בס"ד

פרויקט גמר ברשתות תקשורת
שמות המגישים:

רון אברהם- 208007005

רעי שאול- 325390086

עידן שומסקי- 325693422

ליאור פיכמן- 215455502

חלק ראשון- מענה על השאלות הפתוחות

<u>שאלה 1</u>
להלן גורמים אפשריים לקצב העברה איטי של המשתמש במסגרת שכבת התעבורה:

א. גודל החלון ובקרת עומס:

אם חלון הקבלה rwnd של קבלת ההודעות window size קטן אז השולח מוגבל בכמות נתונים קטנה שהוא יכול לשלוח לפני קבלת כל אישור ACK
<u>איך היינו פותרים:</u> תחילה כדי לזהות את הבעיה היינו משתמשים בווירשארק, ובודקים את השדות של גודל החלון ב TCP Header. אז היינו מגדילים את ה TCP receive buffer כך שנוכל להעביר מידע בקצב מהיר יותר.

ב. אובדן חבילות וביצוע שידורים חוזרים(retransmission):

אם חבילה כלשהי נאבדת מאיזושהי סיבה, tcp ישלח אותה בשנית עם מנגנון retransmission מה שמאט את קצב ההעברה כי כל העברה לוקחת זמן.

<u>כדי לזהות את הבעיה</u> נשתמש בווישארק – ניתן לבדוק את אחוז איבוד חבילות, או ACK-ים כפולים.
כמו כן, ניתן להשתמש בפקודת traceout שלמדנו כדי לעקוב אחרי המסלול של החבילה ולזהות נקודות של עומס וכ"ו.

אפשרויות טיפול בבעיה- שינוי החיבור לאינטרנט קווי/wifi
הפחתת העומס ברשת- ננסה להפחית עומס אולי ע"י שימוש ב VPN

ג. RTT גבוה

נוכל לזהות זאת עם ווישארק והפקודות ping ו traceout
כדי לשפר את זמני ה RTT נבדוק אם אפשר להשתמש בשרת קרוב יותר פיזית. נשפר את החיבור לאינטרנט ואולי נפחית מהעומס על הרשת שאנחנו עובדים עליה נשהה פעולות אשר משתמשות ברשת שלנו בו זמנית יחד איתנו.
(אם לא היה מדובר בקובץ, היינו שוקלים להשתמש בפרוטוקול UDP במקום TCP)

ד. עומס על השרת או על תחנות הקצה

כאשר המחשב או השרת עמוסים אז ההעברה תהיה איטית. (נוכל לזהות ב task manager או ע"י פקודות ייחודיות ב cmd)
נפתור זאת ע"י שימוש בשרת חלופי אם אפשר.

<u>שאלה 2</u>

כאשר לשולח יש עיבוד גבוה יותר משל המקבל, זה יכול לגרום לעומס אצל המקבל.

בס"ד

קיים ב TCP header שדה rwnd (receive window) המייצג את כמות המקום הזמינה ב buffer.
המקבל דואג לעדכן את השדה הזה, וכאשר יש עומס השולח מאט את קצב השליחה או שהוא שולח בחבילות קטנות יותר בהתאם לעומס.

בכך הbuffer של המקבל לא יוצף (overflow).

שאלה 3

בחירת הנתיב היא עניין חשוב בהעברת הנתונים מכיוון שהיא יכולה להשפיע על כלל הרשת.

בחירת נתיב קצר יותר או בעל עלות נמוכה יותר יכולה להקטין את הזמן הדרוש להעברת חבילות מידע מהמקור ליעד, מה שמפחית עיכובים ומשפר זמני תגובה.

נעדיף נתבים שמשתמשים בצורה יעילה ברוחב הפס. בכך האיזון ברחבי הרשת נשמר

ונמנע עומס יתר על אזורים מסוימים.

וכמובן שתמיד נעדיף לבחור נתיבים אמינים להעברה בטוחה ואופטימלית, אשר מעידים על הרשת כאמינה.

כאשר אנו מדברים על בחירת מסלול ברשת מחשבים, נתייחס למספר גורמים חשובים שעלינו לקחת בחשבון:

עלות

כאשר מחפשים מסלול "טוב" מבחינת עלות, נרצה לבחור במסלול שעלות השימוש בו נמוכה יותר, מה שעשוי להביא לחיסכון בהוצאות התפעוליות של הרשת.

מהירות

מסלול טוב מבחינת מהירות הוא זה שמציע את העיכוב הנמוך ביותר ומהירות העברת נתונים גבוהה.
בחירה בנתיב שמזמן את המהירות הטובה ביותר היא קריטית ליישומים בעלי דרישות זמן אמת, כמו שיחות וידאו, משחקי מחשב ושאר יישומים מקוונים.

עומס

בחירת מסלול עם עומס נמוך יותר תורמת לשימור היציבות והאמינות של הרשת, מכיוון שפחות עומס על הקישורים יכול להוביל לפחות השהיות ופחות נקודות תקלה פוטנציאליות.

שאלה 4

MPTCP הוא קבוצה של הרחבות למפרט הפרוטוקול TCP בעזרת MPTCP לקוח יכול להתחבר לאותו מארח יעד עם חיבורים מרובים דרך מתאמי רשת שונים. כך נוצרים חיבורי נתונים חזקים ויעילים בין מארחים שעובדים עם תשתיות רשת קיימות.

שימוש ב MPTCP משפר את ביצועי השרת בכמה דרכים עיקריות-

א. למדנו ש MPTCP בעל אמינות גבוהה ויודע להתמודד טוב עם נפילה של נתיב
אם נתיב אחד נכשל MPTCP ישתמש בנתיבים האחרים להעברת הנתונים ובכך לא יפסיק את החיבור.
ב. אפשרות לחלוקת המידע הכולל- MPTCP יכול לחלק את המידע בין הנתיבים השונים בצורה מאוזנת ובכך להפחית עומס ואז יהיו פחות עיכובים בהעברת המידע.
ג. שיפור זמן ב RTT -בגלל של- MPTCP יש מספר נתיבים הוא יכול לבחור את הנתיב עם הקצב העברה הנמוך ביותר ובכך להפחית את הזמן הכולל של העברה.

ד. גידול ברוחב הפס- MPTCP מאחד את רוחבי הפס של כלל הנתיבים וע"י שימוש בכמה נתיבים בו זמנית הוא מגדיל את מהירות העברה הנתונים.
ה. בחירת הנתיב הכי טוב- MPTCP יכול לבחור את הנתיב הכי טוב להעברת המידע תוך כדי ריצה- הוא ייבחר את הנתיב ע"י שיקלול של הפרמטרים שלו(רוחב פס, אמינות, זמן העברה).

שאלה 5

כאשר נתקלים בבעיות של איבוד חבילות בין שני נתבים ברשת, נבחן בשכבות הרשת והתעבורה מספר גורמים שעשויים להיות אחראים לזה, ונציע להם פתרון מתאים.

שכבת הרשת :

אחת הסיבות הנפוצות לאיבוד חבילות בשכבת הרשת היא עומס יתר על הנתבים או על קטעי הרשת, שלבסוף גורם לתורים ארוכים להתמלא ולאיבוד חבילות.

נובע בעיקר מרשת שאינה יכולה לספק רוחב פס גדול מספיק שיתמודד תחת עומס מסוים.

ותקלות ומחסור בחומרה של כרטיסי הרשת והנתבים.

פתרון אפשרי:

התקנת רכיבי חומרה חדשים ושדרוג רכיבים קיימים (על פי תכנון מקדים שמתחשב בכמות התעבורה) עשויים להקל על נטל העומס, בכך שתסופק קיבולת רוחב פס גדולה יותר.

כך שהרשת תוכל להכיל הרבה יותר חבילות בזמן נתון.

שכבת התקשורת :

כאשר יש אובדן חבילות גבוה בין שני נתבים, ברוב המקרים, זה נובע משימוש בפרוטוקול UDP על פני פרוטוקול TCP.

שימוש בפרוטוקול UDP אומנם מהיר יותר אך אינו מבטיח מסירה אמינה של חבילות.

לכן במקרה של עומס רשת, חבילות עשויות להיאבד מבלי שהמערכת תנסה לשלוח אותן מחדש.

פתרון אפשרי:

הפתרון הנפוץ ביותר הוא מעבר לפרוטוקול TCP במידת הצורך.

TCP הוא אמנם איטי יותר אך הוא מצמצם משמעותית איבוד חבילות.

TCP משתמש במנגנונים אמינים ומתוחכמים שיודעים לשלוח מחדש חבילה שאבדה,

דואגים לא להעמיס נתונים על הצד המקבל ובכך מעביר את כל המידע הנחוץ בצורה איכותית ואמינה.

על ידי יישום שלבים אלה, ניתן לפתור או להפחית באופן משמעותי את בעיית איבוד החבילות ברשת ולשפר את אמינות וביצועי התקשורת בין הנתבים.

בס"ד

חלק שני- מענה על השאלות על המאמרים

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

<u>התרומה העיקרית של המאמר:</u>

אחת העובדות הבולטות הן שהמאמר מציג את הגישה הראשונה שבה אפשר לזהות את מערכת ההפעלה, הדפדפן והיישום של משתמש מתוך תעבורת HTTPS מוצפנת.

במילים פשוטות, במקום שהתוקף ינסה לקרוא או לשנות את התוכן המוצפן בתעבורה ("התקפה אקטיבית") הוא יכול פשוט לנצל את תבניות התעבורה כמו זמן הגעת חבילות, גודל החבילות וההתנהגות של היישומים השונים כדי לשער איזו מערכת הפעלה, דפדפן ויישום המשתמש משתמש בהם, כל זאת מבלי לפרק את המידע המוצפן עצמו.

ניתוח תעבורה בעזרת תכונות חדשות

המאמר מציע תכונות חדשות שמבוססות על ניתוח מעמיק של התנהגות התעבורה המוצפנת. התכונות החדשות הללו כוללות:

1. **התנהגות ה SSL/TLS**
2. **מאפיינים של TCP**
3. **התנהגות התעבורה הלא רציפה של דפדפנים**

ההצעה היא שכל התכונות הללו ביחד עם תכונות בסיסיות יותר המוכרות משיטות קודמות, מאפשרות לזהות בצורה מדויקת יותר את המערכת הפעלה, הדפדפן והיישום שהמשתמש משתמש בהם, אפילו כאשר התעבורה מוצפנת.

שימוש במאגר נתונים מקיף- נעשה שימוש במאגר נתונים של יותר מ-20,000 ssנים(קטעים של תעבורה traffic) מסומנים המכסה מערכות הפעלה שונות (Windows, Ubuntu, macOS) דפדפנים (Chrome, Firefox, Safari, Internet Explorer) ויישומים(YouTube, Facebook, Twitter)

המודל המוצג במאמר משיג דיוק של **96.06%** בסיווג מערכת ההפעלה, הדפדפן והיישום מתוך התעבורה המוצפנת. המודל משתמש בתכונות שונות כדי לזהות את המידע הזה, ובסופו של תהליך, הצליח להגיע לרמת דיוק גבוהה מאוד בסיווג של המידע(פריצת דרך לעומת מה שהושג עד כה).

הדיוק הזה הושג על ידי שימוש בשני סוגי תכונות:

התכונות בסיסיות- תכונות שהן נפוצות בזיהוי תעבורה ברוב המחקרים בתחום. תכונות כאלה כוללות נתונים כמו גודל החבילות, הזמן שעובר בין חבילות ועוד.

בס"ד

התכונות חדשות- תכונות שמומשו במיוחד לצורך המאמר הזה, ומתמקדות בזיהוי דפוסים נוספים בתעבורה, כמו ההתנהגות "הלא רציפה" של הדפדפנים (ההתנהגות בה יש שקט בתעבורה ולאחריו פרץ של פעילות), תכונות ,SSL ותכונות של פרוטוקול TCP

באמצעות שילוב של שתי קבוצות התכונות האלה (הבסיסיות והחדשות), המודל מצליח לזהות את מערכת ההפעלה, הדפדפן והיישום של המשתמש בצורה מאוד מדויקת, תוך שהוא מתמודד עם תעבורה מוצפנת, שמקשה בדרך כלל על זיהוי המידע.

מבחינת אבטחה ופרטיות- הממצאים במאמר מצביעים על כך שהצפנה כמו SSL/TLS לא מספקת בהכרח פרטיות מלאה, משום שתבניות התעבורה שמתקבלות בזמן העברת המידע עדיין עשויות לחשוף מידע על המשתמש.

תוקפים או משווקים יכולים לנצל את הדפוסים האלה כדי להבין פרטים על המשתמש, כמו המערכת בה הוא משתמש, איזה דפדפן יש לו, ואילו אפליקציות הוא פועל בהן. במילים אחרות, אם התוקף יודע לזהות דפוסים מסוימים, הוא יכול לבנות "פרופיל" של המשתמש או להבין את הפעולות שהוא עושה באינטרנט, גם אם הוא משתמש בהצפנה.

כך, גם כאשר יש הצפנה שמגנה על תוכן המידע, לא תמיד נשמרת פרטיות המשתמש, משום שדפוסי השימוש יכולים לחשוף מידע על ההתנהגות של המשתמש ברשת.

פרומפט שהשתמשנו: חילקנו את המאמר לחלקים וביקשנו "תרגם את הקטע"

<u>תכונות ותכונות חדשניות להעברת מידע שהמאמר משתמש בהן:</u>

למעשה, המאמר עושה שימוש בשני סוגים של תכונות של תעבורת רשת לצורך סיווג תעבורה מוצפנת ב HTTPS-

התכונות הבסיסיות הנמצאות בשימוש בסיווג תעבורה בגרסה רגילה:

- **משך הסשן**: הזמן הכולל בו נמשך הסשן בשניות בדכ.
- **מספר החבילות בסשן**: מספר החבילות שהוחלפו במהלך הסשן.
- **זמן הגעת חבילות:** ההפרש בזמן בין הגעת החבילות.
- **גודל: payload** גודל המידע שבחבילות.
- **קצב נתונים:** (Bit rate) הקצב שבו הנתונים מועברים.
- **זמן סיבוב:** (RTT) הזמן שלוקח לחבילה לעבור מכתובת מקור לכתובת יעד ולחזור.
- **כיוון החבילה:** אם החבילה היא נכנסת או יוצאת.
- **קצב נתונים שנשלח על ידי השרת:** הקצב שבו השרת שולח את הנתונים.

התכונות החדשות שהוצעו במאמר כמחקר חדש:

- **תכונות SSL:** פרטים שניתן לחלץ מהפרוטוקולים של SSL/TLS שעוזרים להבדיל בין סוגי תעבורה שונים. ה SSL הוא פרוטוקול הצפנה שדואג להגן על המידע שנשלח בין המחשב של המשתמש לאתר, מה שמקשה על תוקפים לגלות מידע כמו כתובת IP פרטי משתמש, תוכן הבקשות התגובות של האתר ועוד. יחד עם זאת, המאמר מציין שדרך ניתוח של התעבורה המוצפנת אפשר לאסוף תכונות מסוימות שיכולות לעזור בהבחנה בין דפדפנים שונים, מערכות הפעלה ויישומים שונים.

- **תכונות TCP:** פרמטרים שקשורים להתנהגות פרוטוקול ה TCP בתעבורת הרשת, גם הם עוזרים לזהות את סוג התעבורה. הכוונה בתכונות TCP היא לפרמטרים שנבדקים מתוך תעבורת הרשת שמבוצעת בעזרת פרוטוקול TCP שמספק שירותים של אמינות, סדר ואחידות בתקשורת בין מחשבים. תכונות אלו מאפשרות ניתוח של דפוסים בתעבורה שנעשית באמצעות פרוטוקול TCP ומסייעות בזיהוי של מערכת ההפעלה, הדפדפן והיישום.
- תכונת **bursty-** במאמר מתוארת **תכונת bursty של דפדפנים** שהיא התנהגות שבה התעבורה ברשת לא מתרחשת באופן רציף, אלא יש תקופות של פעילות אינטנסיבית (נשלחים הרבה נתונים), שמלוות בשקט בתעבורה (עיכובים בין חבילות נתונים).

תקופות אלו נקראות "**(Peaks)**" הפסגות נובעות מהתנהגות של הדפדפן או היישום

כמו (YouTube) במצבים בהם הוא צריך להוריד נתונים מהר יותר כדי להפעיל את הוידאו בצורה חלקה, ואז להמתין לפרק זמן קצר עד שמגיעים עוד נתונים.
מכאן שהמאמר מזהה את דפוס ההתנהגות הזה כמאפיין ייחודי של דפדפנים במהלך סטרימינג של וידאו, ומציע להשתמש ב-"peaks" האלו כדי לעזור בזיהוי דפדפנים ויישומים שונים.

פרומפט שהשתמשנו: "תרגם את הקטע והדגש את החלקים שבהן מדובר על תכונות התעבורה ותכונות תעבורה חדשניות"

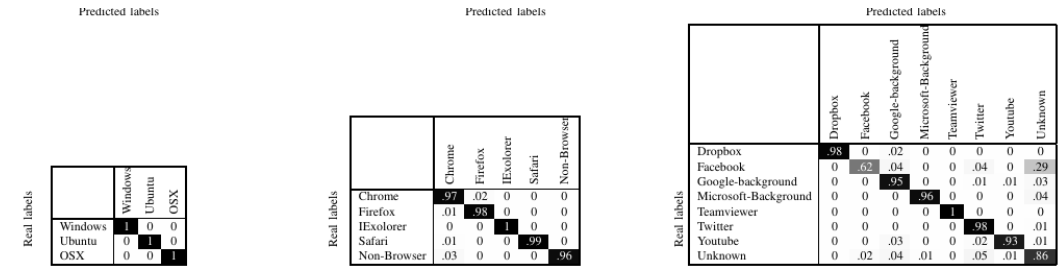
ניתן לסכם את תוצאות המאמר באופן הבא:

1. דיוק ההבחנה: המאמר מראה כי ניתן להבחין בצורה מדויקת במידע על מערכת ההפעלה, הדפדפן והיישום של המשתמש גם כאשר מדובר בתעבורת HTTPS מוצפנת.

עולה כי השימוש ב**מאפיינים בסיסיים** בלבד נותן דיוק של 93.51%

כאשר משלבים **מאפיינים בסיסיים** עם **מאפיינים חדשים**, הדיוק עולה ל **96.06%**

2. לפי ה- confusion matrix ניתן להבחין שההבחנה היא כמעט מושלמת ברוב המקרים.



(b) OS Confusion Matrix (c) Browser Confusion Matrix (d) Application Confusion Matrix

טעויות בהבחנה קרות בעיקר עם הקטגוריה **Unknown** שהיא מייצגת מקרים שבהם המערכת לא מצליחה לקבוע את הקטגוריה בוודאות לדוגמה, כאשר אין מספיק נתונים או שהשילוב של מאפייני התעבורה לא ברור.

תובנות מהתוצאות:

בס"ד

ישנן מספר הבחנות שנעשו-

הבחנת מערכת ההפעלה- ההבחנה במערכת ההפעלה היא **כמעט מושלמת**. כנראה שזה נובע מכך שמאפיינים מסוימים כגון התנהגות SSL/TLS זמן הגעת חבילות, וכו מאפשרים הבחנה טובה בין מערכות הפעלה שונות.

הבחנת דפדפן- גם הבחנה זו כמעט מושלמת, אך ישנם מקרים בהם חבילות תעבורה מצביעות על התנהגות מנותקת או על תעבורת רקע. התנהגות זו, שהיא תכונה של הדפדפן, עוזרת להבדיל בין דפדפנים.

הבחנת יישומים- הבחנה ביישומים מדויקת למדי, אם כי יש טעויות בהבחנה כאשר התעבורה של יישום מתנהגת בצורה דומה לתעבורת רקע. למשל, יישום כמו פייסבוק זוהה כ **Unknown**-ב-29% מהמקרים.

הוספת **מאפיינים חדשים** כמו תכונות SSL/TLS מאפייני TCP והתנהגות מנותקת של הדפדפנים שיפרה באופן משמעותי את הדיוק בהבחנה. מאפיינים אלו עוזרים להבדיל בין תעבורה של מערכות הפעלה, דפדפנים ויישומים שונים.

מסקנות:

מבחינה מעשית, התוצאות מראות שגם בזמן שמדובר בתעבורה מוגנת (HTTPS) עדיין יש אפשרות לגורם עוין לזהות את המידע אודות מערכת ההפעלה, הדפדפן והיישום של המשתמש. תוצאה זו עשויה להוות איום על הפרטיות, שכן תוקפים יכולים להשתמש במידע הזה כדי להתאים את אסטרטגיית ההתקפות שלהם.

לסיכום, המאמר מציג תוצאות שמראות כי ניתן להבחין בצורה מדויקת בתעבורה מוצפנת (HTTPS) ולהסיק ממנה מידע אודות מערכת ההפעלה, הדפדפן והיישום של המשתמש, בעזרת שילוב של מאפיינים בסיסיים וחדשים. תוצאות אלו מהוות צעד משמעותי בהבנת ניתוח תעבורה מוצפנת.

בס"ד

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

סיווג מוקדם של תעבורה (eTC) מהווה תהליך מרכזי בניהול רשתות מודרניות.

כדי לשמור על איכות ויעילות השירות, חיוני שספקי הרשת יזהו מהר את סוגי התעבורה ברשת ויקצו להם עדיפות בהתאם. אתגר נוכחי הוא ש 97% מהתעבורה ברשתות מוצפנת באמצעות TLS,

מה שמקשה מאוד על זיהוי וסיווג נתונים מהחבילות הנעות ברשת.

עד כה, האלגוריתמים הקיימים לסיווג תעבורה השיגו דיוק של 38.4% בלבד.

התרומה העיקרית של המחקר היא פיתוחו של אלגוריתם מתקדם,

ה -(Hybrid Random Forest Traffic Classifier) hRFTC המשלב את היתרונות של אלגוריתמים קודמים ומציע שיפור משמעותי בדיוק הסיווג, עם תוצאה מרשימה של 94.6% דיוק בניסויים ובהשוואות שבוצעו.

המאמר בוחן שני סוגים מרכזיים של מאפייני תעבורה:

1. מאפייני TLS (Transport Layer Security) –

מבוססים על המידע החשוף במהלך לחיצת היד (TLS Handshake).

2. מאפייני זרם סטטיסטיים (Flow-Based) –

ניתוח של גודל חבילות, זמני הגעה ותבניות תקשורת כלליות.

מאפייני TLS בלתי מוצפנים (Packet-Based)

למרות ש-TLS 1.3 ו-ECH מצפינים חלקים משמעותיים מהתעבורה, עדיין יש שדות מסוימים שנותרים גלויים.

אלו הנתונים שהמאמר משתמש:

- גרסת TLS
- קבוצות הצפנה (Cipher Suites)
- אורך ההרחבות (Extension Lengths)
- מפתחות משותפים (Key Share, Pre-Shared Keys)

חידוש:

מחקרים קודמים הניחו שהמידע הזה אינו מספק לסיווג, אבל המאמר מוכיח שניתן לזהות דפוסים גם כאשר SNI מוצפן.

בס"ד

מאפייני זרם סטטיסטיים (Flow-Based)

כדי להתמודד עם המגבלות של TLS בלבד, המאמר מוסיף מדדים דינמיים של זרם התעבורה, הכוללים:

1. גודל חבילות (Packet Sizes)

המאמר בודק את הגודל הממוצע, החציוני והמקסימלי של החבילות.

מחלק את החבילות לקטגוריות על בסיס גדלים (64, 128, 256, 512 בייט).

2. מרווחי זמן בין חבילות (Inter-Packet Times - IPT)

מודדים את הזמן שעובר בין שליחת שתי חבילות עוקבות.

לדוגמה, סטרימינג אודיו (Spotify) מאופיין במרווחי זמן יציבים, בעוד שסטרימינג וידאו (YouTube) מציג מרווחים משתנים.

3. כיוון התעבורה (Upload vs Download)

סוגי תעבורה שונים מפגינים דפוסים שונים של שליחה והורדה:

YouTube: רוב התעבורה בכיוון ההורדה.

Zoom: משדר כמות משמעותית של נתונים גם בהעלאה.

החידוש כאן הוא בשילוב של מאפייני TLS עם נתונים סטטיסטיים של זרמים. זה מה שהופך את hRFTC לכל כך מוצלח.

בס"ד

תוצאות:

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

הטבלה מציגה השוואה בין מספר מסווגי תעבורה על בסיס F-score

העמודות מחולקות לשלוש קבוצות של מסווגים:

Hybrid Classifier - שילוב של מאפייני TLS עם מאפייני זרם סטטיסטיים.

Flow-Based Classifier- מסווגים שמתבססים רק על מאפייני זרם.

Packet-Based Classifiersמסווגים מבוססי חבילות - מסווגים שמסתמכים רק על מידע מתוך חבילות TLS ראשוניות.

המסווגים ההיברידיים כוללים את hRFTC והשוואה למודלים היברידיים קיימים (UW, hC4.5) .

hRFTC הוא האלגוריתם הטוב ביותר – השיג את התוצאות הגבוהות ביותר (94.6% דיוק ממוצע), בעוד שהשיטות הישנות והמסווגים מבוססי חבילות נכשלים לחלוטין.

התובנה העיקרית מהמאמר היא ששימוש במידע היברידי (TLS + זרם) הוא הפתרון הטוב ביותר לניהול וסיווג תעבורה מוצפנת כאשר ECH מופעל!

בס"ד

מאמר שלישי- FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition

<u>התרומה העיקרית של המאמר:</u>

סיווג תעבורת רשת חיוני למטרות כמו ניהול איכות שירות (QoS), אבטחת מידע, ואיתור נזקים.

עם זאת, קיים קושי בזיהוי תעבורת רשת מוצפנת באמצעות שיטות מסורתיות, במיוחד בשימוש בכלים כמו VPN או Tor שמסוות את זהות הנתונים ומקשות על ניתוחם. שיטות קיימות, שבדרך כלל נשענות על איתור מאפיינים ידניים של זרמי נתונים ולמידה מפוקחת, אינן מספקות פתרון מדויק ויעיל כאשר הנתונים מוצפנים.

התרומה המרכזית של המאמר היא הצגת גישה חדשנית לזיהוי תעבורת רשת מוצפנת, דרך המרת זרמי נתונים לתמונות, המכונות "FlowPics"

השיטה מנצלת רשתות נוירונים קונבולוציוניות (CNNs) לזיהוי תבניות בתמונות הללו,

מה שמאפשר זיהוי מדויק של סוגי תעבורה ויישומים אינטרנטיים, גם תחת תנאי הצפנה.

זה פותר את הבעיה של חוסר יעילות שיטות הזיהוי המסורתיות בסביבות מוצפנות, תוך שמירה על פרטיות המשתמשים, מכיוון שלא נדרשת הפענוח של מטען הנתונים עצמו.

<u>תכונות ותכונות חדשניות להעברת מידע שהמאמר משתמש בהן:</u>

המאמר מציע גישה חדשנית לסיווג תעבורה מוצפנת על ידי המרה של זרמי נתונים

(Network Flows) לתמונות ושימוש בלמידה עמוקה (Deep Learning) לסיווגם.

כמו כן, המאמר מחלץ מאפיינים מתעבורת הרשת כדי לבנות ייצוגים ויזואליים של התעבורה המוצפנת. המאפיינים המרכזיים כוללים:

1. **מידע על זמני הגעת חבילות(Packet Timing Information)**
   - הפרשי זמן בין חבילות רצופות (Inter-arrival times)
   - חותמות זמן מוחלטות של החבילות בתוך הזרם.
2. **מידע על גודל החבילות(Packet Size Information)**
   - רצף גדלי החבילות בזרם התעבורה.
   - כיוון החבילות (נכנסות מול יוצאות).
3. **מאפייני זרם(Flow-based Features)**
   - מספר החבילות בזרם.

בס"ד

- משך הזרם.
- סך כל הבתים שנשלחו והתקבלו.

המאפיינים החדשים שהמאמר מציע: (החידוש המרכזי במחקר הוא **המרת נתוני זרמי הרשת לתמונות** לצורך סיווג בעזרת למידה עמוקה).

1. **ייצוג FlowPic**

- המאמר ממיר זרמי תעבורה לתמונות שנקראות FlowPic בהן הגודל והזמן של כל חבילה ממופים לערכי פיקסלים.
- השיטה מאפשרת שימוש ברשתות נוירונים קונבולוציוניות (CNNs) שמסוגלות ללמוד תבניות מתוך התמונה, בדומה לטכנולוגיית זיהוי תמונה.

2. **קידוד תמונה בגודל קבוע**

- כל זרם מנורמל לתמונה בגודל קבוע (למשל, 32×32 פיקסלים), כדי להבטיח עקביות בין זרמים עם אורכים שונים

3. **שימוש ב CNNs לסיווג תעבורה**

- בניגוד לשיטות מסורתיות שמתבססות על מאפיינים סטטיסטיים מחושבים מראש, שיטת FlowPic מאפשרת למודל ללמוד לבד את התבניות ישירות מתוך הנתונים.
- אין צורך בהגדרה ידנית של מאפיינים, מה שמקל על הסיווג ומשפר את הדיוק.

תובנות מהמאפיינים החדשים-

- השיטה מחליפה את הצורך בזיהוי ידני של מאפיינים סטטיסטיים ,ומאפשרת למודל לבצע למידה עמוקה על מנת לזהות תבניות בתעבורה בצורה אוטומטית.
- FlowPic מאפשרת שימוש ברשתות נוירונים קונבולוציוניות (CNNs) מה שמשפר את הדיוק ומקל על זיהוי תעבורה מוצפנת.
- המרת הזרמים לייצוג דמוי תמונה מאפשרת שימוש בטכניקות מוכרות מתחום זיהוי תמונות, גם עבור סיווג תעבורת רשת.

פרומפט שהשתמשנו: "תרגם את הקטע והדגש את החלקים שבהן מדובר על תכונות התעבורה ותכונות תעבורה חדשניות"

בס"ד

להלן התוצאות:

FlowPic בהשוואה למסווגים מסורתיים בסיווג קטגוריות תעבורה (Non-VPN, VPN, Tor)

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

FlowPic טוב יותר מהשיטה הקודמת ב - Non-VPN (85% לעומת 84%)

FlowPic כמעט זהה לשיטה הקודמת בזיהוי VPN (98.4% לעומת 98.6%)

FlowPic חלש יותר ב- Tor (67.8% לעומת 84.3%) ,

מה שמראה כי Tor מצליח להסוות את דפוסי התעבורה בצורה משמעותית יותר.

FlowPic נבדק על קטגוריות ספציפיות של תעבורה בסביבות שונות (ללא הצפנה, VPN, Tor).

Class	Accuracy (%)			
VoIP	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	**99.6**	99.4	48.2
	VPN	95.8	**99.9**	58.1
	Tor	52.1	35.8	**93.3**
Video	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	**99.9**	98.8	83.8
	VPN	54.0	**99.9**	57.8
	Tor	55.3	86.1	**99.9**
File Transfer	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	**98.8**	79.9	60.6
	VPN	65.1	**99.9**	54.5
	Tor	63.1	35.8	**55.8**
Chat	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	**96.2**	78.9	70.3
	VPN	71.7	**99.2**	69.4
	Tor	85.8	93.1	**89.0**
Browsing	Training/Test	Non-VPN	VPN	Tor
	Non-VPN	**90.6**	-	57.2
	VPN	-	-	-
	Tor	76.1	-	**90.6**

שיחות VoIP ווידאו מזוהות בדיוק כמעט מושלם (99.9%-99.6%) גם ללא VPN וגם עם VPN.

אבל ברגע שמוסיפים Tor, הזיהוי נופל משמעותית (יורד ל-48.2%)

העברת קבצים וצ'אט מזוהים טוב מאוד ללא הצפנה (98.8%-96.2%) .

גם גלישה כללית מזוהה פחות טוב עם Tor (57.2%), כי מדובר בתעבורה מגוונת יותר.

סיכום ותובנות:

FlowPic משיג תוצאות מצוינות בסיווג תעבורה לא מוצפנת ותעבורת VPN ועולה על כל השיטות הקודמות. הוא מזהה יישומים ספציפיים בדיוק כמעט מושלם של 99.7%.
FlowPic עדיין מתקשה בזיהוי תעבורת Tor (67.8%), אך עדיין מציג שיפור מסוים ביחס לגישות מסורתיות.

זהו כלי חזק לניהול רשתות, אבטחת מידע ופיקוח תעבורה, במיוחד כאשר רוצים לזהות תעבורה מוצפנת.
FlowPic מהווה פריצת דרך בעולם סיווג התעבורה ומאפשר לזהות דפוסים מוצפנים בעזרת למידת מכונה מבוססת תמונות.

בס"ד

קישור אל התעבורה שהקלטנו עבור כל סוג תעבורה:

https://drive.google.com/drive/folders/1VnhL0cE8ubw06vcc9PBTO3wL2omlcBPJ?usp=drive_link

קישור אל הdataset

https://drive.google.com/file/d/1F7s170qiJIAhP1AzJIcY-eFmTMehYnoa/view?usp=drive_link

קישור לגיטהאב:

https://github.com/idan200402/communicationsFP

כיצד הפקנו את הנתונים מהתעבורה:

עבדנו עם הממשק של tshark כדי להוציא את הנתונים המעניינים דרך ה.linux command דוגמאות לחלק מהקוד שבעזרתו הצלחנו להפיק את קבצי הטקסט לגרפים.

```
tshark -r chrome.pcapng -Y "tls" -T fields -e tls.record.version -e tls.handshake.type -e tls.record.length > chrome_tls.txt

tshark -r spotify.pcapng -Y "tls" -T fields -e tls.record.version -e tls.handshake.type -e tls.record.length > spotify_tls.txt

for app in chrome firefox youtube soundcloud zoom; do

tshark -r "$app.pcapng" -Y "tcp" | wc -l > "${app}_tcp_count.txt" tshark -r "$app.pcapng" -Y "udp" | wc -l > "${app}_udp_count.txt"

done

for app in chrome firefox youtube soundcloud; do

tshark -r "$app.pcapng" -Y "tls" | wc -l > "${app}_tls_count.txt"

done

for app in youtube soundcloud; do

tshark -r "$app.pcapng" -Y "tcp" -T fields -e tcp.window_size > "${app}_window_sizes.txt"

done
```

```
project/datacapturing$ tshark -r chrome.pcapng -T fields -e frame.len > chrome_s
izes.txt
tshark -r firefox.pcapng -T fields -e frame.len > firefox_sizes.txt
tshark -r youtube.pcapng -T fields -e frame.len > youtube_sizes.txt
tshark -r soundcloud.pcapng -T fields -e frame.len > soundcloud_sizes.txt
tshark -r zoom.pcapng -T fields -e frame.len > zoom_sizes.txt
```

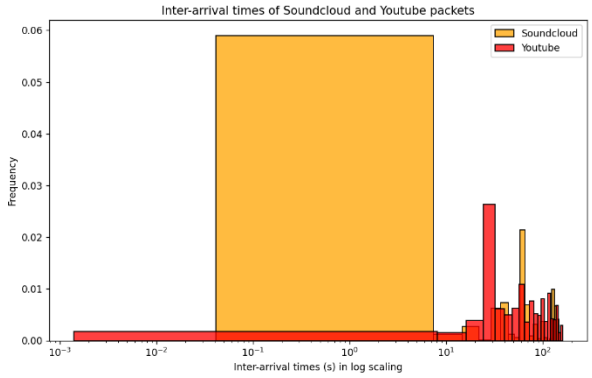
```
project/datacapturing$ tshark -r chrome.pcapng -T fields -e frame.time_epoch > c
hrome_times.txt
tshark -r firefox.pcapng -T fields -e frame.time_epoch > firefox_times.txt
tshark -r youtube.pcapng -T fields -e frame.time_epoch > youtube_times.txt
tshark -r soundcloud.pcapng -T fields -e frame.time_epoch > soundcloud_times.txt

tshark -r zoom.pcapng -T fields -e frame.time_epoch > zoom_times.txt
idan2004@idan2004-VirtualBox:~/Desktop/artel year b/communacation networks/final
project/datacapturing$ tshark -r chrome.pcapng -q -z io,phs > chrome_flow.txt
tshark -r firefox.pcapng -q -z io,phs > firefox_flow.txt
tshark -r youtube.pcapng -q -z io,phs > youtube_flow.txt
tshark -r soundcloud.pcapng -q -z io,phs > soundcloud_flow.txt
tshark -r zoom.pcapng -q -z io,phs > zoom_flow.txt
idan2004@idan2004-VirtualBox:~/Desktop/artel year b/communacation networks/final
project/datacapturing$
```

בס"ד

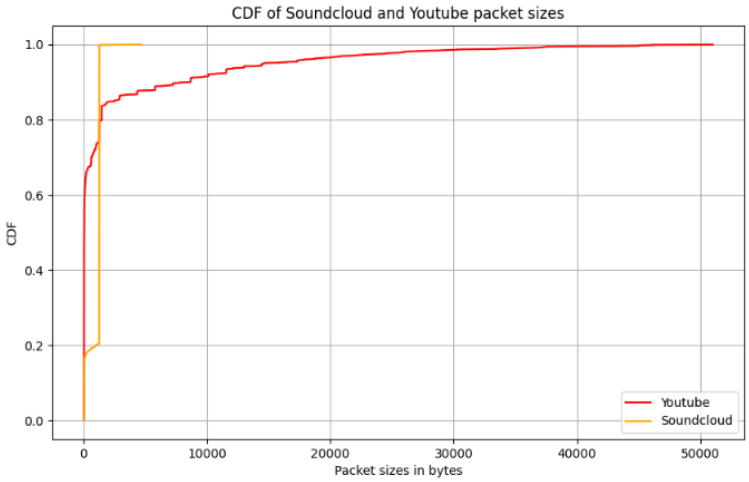
תמלול של הגרפים שהכנו:

בחרנו במספר גרפים אשר משווה בין האפליקציות הרלוונטיות כדי למנוע עומס על הקורא ולהראות בצורה מיטבית את העקרונות וההבדלים עליהם דנו בהרצאות והתרגולים.

נשים לב כי רוב התוצאות אינן תואמות למה שלמדנו בתאוריה וזאת בגלל טכניקות mitigation שעליהן נדון בחלק הבא.

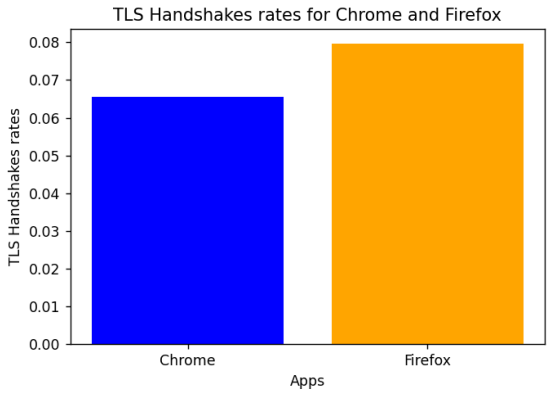


הגרף מתאר את הפרשי ההגעה בין חבילה iלחבילה i+1 לכל i שמתאים לאורך החבילות פחות1 של החבילות של יוטיוב וסאונדקלאוד (הקלטה של שידור וידאו ושידור אודיו) . כפי שניתן לראות כל חבילה מהתעבורה של סאונדקלאוד רוב החבילות מגיעות בהפרש עקבי בין 0.1 לשניה מהחבילה שלפניה , וחלק קטן בהפרשים יותר גדולים. לעומת התעבורה של יוטיוב שהפרשי החבילות הוא לא קבוע אלא מתפרש לאורך כל הספקטרום מ0.001 שניות אל 100 שניות הפרש. לסיכום , הזרימת תעבורה של שידור אאודיו הוא יותר קבוע ויציב לעומת תעבורה של שידור וידאו שהוא עם יותר קפיצות spikes , נובע מאופי השידור שהוא אדאפטיבי של יוטיוב.

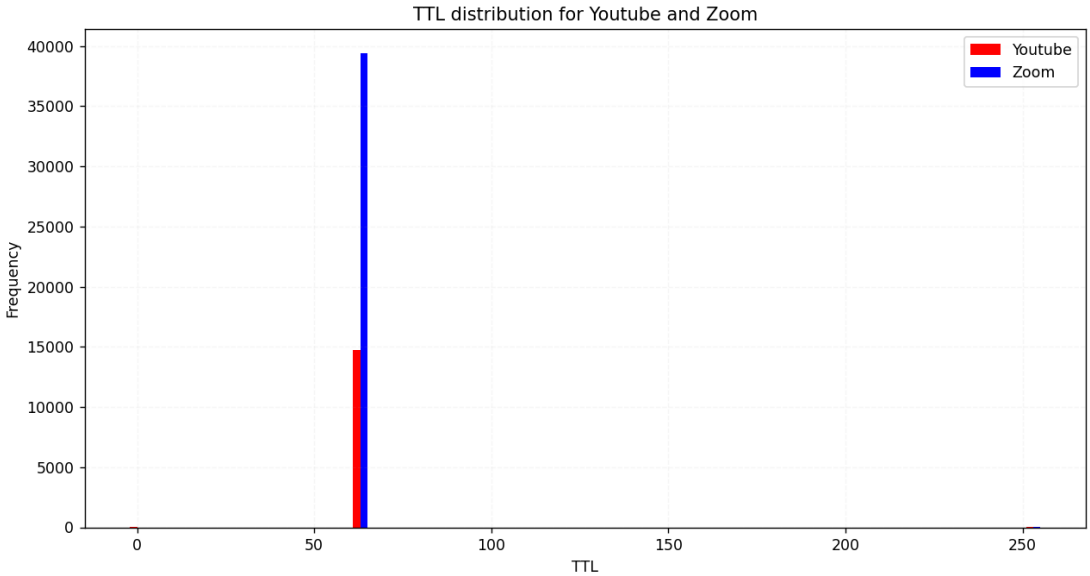


בס"ד

הגרף מתאר פונקציית הצטברות של הגודל חבילות של יוטיוב וסאונדקלאוד. נתבונן בפונקציית ההצטברות של סאונדקלאונד ונשים לב שהיא מגיעה לשיא שלה בצורה חדה יותר מ youtube ,לעומת הפונקציית הצטברות של יוטיוב שכוללת מגוון רחב יותר של חבילות ובאופן כללי מגיעה לגדלי החבילות גדולים יותר. העליה החדה של פונצקיית סאונדקלאוד מצביעה על כך שהחבילות יחסית קטנות ואחידות בגודלן , לעומת הפונקציה של יוטיוב שעולה בצורה לוגרתמית ומצביעה על כך שיש מגוון רחב יותר של חבילות ורוב החבילות הינן חבילות גדולות , כנראה חבילות שמכילות חלקי תוכן של הסרטון.



הגרף הזה משווה בין התדירות של לחיצות ידיים של חיבור TLS של דפדפן כרום ודפדפן פיירפוקס . כפי שניתן לראות הדפדפן פיירפוקס מבצע יותר לחיצות ידיים לעומת כרום . אין הבדל משמעותי , כן ידוע כי הדפדפן פיורפוקס הוא יותר בטוח(secure) מאשר כרום.

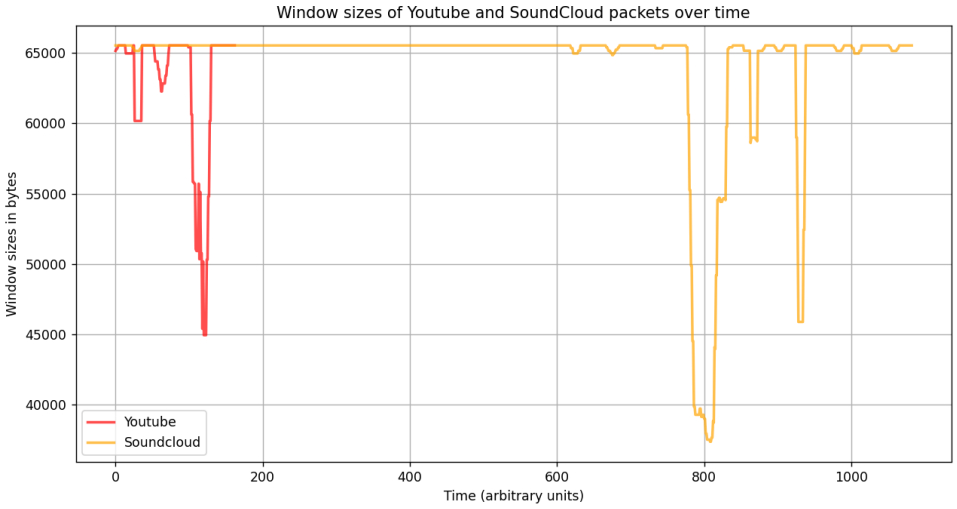


הגרף עמודות הזה מציג את ההתפלגות דגל TTL אצל החבילות של תעבורת יוטיוב וזום.

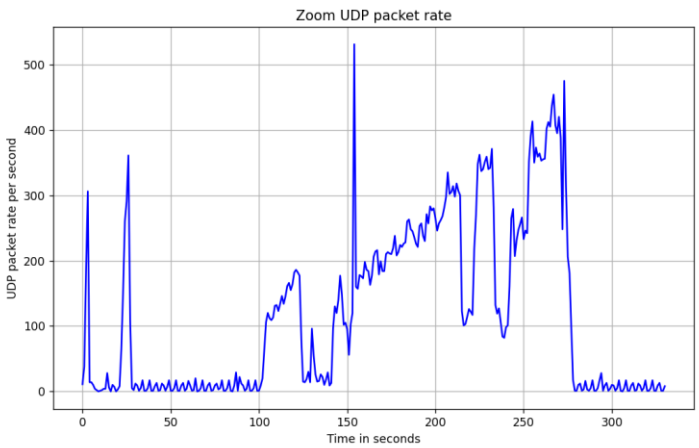
כפי שניתן לראות רוב החבילות נשלחות עם ttl של 64 . המספר 64 הוא המספר ttl הדיפולטיבי של שרתים מבוססים על מערכת הפעלה linux זאת אומרת שהחבילה נשלחה

בס"ד

משרת שקרוב אלינו ולכן קיבלנו את החבילה עם ttl 64. (בזום המגנון שונה אבל העיקרון של ה tttlזהה) . נשים לב שקיימות גם מעט חבילות עם ttl 255 ככל הנראה אלו חבילות מפרוטוקול ICMP שמטרתן היא להגיע ליעד לא משנה מה ולכן הם קיבלו את הזמן המקסימאלי לחיות . ניתן לראות שגם קיימות מעט מאוד חבילות של יוטיוב שנשלחו עם ttl של 0 ככל הנראה טעות .



הגרף מתאר את שינוי גודל החלון בפרוטוקול TCP לאורך ההקלטה של יוטיוב וסאונדקלאוד . כפי שניתן לראות גם יוטיוב וגם סאונדקלאוד מתחילות עם גודל חלון יחסית גדול. כפי שניתן לראות לאורך השימוש בשני האפליקציות קיימות ירידות בגודל החלון , המצב הזה קורא כאשר הלקוח ,במקרה הזה אנחנו , לא יכול להתמודד ולעבד את הכמות של הנתונים , ככל המקרה בגלל עומס ברשת. לאחר מכן ניתן לראות שהגודל חלון חוזר להיות גדול וקיימות נפילות חוזרות . למרות שבשתי התעבורות קיימות ירידותת חדות של גודל החלון , נראה כי המשתמש יכול לעבד נתונים של

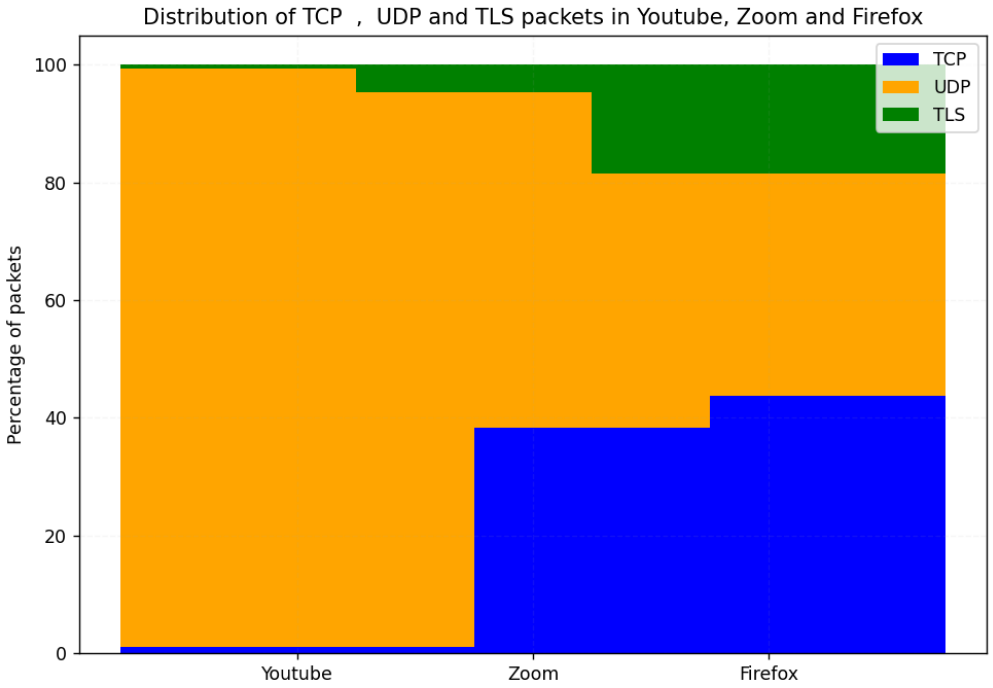


בס"ד

סאונדקלאוד בצורה יותר טובה מאשר יוטיוב מכיוון שברוב הזמן גודל החלון של סאונדקלאוד נשאר מקסימלי .

הגרף מתאר את מספר של חבילות מפרוטוקול udp לשנייה לאורך סשן זום . ניתן לראות שב50 שניות הראשונות יש קפיצה של החבילות לאור טעינת הממשק זום וחיבור מיקרופון ווידאו . מ50 עד 100 לא בוצעה שום פעולה , החל מ100 שניות עד 280 שניות ניתן לראות עליה משמעותית של החבילות לאור התחלת שיחה בזום. עליה מאוד חדה ב150 שניות שמצביעה על עליה באיכות הוידאו בזמן אמת או פיצוי על איבודי חבילה רבים. מ150 עד 280 יש תנודות יחסית נמוכות שמצביעות על שיוניים באיכות השידור בזמן אמת כתלות בעומס רשת. החל מ280 יש ירידה חדה שמצביעה על סיום השיחה.

נשים לב כי הגרף לא מתאר בצורה ישירה את אופי הסשן מפני שקיימות חבילות נוספת שמטרתן לבלבל את התוקף הפוטנציאלי.



הגרף הזה מתאר את האחוזים של פרוטוקלי , TCP UDP ו TLS בחלק מתעבורות של זום יוטיוב ופיירפוקס . נתבונן בעמודה של יוטיוב , כפי שציפינו הפרוטוקול הדומיננטי הוא UDP , או בשמו המדוייק QUIC שמבוסס על UDP . בנוסף יש מעט של TCP ו TLS שכן פרוטוקולQUIC משתמש בהצפנה ותקשורת אמינה ולכן קיים כמות כלכך מזערית שלו.
בעמודה של זום , ניתן לראות שמרבית החבילות הן מפרוטוקול UDP וTCP . שכן שיחות וידאו ואאודיו בזמן אמת מסתמכות על UDP וTCP עבור ניהול הסשן , הודעות בצאט , ו"תכנית חלופית " למצבים בהם העומס רשת גבוה מדי. ושימוש מועט בTLS על מנת הצפנת מידע מעל TCP

<u>חלק 3 שאלה 4</u>

בס"ד

לדעתנו הדרך הכי נכונה לענות של שני החלקים הוא לבנות שני מודלים אשר מתאמנים על שני מאגרי נתונים לפי כל דרישה בהתאמה . אך אחרי חיפוש מעמיק ברשת , לא מצאנו מאגר נתונים מספק עבור החלק הראשון , בניגוד לחלק השני שמצאנו מאגר מידע אשר עונה על הדרישות. לכן עבור החלק הראשון , נסביר מפורטות ונענה על השאלות מכל ההבטים. עבור החלק השני , נאמן מודל למידת מכונה על המאגר המתאים והמטרה שלנו שהוא יצליח לנחש את סוג התעבורה לפי הפיצ'רים המוזכרים.

<u>חלק ראשון:</u>

התוקף יודע גודל של כל חבילה , מתי כל חבילה הגיעה ונשלחה ו flow id מוצפן.
למרות שהתוכן של כל חבילה מוצפן ואין לנו גישה אליו , בעזרת המטאדטא הזו אנחנו יכולים לדעת על הפעילות של המשתמש ובאילו אפליקציות או אתרים הוא משתמש (נתייחס אל גלישה באתר אינטרנט , שיחה בזמן אמת (זום) , צפייה בוידאו והאזנה לאאודיו).

ה flow id יכול לעזור לנו המון גם אם אנחנו יודעים מה כל תא בקבוצה הסדורה מכיוון שהפונקציית הצפנה היא חח"ע אז כל flow id יהיה יחודיי , לכן , נוכל ליצור קבוצת של כל flow id ולכל קבוצה שניצור נוכל לחשב את משך הקשר , מספר החבילות שהועברו , התפלגות של גודל החבילות וזמני הגעה פנימיים של כל חבילה.

בעזרת המידע הזה יהיה לתוקף אינטואיציה חזקה באילו אתרים או אפליקציות המשתמש גולש. ניתן כמה דוגמאות :
1. אם התוקף מזהה שהרוחב פס גדול , אז הוא יוכל להסיק שמדובר בשיחת זמן אמת או שידור של סרטון וידאו שידוע שהם דורשים רוחב פס גדול , לעומת רוחב פס קטן , שככל הנראה המשתמש גולש באתרי אינטרנט.

2. תדירות של לחיצות ידיים של פרוטוקול TLS. אם התוקף מזהה תדירות רבה של לחיצות ידיים יהיה ניתן להסיק שהמשתמש גולש בדפדפן אינטרנט והוא פותח אתרי אינטרנט שונים , וככל הנראה לא מדובר בצפייה בשידור וידאו וכדומה.

כעת נשתמש בידע שצברנו בהרצאות ובהכנת המטלה כדי לסכם לאילו מסקנות התוקף יכול להגיע בעזרת הנתונים .

גלישת באתרי אינטרנט (כרום פיירפוקס): גודל חבילות יחסית קטן , עם זמני הגעת חבילות שרירותי אבל ככל הנראה לא רציף , משך של כל flow נמוך , זאת אומרת שכל חיבור מתקיים זמן קצר , רוחב פס קצר מאוד.
שידור וידאו (יוטיוב): גודל חבילות גדול , זמני הגעת חבילות אינו רציף בגלל ה adaptive streaming , משך של כל חיבור הוא ארוך מאוד (דקות ארוכות) , ודורש רוחב פס ארוך .

שידור אאודיו (ספוטייפי , סאונדקלאוד): גודל חבילה מעט יותר קטן משל יוטיוב , זמני הגעה רציפיים , שידור יציב , זמן חיבור כמו של יוטיוב , נמשך מספר דקות .
תקשורת בזמן אמת וידאו ואאודיו (זום): גודל חבילות משתנה , ההגעת חבילות רציפה , משך זמן של flowהוא כמה שניות בעקבות פרוטוקול .NAT ומה שמסגיר את סוג התעובר הוא השליחת חבילות מהלקוח אל השרת והפוך בנוסף ל.low latency

כפי שכתבנו , התוקף יכול להצליח לגלות באילו אתרים המשתמש גולש גם אם הוא לא יודע פרטים קונקרטיים כמו תוכן החבילה , הIP של השרת ועוד. מספיק לו לדעת דפוסים חוזרים של תעבורה בשביל לגלות . כעת נדון בדרכים שבהן ניתן לבלבל את התוקף על ידי שינוי בדפוסי ההתנהגות וכך לתוקף לא יהיו את ההתנהגויות האמיתיות של כל תעבורה .

טכניקות כלליות שמשתמשים בהן עבור כל האפליקציות :

בס"ד

1. להוסיף מידע רנדומלי לכל חבילה על מנת לשנות על ההתפלגות גדלים האמיתית שלהן וכך להטעות את התוקף שלא יוכל להסיק מההתפלגות חבילות מסקנות מהימנות .
2. להוסיף שהיות מלאכותיות בין שליחה של חבילות בצורה שרירותית על מנת לבלבל את התוקף ולא יוכל להסיק מסקנות מה inter arrival times.
3. הצפנה של DNS. מצפין גם הודעות requestשל המשתמש כך שתוקף לא יכול לראות לאן המשתמש רוצה להגיע .

כעת נבחר אפליקציה שמסווגת כאחד מסוגי התעבורה שחקרנו ונדון מאילו טכניקות mitigation הן משתמשות.

1. יוטיוב: הבעיה היא שגודל חבילות הוא גדול ומשך חיבור הוא יחסית ארוך.

- שימוש בQUIC מאשר TLS מעל TCP . השימוש בפרוטוקול QUIC מוכח בכך שהוא מצפין יותר מטאדטא בצורה יותר יעילה ולכן יהיה קשה יותר לתוקף לזהות את תוכנן.
- Adaptive bitrate streaming, יוטיוב משנה את האיכות שידור שלה בהתאם לרוחב הפס באופן דינאמי כך שיהיה קשה לתוקף לזהות התנהגות מסויימת.
- יוטיוב מוסיפה חלקים גדולים של מידע אל החבילות כך שהתוקף לא יוכל לזהות שינויים באיכות שידור ולהסיק מהן דברים .

2. סאונדקלאוד: הבעיה היא ששידור אאודיו יש זרימה יציבה וארוכה עם הגעת חבילות רציפה .

- האפליקציה יודעת לבחור בצורה רנדומלית קטעי קול לחיבור כדי לשבור את הדפוסי זרימה
- על ידי הצפנת domains התוקף לא יודע אילו שירים המשתמש שומע
- הוספת רעש לחבילה כמו ביוטיוב.

3. גלישה בדפדפן: יש התפרצויות של חבילות , הגעה לא רציפה , גודל חבילות קטן .

- הצפנת DNS , לא תהיה גישה לתוקף לאתרים מסויימים.
- הוספת רעש כמו ביוטיוב , כך התוקף לא מזהה חבילות קטנות .
- Multiplexing דפדפנים יכולים לקחת כמה בקשות ולשגר אותם ביחד דרך חיבור יחיד וזה שובר את ההתנהגות של גלישה בדפדפן.

4. Zoom: זרימת חבילות רציפה ודו כיוונית

- בגלל שזום משנה את האיכות אאודיו ווידאו בצורה דינאמית זה מקשה על התוקף לזהות דפוסים.
- זורם מוסיפה חבילות נוספות כדי לבלבל את התוקף ולשבור את ההתנהגות הרגילה

- שימוש בP2P במהלך השיחה הקשר עובד מלהיות מרוכז בשרת של זום , לקשר בין שני לקוחות . ובגלל שהקשר הזה גם לא קשור יותר לשרתים של זום וגם הרבה יותר תלוי במצב בשרת של כל אחד המשתתפים יהיה קשה לתוקף לזהות התנהגות מסויימת.

המסקנות שלנו מהעמקה על תקיפות:

אי אפשר להיות בטוחים לגמרי מתקיפה (לפחות כרגע) , אך אפשר להקטין את הסיכויים לתקיפה על ידי כל האפשרויות mitigationשהצגנו . תמיד התוקף יכול לאמן מודל אשר מנחש סטטיסטית את האפליקציה כך המטרה שלנו היא למנוע זאת.

<u>חלק שני:</u>

אחרי חיפוש במרשתת מצאנו dataset אשר מכיל את הפיצ'רים של גודל החבילה וזמן ההגעה של כל חבילה .

https://www.kaggle.com/datasets/inhngcn/https-traffic-classification?resource=download

כל שורה במאגר נתונים מתארת flowברשת בין הלקוח לשרת.

לפי הנתונים , לתוקף יש גישה רק לגודל החבילה , זמן ההגעה. בשביל לאמן את המודל אנחנו צריכים להוציא את הסוג של סוג חבילה ולכן לעבוד רק עם העמודות הבאות

- BYTES מייצג את גודל החבילות שעברו בסשן הנוכחי
- PKT_TIMES מייצג מערך זמני הגעה של כל החבילות בסשן הנוכחי
- TYPE מייצג את סוג התעבורה (אות לכל סוג תעבורה).

שמנו לב שאין סוג זרימה של שיחת וידאו בזמן אמת , לכן החלפנו את הסוג הזה בסוג של שידור וידאו בזמן אמת (כגון יוטיוב לייב).

מהלך עבודה:

בגלל שהקובץ גדול מדי להיות בגיטהאב , אז שמרנו אותו בחשבון גוגל דרייב , נתנו הרשאת קישור ציבורית וכתבנו סקריפט שיוריד את המאגר נתונים בעזרת ספריית gdown

לאחר מכן הורדנו את כל העמודות עם התכונות שלא ניתנו לנו והורדנו את כל השורות שבהן היה ערך NULL . לאחר מכן , לכל סוג תעבורה קבענו מספר חד ערכי והחלפנו את האות במספר. לאחר מכן החלפנו את המחרוזת של זמני ההגעה במערך פורמלי. לבסוף יצרנו שלוש עמודות עמודות של נתונים סטטיסטים כגון ממוצע התפלגות נורמלית וכמות הגעות והחלפנו אותן בעמודה של המערך חבילות ההתחלתי. אחרי ה preprocessing למאגר מידע התחלנו את התהליך של היצירת מודל , קבילת משתנים תלויים ובלתי תלויים , חלוקה לקבוצת אימון ומבחן . אימון המודל ובדיקת דיוק על הקבוצת מבחן .

בס"ד

עד כה הסברנו כיצד לזהות היטב את סוג התעבורה , אך לא נדו בנושא זיהוי של אתרים ספציפיים.
אחת הדרכים , הן לבנות מודל שמנבא את שם האתר. התוקף אוסף מאפיינים כמו גודל חבילה ,
זמני הגעה פנימיים ואת הוליום של הזרימה , מאמן את המודל על הנתונים האלה ומנבא תוצאה.

עוד דרך לגלות על האתר זה אם יש שימוש בפרוטוקול DNS (לא מוצפן) , הלקוח שולח שאילתת
DNS ששם האתר מופיע בה , התוקף יכול להשיג גישה אליה ולגלות את שם האתר.