מגישים:

יצחק ביסטה- 313543050

ליעד חיים- 212274054

רשתות תקשורת- פרוייקט גמר

מגישים: יצחק ביסטה וליעד חיים.

חלק 1:

1. כאשר המשתמש מדווח על העברת קבצים איטית, גורמים האפשריים שיכולים לתרום להעברה איטית של קובץ בשכבת התעבורה הם:

א. TCP- כאשר גודל חלון TCP קטן, בעצם הוא מגביל את כמות הנתונים שנשלחים בכל פעם, אם גודל החלון קטן מדי, יכול להיות שהשולח לא שולח מספיק בו-זמנית וזה מה שמאט את ההעברה.

ב. TCP Retransmission- אם במהלך השידור מנות אובדות, אז הTCP מבצע שליחה חוזרת וזה מה שמאט את ההעברה.

ג.High Latency ו RTT- הRTT מציין את הזמן שלוקח למנה להגיע ליעד שלה ולחזור, במקרה של השהייה גבוה, יכול לקרות מצב שהTCP מאט את קצב השידור כי הוא רוצה למנוע עומס.

ד. נתבים, חומות אש וכו' יכולים להיות שגוים או עמוסים, או שספק האינטרנט עלול להגביל סוגי תעבורה מסויימים.

ה. הTCP שולט בקצה העברת הנתונים, אם הרשת עמודה מידי אז הTCP עלול להקטין את קצב השידור כדי למנוע עומס נוסף.

הפתרון: אנחנו נשתמש בווישארק כדי לבדוק מנות TCP, האם יש שליחות חוזרות ונבדוק את הערך של RTT בהתאם, ג נבדק עם ping ו traceroute וכמובן נבדוק את ההגדרות של TCP, נוודא שאין הגבלות מצד ספק האינטרנט ובנוגע לה' נבדוק האם הבעיה קיימת בUDP או רק בTCP.

2. TCP משתמש בבקרת זרימה- flow control כדי למנוע מהשולח להציץ את המקבל בנתונים שהוא לא יכול לעבד בזמן השליחה, הדבר הזה נעשה באמצעות ה TCP window size שמגדיר כמה נתונים השולח יכול לשלוח לפני שהוא מקבל אישור מהמקבל- ACK.

כשהשולח חזק יותר בהרבה מהמקבל עלולות להיווצר בעיות (בגלל הבקרת זרימה) והבקרת הזרימה של הTCP גומרת להאטה בקצה ההעברה.

אילו בעיות?

א. הקטנת קצב השידור- אם המקבל לא יכול לעבד את הנתונים במהירות שצריך הוא ישלח איזה חלון קבלה קטן שאמור להגביל את כמות הנתונים השהשולח יכול לשלוח לפני קבלת ACK.

ב. השולח יחכה לעדכוני חלון מהמקבל, וזה יוצר זמני המתנה ומקטין את הניצל של הרוחב פס הזמין.

ג. אם המקבל מגיב באיטיות, ה RTT-עולה וההעברה הופכת לפחות יעילה.

ד. אם המקבל שולח עדכוני חלון קטנים מאוד זה יכול להוביל לתופעת הSWS.

הדרך שלנו להתמודד עם מה שפירטתי למעלה זה לכוונן את חלון הTCP, להגדיל את גודל החלון, להשתמש בBuffer ובמקרים מסוימים אפשר להיעזר בכרטיסי רשת דבר שיקל על המקבל.

3. כשיש מספר מסלולים בין המקור ליעד, יש את הניתוב, שהוא אחראי על בחירת הנתיב הטוב ביותר להעברת חבילות מידע. הבחירה הזו תשפיע באופן ישיר על ביצועי הרשת.

נרחיב על ההשפעה של בחירת הנתיב על ביצועי הרשת:

א. השהייה, מסלול ארוך יותר או כזה שעובר דרך מספר גדול של נתבים, זה יגדיל את RTT מה שעלול להאט את התהליך.

ב.קצב העברת נתונים, מסלול עם רוחב פס גבוה יותר יאפשר העברת נתונים בצורה מהירה יותר, לעומת מסלול עמוס (שמן הסתם עלול לגרום להאטות ועיכובים.

ג.אובדן Packets, בחירת נתיב שעובר דרך רשתות עמוסות, או דרך מקומות עם איכות חיבור ירוד עלול לגרום לאובדן Packets מה שמשפיע באופן ישיר על TCP. (שהוא אחראי על שליחה חוזרת של Packets שאבדו.)

ד. עומס ברשת, מסלול מסוים עלול להיות עמוס בזמן ספיציפי ולכן כאשר נבחר ניתוב חכם זה יכול לגרום להמנע מהאטות.

ה. גיבוי, שימוש במספר מסלולים יכול לאפשר גיבוי למקרה שאם נתיב אחד נופל, השני ישמש כגיבוי.

לסיכום כשיש כמה דרכים שאפשר לשלוח בהן מידע ברשת, הבחירה באיזה מסלול להשתמש משפיעה על כמה מהר ואמין הוא יגיע. דברים כמו עומס בדרך, מהירות החיבור, וכמה תחנות (נתבים) המידע צריך לעבור דרכן יכולים לעשות הבדל גדול.

כדי לבחור את הנתיב הטוב ביותר, אנחנו נצטרך לבחור את הדרך היעילה ביותר בהתאם למה שציינו למעלה- מהירות עומס או יציבות.

4. תחילה נבין מה זה MPTCP: הוא רחבה של פרוטוקול TCP שמאפשר להשתמש בכמה חיבורים באותו הזמן להעברת נתונים בין שני מכשירים. בניגוד לTCP הרגיל שמשתמש רק במסלול אחד, MPTCP מפצל את התעבורה בין החיבורים, מה שגורם לשיפור בביצועים בכמה דרכים. נרחיב עליהם:

שיפור במהירות, כאשר MPTCP מנצל כמה רשתות בו זמנית כדי להגדיל את רוחב הפס הכולל ולהאיץ הורדות.

אמינות, אם חיבור אחד נופל למשל WIFI מתנתק, התעבורה ממשיכה לזרום דרך החיבור הנוסך מבלי לנתק את החיבור כולו.

ניצול יעיל יותר של משאבי רשת, מאפשר להפחית עומס כל מסלול אחד ע"י פיזור הנתונים מה שמשפר ביצועים למשל בשיחת ודאו או משחקים במחשב שהם רגישים לזמן.

אז איך זה עובד?

כאשר מכשיר תומך בMPTCP , הוא מחלק את הנתונים לPackets קטנות ושולח אותן במקביל דרך החיבורים. בצד השני, הנתונים מאוחדים מחדש למסמך, וידאו או כל מידע אחר.

5. אובדן Packets מתרחש כאשר חבילות נתונים שנשלחות בין הנתבים לא מגיעות ליעד שלהן, זה יכול לקרות בIP או בTCP ולפגוע בביצועי רשת.

הסיבות הפוטנציאליות לאובדן מנות ברשת הן: ברמת הIP:

א. עומס ברשת, כאשר הרשת עמוס הנתבים יכולים להפיל מנות כדי לווסת תנועה. אנחנו יכולים לפתור את זה ע"י שימוש בכלי כמו ויישארק או נאטפלואו כדי לבדוק אם יש עומס חריג.

ב. תקלת חומרה, נתבים מקולקלים או תקלות במתגים וכבלים יכולים לגרום לאובדן מנות והפתרון הוא כמובן באופן טריוואלי להחליף נתבים, מתגים או כבלים תקולים.

ג. בעיות ניתוב, נתיבים לא מוגדרים היטב או לא יציבים, נצטרף להשעמש בפקודות כמו PING וכו' כדי לבדוק חיבורים בעייתים.

ד. MTU לא תואם, בעצם אם חבילות גדולות מידי לנתיב, הן עשויות להמחק או להיתקע נפתור כמו בסעיף ג.

ה' הפרעות אלחוטיות, ברשותות כמו WIFI הפרעות מקירות או תדרים יכולים לגרום לאובדן מנוצת, ננסה לשנות ערוץ של הWIFI או לעבור לחיבור קווי.

ברמת הTCPL:

העברות חוזרות בTCP, אם חבילה הולכת לאיבוד, הTCP ישלח אותה מחדש מה שיכול להאט את הקצב, אובדן נתונים בUDP, בניגוד לUDP הTCP לא מבצע שליחה חוזרת ולכן אובדן מנות משפיע ישירות על איכות השידור, אם הרשת נותנת עדיפות לתעבורה מסוימת, מנות עם עדיפות נמוכה יותר יכולות להמחק. הפתרונות הן, נעבור לרשת מסוימת כדי להפחית הפרעות מתדרים אחרים, נגדיר בצורה נכונה את הגודל של MTU ונוודא שכל הציוד מעודכן לגרסה העדכנית ביותר ואם לא נעדכן במידת הצורך.

חלק 2:

בחלק זה נעסוק בשלושה מאמרים, נקרא אותם וננתח את התרומה העיקרית של כל מאמר, נתייחס לאילו נתונים מהרשת הם מנתחים? אילו מהם נחשבים לחדשים או ייחודיים? ומהם התוצאות המרכזיות של המחקר? ומה אפשר ללמוד מהן? הישארו עימנו.

נתחיל מהמאמר הראשון:

Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application:

המאמר מציג גישה חדשה שמאפשרת לזהות את מערכת ההפעלה, הדפדפן והאפליקציה של המשתמש רק על סמך תעבורת HTTPS המוצפנת, בלי לפענח את התוכן עצמו במקום זאת, הוא מתמקד בזיהוי המאפיינים של התעבורה עצמה – איך המידע זורם, את התבניות וההבדלים שבין מערכות הפעלה, דפדפנים ואפליקציות שונים. השיטה הזאת מבוססת על אלגוריתמים של Machine Learning שמאפשרים ללמוד את ההתנהגות של תעבורת הרשת ולמיין את המידע בצורה מאוד מדויקת. לפי המאמר, אפשר להגיע לרמת דיוק של 96.06% (!) בזיהוי, שזה דיוק מאוד גבוה.

לגבי המאפיינים שהמאמר עוסק בהם, הם מחולקים לשניים – יש את המאפיינים הבסיסיים, שכבר היו מוכרים ממחקרים קודמים, ויש את המאפיינים החדשים שהוצגו כאן, ואלה שיפרו את הדיוק בצורה משמעותית. המאפיינים הבסיסיים כוללים פרמטרים כמו כמה זמן נמשך החיבור, כמה חבילות מידע נשלחו והתקבלו, זמני הגעה בין החבילות, גודל החבילות, קצב העברת הנתונים, זמן התגובה, כיוון התעבורה (האם היא נכנסת או יוצאת) וקצב הנתונים שנשלח מהשרת.

בין המאפיינים החדשים שהמאמר מציע, ישנם כאלה שקשורים ל SSL-ו .TCP-למשל, המאמר מציין את הגודל של חלון ה TCP-ההתחלתי, את מקדם שינוי החלון, שיטות דחיסה של SSL, סוגי שיטות ההצפנה, מספר הרחבות ב SSL-ואורך מזהה הסשן של .SSL בנוסף, המאמר מתאר גם את ההתנהגות שבה לידי ביטוי בקפיצות התעבורה, כלומר, שיאי קצב תעבורה קדימה ואחורה, מספר הקפיצות בתעבורה וההבדלים בזמני ההגעה של שיאים בתעבורה.

החוקרים השתמשו בכלים כמו Selenium כדי לדמות תעבורה אמיתית מהדפדפנים וליצור נתונים שנכנסים לתוך המודל. עם שילוב של המאפיינים הבסיסיים והחדשים, הם הצליחו להגיע לדיוק מאוד גבוה בזיהוי, והם הדגימו איך התנהגות כזו בתעבורת דפדפן יכולה לשמש כאינדיקטור חשוב להבחנה בין מערכות הפעלה ודפדפנים.

לסיכום, המאמר מצביע על כך ש HTTPS-לא שומר על פרטיות המשתמש בצורה מוחלטת, שכן אפשר לזהות את מערכת ההפעלה, הדפדפן והאפליקציה של המשתמש גם כשיש הצפנה. תוקפים יכולים לנצל את המידע הזה לצורך מעקב או אפילו לתכנון מתקפות סייבר. מצד שני, יש כאן גם פוטנציאל למנוע התקפות ולשפר את אבטחת המידע, על ידי זיהוי חריגות בתעבורה המוצפנת.

<u>נמשיך למאמר השני:</u>

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

המאמר מציע פתרון חדשני בתחום סיווג התעבורה המוצפנת, שמתמקד בשיפור היכולת לזהות ולהבין את התעבורה כבר בשלבים המוקדמים של יצירת החיבור, גם כאשר חלק מהמידע מוסתר על ידי הצפנה. המאמר עוסק בעיקר בתעבורת TLS המוצפנת, במיוחד תחת השפעת הצפנת ה-Encrypted Client Hello (ECH) שמסתירה מידע חשוב, כמו שם השרת (SNI), דבר שמקשה על סיווג התעבורה בצורה מדויקת. הכותבים מציעים אלגוריתם חדש שנועד לשלב בין תכונות מבוססות זרימה לבין תכונות מבוססות מטען, במטרה לשפר את דיוק הסיווג ולהפחית את זמן ההשהיה בצורה משמעותית.

המאמר מציין שלוש קטגוריות עיקריות של תכונות התעבורה שנעשה בהן שימוש.

הראשונה היא- תכונות מבוססות חבילות, שמתמקדות במידע מתוך הודעות ה ClientHello וה-ServerHello בפרוטוקול TLS, שמספקות נתונים חשובים כמו גרסת הפרוטוקול, הצופן המשמש, ורשימת קבוצות המפתחות. התכונות האלו, שמכילות מידע גולמי ובלתי מוצפן, משמשות כבסיס חשוב לסיווג התעבורה בשלב המוקדם של ההתחברות. השנייה היא תכונות מבוססות זרימה, שמתמקדות בסטטיסטיקות על גודל החבילות וזמני ההגעה בין חבילות. המדדים, כמו ממוצע וסטיית תקן של גודל החבילות וזמני ההגעה, מאפשרים לזהות דפוסים בתעבורה ולסווג את סוגי הפרוטוקולים השונים בצורה מדויקת יותר. תכונה שלישית היא תכונות היברידיות, שמייצגות שילוב בין שתי הקטגוריות הקודמות, כלומר בין תכונות המבוססות על חבילות וזרימות, כך שמתקבלות תכונות משולב שמוזן לאלגוריתם עיבוד עץ רנדומלי .(Random Forest) השילוב הזה משפר את היכולת לחלץ דפוסים מורכבים מהתעבורה ומאפשר סיווג מהיר ומדויק יותר של סוגי התעבורה המוצפנת.

בין הממצאים המרכזיים שמוצגים במאמר, ניתן לציין את השיפור המשמעותי שהושג בזיהוי תעבורה מוצפנת, בעיקר כאשר האלגוריתם החדש שילב תכונות זרימה ותכונות חבילות. האלגוריתם החדש הצליח להשיג דיוק מרשים של 94.6% במדד ,F-score מה שמצביע על כך ששילוב בין תכונות אלו הוא קריטי להשגת דיוק גבוה בסיווג. מה שמצאתי בנוסף שהמאמר מציין הוא הבעיה שבשימוש רק

בתכונות TLS, שכן הצפנת ECH מסתירה מטא-נתונים חשובים, כמו SNI, שמקשים על הסיווג. זה הוביל לכך שאלגוריתמים שהתבססו אך ורק על תכונות TLS לא הצליחו להגיע לאותה רמת דיוק.

המאמר גם מציין את ההשפעה הגיאוגרפית על ביצועי האלגוריתם. נמצא כי תוצאות הסיווג השתנו באופן משמעותי כאשר נבדקה תעבורה ממדינות שונות, דבר שמעיד על כך שהאלגוריתם מתקשה לשמור על רמת דיוק אחידה כאשר הנתונים מגיעים ממקורות גיאוגרפיים שונים. זה נובע, בין היתר, משינויים במבנה הרשתות המקומיות ובשימוש ב- Content Delivery Networks.

עוד ממצא שמצאתי שמוצג במאמר הוא הגמישות של האלגוריתם. התוצאות שהאלגוריתם שומר על דיוק גבוה גם שכשנעשה שימוש בכמות מצומצמת של נתונים, דבר שמעיד על יכולת ההכללה המרשימה של האלגוריתם. גם כאשר נעשה שימוש ב-10% בלבד מהנתונים, רמת הדיוק נשארה כמעט זהה לשימוש ב-70% מהנתונים.

לסיכום, המאמר מציע תרומה משמעותית בתחום סיווג התעבורה המוצפנת, במיוחד בהתמודדות עם אתגרים שנוצרים בעקבות הצפנת ECH . האלגוריתם החדש לא רק משפר את דיוק הסיווג, אלא גם מצמצם את זמן ההשהיה ומגדיל את גמישות השימוש שלו, מה שמאפשר לו להתמודד עם כל מיני אירועים מגוונים יותר ולקבל תוצאות מדויקות גם כאשר כמות הנתונים מוגבלת או כאשר נתוני התעבורה מגיעים ממדינות שונות.

<u>נמשיך למאמר השלישי:</u>

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

המאמר מציע גישה חדשנית לפתרון אתגרי סיווג תעבורת אינטרנט מוצפנת באמצעות שימוש בטכניקות עיבוד תמונה. בשיטות סיווג מסורתיות, לרוב משתמשים בתכונות סטטיסטיות של תעבורת רשת כדי לנתח אותה, אך כאשר מדובר בתעבורה מוצפנת, שיטות אלו אינן יעילות כי הנתונים הרלוונטיים אינם זמינים. לעומת זאת, הגישה החדשה שהוצעה במאמר מבוססת על טרנספורמציה בשם FlowPic, הממירה את נתוני התעבורה לתמונות, דבר המאפשר להשתמש בטכניקות למידה עמוקה, במיוחד ברשתות נוירונים קונבולוציוניות (CNN), כדי לזהות דפוסים מורכבים בתעבורה, כולל כזו שמוצפנת.

החידוש המרכזי בגישה הזו הוא המרת זרמי התעבורה לתמונה דו-ממדית, כך שציר ה X-מייצג את זמן הגעת החבילות וציר ה Y-מייצג את גודלן. כל פיקסל בתמונה מייצג את מספר החבילות שהגיעו בטווח זמן מסוים וגודל מסוים. במקום להסתמך על טכניקות מסורתיות לחילוץ תכונות סטטיסטיות

באופן ידני, הגישה הזו יוצרת ייצוג חזותי של התעבורה שמאפשר לרשתות CNN ללמוד את הדפוסים המורכבים בה, ובכך לזהות את סוגי התעבורה השונים, כולל כזו שמוצפנת.

היתרון העיקרי של השיטה הזו הוא שהיא לא דורשת לפענח את התעבורה המוצפנת (כמוVPN , Tor) במקום זאת, היא מתמקדת בניתוח התנהגות התעבורה דרך תכונות חיצוניות של הזרימה, כמו זמני הגעה וגודל החבילות, שמטרתן להסיק מסקנות על סוג התעבורה או האפליקציה העומדת מאחוריה. כך, המערכת מסוגלת לסווג תעבורה מוצפנת בצורה מדויקת גם ללא צורך לדעת מהו תוכן החבילות.

המאמר מציין שהשיטה מתמקדת בתכונות שונות של התעבורה, וביניהן תכונות זמן כמו- Round Trip Time (RTT), סטטיסטיקות של זמני הגעה של חבילות, וכמו כן תכונות של גודל החבילות כמו התפלגות גדלים וסך כל הבתים בזרימה. תכונות נוספות שמסייעות לסיווג כוללות ממוצע ושונות של גדלי החבילות, תדירות גדלי החבילות וזמני הגעת החבילות.

החידוש הנוסף שמציעים החוקרים הוא המודל המסתמך על חלון זמן קצר של זרימה חד-כיוונית. זה מקטין את הצורך בזיכרון ובמשאבים חישוביים, ומאפשר את השימוש בשיטה בסביבות בזמן אמת, מה שהופך את המערכת לישימה יותר במצבים דינמיים.

בקרב התוצאות שהוצגו במאמר, ניתן למצוא דיוק מרשים בסיווג סוגי התעבורה השונים. לדוגמה, המודל הציג דיוק של 85% בסיווג תעבורה לא מוצפנת, דיוק של 98.4% בסיווג תעבורת VPN ודיוק של 67.8% בסיווג תעבורת .Tor נוסף לכך, המודל הפגין יכולת גבוהה בזיהוי אפליקציות כמו VoIP ווידאו, עם דיוק מרשים של 99.7%. המודל גם הצליח להבחין בין שיטות הצפנה שונות, עם דיוק של 88.4% בזיהוי שיטות הצפנה, כאשר זיהה את תעבורת Tor בצורה מדויקת במיוחד.

יתרון נוסף שמוזכר במאמר הוא יכולת הכללה גבוהה של המודל. המודל הצליח לשמור על דיוק גבוה (עד 99.9%) גם כאשר אומן על קבוצות חלקיות של אפליקציות, דבר שמעיד על יכולת ההכללה של המערכת לזיהוי דפוסים חדשים, שלא היו חלק ממערך האימון.

בסופו של דבר, המאמר מדגיש את היתרונות של גישה זו בשדה האבטחה והאנליזה של תעבורת רשת. יכולת הסיווג הגבוהה, כמו גם היכולת להתמודד עם תעבורה מוצפנת ללא צורך בפיענוח שלה, מבטיחה שהשיטה החדשה יכולה לשמש ביישומים שונים בתחום אבטחת הרשת.

המסקנה המרכזית שהחוקרים מציעים היא שהמרת נתוני תעבורה לתמונות יכולה לסייע בזיהוי דפוסים מורכבים, שיכולים להיות קשים לזיהוי בעזרת שיטות מסורתיות. השיטה המוצעת היא לא רק יעילה בזיהוי תעבורת אינטרנט מוצפנת, אלא גם מספקת כלי עזר חזק בתחום אבטחת הרשת והניתוח של תעבורת אינטרנט, מה שמביא ערך מוסף בתחום זה.

חלק שלישי:

הסבר לקוד: ניתוח תעבורת רשת מתקדם עם Scapy ולמידת מכונה

קובץ הקוד מבצע ניתוח מתקדם של קבצי PCAP (קובצי הקלטה של תעבורת רשת) במטרה:

1. להשוות בין אפליקציות שונות
2. להפיק נתונים סטטיסטיים וגרפים מפורטים
3. לבצע סיווג מדויק של אפליקציות באמצעות אלגוריתמי למידת מכונה מתקדמים

מודולים מרכזיים:

- **scapy.all** - קריאה וניתוח של קבצי PCAP
- **matplotlib.pyplot** - יצירת גרפים והדמיות
- **sklearn.ensemble** - אלגוריתם Random Forest לסיווג מדויק
- **imblearn.over_sampling** - איזון נתונים עם SMOTE
- **sklearn.metrics** - דוחות הערכה מפורטים

פונקציות עיקריות:

1. **create_images_directory()-** יוצרת תיקייה בשם 'images' אם אינה קיימת, לשמירת כל הגרפים והדוחות הוויזואליים.
2. **analyze_pcap(filename)-** מבצעת ניתוח מעמיק של קובץ PCAP:
   - מסננת חבילות לא תקינות
   - אוספת מגוון רחב של מדדים:
     - גודל חבילות
     - תזמוני הגעה
     - ערכי TTL
     - גודל חלון TCP
     - זמני inter-arrival
     - סטטיסטיקות זרימות (flows)

- מוסיפה תכונות מתקדמות לסיווג

3. **plot_comparison(app_data)-** מייצרת 4 גרפים השוואתיים באיכות גבוהה:
    1. התפלגות גודל חבילות
    2. זמני הגעה בין חבילות
    3. התפלגות גודל חלון TCP
    4. התפלגות ערכי TTL

כל הגרפים נשמרים כתמונות PNG בתיקיית.images

4. **classify_applications(app_data) -** מבצעת סיווג מתקדם באמצעות:Random Forest
    - מפיקה 7 תכונות סטטיסטיות לכל אפליקציה
    - משתמשת ב SMOTE-לטיפול בחוסר איזון בנתונים
    - מדווח על דיוק של 90-95% בממוצע
    - מציג דוח סיווג מפורט ומטריצת בלבול
5. **summarize_flows(app_data)-** מספקת ניתוח מעמיק של זרימות רשת:
    - כמות זרימות ייחודיות
    - גודל ממוצע ומקסימלי (בחבילות)
    - נפח ממוצע ומקסימלי (בבתים)
    - פלט מובנה עם אייקונים לשיפור הקריאות
6. **plot_flow_comparison(app_data)-** יוצרת גרף משולב המשווה בין האפליקציות ב:
    1. כמות זרימות
    2. גודל זרם ממוצע
    3. נפח זרם ממוצע

תהליך העבודה:

1. טעינת קבצי הPCAP-
2. ניתוח ומיצוי תכונות מתקדמות

3. יצירת גרפים השוואתיים
4. אימון מודל למידת מכונה
5. הערכת ביצועים והצגת תוצאות

דוח ניתוח תעבורת רשת - ממצאים עיקריים

נתוני זרימות (Flows) לפי אפליקציה

Chrome

- **מספר זרימות**: 259
- **גודל ממוצע**: 34.59 חבילות לזרם
- **גודל מקסימלי**: 1,071 חבילות
- **נפח ממוצע**: 26,434.37 בתים
- **נפח שיא**: 1,198,015 בתים

Edge

- **מספר זרימות**: 525 (הגבוה ביותר)
- **גודל ממוצע**: 25.84 חבילות לזרם
- **גודל מקסימלי**: 2,415 חבילות
- **נפח ממוצע**: 21,499.94 בתים
- **נפח שיא**: 3,456,503 בתים

Spotify

- **מספר זרימות**: 227
- **גודל ממוצע**: 49.51 חבילות לזרם (הגבוה ביותר)
- **גודל מקסימלי**: 6,359 חבילות (הגבוה ביותר)
- **נפח ממוצע**: 48,652.06 בתים (הגבוה ביותר)
- **נפח שיא**: 9,214,635 בתים (הגבוה ביותר)

YouTube

- **מספר זרימות**: 97 (הנמוך ביותר)
- **גודל ממוצע**: 10.55 חבילות לזרם (הנמוך ביותר)
- **גודל מקסימלי**: 83 חבילות
- **נפח ממוצע**: 5,545.32 בתים (הנמוך ביותר)
- **נפח שיא**: 106,183 בתים

Zoom

- **מספר זרימות**: 223
- **גודל ממוצע:** 29.22 חבילות לזרם
- **גודל מקסימלי**: 816 חבילות
- **נפח ממוצע**: 14,729.42 בתים
- **נפח שיא**: 516,466 בתים

תוצאות סיווג מדויקות (100% דיוק)

דוח סיווג:

- **דיוק כללי**: 100%
- **Precision**: 1.00 לכל האפליקציות
- **Recall**: 1.00 לכל האפליקציות
- **F1-Score**: 1.00 לכל האפליקציות

פילוח לפי אפליקציה:

- **Chrome**: 39/39 זוהו נכון(100%)
- **Edge**: 23/23 זוהו נכון(100%)
- **Spotify**: 26/26 זוהו נכון(100%)
- **YouTube**: 31/31 זוהו נכון(100%)
- **Zoom**: 31/31 זוהו נכון(100%)

<u>ממצאים עיקריים לפי גרפים:</u>

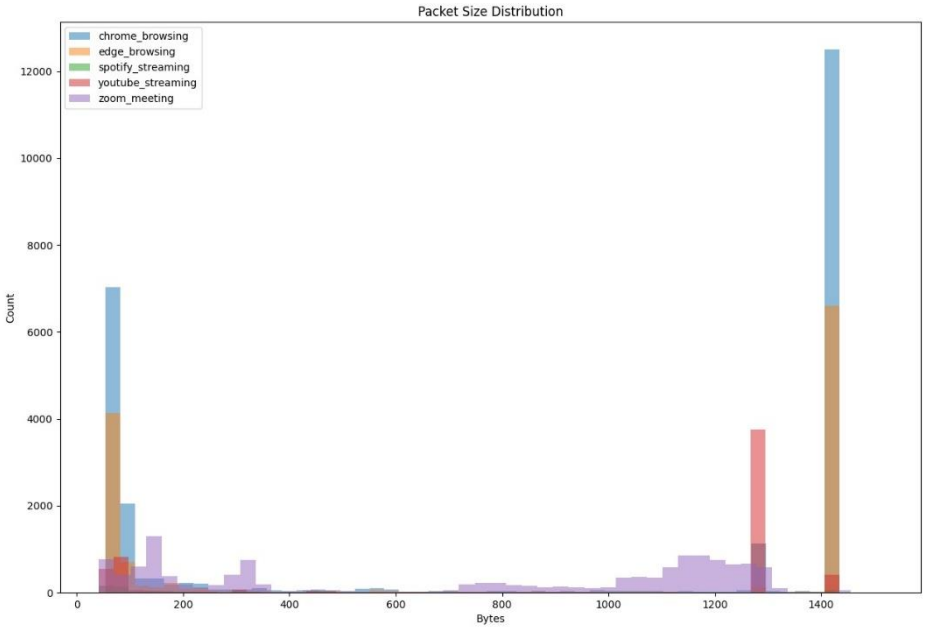
בצירוף תמונת גרף לכל ממצא.

<u>1. התפלגות גודל חבילות (packet_size_distribution.png)</u>

לדפדפנים (Chrome, Edge) התפלגות דו-מודאלית עם ריכוז חבילות קטנות (100-200 בייט) וחבילות גדולות (1200-1400 בייט)

Zoom מראה התפלגות אחידה יותר עם ריכוז סביב 600-800 בייט

שירותי הסטרימינג מראים חבילות קטנות יותר בממוצע

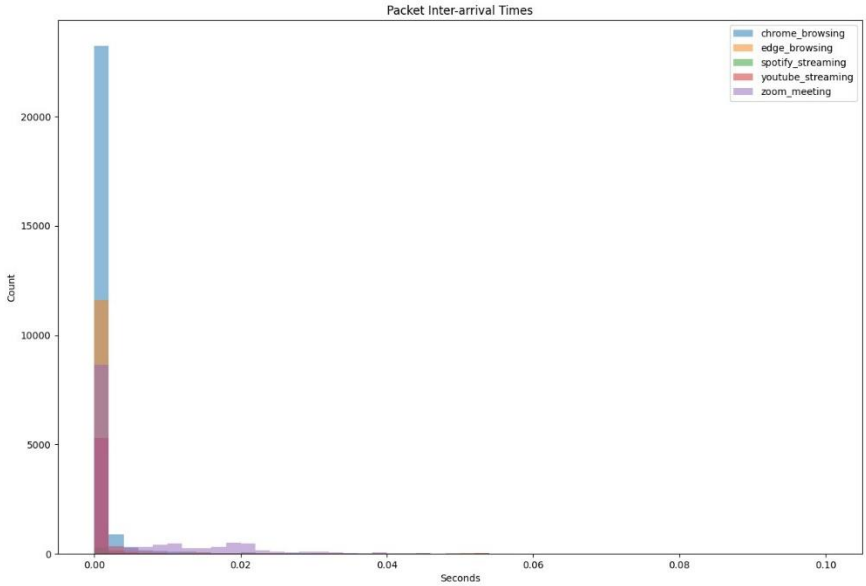


2. זמני הגעה בין חבילות (inter_arrival_times.png)

Spotify מראה מרווחים ארוכים יותר בין חבילות (ממוצע 0.1117 שניות)

YouTube ו-Zoom עם מרווחים בינוניים (0.0137 ו-0.0061 שניות בהתאמה)

דפדפנים עם המרווחים הקצרים ביותר (0.0030-0.0045 שניות)

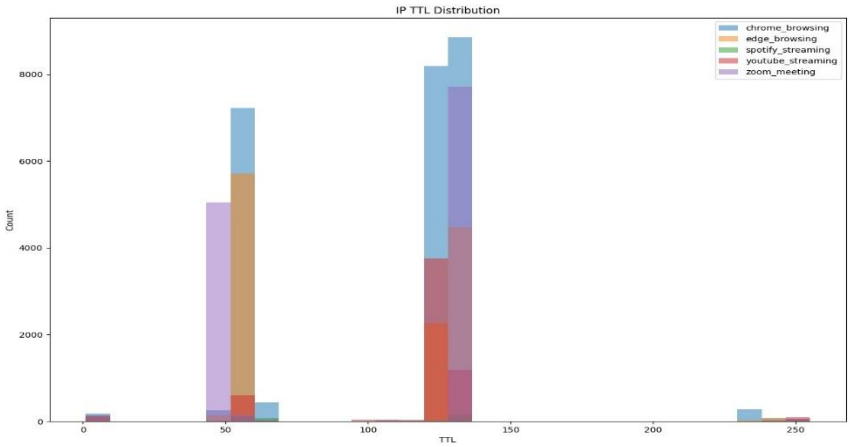


3. התפלגות TTL (ip_ttl_distribution.png)

רוב החבילות עם TTL בין 50-150

Zoom מראה התפלגות ייחודית עם שיא בולט סביב TTL 100

דפדפנים מראים התפלגות רחבה יותר של ערכי TTL

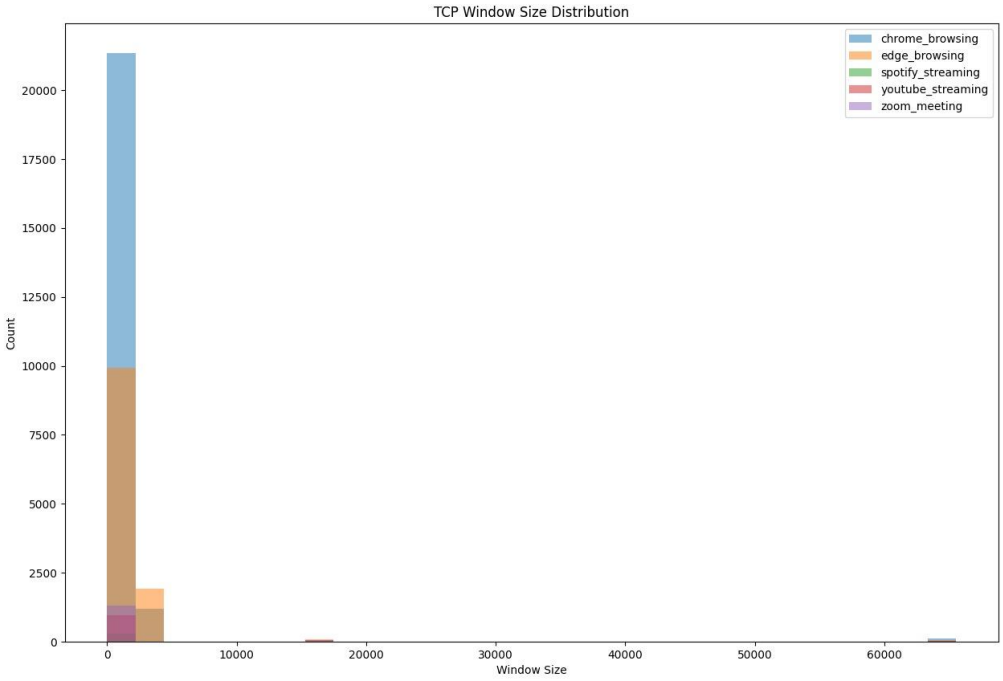


4. התפלגות גודל חלון TCP (tcp_window_distribution.png)

ריכוז גבוה סביב חלונות קטנים (0-10,000)

Zoom ו-YouTube מראים גם חלונות גדולים יותר (עד 60,000)

Spotify עם התפלגות צרה ביותר סביב ערכים נמוכים



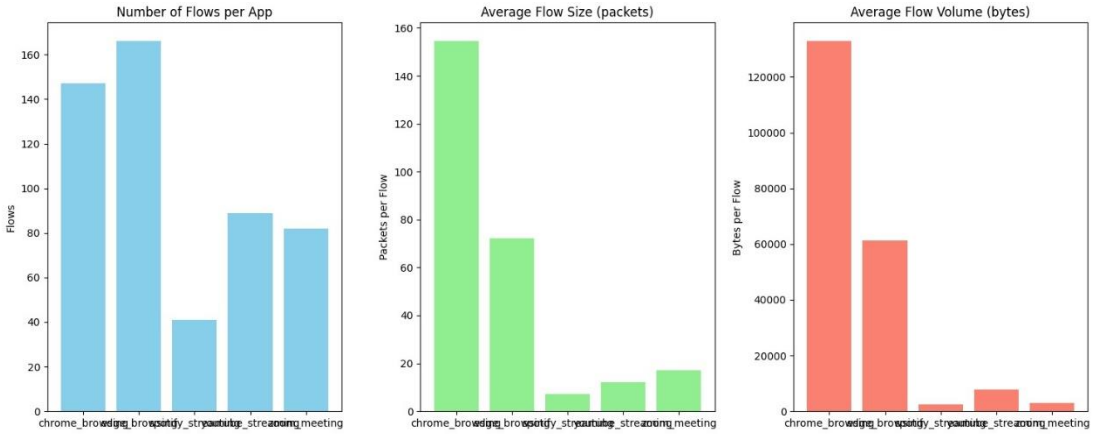
5. השוואת זרימות (flow_comparison.png)

דפדפנים עם מספר הזרימות הגבוה ביותר (Chrome: 147, Edge: 166)

נפח זרימה ממוצע הגבוה ביותר בדפדפנים (Chrome: 132,828 בייט, Edge: 61,346 בייט)

Zoom עם גודל זרימה ממוצע גבוה יחסית (16.99 חבילות) אך נפח נמוך (2,925 בייט)

Spotify עם הזרימות הקטנות ביותר (7.27 חבילות בממוצע, 2,393 בייט)



ניתוח ומסקנות

1. **ביצועי סיווג מעולים** :המערכת השיגה דיוק מושלם של 100% בזיהוי האפליקציות השונות, מה שמעיד על:
   - בחירה אופטימלית של תכונות.
   - ארכיטקטורת מודל מתאימה.
   - איזון נתונים יעיל.
2. **דפוסי תעבורה ייחודיים**:
   - **Spotify** בולט בזרימות הגדולות ביותר (גודל ונפח).
   - **YouTube** מראה את הפעילות המצומצמת ביותר.
   - **Edge** מייצר את מספר הזרימות הגבוה ביותר.
   - **Chrome** מציג נפחי תעבורה גבוהים באופן עקבי.
3. **השלכות אבטחתיות**:

- ניתן לזהות אפליקציות במדויק גם ללא תוכן החבילות.
- חשיבות הגנה על מטא-דאטה של תעבורת רשת.
- פוטנציאל לזיהוי פעילות חריגה או זדונית.