ב"ה

פרוייקט גמר רשתות תקשורת- עטרת ון-לואן 325894160 וטליה ולרשטיין 324866680

שאלה 1-

ישנם כמה גורמים שעלולים לגרום להעברת קבצים איטית בשכבת התעבורה-

-גודל החלון- כאשר גודל החלון של שליחת החבילות קטן, אז העברת הקבצים תהיה איטית. בכדי לשפר את קצב העברת הנתונים נוכל להגדיל את גודל החלון.

-אובדן חבילות- אובדן חבילות גורם ל TCP לבצוע שידור מחדש והקטנת גודל החלון, מה שמאט את הקצב

-טיימר קטן מדי- הלקוח יצפה לקבל ACK בזמן קצר מדי לזמן האמיתי שייקח ולכן לפני שהACK יספיק לחזור, למרות שאכן נשלח כרצוי הלקוח יראה שלא חזר האישור וכבר ינסה שוב לשלוח את ההודעות מה שיאט את הקצב. בכדי לפתור את זה נוכל להשתמש במערכת שתגדיל ותקטין את הטיימר בהתאם למצב הרשת לפי ה-RTT או להגדיל "ידנית" את הטיימר.
זמן תגובה ארוך בין שליחת מנה לקבלת האישור מאט את הקצב הכולל

-אם בין המקור ליעד יש קצב שידור לא מאוזן אז עלול לאחד מהם להיות כמות מידע מעבר למה שהוא יכול לעבד מה שיגרום להאטת העברת הנתונים. נפתור ע"י Flow Control .

-Congestion Control – כאשר זוהה עומס ברשת, מנגנון זה ב TCP מצמצם את קצב שידור החבילות שלו בעזרת המנגנונים שברשותו. וזה יכול להאט את העברת הנתונים.

שאלה 2-

מנגנון Flow Control מתמקד במקבל, משפיע על העברת נתונים בכך שהוא מוודא שלא ייווצר מצב בו המקור- השולח, פוגע במקבל-יעד בכך ש"מעמיס" יותר מדי נתונים\במהירות גדולה מידי על המקבל, כך שהמקבל לא יעמוד בכך וייווצר תור ארוך או יאבדו חבילות דבר שיגרום לקצב העברת נתונים איטי. אז השפעת מנגנון Flow Control על העברת הנתונים הוא:

- שיפור העברת הנתונים בצורה יציבה- מוודא שלא יהיה עומס על המקבל וכך לא יאבדו נתונים\ יהיו איטיים מאוד.

- האטת קצב שידור השולח- אם יכולת עיבוד המקבל נמוכה מיכולות העיבוד של השולח, אז השולח יצטרך להאט ולהתאים עצמו לקצב העברת הנתונים בו יוכל המקבל לעמוד ולתפקד טוב.

לכן, גם אם לשולח יש כוח עיבוד גבוה יותר משל המקבל, קצב העברת הנתונים- כאשר קיים מנגנון Flow Control, יהיה לפי המקבל- לפי כמות הבתים שיכול עוד המקבל לקבל, דבר שתלוי במהירות העיבוד שלו. ובמקרה לשנו עלול להאט את קצת העברת הנתונים.

שאלה 3-

תפקיד הניתוב ברשת משמעותי מאוד, הוא קובע את מסלול כל חבילה ברשת לפי גודלה ויעדה בצורה אופטימלית. בחירת ניתוב משפיעה על ביצועי הרשת בכך שעבור כל חבילה מחדש בוחרת- באופן דינמי נתיב מתאים בהתאם למצב הרשת הנוכחי ולצורך החבילה. ע"י זה ניתן למנוע עומסים ולאפשר ביצועי רשת טובים יותר וממוקדים יותר לצורך כל חבילה והרשת.

הניתוב יכול להשפיע על עומס הרשת בכך שיבחר מחדש- אם כל הניתובים יהיו דרך מסלול יחיד העומס ברשת יהיה עצום ויגרום להאטה גדולה ולאיבוד חבילות מרובות. בחירת ניתוב יכולה להיות

ב"ה

מושפעת מעלות שימוש בנתיב, מביטחון בשימוש בנתיב, מספר הקפיצות, מרוחב פס של נתיב, מזמינות נתיב ועיכובים- delay זמן שלוקח לחבילה לעבור מהנתיב. את כל המשפיעים האלו יש לקחת בחשבון בכל החלטת ניתוב בהתאם לצורך החבילה.

שאלה 4-

MPTCP משפר את ביצועי הרשת בכך שמאפשר שימוש במספר נתיבים במקביל לעברת חבילות. דבר זה גורם לכך שיהיה אפשר להעביר את כלל הנתונים בצורה מהירה יותר בשימוש בכמה נתיבים, בנוסף העומסים על נתיבים יכולים לפחות בכך שהחבילות מחולקות בין כמה נתיבים והעומס מתחלק ביניהם, MPTCP כולל אלגוריתמים לאיזון עומסים חכם. רוחב הפס הכולל גדול יותר כאשר כמה נתיבים "תורמים" לרוחב הפס של החיבור, וגם, כאשר נתיב יחיד כושל, המערכת לא תאבד את החיבור כיוון שתוכל להיתמך בנתיבים האחרים, והמעבר נעשה בצורה אוטומטית וחלקה. דבר זה ייתן סיכויים נמוכים יותר לאיבוד מידע בחיבור ויגדיל את רמת האמינות של החיבור. שימוש בנספר נתיבים מקבילים יאפשר גם הפחתת זמן השהיה, בכך שהמערכת תבחר מבין הנתיבים האפשריים את הנתיב המהיר מבניהם ברגע נתון. מה שיגרום לביצועי הרשת להיות מהירים יותר. MTCP כולל ניהול נתיבים עבור כל נתיב בנפרד וזה יאפשר לו להבטיח שהחבילות יתפלגו בצורה יעילה בין כל הנתיבים.

שאלה 5-

הגורמים האפשריים לאובדן חבילות בשכבות הרשת והתעבורה הם-

-עומס ברשת- בנתבים, יגרום לנתבים "לזרוק" חבילות שיגיעו אליהם כאשר התור של הנתב מלא, זה קורה כשקצב הכניסה עולה על קצב היציאה.

-חוסר יציבות של חיבור הרשת- יכול לגרום לחבילות להיאבד בדרך

-חלון שליחה גדול מדי ליכולות הקבלה והעיבוד של צד המקבל יגרום לאיבוד חבילות כאשר יגיעו אל המקבל.

-גודל Buffer לא מתאים שלא יכול להתמודד עם פרצי תעבורה(עלייה פתאומית של כמות הנתונים שעוברים ברשת בפרק זמן קצר), אם הוא קטן מידי הוא יתמלא ויזרוק חבילות ואם הוא גדול מידי חבילות עלולות להיות מאוחסנות בbuffer יותר מידי זמן ולהגיע מאוחר מידי

-בעיות ברמת החומרה- כשלים בכרטיסי רשת, זיכרון או מעבד של הנתב

שלבים מומלצים לפתרון הבעיה- ננסה לבדוק האם העומס ברשת-ניתן ע"י traceroute לזהות היכן בדיוק מתרחש האובדן ולבדוק אם יש השהיות חריגות בנקודות מסוימות בנתיב. אחרת, ננסה להקטין את חלון השליחה. ננסה להתאים את גודל הbuffer כדי שיוכל להתמודד טוב יותר עם פרצי תעבורה, (אבל לא נגדיל יותר מידי)

חלק 2- מאמרים

מאמר ראשון- Analyzing HTTPS Encrypted Traffic to Identify User’s Operating System, Browser and Application

ב"ה

תרומת המאמר-

המאמר הוא העבודה הראשונה שמראה יכולת לזיהוי מערכת הפעלה, דפדפן ואפליקצייה של משתמשים רק ע"י תעבורת HTTPS שלו. (תעבורה מוצפנת).
המאמר מציג תכונות חדשות שמנצלות את ההתנהגות המתפרצת של דפדפנים ואת התנהגות SSL. (התנהגות מתפרצת- הכוונה לקטע של תעבורת רשת כאשר לפניו ואחריו אין\כמעט אין פעילות רשת פעילה.) בעזרת תכונות חדשות אלו החוקרים שיפרו את הדיוק של זיהוי פרטי המשתמש (מערכת הפעלה, דפדפן ואפליקציה).
בנוסף במאמר החוקרים מתארים מאגר נתונים שיצרו המכיל יותר מ-20,000 דוגמאות של תעבורות רשת מתוייגות שבתעבורות אלו ניתן לדעת בוודאות את מערכת ההפעלה, דפדפן ואפלקציה של המשתמש.
המאמר תורם לנו בכך שמציג שהצפנה אין משמעותה פרטיות. המאמר מציג את פגיעות פרטיות המשתמש אף על פי שהמשתמש משתמש בתעבורה מוצפנת.

תכונות תעבורה שבהן משתמש המאמר הן-

- מספר חבילות ובייטים של חבילות יוצאות מהמשתמש ונכנסות אליו (קדימה\אחורה).
- זמני הגעה בין חבילה אחת לזו שאחריה קדימה ואחורה (מינימום, מקסימום, ממוצע וסטיית תקן)
- מינימום, מקסימום, ממוצע, סטיית תקן ושונות על גודלי החבילות קדימה ואחורה.
- ערכי TTL
- מספר חבילות כולל

תכונות חדשות-

- תכונות TCP – גודל חלון התחלתי, גורם שינוי גודל חלון, חבילות keep alive וגודל מקטע מקסימלי.
- תכונות SSL – שיטות דחיסה , מספר הרחבות(תוספות של הפרוטוקול הבסיסי), שיטות הצפנה, אורך מזהה, גרסת SSL קדימה(חבילות יוצאות מהמשתמש).
- תכונות מבוססות התנהגות מתפרצת- מספר התפרצויות קדימה ואחורה, תפוקה מקסימלית, סטיית תקן וממוצעת שיאים אחורה וקדימה (שיא- תיאור התנהגות מתפרצת), הפרשי זמני הגעה בין שיאים (מקסימלי, ממוצע, סטיית תקן קדימה ואחורה)

תוצאות עיקריות-

שילוב תכונות הבסיס עם התכונות החדשות מגדיל את רמת הדיוק בזיהוי מערכת ההפעלה, דפדפן ואפליקציה של המשתמש כולן ביחד וכל זיהוי של אחד מהם לחוד. אחוז הדיוק הוא 96.06% עבור זיהוי שלושת המאפיינים יחד.
בנוסף, אף על פי ששילוב התכונות מספק דיוק גבוה מאוד עדיין ישנן טעויות בסיווג שילובים ספציפיים, בעיקר בין שילובים דומים או עם קטגוריית "unknown". למרות הטעויות, העובדה שהטעויות בין שילובים דומים מעידה על כך שהשיטה טובה כי הבלבול קורה במקומות הגיוניים ואכן מבלבלים. הדיוק המתקבל עדיין מאפשר קבלת מידע משמעותי על המשתמש.

תובנות מהתוצאות-

התוצאות מראות כי הצפנה אינה פרטיות, גם כאשר התוכן מוצפן, תוקף פסיבי יכול להשיג מידע משמעותי על המשתמש.

ב"ה

וניתן לזהות בדיוק גבוה על אף ההצפנה את מערכת ההפעלה, דפדפן ואפליקציה רק מתוך ניתוח תעבורת HTTPS, ללא צורך בפיענוח התוכן עצמו.

שילוב התכונות החדשות שהציעו החוקרים אכן תרמו לשיפור משמעותי בדיוק הסיווג.

הסיווג של מערכות ההפעלה הוא המדוייק ביותר מכיוון שיש הבדלים ברורים שבהן כל מערכת הפעלה מיישמת פרוטוקולי רשת, ניתן לראות בגרפים.

מאמר שני- Early_Traffic_Classification_With_Encrypted_ClientHello_A_Multi-Country Study

תרומות המאמר-

המאמר מציג אלגוריתם חדש מוביל בתחום שנקרא hRFTC שמשלב ניתוח של חלקי תקשורת שעדיין אינם מוצפנים יחד עם מאפיינים סטטיסטיים של זרימת התעבורה (גודל חבילות וזמנים בין חבילות). האלגוריתם מתמודד עם פרוטוקול הצפנה חדש ECH. במאמר ישנה הוכחה לכך שהאלגוריתם מספק שיפור משמעותי בדיוק סיווג התעבורה המוצפנת. האלגוריתם מתמודד עם נתונים מגוונים ומצליח לספק תוצאות טובות מכמות מועטה של נתונים. המאמר מציע שיפור לאלגוריתם כך שיתמוך בפרוטוקול QUIC .

המאמר מפרט על מאגר נתונים מקיף של תעבורת TLS ממדינות שונות בכולל יותר מ- 600,000 זרמי TLS שמחולקים ל-19 סוגי תעבורה. זהו מאגר פתוח ומקיף ביותר בתחום שאותו יצרו החוקרים בעלי המאמר.
תרומות אלו של המאמר חשובות כיוון שהן מציעות פתרון לאתגר של שמירה על איכות שירות ברשתות בעידן של הצפנה מוגברת, מה שמאפשר לספקיות אינטרנט לספק חווית משתמש טובה תוך כיבוד פרטיות המשתמשים.

תכונות תעבורה שבהן משתמש המאמר הן-

מאפיינים מבוססי חבילות-

- גרסת פרוטוקול TLS- גרסת ה TLS שהלקוח מבקש להשתמש בה.
- ערכות הצפנה
- אורך הרחבות- מציין את האורך הכולל של כל הרחבות הTLS בהודעת ה ClientHello
- הרחבות ספציפיות- הרחבה שמכילה את המפתח הציבורי של הלקוח לביצוע החלפת מפתחות., מאפשר שימוש מחדש במפתחות קודמים להתחברות מהירה יותר. , רשימת גרסאות הTLS שהלקוח תומך בהן.
- הרחבות GREASE- ערכים אקראיים שנשלחים כדי לבדוק את התאמת השרת להרחבות עתידיות של הפרוטוקול.
- גרסת QUIC- גרסת הפרוטוקול QUIC שהלקוח משתמש בה.
- מספר חבילות QUIC

מאפיינים מבוססי זרימה-

- סטטיסטיקה של גודל חבילות- ממוצע, סטיית תקן, מינימום , מקסימום, אחוזונים וסכום של גדלי חבילות.

ב"ה

- סטטיסטיקה של זמן בין חבילה לחבילה שאחריה- ממוצע, סטיית תקן, מינימום , מקסימום, אחוזונים וסכום של זמנים בין חבילות.

תכונות חדשות-

- קריטריון בחירת חבילות חדשני- האלגוריתם מנתח את כל החבילות מהשרת ללקוח עד החבילה הראשונה שמכילה נתוני אפליקציה ממשיים (לא רק נתוני הקמת חיבור).
- הרחבה האלגוריתם ל-QUIC
- גישה היברידית מותאמת- שילוב של מאפיינים מבוססי זרימה וחבילות מאפשרים לאלגוריתם להתמודד עם הצפנת ClientHello שבעיקר משפיעה על מאפיינים מבוססי חבילות.

תוצאות עיקריות-

ביצוע האלגוריתם המוצג במאמר על פני אלגוריתמים אחרים קודמים מאז שימוש ב ECH, מניב דיוק סיווג התעבורה המוצפנת של 94.6% תוצאה שטובה משמעותית מתוצאות האלגוריתמים האחרים שמניבים דיוק רק של 38.4% .
ישנם שינויים בביצועים בין מקומות גאוגרפיים דבר שמצריך אימון האלגוריתם על נתונים מהמיקום הספציפי.

תובנות מהתוצאות-

האלגוריתם החדשני אכן משפר משמעותית את דיוק הסיווג של התעבורה המוצפנת למרות ההצפנה המוגברת של ECH ע"י שימוש במאפיינים שאינם מוצפנים.
מודולים היברידיים יעילים יותר- שילוב תכונות של זרימה בנוסף לתכונות מבוססות חבילות הביא לדיוק בסיווג . נדרשת התאמה גאוגרפית של האלגוריתם. ו- ההרחבה לQUIC מגביל את יכולות הסיווג אך עדיין לא מבטל לחלוטין את היכולות.

מאמר 3- FlowPic_Encrypted_Internet_Traffic_Classification_is_as_Easy_as_Image Recognition

תרומות המאמר-

המאמר מציג שיטה חדשה לסיווג תעבורת אינטרנט מוצפנת באמצעות המרת זרמי נתונים לתמונות ושימוש ברשתות נוירונים קונבנציונליות לזיהוי קטגוריות של תעבורה וזיהוי אפליקציות ספציפיות.
שיטה זו שמציג המאמר מתמודדת עם תעבורה מוצפנת ומאפשרת סיווג מדוייק עבורה, ומציגה ביצועים טובים יותר משמעותית מהשיטות הקיימות והשיגה דיוק גבוה במיוחד של 99.7% בזיהוי אפליקציות ועוד, השיטה אינה מסתמכת על תוכן החבילות אלא על נתוני זרימה בלבד ובכך שומרת על פרטיות המשתמשים.

תכונות תעבורה שבהן משתמש המאמר הן-

- גודל החבילות- כמות בייטים שמכילה כל חבילה
- זמני הגעה- זמן קבלה של כל חבילה- היסטוגרמה(האם יש מרווים קבועים או שהחב' נשלחות במקבצים) ושונות.

ב"ה

תכונות חדשות-

- היסטוגרמה דו-מיימדית- ע"י שילוב של זמני הגעה + גודל חבילות.
- עיבוד מבוסס תמונה-בניית תמונה מההסטוגרמה הדו-מיימדית.
- שימוש בחלון זמן קצר וחד-כיווני במקום זרימה דו-כיוונית מלאה.
- אי תלות בתוכן החבילות
- לתת לרשת העצבית- CNN ללמוד בעצמה אילו דפוסים חשובים בנתונים.

תוצאות עיקריות-

במאמר ניתן לראות תוצאות סיווג קטגוריות תעבורה- עבור NON-VPN- 85% דיוק, עבור VPN – 98.4% דיוק, עבור Tor – 67.8% דיוק.

זיהוי יישומים- דיוק של 99.7% בסיווג 10 יישומי וידאו.

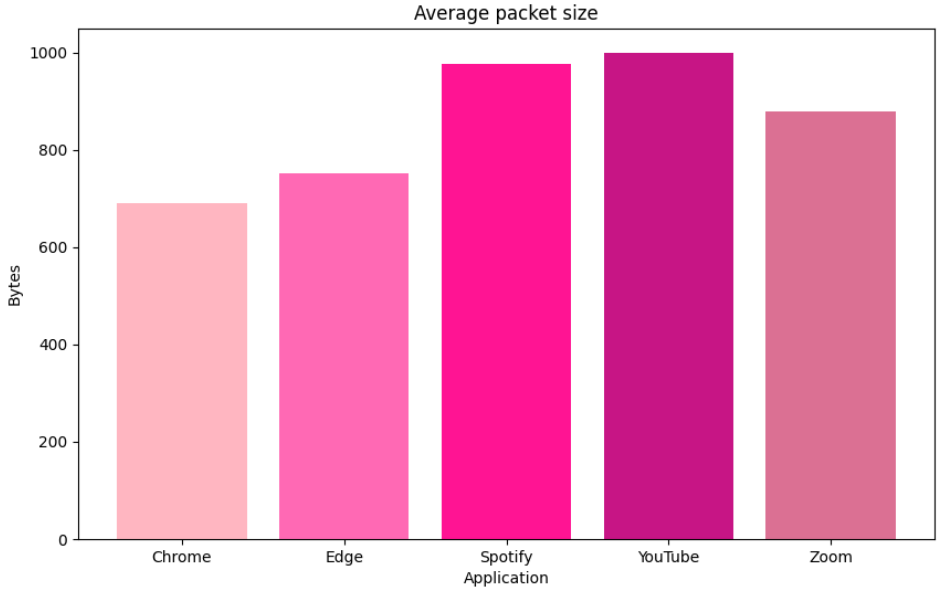
סיווג קטגוריה מול כל השאר- עבור NON-VPN- 97% דיוק, עבור VPN – 99.7% דיוק, עבור Tor – 85.7% דיוק.

סיווג טכניקות הצפנה- דיוק כולל 88.4%.

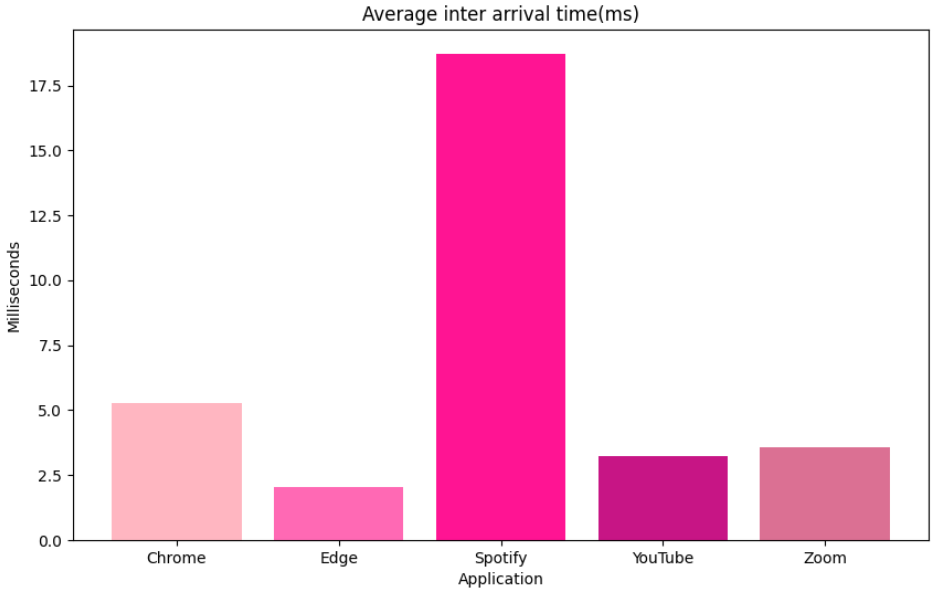
סיווג יישומים לא מוכרים- סיווג בהצלחה יישומים שלא היו חלק מנתוני האימון- דיוק מעל 80%.

תובנות-
שיטת ההמרה לתמונות מסווגת תעבורה מוצפנת ביעילות תוך שמירה על פרטיות ללא בדיקת תוכן החבילות, היא עמידה בקטגוריות הצפנה ותעבורה שונות, יכולת החקירה של התמונות מספקת סיווג טוב. המודל יכול לסווג נכון יישומים חדשים שלא נראו במהלך האימון מה שמצביע על כך שהוא לומד את המאפיינים המהותיים של קטגוריית תעבורה ולא התנהגויות ספציפיות של יישומים.

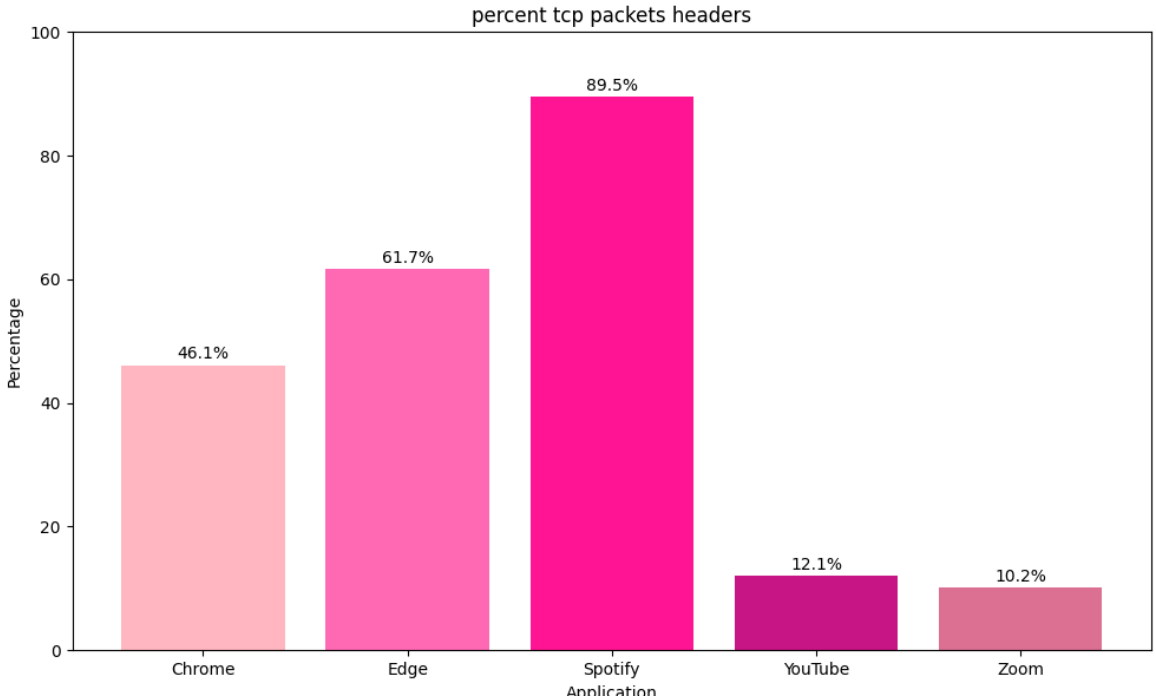
חלק 3- בחלק זה נעזרנו בבינה המלאכותית, בשאלות על עבודה עם הקלטות ווירשארק בקוד.



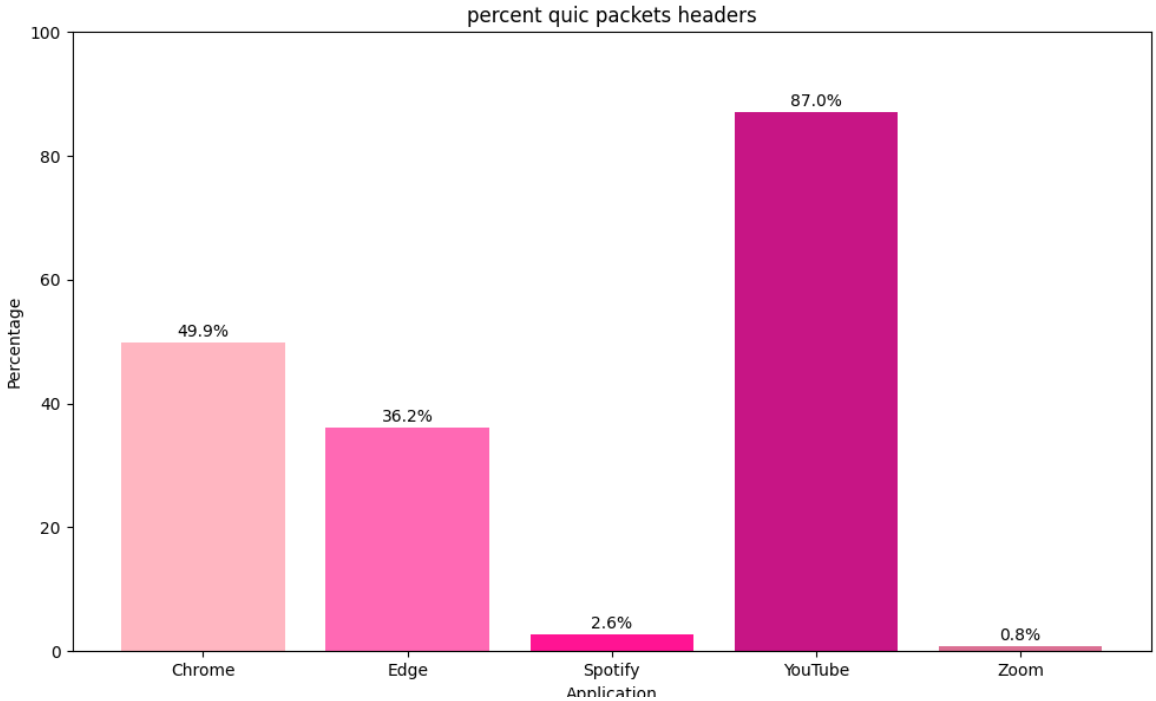
השוואת גודל ממוצע של החבילות בין כל הקלטה של הרצת האפליקציות. ניתן לראות כי גודל החבילות הממוצע של אפליקציות סטרימינג- שידור תוכן בצורה רציפה גדול יותר מאשר גודל החבילות של הקלטת גלישה בדפדפן אינטרנט. ניתן לראות כי ממוצע גודל החבילות של גלישה ביוטיוב הוא הגבוה ביותר ואילו ממוצע גודל החבילות בגלישת דפדפן chrom הוא הקטן ביותר.



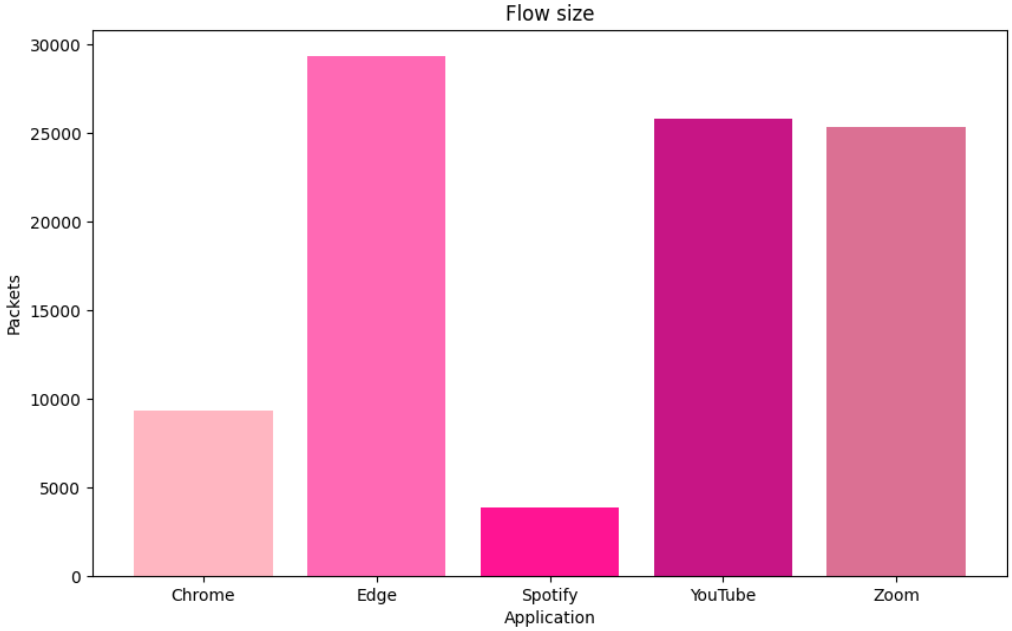
הגרף מציג את הזמן הממוצע במילישניות בין חבילה לחבילה של חבילות עוקבות. ניתן לראות כי בתעבורת הרשת בהרצת spotify ממוצע הפרשי זמנים של הגעה בין חבילות גדול יותר מכל שאר הרצות האפלקציה האחרות. ואילו בהרצת דפדפן Edge Microsoft ניתן לראות שממוצע הפרשי הזמנים של הגעה בין חבילות עוקבות הוא הקטן ביותר. הדבר נכון מכיוון שספוטיפי לא דורש הרבה נתונים ולכן יכול להרשות לעצמו עיכובים בשליחה.



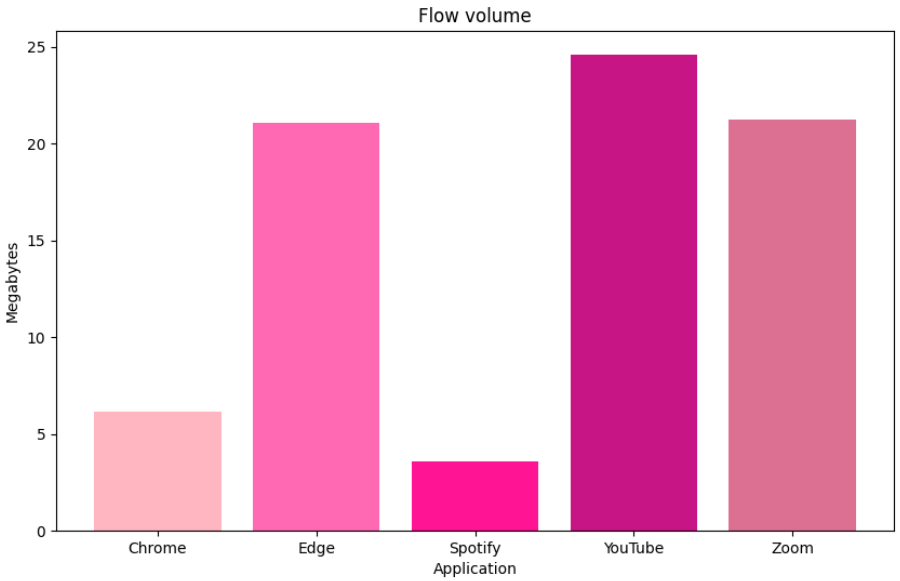
הגרף הנ"ל מציג את אחוזי החבילות שמכילות headers של TCP מתוך כלל החבילות של הקלטות הרצת האפליקציות. ניתן לראות כי בהקלטת הרצת spotify ו-Edge Microsoft מעל לחצי מהחבילות מכילות headers של TCP משמע מעל לחצי מהחבילות משתמשות בפרוטוקול TCP – מועברות בצורה מהימנה. ואילו youtube ו-Zoom מכילות אחוזים מועטים של חבילות בעלות header של פרוטוקול TCP וזה אכן כמצופה מכיוון ששירות וידאו בכדי לקבל איכות צפייה טובה ורציפה אינו משתמש בפרוטוקול TCP ויעדיף להשתמש בפרוטוקולים כמו QUIC , DATA ועוד שהם מעל UDP.



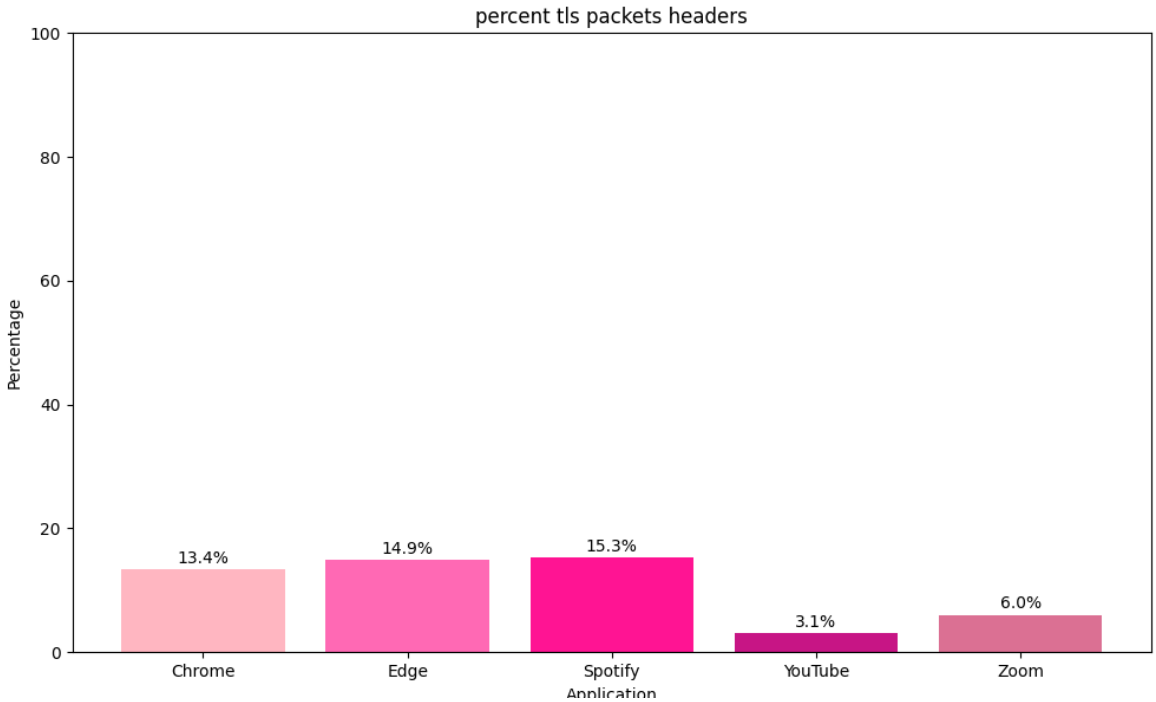
גרף זה מציג את אחוזי החבילות שמכילות headers של פרוטוקול QUIC מתוך כלל החבילות של הקלטות הרצת האפליקציות. ניתן לראות כי אחוזי החבילות שמכילות header של פרוטוקול QUIC בהרצת אפליקציית youtube הם הגבוהים ביותר מכיוון ששירות youtube אכן נדרש להשתמש בפרוטוקול מהיר ושלא מהימן כמו TCP בכדי לספק איכות צפיית וידאו איכותית.



גרף זה מציג את המספר הכולל של חבילות שהועברו עבור כל אפליקציה. הדבר תלוי ומשתנה בהתאם לזמן שימוש של המשתמש באפליקציה וגם בכמות החבילות שדורשת הפעילות הספציפית בשירותי האפליקציה. הגרף מתאר את צורת השימוש שלנו בהרצת כל אחת מהאפליקציות הנ"ל. מדד זה נמוך מאוד בspotify מכיוון שהוא צורך פחות נתונים ומשתמש בשיטות דחיסה יעילות, הוא צורך פחות חבילות מדפדפנים שמורידים מגוון תוכן.



גרף זה מתאר את כמות המידע הכולל במגה-בייטים שהועבר בהרצת כל אפליקציה.
הגרף מציג סכום כולל של כל גדלי החבילות זהו נפח הזרימה עבור תעבורת רשת זו.
גם בגרף זה התוצאות הן בהתאם לשימוש שלנו בכל אפליקציה. ניתן לראות כי נפח הזרימה של תעבורת הרשת בהרצת אפליקציית יוטיוב הוא הגדול ביותר משמע כמות המידע הכולל בהפעלת יוטיוב הוא הגדול ביותר.



גרף זה מציג את אחוזי החבילות שמכילות headers של פרוטוקול TLS מתוך כלל החבילות של הקלטות הרצת האפליקציות. ניתן לראות כי אחוזי החבילות שמכילות header של פרוטוקול TLS בהרצת אפליקציית spotify הם הגבוהים ביותר ואילו של יוטיוב הם האחוזים הקטנים ביותר כי יוטיוב משתמש בQUIC. האפליקציות שמשתמשות בגלישת דפדפן משתמשות יותר בהצפנה-TLS .

ב"ה

הקוד שבנינו מנתח ומפיק מתוך הפקטות של כל הקלטת תעברורה את- מספר הפקטות הכולל של ההקלטה, את מספר הפקטות שמכילות headwrs של IP, TCP, TLS, QUIC, הקוד מחשב את ממוצע גודלי החבילות, את ממוצע הפרש הזמנים בין כל שתי חבילות עוקבות, את גודל הזרימה ונפח הזרימה.

עבור הרצת דפדפן כרום-

```
===Chrome summary===
packet count: 9346
ip header: 9342
tcp header: 4312
tls header: 1254
quic header: 4663
avg packet size: 690.498
avg inter arrival time: 5.26977
flow volume(MB): 6.154
flow size: 9346
```

עבור הרצת דפדפן אדג'-

```
===Edge summary===
packet count: 29339
ip header: 29337
tcp header: 18090
tls header: 4371
quic header: 10607
avg packet size: 752.374
avg inter arrival time: 2.0416
flow volume(MB): 21.051
flow size: 29339
```

עבור הרצת ספוטיפי-

```
===Spotify summary===
packet count: 3861
ip header: 3855
tcp header: 3456
tls header: 592
quic header: 102
avg packet size: 977.439
avg inter arrival time: 18.71261
flow volume(MB): 3.599
flow size: 3861
```

עבור הרצת יוטיוב-

```
===YouTube summary===
packet count: 25795
ip header: 25785
tcp header: 3122
tls header: 802
quic header: 22450
avg packet size: 999.808
avg inter arrival time: 3.21972
flow volume(MB): 24.595
flow size: 25795
```

עבור הרצת זום-

```
===Zoom summary===
packet count: 25332
ip header: 25318
tcp header: 2576
tls header: 1529
quic header: 195
avg packet size: 878.392
avg inter arrival time: 3.58366
flow volume(MB): 21.221
flow size: 25332
```

ב"ה

שאלה 4-

אפשרות ראשונה בה התוקף יודע עבור כל חבילה את גודלה, את חותמת זמן הלכידה שלה וגיבוב tuple 4 תאפשר לתוקף לדעת בסבירות גבוהה מאוד לזהות שימוש באפליקציות שונות.
מכיוון שידיעת הtuple 4 יכולה לעזור לבנות פרופיל תעבורה, גם בלי גישה לתוכן, התוקף יכול לנתח את תבניות התעבורה (גודל חבילות, קצב, כיוון..) על ידי קיבוץ חבילות לפי הhash של ה-flow ID, ולהבין את התנהגות האפליקציה.
התוקף יכול להשוות את הנתונים שלו לחתימות ידועות מראש של אפליקציות פופולריות.
למרות שהוא רואה רק את הhash של כתובות הIP הוא יכול לדעת מתי משתמש מתקשר עם אותו שרת, יכול לראות מתי משתמש מתחיל ומסיים תקשורת עם שרת, גם אם התהליך מוצפן (על ידי תבניות קבועות של חבילות בגדלים ספציפיים או על ידי סדר החיבורים, לדוגמא לרוב מתחיל בחיבור לDNS ועוד..)

אפשרות שנייה בה התוקף יודע עבור כל חבילה את גודלה ואת חותמת זמן הלכידה בלבד, עדיין תאפשר לתוקף לזהות שימוש באפליקציה שונות במידה מסוימת, אבל פחות מדויק.
מכיוון של ידי ידיעת גודלי החבילות וזמני הגעתן נוכל לחשב התפלגות של הגדלים והזמנים וליצור חתימה ייחודית לכל אפליקציה גם ללא ידיעה לאיזה שרת הם שייכים, נוכל לחשב גם ממוצע וסטיות תקן ולזהות דפוסים ברורים של תבניות תעבורה (לדוגמא, עבור שירותי סטרימינג נרצה לראות פרצי חבילות גדולים באופן קבוע ועבור גלישה רגילה נרצה פרצי חבילות לא סדירים)
יכולת הזיהוי של התוקף תהיה נמוכה יותר מאפשר באפשרות א', אך עדיין אפשרית, במיוחד לאפליקציות עם דפוסי תעבורה מובהקים.
כתבנו קוד על אפשרות זו, שע"י ניתוח גודלי החבילות וזמני לכידתן, מנסה לשער מה הפעולה שאת תעבורת הרשת שלה הקליט המשתמש. בעזרת גודלי החבילות חישבנו את ממוצע גדלי החבילות. בעזרת זמני הלכידה, חישבנו את סטיית התקן של הזמנים בין כל 2 חבילות עוקבות. ובעזרת שני הנתונים יחד חישבנו את ה- bitrate – כמות הביטים המועברים בשנייה- (כאשר bitrate גדול אז מועברים הרבה נתונים מה שמאפשר איכותה גבוהה יותר). בקוד זה הסתמכנו על נתונים שקיבלנו מ-AI, ביקשנו נתונים סטנדרטים של גדלי חבילות, סטיית תקן של זמנים בין חבילות וכמות ביטים לשנייה של הפעלת- Video Streaming, Video Cals, Audio Streaming, Internet Browsing. לפי הנתונים שקיבלנו ניסינו לסווג את ההקלטות, אחוז ההצלחה הם 60% בהקלטות שלנו הקוד מצליח לזהות את הפעלת Chrom, YouTube, Zoom. נתוני ה-AI הם הטווחים בעבור ממוצע גודלי חבילות, סטיית תקן של זמנים בין חבילות עוקבות, וכמות ביטים לפנייה וכן ככתוב בקוד-

```
if 950 < avg_packet_size < 1460 and 10 < std_time_between < 25 and 2 < bitrate < 8:
    print("The identified traffic type is: Video Streaming\n")
elif 500 < avg_packet_size < 1200 and 5 < std_time_between < 25 and 0.5 < bitrate < 3:
    print("The identified traffic type is: Video Calls\n")
elif 150 < avg_packet_size < 450 and 3 < std_time_between < 8 and 0.1 < bitrate < 0.3:
    print("The identified traffic type is: Audio Streaming\n")
elif 600 < avg_packet_size < 1400 and 50 < std_time_between and 0.1 < bitrate < 5:
    print("The identified traffic type is: Internet Browsing\n")
```

```
analyzing Spotify...
=== summary ===
avg_packet_size: 977.439
std_time_between: 182.448
bitrate: 0.418
The identified traffic type is: Internet Browsing
```

```
analyzing Chrome...
=== summary ===
avg_packet_size: 690.498
std_time_between: 60.912
bitrate: 1.048
The identified traffic type is: Internet Browsing
```

```
analyzing YouTube...
=== summary ===
avg_packet_size: 999.808
std_time_between: 13.926
bitrate: 2.484
The identified traffic type is: Video Streaming
```

```
analyzing Edge...
=== summary ===
avg_packet_size: 752.374
std_time_between: 23.264
bitrate: 2.948
The identified traffic type is: Video Cals
```

```
analyzing Zoom...
=== summary ===
avg_packet_size: 878.392
std_time_between: 20.082
bitrate: 1.961
The identified traffic type is: Video Cals
```

כיצד נוכל למתן התקפה זו?
עבור אפשרות א'- נוכל לשלוח חבילות שאין בהם מידע אמיתי, רק ליצור רעש במקביל לתעבורה האמיתית ובכך להקשות על התוקף להות מתי יש תקשורת אמיתית.
בנוסף אפשר להשתמש ברשתות אנונימיות כמו Tor, התעבורה תעבור דרך מספר שרתים שכל אחד מהם רואה רק חלק מהמסלול ובכך להקשות על מעקב אחרי החבילות. אפשר לעשות החלפת נתיבים כלומר לשנות את כתובת הIP או הפורטים באופן תדיר, ובכך להקשות על יצירת פרופיל מתמשך של התעבורה.
עבור אפשרות ב'- נוכל לרפד חבילות כלומר במקום לשלוח חבילות בגדלים שונים, מגדילים את כל החבילות לגודל אחיד ובכך להפחית את היכולת של התוקף לזהות סוגי תוכן.
בנוסף אפשר במקום לשלוח נתונים כשצריך, לשלוח אותם בקצב קבוע וידוע מראש ללא תלות בכמות המידע האמיתית שצריך להעביר ובכך להסתיר את דפוסי הזמן שיכולים לרמז על אופי הפעילות.
- שימוש בכמה מדרכי המיתון יביא להגנה טובה יותר מאשר כל אחד מהם בנפרד, זה יצור שכבות הגנה שיקשו מאוד על התוקף לזהות את האפליקציות בשימוש.