet Times – IPT), מספר החבילות בכל כיוון (בהורדה/בהעלאה) והתפלגות גדלי חבילות (Histogram-based PS).

מהם הממצאים המרכזיים (כולל תרשימים), ואילו תובנות עולות מהם?

ממצאים:

- האלגוריתמים RB-RF, MATEC, BGRUA השיגו F-score נמוך (כ-40%) מכיוון שהגדרות TLS דומות בין שירותים שונים תחת ECH. (מופיע בתחילת המאמר)
- hRFTC השיג שיפור של 56.3% לעומת RB-RF, שיפור של 63.3% לעומת MATEC, ושיפור של 59.6% לעומת BGRUA. (טבלה 11 במאמר)

TABLE 11. Full dataset per class F-score for different classifiers.

Class	F-score [%]						
	Hybrid Classifiers			Flow-based Classifier	Packet-based Classifiers		
	hRFTC [proposed]	**UW** [35]	**hC4.5** [34]	**CESNET** [63]	**RB-RF** [24]	**MATEC** [33]	**BGRUA** [32]
BA-AppleMusic	**92.1**	89.5	80.2	89.2	25.5	13.1	14.5
BA-SoundCloud	**99.6**	98.9	97.8	98.7	84.4	81.8	82.0
BA-Spotify	**93.6**	90.8	89.0	88.5	16.3	0.0	3.6
BA-VkMusic	**95.7**	89.7	88.5	91.8	2.6	2.1	3.2
BA-YandexMusic	**98.5**	93.2	93.7	92.5	1.8	0.2	0.1
LV-Facebook	**100.0**	99.7	99.8	99.8	**100.0**	**100.0**	**100.0**
LV-YouTube	**100.0**	**100.0**	99.9	**100.0**	**100.0**	99.0	98.4
SBV-Instagram	**89.7**	74.7	76.5	78.8	10.0	6.3	6.4
SBV-TikTok	**93.3**	81.8	81.8	76.3	38.3	34.3	34.5
SBV-VkClips	**95.7**	94.0	91.3	92.4	53.2	37.7	46.0
SBV-YouTube	**98.2**	96.6	94.7	96.4	1.1	0.2	0.2
BV-Facebook	**87.7**	78.2	79.7	77.6	5.6	3.2	3.8
BV-Kinopoisk	**94.1**	84.1	85.8	89.8	5.4	4.0	4.1
BV-Netflix	**98.5**	97.2	95.2	93.7	50.7	52.3	56.1
BV-PrimeVideo	**91.3**	86.7	84.1	84.7	32.5	24.7	26.8
BV-Vimeo	**94.8**	90.5	90.2	81.4	72.0	19.5	68.6
BV-VkVideo	**88.6**	80.5	80.4	79.7	10.5	0.0	0.1
BV-YouTube	**85.9**	84.3	77.0	78.5	22.3	19.6	20.2
Web (known)	**99.7**	99.5	99.4	99.4	98.0	98.0	98.0
Macro-F-score (average)	**94.6**	89.9	88.7	88.9	38.4	31.4	35.1

LV is Live Video, (S)BV is (Short) Buffered Video, and BA is Buffered Audio.

- כאשר האלגוריתם אומן על מדינה אחת ונבדק על מדינה אחרת, הדיוק ירד בכ-30%. (טבלה 14 במאמר)

TABLE 14. TC quality depending on training locations.

Test Country	Share in Dataset	Training Country	Classifier Macro F-score [%]		
			hRFTC	hC4.5	UW
Germany	18.8%	Others	38.4	26.9	19.5
Kazakhstan	3.0%	Others	57.3	32.3	27.5
Russia	29.2%	Others	49.8	35.6	20.9
Spain	16.3%	Others	38.5	34.4	12.6
Turkey	25.2%	Others	35.1	26.0	16.4
USA	7.5%	Others	49.2	41.4	21.3

תובנות מרכזיות:

1. מאפייני TLS בלבד אינם מספקים דיוק גבוה תחת ECH, ולכן יש לשלב מאפייני זרם סטטיסטיים.
2. hRFTC משיג תוצאות טובות יותר מכל האלגוריתמים הקיימים, עם 94.6% דיוק, בזכות שילוב גישות שונות.
3. יש לאמן מחדש את האלגוריתם במדינות שונות, משום שתבניות תעבורה תלויות בגורמים מקומיים כמו ספקי אינטרנט.

מאמר 3:

Analyzing HTTPS encrypted traffic to IDE

1. התרומה המרכזית של המאמר:

בהצגת שיטה שמאפשרת לתוקף פסיבי לזהות במדויק את מערכת ההפעלה, הדפדפן והאפליקציה של המשתמש רק מנתוני תעבורה של https באמצעות טכניקות למידת מכונה:

1. הצעת שיטת חילוץ תכונות ומאפיינים חדשנית החורגת מסטטיסטיקות תנועה מסורתיות.
2. שילוב בין "מאפייני בסיס" (המשמשים באופן נרחב בסיווג תעבורה) לבין קבוצת מאפיינים חדשה הלוכדת את ההתנהגות של SSL/TLS ואת האופי המתפרץ של תעבורת דפדפן.
3. השגת דיוק גבוהה של 96.06% על סך הנתונים הכולל למעלה מ-20,000 תהליכים מתויגים.

מאמר זה משמעותי מכיוון שהוא מוכיח שגם עם התעבורה מוצפנת, תוקף פסיבי עדיין יכול להסיק פרטים רגישים על המערכת והפעילויות של המשתמש.

2. תכונות התעבורה- בסיסיות וחדשות:

המחקר משתמש בשני סוגי תכונות:

<u>תכונות בסיסיות:</u> סטטיסטיקות על חבילות: מספר החבילות, סך הבייטים בזרם הכניסה והיציאה.

<u>סטטיסטיקות על זמני הגעה:</u> מינימום, מקסימום, ממוצע וסטיית תקן לזמני ההפרדה בין החבילות.

<u>סטטיסטיקות על גודל החבילה:</u> מינימום, מקסימום, ממוצע ושונות.

<u>תכונות TCP ספציפיות:</u> גודל חלון התחלתי, והרחבת חלון (Window Scaling).

<u>תכונות חדשות (novel features):</u>

1. תכונות SSL/TLS: שיטות דחיסה של SSL, מספר ההרחבות של SSL, שיטות הצפנה של SSL , אורך מזהה התהליך של SSL.
2. מאפייני התנהגות מתפרצת של דפדפנים: מספר ההתפרצויות בתעבורה קדימה ואחורה - דפדפנים נוטים להציג התפרצויות של תעבורה עקב טעינה מקבילה של רכיבי דף אינטרנט , קצב שיא (מינימום, מקסימום, ממוצע וסטיית תקן) בכל כיוון: מסייע באפיון התנהגות הדפדפן בעת שליפת דפי אינטרנט, סטטיסטיקות זמן הגעה בין התפרצויות: מודדות את התזמון בין פרצי נתונים כדי לזהות דפוסים ייחודיים לדפדפן וליישום. מספר מנות keep-alive: דפדפנים ויישומים מסוימים שולחים מנות keep-alive בקצבים שונים. גודל מקטע מקסימלי (MSS) ב-TCP: ערכי MSS עשויים להשתנות בין מערכות הפעלה ופרוטוקולי רשת, דבר המסייע בסיווג.

3. התוצאות המרכזיות והמסקנות:

אחוזי דיוק:

1. שימוש רק במאפייני הבסיס: ~93.51%.
2. שימוש רק במאפיינים החדשים: דיוק דומה למאפייני הבסיס.
3. שילוב של מאפייני הבסיס והמאפיינים החדשים: דיוק של **96.06%** עבור הצירוף ⟨מערכת הפעלה, דפדפן, יישום⟩.

תוצאות מפורטות:

1. סיווג מערכת הפעלה: דיוק מושלם.
2. סיווג דפדפן: כמעט דיוק מושלם.
3. סיווג יישום: היו מספר טעויות סיווג, בעיקר בתוויות שסומנו כ"לא ידועות".

סקיצות:

סקיצה 1 – סטטיסטיקות dataset:

(a) סטטיסטיקות הdataset: מציג את התפלגות 30 הקטגוריות השונות (שילובי מערכת הפעלה, דפדפן ויישום).

(b) סטטיסטיקות מערכת הפעלה: התפלגות מערכות ההפעלה (Windows, OSX, Ubuntu).

(c) סטטיסטיקות דפדפנים: התפלגות הדפדפנים כגון Chrome, Firefox, Internet Explorer, Safari.

(d) סטטיסטיקות יישומים: התפלגות היישומים, כולל Twitter, YouTube, Facebook, לצד סוגי תעבורה רקעית.

סקיצה 2- תוצאות דיוק:

גרפים להשוואה המראים דיוק בשימוש:

במאפייני בסיס בלבד: ~93.52%.

במאפיינים החדשים בלבד: ביצועים דומים למאפייני הבסיס.

בשילוב המאפיינים: הביצועים הטובים ביותר עם דיוק של **96.06%**.

סקיצה 3 – דוגמה להתנהגות מתפרצת:

מציג גרף על ציר הזמן המדגים כיצד התפרצויות בתעבורה בולטות, מאפיין זה ממחיש כיצד המאפיינים החדשים משקפים את ההתנהגות של תעבורה שנוצרת על ידי דפדפנים.

סקיצה 4 – Confusion matrices:

מטריצת בלבול של הצירופים: רוב טעויות הסיווג הן מינוריות ומתרחשות בין צירופים דומים.

מטריצת בלבול של מערכת ההפעלה: מראה דיוק כמעט מושלם.

מטריצות בלבול של דפדפן ויישום: בעוד שסיווג דפדפנים כמעט מושלם, ישנן מספר טעויות בסיווג יישומים, בעיקר בגלל תוויות שסומנו כ"לא ידועות".

לסיכום ,התובנות:

1. **יש היתכנות של התקפות פאסיביות, כלומר גם מתוקף פסיבי:** למרות הצפנת HTTPS, תוקף מאזין יכול לנצל דפוסים סטטיסטיים בתעבורה כדי להסיק באופן מהימן את מערכת ההפעלה, הדפדפן והיישום של המשתמש.
2. **ערך המאפיינים החדשים:** שילוב התנהגות SSL/TLS והתפרצותיות במערך המאפיינים משפר באופן משמעותי את הדיוק.
3. **השלכות על פרטיות:** הממצאים מצביעים על סיכון פרטיות פוטנציאלי, שבו מעקב פאסיבי עשוי לחשוף מידע מפורט על המערכת והפעילות המקוונת של המשתמש, מה שעלול לשמש לתקיפות ממוקדות או מעקב.

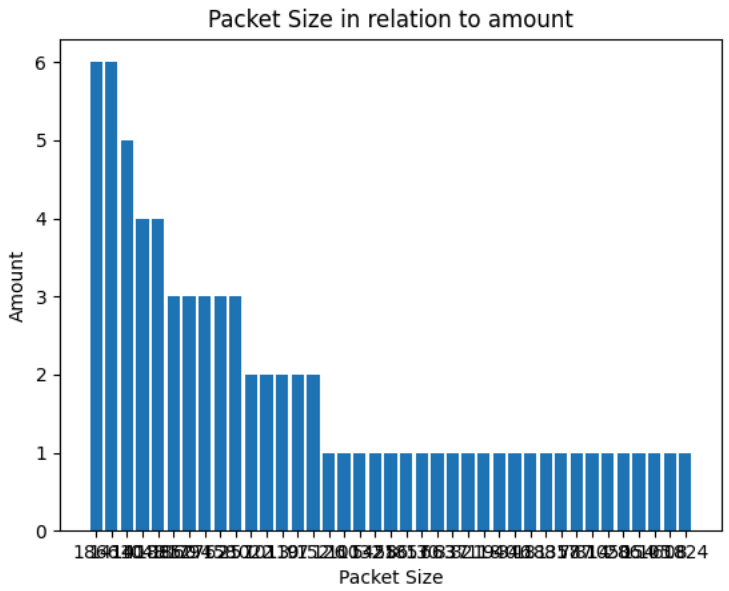
חלק 3:

שאלות 1+2+3

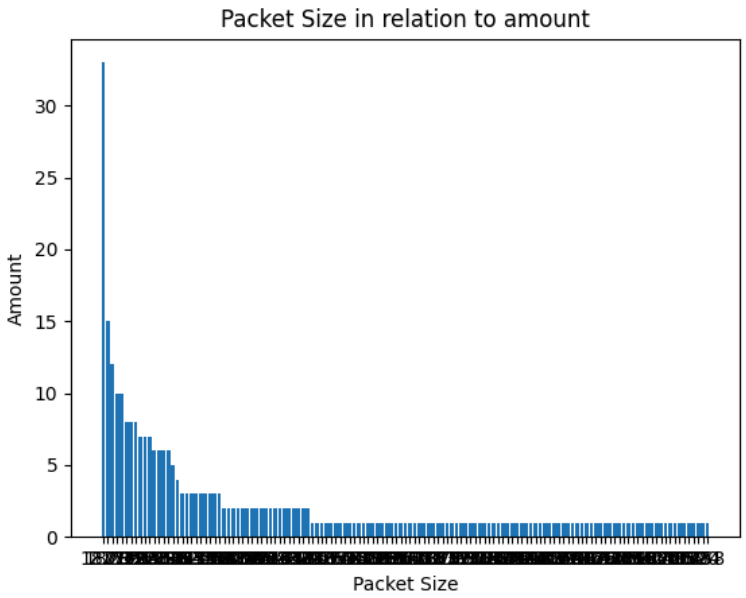
הרצנו 5 בדיקות על 5 מקרים שונים: Google Chrome בלבד על ויקיפדיה, Microsoft Edge בלבד על ויקפדיה, ספוטפיי, יוטיוב וגוגל מיט. להלן הגרפים והניתוחים:

Packet Sizes

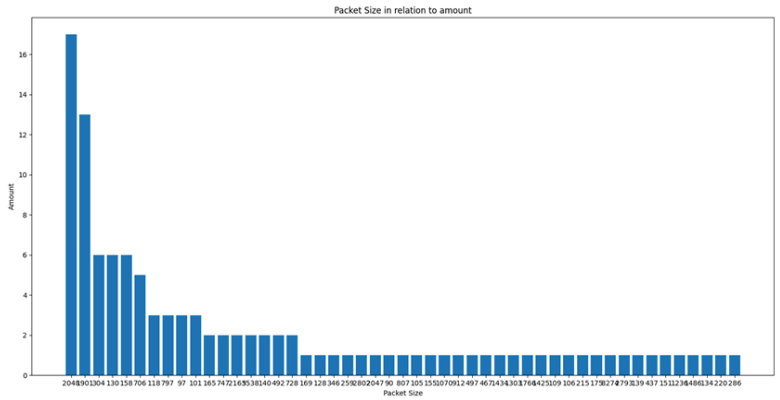
Google Chrome (Wikipedia)-



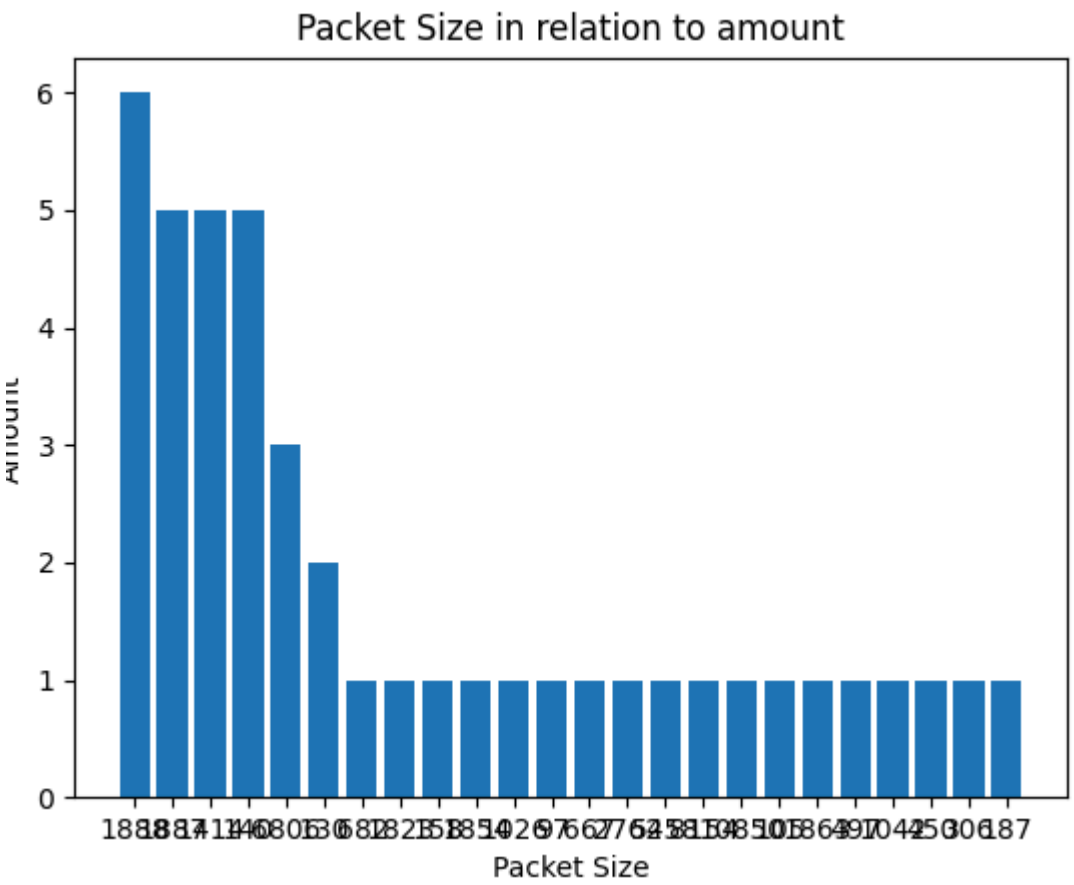
Microsoft Edge (Wikipedia)-



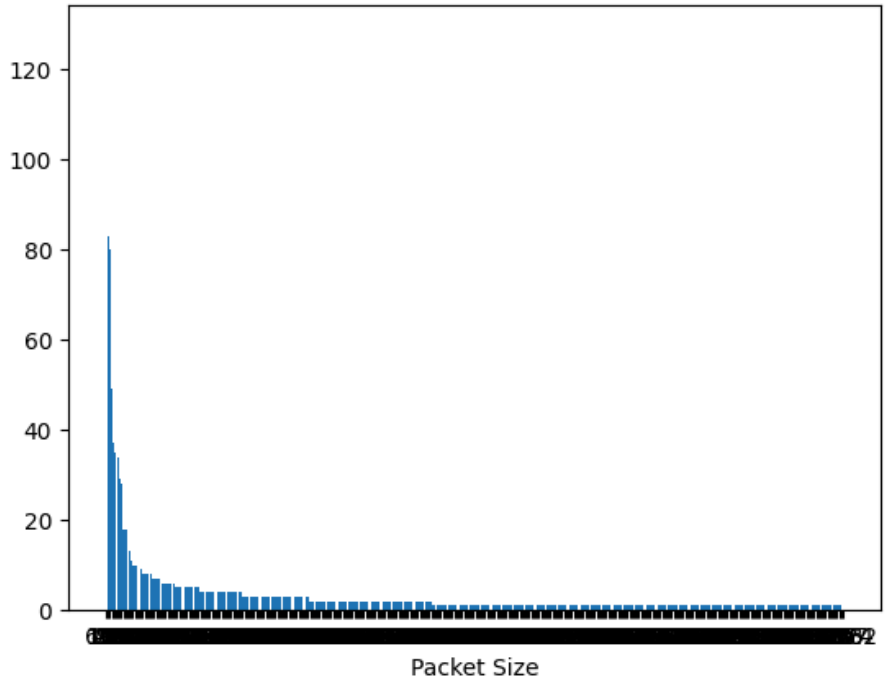
Spotify (Microsoft Edge)-



YouTube (Microsoft Edge)-



Google Meet(Microsoft Edge)-



Google Chrome- ההתפלגות מציגה מספר גבוה של מנות קטנות מאוד, ופחות מנות קטנות, ניתן להניח ש- Chrome שולח הרבה מאוד מנות קטנות כדי להבטיח טעינה מהירה יותר של דפי אינטרנט ומקטין את ההשהיה.

Microsoft Edge- באופן דומה לכרום, משתמש בכמות גדולה של פאקטות קטנות ופחות בגדולות, אך עם פחות תנודתיות (variance) בגודל המנות, רומז לכך ש-Edge נוטה להשתמש ביותר מנות בגודל אחיד (פחות שונות בגודל), מה שעשוי לרמוז על אסטרטגיית תקשורת יותר "מאורגנת" או אופטימיזציה שונה לניהול הרשת.

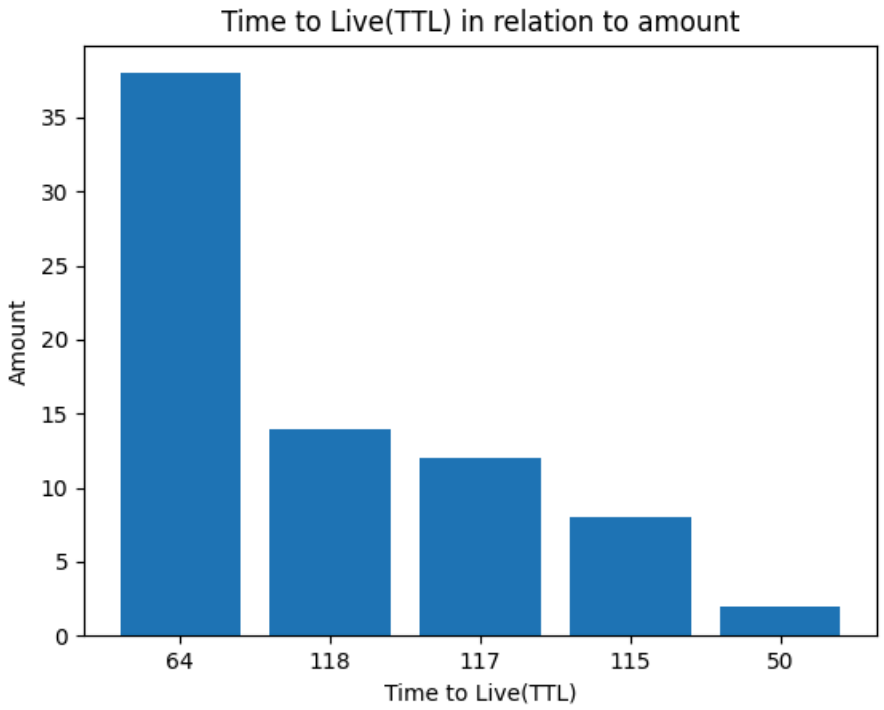
Spotify- יש ריכוז גדול מאוד של חבילות בגודל קטן עד בינוני, ורוב הפאקטות בגודל דומה, נובע מכך שספוטיפיי מספק מנגנון סטרימינג יציב שמעדיף לשלוח פאקטות קטנות בתדירות גבוהה כדי למנוע עומס על הרשת

YouTube- הרוב מנות קטנות, אך יש שונות גבוהה בגודל הפאקטות. נובע מכך שיוטיוב מתאים את גודל הפאקטות בזמן אמת לפי תנאי הרשת, מה שמאפשר התאמת איכות דינמית.

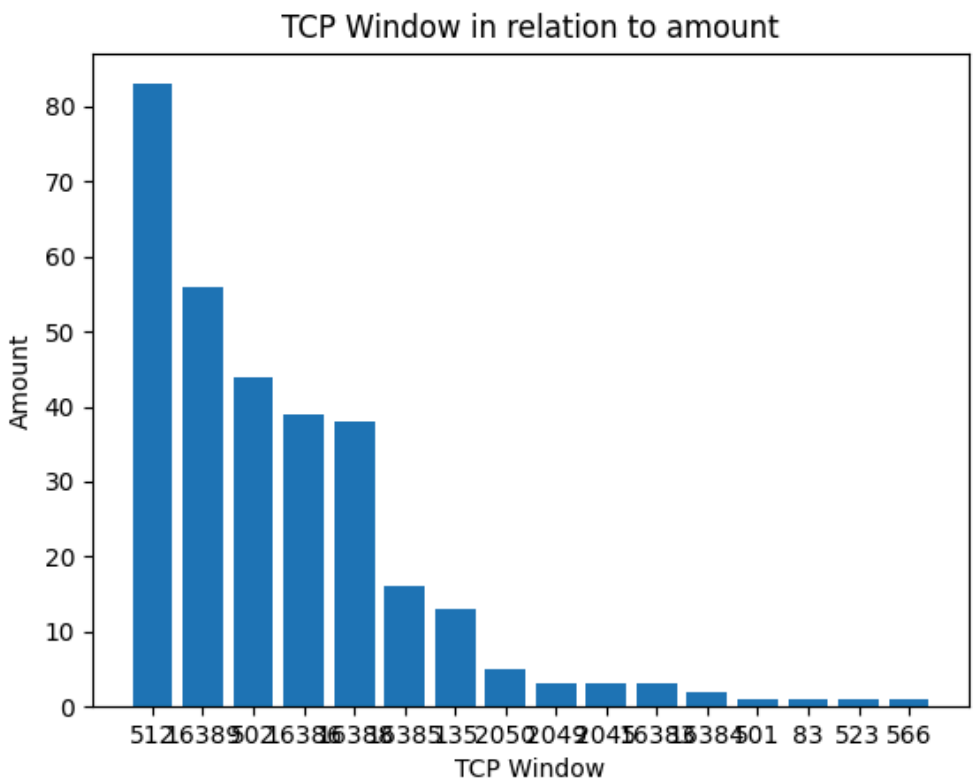
Google Meet- רוב הפאקטות קטנות מאוד, נובע מכך שמשתמשים בפרוטוקול UDP שמבצע העברת חבילות במהירות גם אם חלק מהחבילות יאבדו בדרך על מנת לאפשר תקשורת.

IP Headers

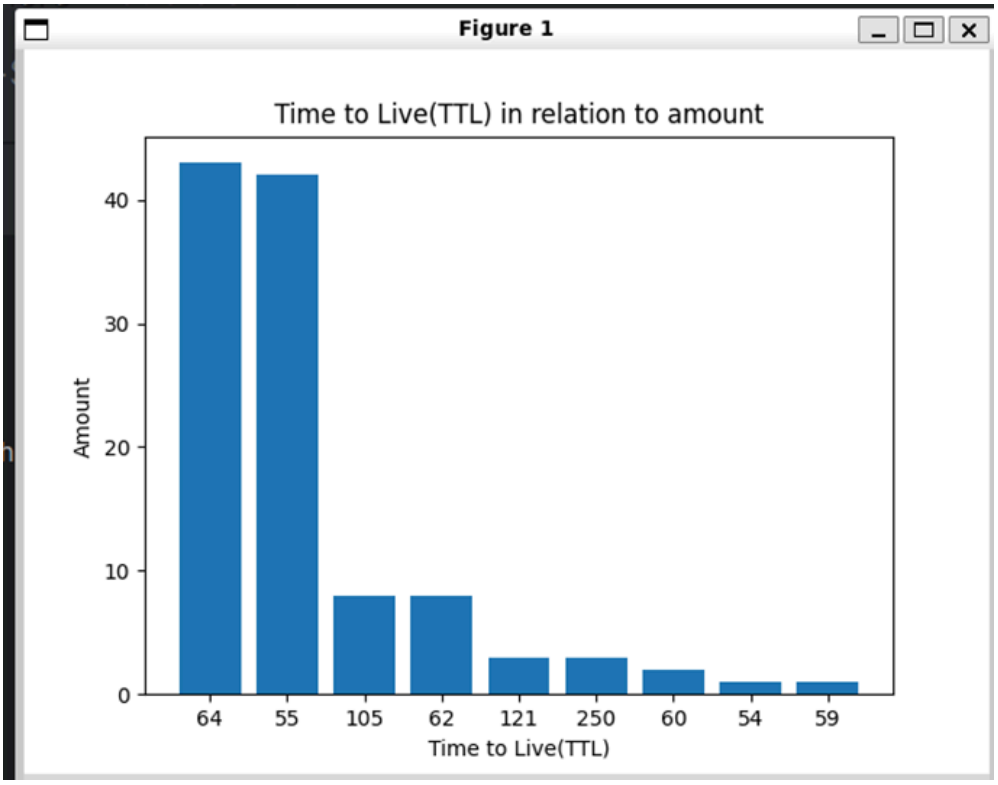
Google Chrome(Wikipedia)-



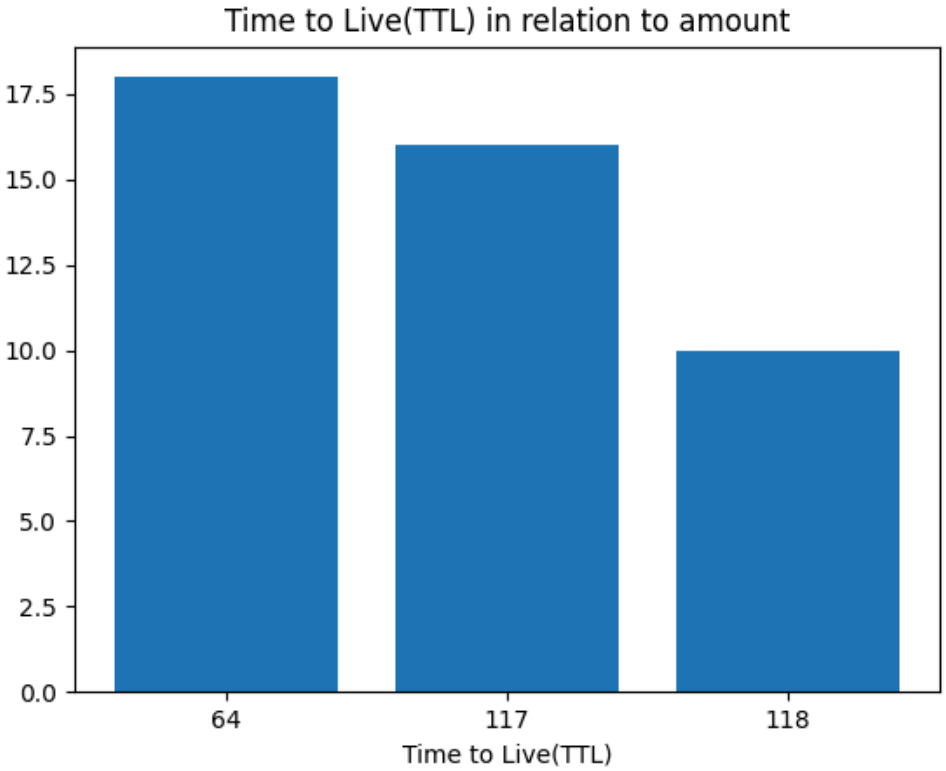
Microsoft Edge(Wikipedia)-



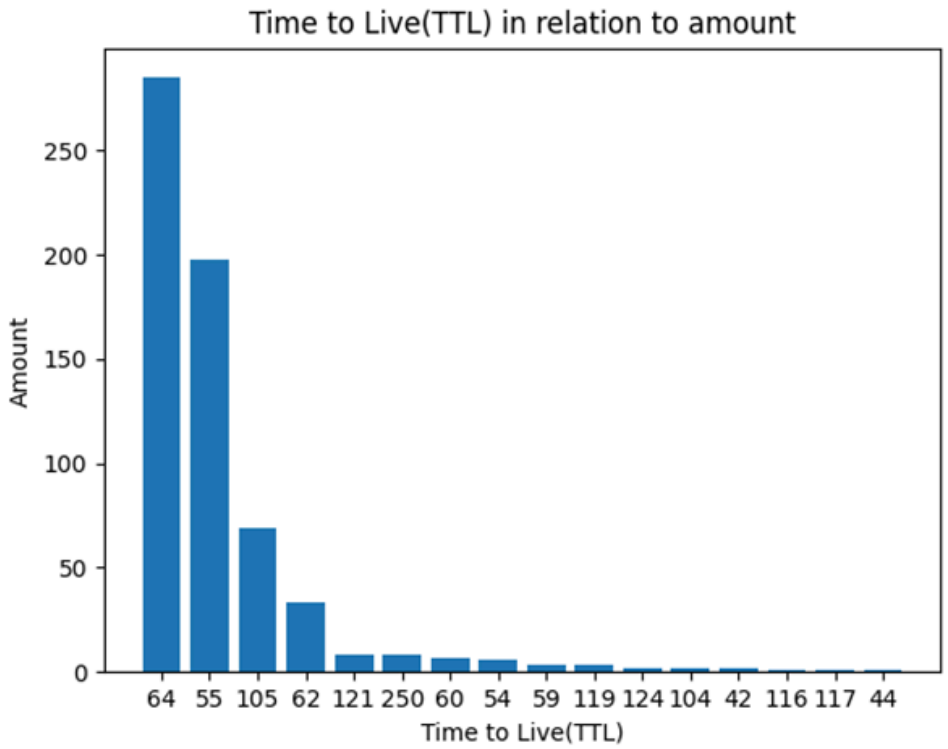
Spotify(Microsoft Edge)-



YouTube(Microsoft Edge)-



Google Meet(Microsoft Edge)-



Google Chrome- מופיע הכי הרבה ttl 64, עשוי להצביע על כך שהרבה מהשרתים שהדפדפן מתקשר איתם משתמשים במערכת הפעלה מבוססת לינוקס, ttl 55 מצביע על כך שהחבילות עוברות 9-10 נתבים לפני שהן מגיעות ליעדן.

Microsoft Edge- תבנית TTL דומה מאוד לזו של TTL 64. Google Chrome הוא הנפוץ ביותר, ויתר הערכים כמעט זהים. הדבר מעיד על כך ששני הדפדפנים פונים מתנהגים בדרך דומה, כי Microsoft Edge מבוסס על Google Chrome.

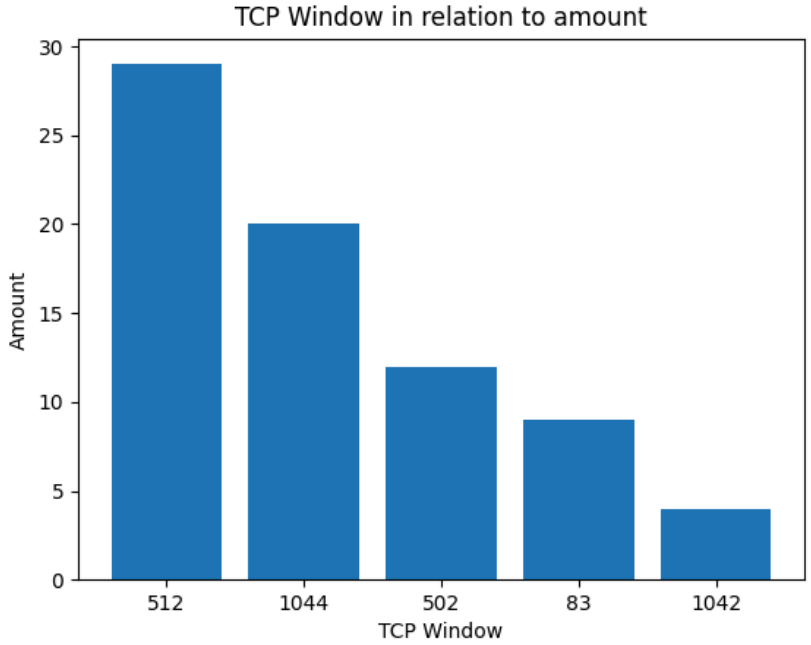
Spotify- מגוון רחב של TTL בניגוד לדפדפנים, ניתן ללמוד מכך שספוטיפיי משתמש בעיקר בשרתים של לינוקס ושהמשתמש מרוחק יחסית מהשרתים.

YouTube- מציג בדומה לספוטיפיי מגוון רחב של TTL מצביע על רשת שרתים גלובלית שמשרת משתמשים מכל העולם

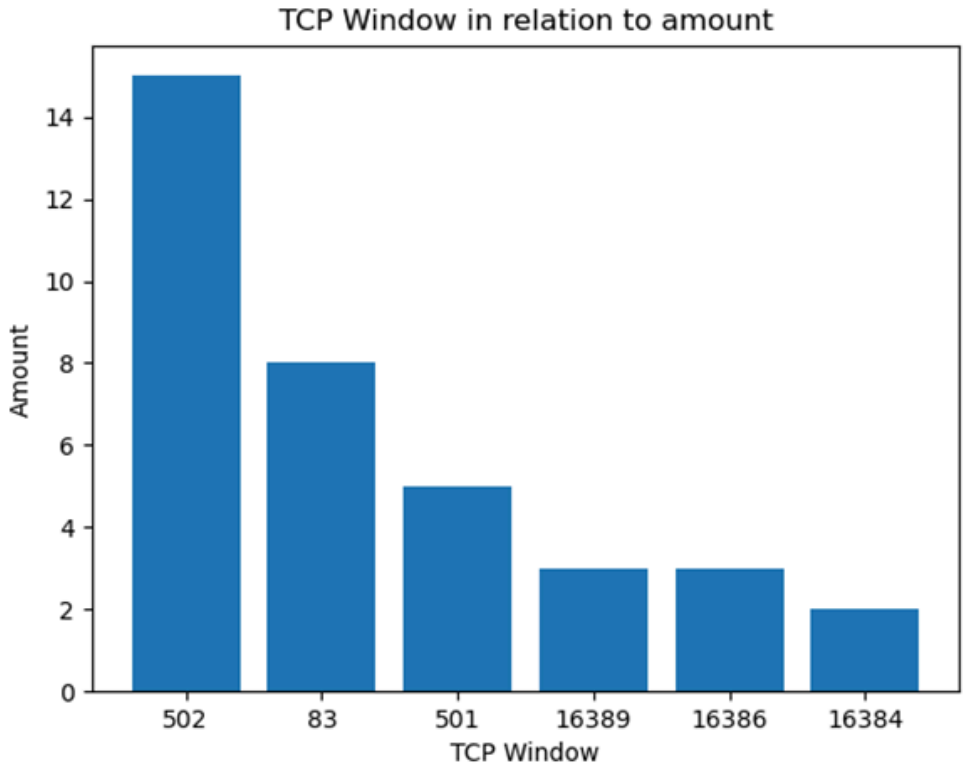
Google Meet– בעיקר מציג TTL 64 ו-TTL 55 בנוסף לסוגים שונים של TTL, מצביע על רשת גלובלית ושימוש בפרוטוקלים שונים לנהל תקשורת בזמן אמת.

TCP WINDOW

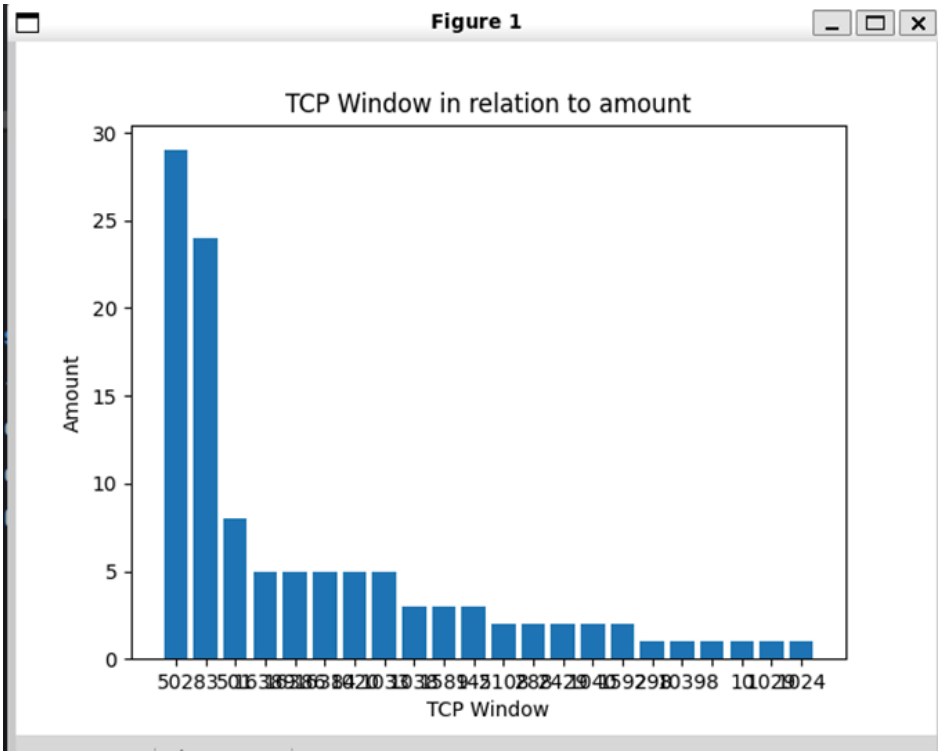
Google Chrome(Wikipedia)-



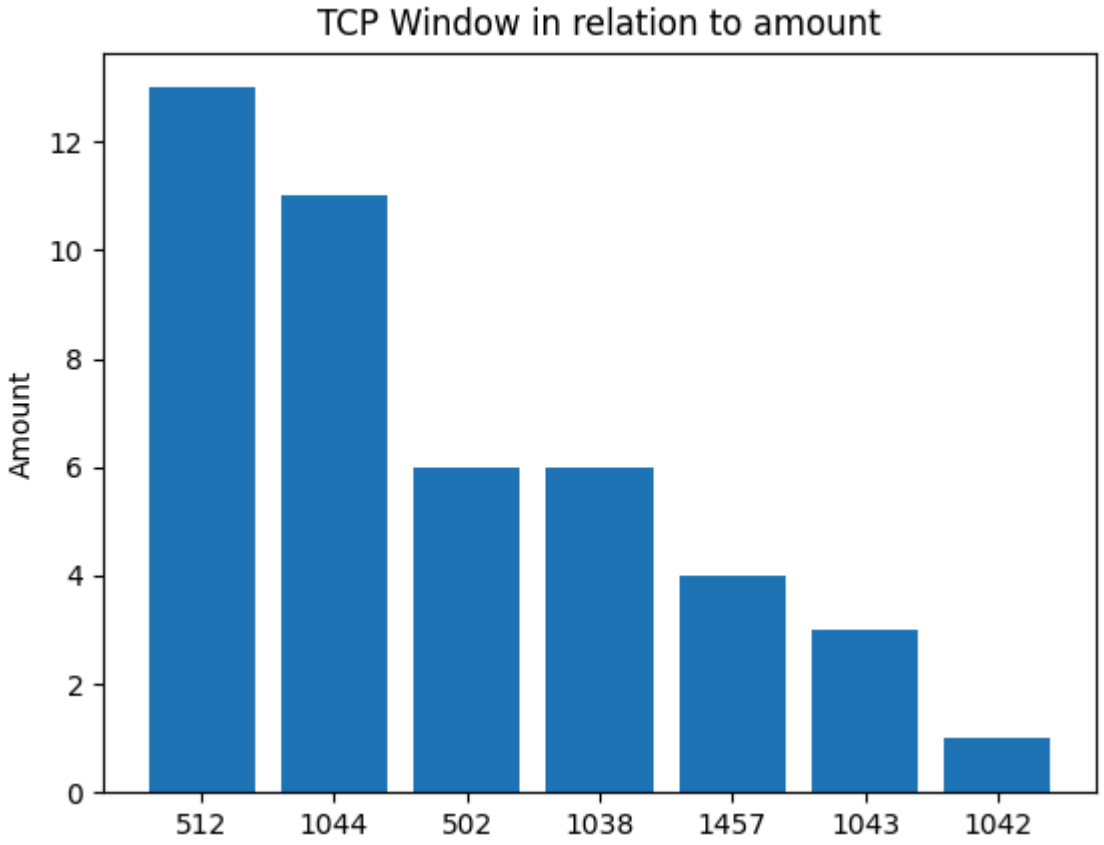
Microsoft Edge(Wikipedia)-



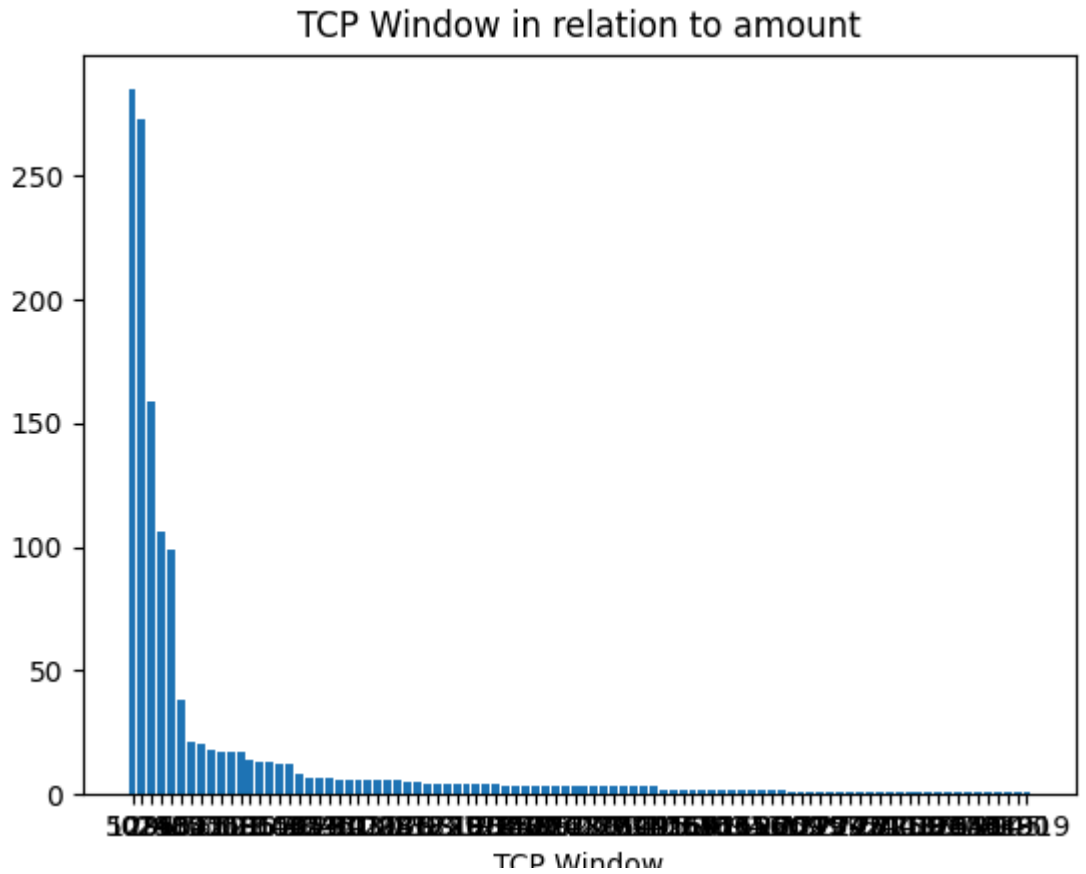
Spotify(Microsoft Edge)-



YouTube(Microsoft Edge)-



Google Meet(Microsoft Edge)-



Google Chrome- מאופיין במספר חיבורים במקביל, משתמש בעיקר בhttp, חלק מהחלונות מופיעים בתדירויות גבוהות, ככל הנראה מהגדרות סטנדרטיות של TCP.

Microsoft Edge- מראה התנהגות דומה לChrome, כי הם משתמשים באותו בסיס (Chromium).

Spotify- מתפלג על סוגים שונים של TCP הצפיפות נמוכה יותר כי מדובר בהעברת חבילות מדורגת במקום בצורה מסיבית כפי שמתממשת בדפדפן, משתמש בTCP כדי למנוע איבוד נתונים.

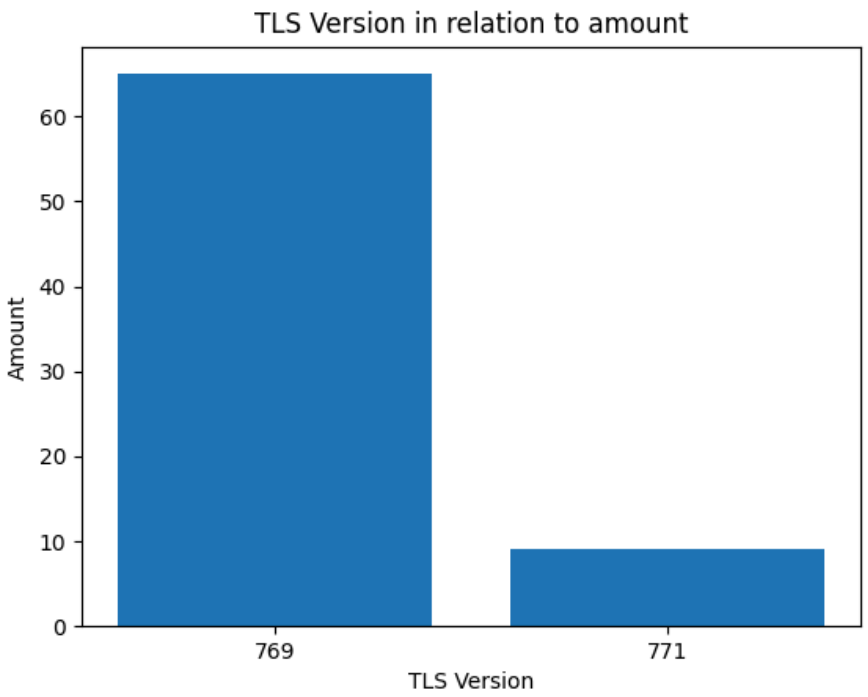
YouTube- מתפלג על סוגי TCP רבים, אך פחות מספוטיפיי ככל הנראה על מנת להתאים את איכות הסרטון בהתאם לשינויים.

Google Meet– מאופיין בקפיצות חדות בין הפרוטוקולים, דפוס בלתי יציב באופן כללי, נובע מכך ששיחות וידאו דורשות העברת מידע בזמן אמת.

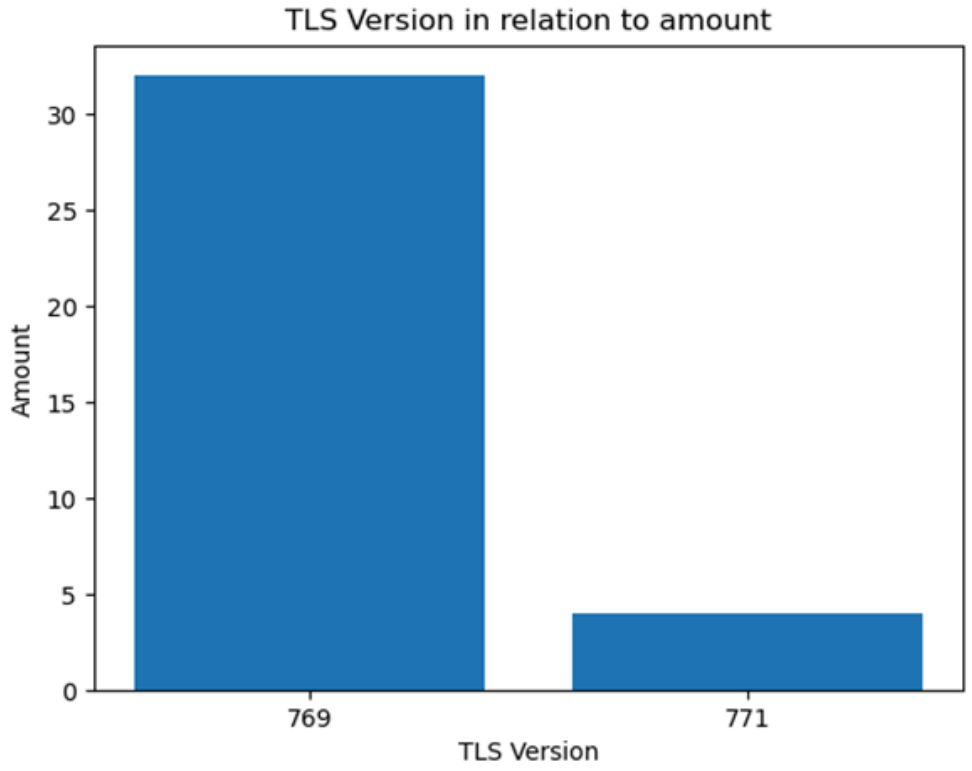
חלון TCP הוא מדד לכמות הנתונים שהמחשב יכול לקבל לפני שהוא שולח אישור (ACK) חזרה לשולח. 502 מופיע בתדירות הגבוהה ביותר, מה שמעיד שרוב החיבורים משתמשים בגודל חלון קטן יחסית, ייתכן בשל חיבורים מוגבלים ברוחב פס או מגבלות של השרתים. ערכים נפוצים כמו 5028, 35016, 16384 וכו' מעידים על כך שהתוכנה מנסה לנצל את רוחב הפס באופן מיטבי. ניתן ללמוד מכך שיישומי סטרימינג ווידאו מנסים לשמור על קצב זרימה יציב באמצעות התאמת גודל החלון TCP, מכיוון שחלונות גדולים מרמזים על חיבור יציב ורוחב פס קבוע.

TLS VERSION

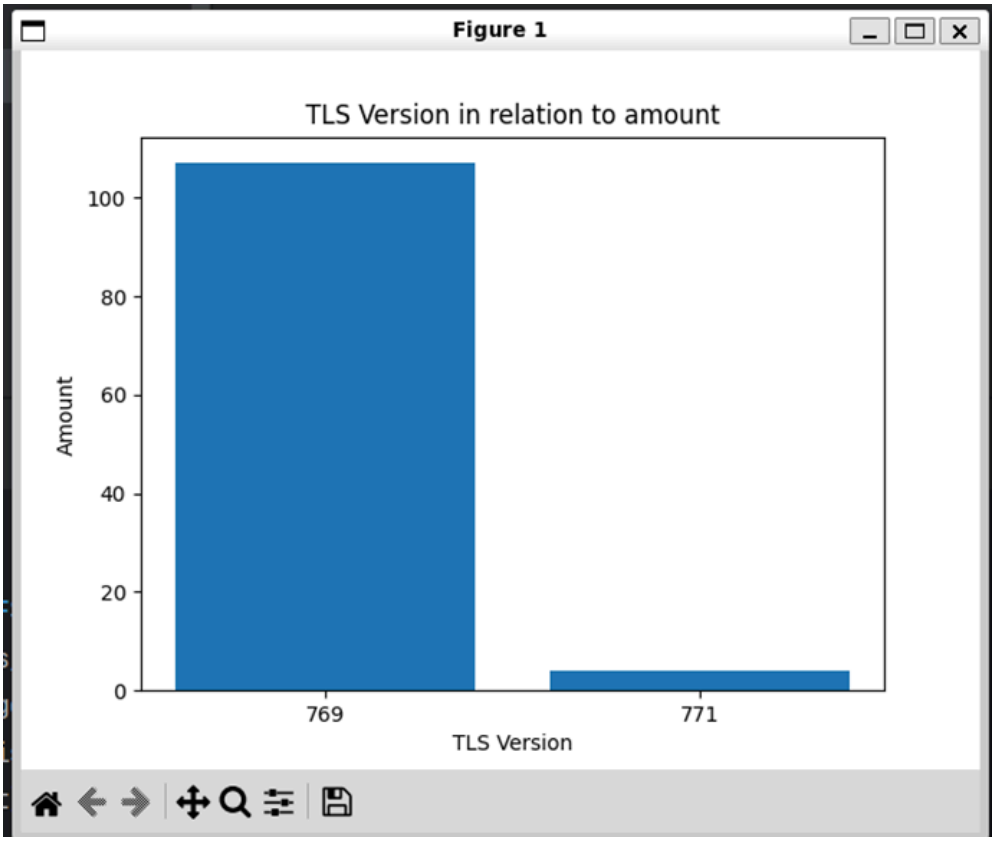
Google Chrome(Wikipedia)-



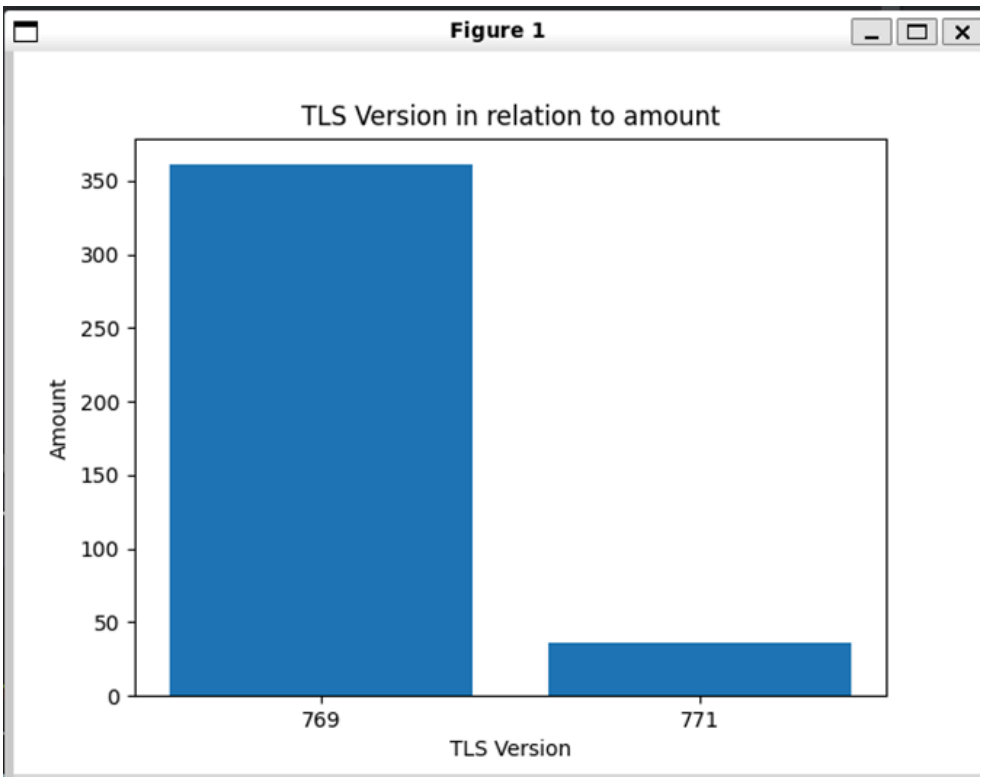
Microsoft Edge(Wikipedia)-



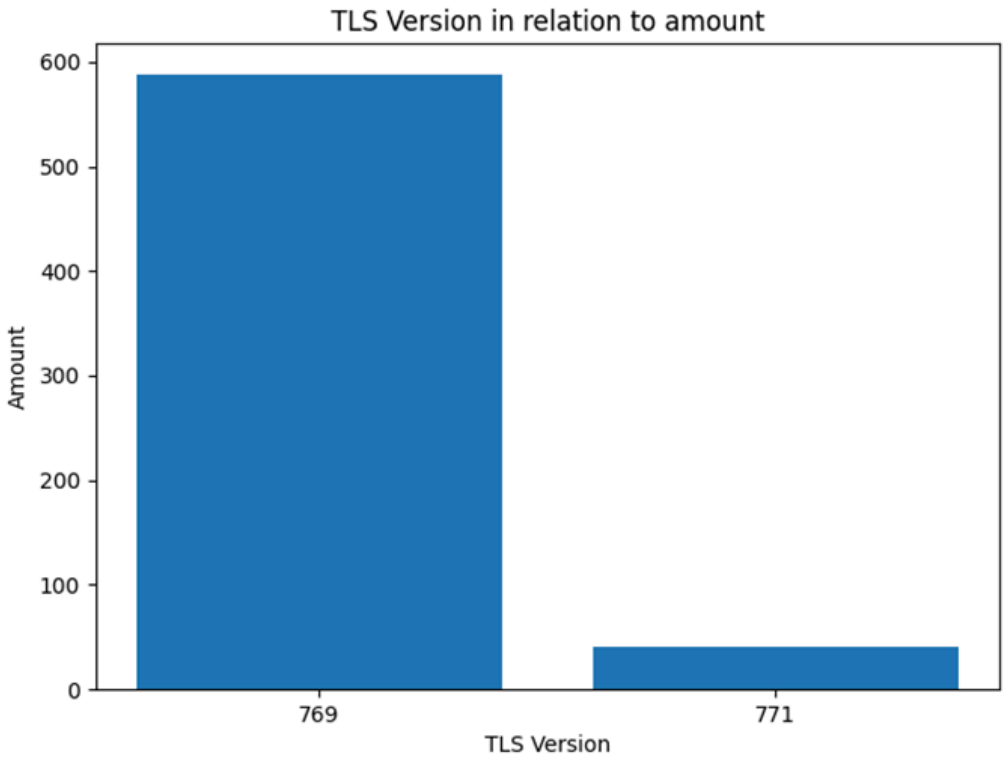
Spotify(Microsoft Edge)-



YouTube(Microsoft Edge)-



Google Meet(Microsoft Edge)-



פרוטוקול TLS הוא פרוטוקול האבטחה שמצפין את התקשורת בין הדפדפן לשרת. רוב התקשורת נעשית באמצעות **TLS 1.0**, מה שמעיד על חיבור שאינו מאובטח, ניתן ללמוד שהאפליקציות משתמשות בחבילות ישנות של TLS. המדד של TLS נשאר זהה, יתכן וזה נובע מכך שנשמר במטמון אותה הגדרה של TLS בכל הקלטה.

חלק 3 שאלה 4:

הנחות יסוד לתשובה:
להבנתנו המשימה היא לאסוף את תעבורת הרשת תוך כדי שימוש באפליקציות כלשהן על ידי כלים כגון Wireshark, כך שבשני התרחישים נתוני התעבורה מוגבלים ומדמים הצפנה או תעבורה אנונימית, ועלינו להשתמש בנתונים הרלוונטים כדי לנתח ולזהות דפוסים המאפיינים אפליקציות ואתרים, כדי לדעת האם בהינתן הנתונים האלה בלבד, תוקף יוכל לזהות מהו האתר או האפליקציה בו ביקרו.

שני התרחישים:
1. ידוע הגודל של כל מנה, חותמת הזמן שלה וערך ה-hash של מזהה הזרימה (source IP, dest IP, source port, dest port).
2. ידוע רק את גודל המנה ואת חותמת הזמן שלה.

- מצורפים בZIP כל קבצי הנתונים (CSV) מתוך ווירשארק, יחד עם ההקלטות.
- כיון שדרוש לנו ניתוח נתונים מתקדם לאלפי פאקטות, השתמשנו בכלי AI לעזרה בניתוח הנתונים.

שיחת וידאו באפליקציית ווצאפ (תרחיש 1):

לפי הנתונים בקובץ, ניתן לראות כמה דברים עיקריים שיכולים להעיד על דפוס שימוש ייחודי של האפליקציה:
1. אינטראקציה דו-כיוונית עם פורטים משתנים:
נראה כי התעבורה נעה בין שני כתובות IP עיקריות: אחת פנימית (למשל 192.168.1.102) ואחת חיצונית (למשל 57.144.121.48).
יש חילופי תעבורה עם שימוש חוזר בפורטים: למשל, "destination port" עבור החבילות היוצאות הוא רוב הפעמים 3478. מספר חבילות חוזרות מופיעות עם פורט זה.

2. קצב תעבורה מהיר וחזרות מרובות:
בניתוח של עמודת "time" ובגודל הפאקטות רואים שהפעולות מתבצעות במרווחי זמן קצרים מאוד תוך שידור חבילות בגודל די קבוע יחסית (למשל 903 בתדירות גבוהה).
החזרה של חבילות בגודל דומה ובפורטים קבועים מצביעה על אפליקציה שמשדרת נתונים בקצב מהיר (לדוגמא: עדכון מיקום, נתוני מדיה או שידורי וידאו/אודיו).

מסקנות:
1. החבילות נשלחות וקולטות בקצב גבוה, עם זמן תגובה קטן מאוד, מה שמעיד על תקשורת בזמן אמת.
2. כתובת הIP הפנימית כנראה מייצגת את המכשיר המקומי שמשדר וקולט נתונים, בעוד שכתובת הIP החיצונית היא של שרת חיצוני שמנהל את התקשורת.

על סמך הנתונים ניתן להסיק כי האפליקציה בה נעשה שימוש מפעילה תקשורת בזמן אמת, הדפוס מתאפיין בקצב תעבורה גבוה, חבילות בגודל קבוע יחסית והעברה דו-כיוונית בין מכשיר מקומי לשרת חיצוני עם שימוש חוזר בפורטים סטנדרטיים (כמו 3478).

שיחת וידאו באפליקציית ווצאפ(תרחיש 2):

לפי הנתונים בקובץ, אפשר לזהות כמה מאפיינים שמצביעים על דפוס התנהגות ייחודי של האפליקציה:
תדירות קבועה של חבילות:
ניתן לראות שיש מרווחי זמן יחסית קבועים בין שליחת חבילות – מה שמעיד על פעילות עקבית מתוזמנת, או עדכוני מצב תקופתיים.

גודל חבילות עקבי:
רוב החבילות בעלות גודל דומה, זה יכול להעיד על כך שהאפליקציה מעבירה נתונים פחות או יותר בגודל אחיד, כמו מידע סטנדרטי של עדכוני מצב, הודעות קצרות, וכדו'.

אופי הפרוטוקולים:
בהתאם לעובדה שמדובר בנתונים מ-Wireshark, ייתכן שמשתמשים בפרוטוקול UDP או TCP עם חיבור מתמשך. שימוש בפרוטוקולים אלה, יחד עם התדירות והעקביות שצוינו, נפוץ באפליקציות שמצריכות תקשורת בזמן אמת – כמו אפליקציות צ'אט, או מערכות בקרה ועדכון סטטוס.

לסיכום, הדפוס מעיד על אפליקציה המבצעת עדכונים תקופתיים ושומרת על חיבור רציף עם השרת, דבר האופייני ליישומים בזמן אמת כמו יישומי תקשורת.

הפעלת סרטון ביוטיוב דרך דפדפן (תרחיש 1):

1. ניתן לראות כתובת IP פרטית (10.100.102.3) המייצגת את המחשב המקומי, וכתובת IP ציבורית (37.60.47.165) של שרת חיצוני.
2. פורט יעד נפוץ: 8000, 1212, 5353 – פורטים מסוימים יכולים להעיד על שירותי סטרימינג, תקשורת מול שרתים ספציפיים, או פרוטוקולי DNS.
3. אורך חבילות ממוצע: 1103 בתים – ייתכן ומדובר בתעבורה שמערבת שליחת נתונים בנפחים משתנים, אך קיימת חזרתיות בגודל 1454 בתים.
4. זמני ההעברה מצביעים על תגובות מהירות , דבר המעיד על דפוס של בקשת תגובה מהירה.
5. טווחי פורטים גבוהים (58000-59000) – לעיתים מעידים על חיבורי UDP/TCP דינמיים, ייתכן עבור וידאו או אודיו.

ניתוח:

1. נראה שלקוח יוזם בקשות אל השרת החיצוני, והשרת מגיב חזרה.
2. קיימים מרווחים קבועים יחסית בין חלק מהחבילות. דפוס כזה יכול להעיד על סטרימינג או דגימה חיה (כגון וידאו או קול), שבהם חבילות נשלחות בתדירות קבועה.
3. העובדה שהשרת תמיד משתמש בפורט 1212 מעיד על כך שזה כנראה פורט שמיועד לאפליקציה זו, דבר שיכול להצביע על שירות פרטי או יישום עם פרוטוקול מותאם אישית.
4. ההחלפה המהירה של חבילות, עם הבדלי זמן קטנים מאוד, מעידה על אינטראקציה בזמן אמת.. התקשורת כוללת תעבורה מהירה ותגובה מיידית, מה שיכול להעיד על אפליקציה שדורשת עדכונים בזמן אמת או תגובה מהירה.
5. ניתן לראות דפוס מחזורי בתעבורת הנתונים, מה שמחזק את הטענה שמדובר באפליקציה שמעבירה נתונים באופן רציף, כמו סטרימינג של וידאו או אודיו. ניתן לראות קצב נתונים שמשתנה אבל עם מחזוריות ברורה, פורטים גבוהים המשמשים לעיתים לתקשורת דינמית, אורך חבילות שנראה עקבי יחסית, עם חזרות על ערכים מסוימים.
6. קצב הנתונים: ממוצע- 231KB לשנייה בערך. מקסימום -2MB לשנייה. סטיית תקן גבוהה – מעידה על קפיצות גדולות בקצב ההעברה, מה שמאפיין וידאו יותר מאשר אודיו. קפיצות חדות – נפוץ בסטרימינג של וידאו, שבו נתונים נשלחים בגושים גדולים, ולא בצורה רציפה כמו באודיו.

למסקנה:

בהתבסס על קצב הנתונים הגבוה, הקפיצות בנפח ההעברה והפרופיל הכללי של התעבורה, מדובר ככל הנראה בסטרימינג של וידאו, ולא רק אודיו.

הפעלת סרטון יוטיוב דרך דפדפן (תרחיש 2):

מנתוני התעבורה אפשר להבחין בכמה מאפיינים שמעידים על דפוס התנהגות אופייני לאפליקציה הפועלת בזמן אמת:

1. תעבורה קבועה ועקבית: רואים שליחה של חבילות קטנות במרווחים קבועים מה שמעיד על כך שהאפליקציה שומרת על קשר רציף ומתעדכנת באופן סדיר (מה שמאפיין לרוב שירותי תקשורת בזמן אמת, כמו סרטוני וידאו).
2. שימוש בפרוטוקולים מהירים יותר(עם שהייה נמוכה) (כמו UDP): זה מרמז על העדפה לתעבורה מהירה על חשבון אמינות מוחלטת (כמו בTCP) תכונה נצרכת ליישומים שבהם העיכובים משפיעים על חוויית המשתמש (סרטוני וידאו למשל).
3. תקשורת דו-כיוונית (client-server): הנתונים מציגים תעבורת מידע משני הכיוונים, כלומר גם הלקוח וגם השרת מעבירים נתונים בצורה סדירה ורציפה מה שמעיד על צורך בסנכרון מתמיד ובשמירה על חיבור פתוח בזמן אמת.

לסיכום אין בידינו בתרחיש זה מידע על סוג האפליקציה, אבל הדפוס יכול להעיד על אפליקציה שבה העברת נתונים בזמן אמת היא קריטית, שזה אפליקציות כמו שיחות קוליות או שירותי וידאו, יתכן גם אפליקציה להודעות מיידיות עם עדכונים רציפים. כלומר לא ניתן לדעת באופן חד משמעי שמדובר בגלישה באפליקציית יוטיוב.

כדי לזהות דפוסי תעבורה, נבחן את הדברים הבאים:

1. התפלגות גודל החבילות.
2. קצב שליחת החבילות לאורך הזמן.
3. האם יש תקופות של תעבורה אינטנסיבית ואז ירידה בפעילות?
4. האם יש חזרות על מבנים אופייניים שיכולים לרמוז על סוג הפעילות?

אורך החבילות:

נראה שהתפלגות אורך החבילות אינה אחידה, יש חבילות קטנות מאוד אבל גם קבוצה משמעותית של חבילות גדולות יותר.
קצב שליחת החבילות:

בגרף ניתן לראות שיש מקטעים של שליחת חבילות באופן מהיר יותר, שיכול להעיד על הזרמת מדיה (Streaming), כמו צפייה בסרטון.
מקטעים חוזרים של קצב תעבורה גבוה ונמוך: (שיכול להצביע על טעינה ראשונית של נתונים ואז ניגון מדיה).

יש שינויים חדים בקצב שליחת החבילות, עם פרקי זמן של עומס גבוה ואחריהם ירידה יחסית.
זה דפוס שמאפיין יישומים של שירותי סטרימינג, שבהם יש הורדה ראשונית של וידאו (Buffering) ולאחר מכן תעבורה תקופתית להשלמת הנתונים.

למסקנה:

1. יש חבילות בגדלים שונים, כולל גדולות (כנראה וידאו) וקטנות (בקרת חיבור).
2. תעבורה מחזורית – עומס גבוה לפרקי זמן קצרים, ואז ירידה.
3. התנהגות זו מתאימה לצפייה בסרטוני וידאו אונליין.

הפעלת שמע בספוטיפיי (דרך האפליקציה) (תרחיש 1):

לפי הנתונים, ניתן לראות כמה דברים עיקריים שיכולים להעיד על דפוס שימוש ייחודי של האפליקציה:

1. רוב התעבורה עוברת דרך פורט 443, הפורט הסטנדרטי לשיחות HTTPS. ספוטיפיי משתמשת בהצפנה להעברת המידע, בין אם מדובר בשידור סטרימינג של מוזיקה או בפעולות אחרות.
2. נצפו 27 כתובות יעד ייחודיות, מה שמעיד על כך שהאפליקציה מתקשרת עם מספר שרתים שונים – תכונה אופיינית לאפליקציות שידור מדיה, המשתמשות ברשת שרתים מבוזרת כדי לייעל את השידור וההזרמה של התוכן.
3. הערך החציוני של גודל החבילות: 1465 בתים; אורך מקסימלי: 1465 בתים
מצב זה מתאים להזרמת נתונים בכמויות גדולות ובאיכות גבוהה, כמו מוזיקה בשירות סטרימינג.
4. הסתכלות על שליחת החבילות לאורך הזמן מעידה על פעילות לא אחידה – ישנם זמנים עם פעילות מוגברת של תעבורה וזמנים של פחות תעבורה. תבנית זו אופיינית לאפליקציות סטרימינג שבהן ההזרמה מתבצעת בפרקים קצרים של נתונים, לצד שליחת פקודות קטנות (כגון ACKs).
5. כתובת ה-IP המקומית (192.168.1.102) שלחה את רוב החבילות (1096 חבילות), דבר המעיד על תקשורת אינטנסיבית מהמשתמש כלפי מספר שרתים חיצוניים שמספקים את שירות ההזרמה.

השילוב של תקשורת מוצפנת, העברת נתונים בגודל מקסימלי בפרצים ורשת שרתים מבוזרת מעידים על מאפייני האפליקציה שבה השתמשו. תכונות אלו מתאימות מאוד לשירות הזרמת מוזיקה איכותי, לכן ניתן להסיק מתוך הנתונים שמדובר באפליקציית ספוטיפיי.

⚫הפעלת שמע בספוטיפיי (דרך האפליקציה) (תרחיש 2):

ניתן לזהות דפוס שבו נראה כי ישנה תקשורת בזמן אמת עם העברת נתונים שמשתנה בקצב – למשל, הודעות קטנות שמטרתן לשמור על החיבור ובמקביל שידור "באצ'ים" של נתונים כבדים יותר (שיכולים להיות פריימים של וידאו או קטעי קול).

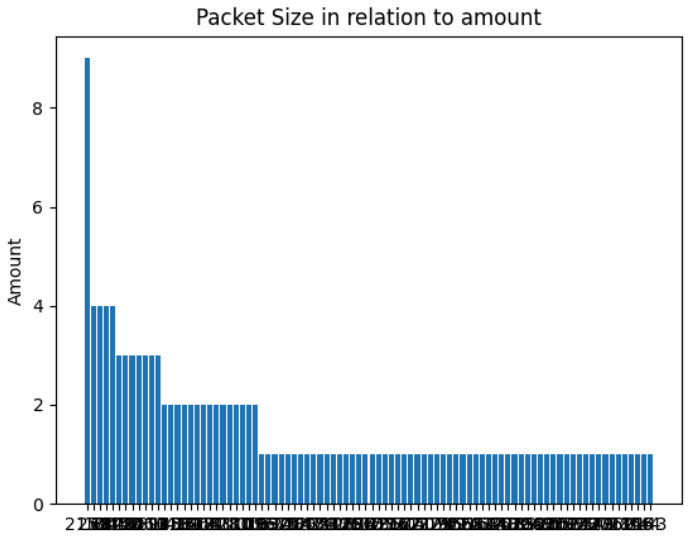
התבנית הזו אופיינית לאפליקציות תקשורת בזמן אמת, כלומר, למרות שאין בידינו את המידע המפורש לגבי שם האפליקציה, דפוס ההתנהגות בתעבורה מצביע על כך שמדובר ככל הנראה באפליקציה להעברת שיחות או שידורי מדיה בזמן אמת. כלומר ניתן לזהות שמדובר באפליקציית שידור מדיה, אך לא לזהות את ספוטיפיי באופן ספציפי.

כיצד ניתן לצמצם את האיום?

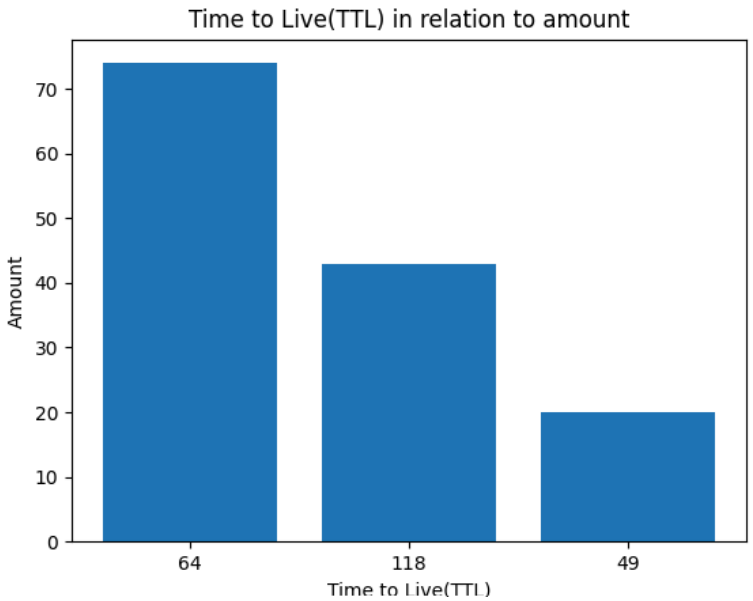
אפשר להקטין את הסיכוי שתוקף יזהה את האפליקציה על ידי הוספת נתונים מזויפים ("ריפוד"-padding) לכל חבילה, כך שכל חבילה תגיע בגודל אחיד ולא תחשוף דפוס ייחודי, וכן על ידי הוספת עיכובים אקראיים בין השליחות של החבילות, כך שהזמנים בין החבילות יהיו משתנים ולא יעידו על תבנית קבועה, ובנוסף לעדכן את הפרוטוקולים שבהם נעשית התקשורת כדי להסתיר עוד פרטים מהמטא-דאטה, מה שמקשה על מי שמנתח את התעבורה לזהות בבירור את האפליקציה שבה נעשית השימוש.

שאלת בונוס

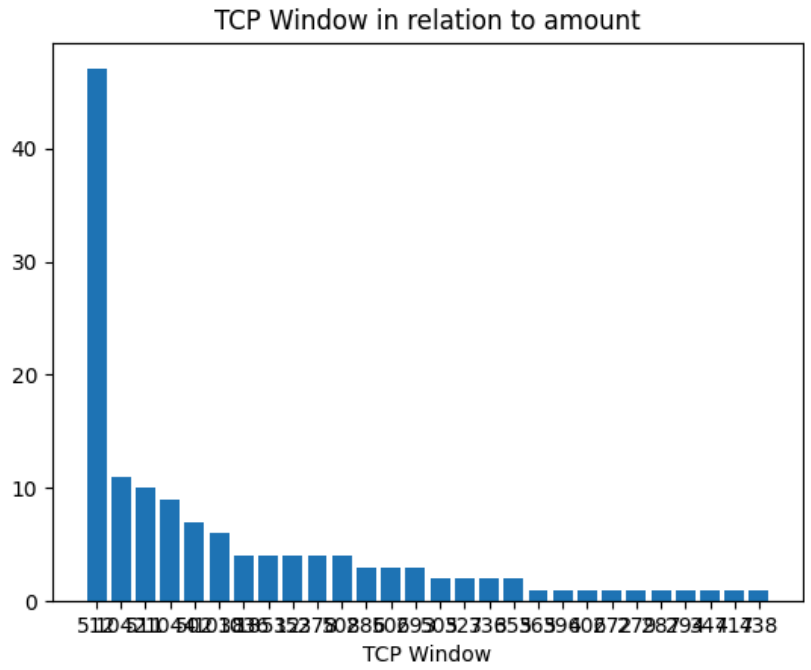
בשאלה זו פתחנו ספוטיפיי ובמקביל שלחנו הודעות דרך ווטסאפ, כעת ננתח את התעבורה ונשווה לממצאים שקיבלנו בתרגילים הקודמים-



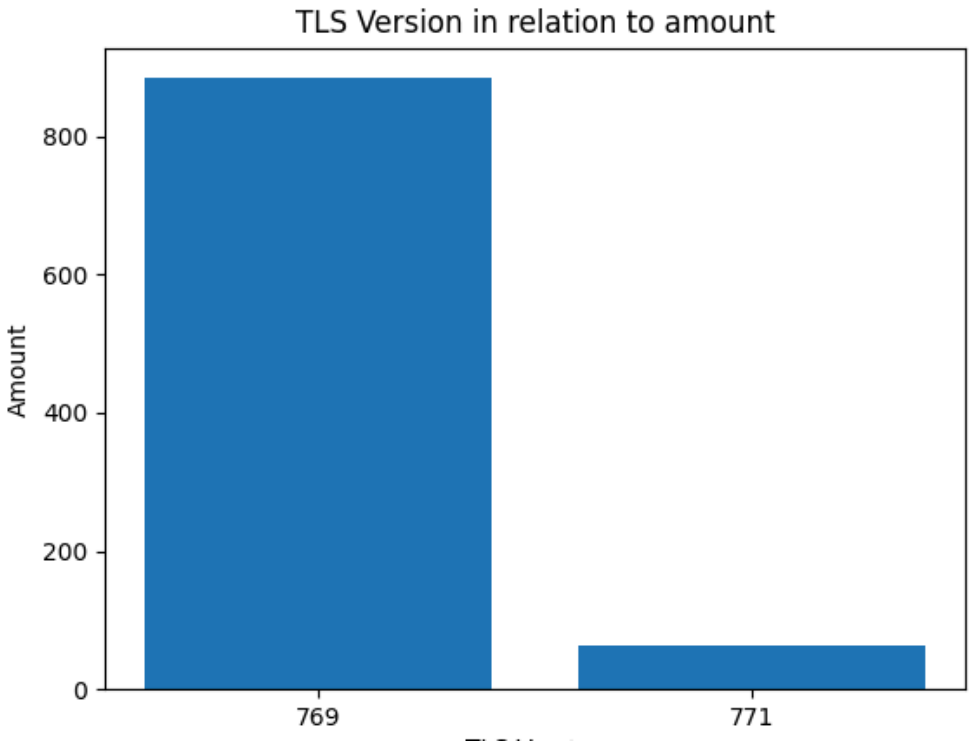
אנו רואים כאן תמונה דומה לגודל הפאקטות של ספוטיפיי בלבד, אך כמות גדולה יותר מנות קטנות, מכיוון שווטסאפ שולח הודעות טקסט בUDP או בTCP עם הצפנה כך שהמנות קטנות יותר, ומפוזרות על טווח רחב יותר.



כאן יש כמות TTL קטנה משמעותית מאשר שספוטיפיי מופעל לבדו, כשספוטיפיי מופעל לבד הוא מושך נתונים ממגוון שרתים כדי לשפר ביצועים, אך כשהוא מופעל יחד עם ווצאפ ייתכן שחלק מיישומי ספוטיפיי נשארים יציבים מול אותם שרתים, מה שמפחית את כמות הTTL.



באופן כללי ספוטיפיי מתפלג על סוגים שונים של TCP, כי הוא שולח את המידע לכמה מוקדים על מנת לשפר ביצועים השימוש בווטסאפ במקביל רק הגביר יותר



נשארנו עם חותמת TLS זהה לכל הבדיקות.