2025

פרויקט גמר רשתות תקשורת

מגישים:
איתן בארי. סביל פאדל. רוזול זקריה. תאיר מרזיב.

תוכן עניינים

חלק 1- שאלות טכניות

:1

כאשר משתמש מדווח על העברה איטית של קבצים, נבדוק את הגורמים הבאים בשכבת התעבורה:

- **גודל חלון ה – TCP (Window Size) -** אם החלון קטן מדי, שליחת הנתונים תהיה מוגבלת למרות שיש רוחב פס זמין.
- **השהיות** – חיבור עם זמן השהייה גבוה עלול לגרום להורדת קצב ההעברה (בעיקר בפרוטוקול TCP, שמשתמש במנגנון לשליטה בקצב ההעברה).
- **איבוד חבילות (packets) -** גורם ל TCP לבצע שליחה חוזרת של הנתונים, מה שמאט את ההעברה.
- **עומס ברשת** – נתבים עמוסים, רוחב פס מוגבל, או תורים עמוסים ברמה שהרשת מתקשה להתמודד, עלולים לגרום להאטה.
- אם מוגדרים כללים שמגבילים את רוחב הפס עבור פרוטוקול ההעברה זה עשוי להאט את התהליך.

פתרונות אפשריים:

- **בדיקת חלון TCP ואיבוד חבילות -** ניתן להשתמש בפקודות ב Wireshark כדי לנתח את ה Window size ולשנות אותו בהתאם, ניתן גם לבדוק אם יש שידורים חוזרים וכך נבין שיש חבילות שנאבדו וננסה למנוע את זה.
- **בדיקת השהיות ואיבוד חבילות -** ניתן להשתמש ב ping או traceroute כדי לזהות בעיות השהייה ולאתר נתבים שמייצרים עומס וגורמים לאיבוד חבילות. אם מזהים עומס כדאי לנסות להפעיל מחדש את הנתבים או להחליף כבל רשת.
- **בדיקת הגדרות** כדי לוודא שאין מגבלות שמונעות ניצול מלא של רוחב הפס, אם רוחב הפס מוגבל כדאי לנסות להעביר את הקובץ דרך VPN .
- **שימוש בפרוטוקול UDP במידת הצורך** – אם ההעברה תלויה ב TCP אך אינה דורשת אימות (כמו סטרימינג או וידאו), אפשר לשקול שימוש בUDP.

2 . בקרת הזרימה של TCP נועדה למנוע עומס על המקבל ע"י התאמת קצב השליחה ליכולת העיבוד שלו.
ההשפעות במצב שבו השולח חזק יותר מהמקבל יכולות להיות:

- **הגבלת קצב השידור** – אם המקבל איטי ואינו מסוגל לעבד נתונים מהר, הוא ישלח ערך קטן עבור חלון הקבלה. השולח, למרות יכולות העיבוד הגבוהות שלו, ייאלץ להמתין לאישורים (ACKs)ולשלוח נתונים בקצב איטי יותר.
- **איבוד נתונים -** השולח עלול לייצר נתונים במהירות גבוהה, אך בגלל מגבלת העיבוד של המקלט, הנתונים ימתינו בbuffer של המקבל או אפילו יאבדו אם הbuffer יתמלא.
- **יעילות ההעברה -** אם המקבל מעדכן את גודל חלון הקבלה באופן חלקי או בקצב איטי ,השולח (למרות שיש לו כוח עיבוד גבוה) עלול לשלוח חבילות קטנות במקום לנצל את רוחב הפס באופן אופטימלי, מה שגורם ליעילות נמוכה של ההעברה.

- **עומס על הרשת** – כאשר יש כמה מקבלים איטיים ברשת, הם עשויים לגרום להצטברות נתונים בנתבים, זה עלול לגרום לאיבוד חבילות ולזמני השהיה ארוכים יותר עקב צורך בשליחת חבילות חוזרות.

3. כאשר קיימים מספר נתיבים בין מקור ליעד, מנגנון הניתוב (Routing) ברשת צריך לבחור את הנתיב האופטימלי להעברת המידע. בחירת הנתיב משפיעה ישירות על מהירות, אמינות, ועומס הרשת.

כיצד בחירת הנתיב משפיעה על ביצועי הרשת?

- **זמן השהיה** – נתיבים קצרים יותר או כאלה עם פחות hops יובילו לזמן תגובה מהיר יותר. לעומת זאת, אם הנתיב שנבחר עובר דרך רשתות עמוסות, זמן ההשהיה יגדל.
- **רוחב פס** – נתיבים מסוימים תומכים ברוחב פס גבוה יותר מאחרים. אם הנתיב שנבחר מגביל את רוחב הפס, מהירות ההעברה תרד.
- **אמינות** – נתיב שעובר דרך רשתות לא יציבות או כאלה עם אובדן חבילות גבוה יגרום להעברות כושלות או לצורך בשליחה חוזרת, מה שיכול להוריד את הביצועים.
- **עומס הרשת** – אם נתיב מסוים עמוס בחבילות תנועה, הביצועים יורדים בגלל זמני עיכוב והמתנה ארוכים יותר.
- **אבטחה** – נתיב שעובר דרך רשתות פחות מאובטחות עלול לחשוף את התעבורה לתקיפות. לפעמים נתיבים מאובטחים עדיפים על מהירים יותר.

גורמים שיש לקחת בחשבון בהחלטות ניתוב:

- **מדדי עלות** – לחשב את ה"עלות" של כל נתיב לפי פרמטרים כמו מספר קפיצות, רוחב פס וזמן השהיה.
- **איזון עומסים** – לאפשר פיזור תעבורה על פני מספר נתיבים כדי למנוע עומס בנקודה אחת.
- **הימנעות ממסלולים עמוסים** – לשנות נתיב באופן דינמי בהתאם לעומס הנוכחי ברשת.
- תמיכה באיכות שירות – רשתות מסוימות נותנות עדיפות לתעבורה קריטית ובוחרות נתיבים עם פחות השהיה.
- **פתרון חלופי -** במקרה של נתיב שנכשל, פרוטוקולי ניתוב צריכים לספק נתיב חלופי כדי להבטיח המשכיות.

4. MPTCP הוא הרחבה של פרוטוקול TCP המאפשרת לחיבור יחיד להשתמש במספר נתיבים בו זמנית להעברת נתונים. הדבר משפר את ביצועי הרשת בכמה מובנים:

- **מהירות גבוהה יותר -** בMPTCP חיבור יכול לנצל כמה קישורי רשת במקביל (לדוגמה Wi-Fi ו-G4). במקום להיות מוגבל לרוחב הפס רק של אחד החיבורים, אפשר לפצל את התעבורה על כמה מסלולים במקביל, מה שמגדיל את קצב ההעברה הכולל.
- **הפחתת השהיה ושיפור מהירות התגובה -** כאשר MPTCP מזהה נתיב עם זמן השהיה קצר יותר, הוא מפנה אליו יותר תעבורה כדי לשפר את מהירות התגובה של החיבור. אם נתיב מסוים הופך לאיטי או עמוס, הפרוטוקול יכול להעביר חלק מהתעבורה לנתיב אחר באופן דינמי.
- **חיבור יציב יותר -** אם נתיב אחד נופל (למשל, ניתוק מחיבור ה-Wi-Fi), החיבור נמשך דרך נתיב אחר בלי שהמשתמש ירגיש. במקום שיהיה ניתוק מוחלט, המערכת פשוט מעבירה את הנתונים דרך החיבורים הפעילים.
- **איזון עומסים וחלוקת תעבורה -** MPTCP מאפשר לחלק תעבורה בצורה חכמה על פני נתיבים שונים בהתאם לעומס שיש בכל נתיב. כך אפשר למנוע עומסים ברשת ולשפר את ביצועי החיבור.

5. אם מזהים אובדן חבילות משמעותי בין שני נתבים, כדאי לבדוק גורמים אפשריים, גם בשכבת הרשת וגם בשכבת התעבורה.

גורמים בשכבת הרשת:

- **עומס ברשת** – הנתבים מקבלים יותר חבילות ממה שהם יכולים לטפל, מה שמוביל לזריקת חבילות.
  בעיות ניתוב – ניתוב שגוי, מעגלי ניתוב, או טבלאות ניתוב לא מעודכנות עלולים לגרום לאובדן חבילות.
  הפרעות פיזיות - במיוחד ברשתות אלחוטיות (Wi-Fi) או בסיבים אופטיים, הפרעות אלקטרומגנטיות יכולות לפגוע בשלמות החבילות.
- **MTU לא תואם** – אם חבילה גדולה מדי ואין מנגנון Fragmentation, החבילות יכולות להיזרק.

פתרונות:

- **בדיקת עומס ברשת** – שימוש בQoS כדי לתת עדיפות לתעבורה קריטית.
- **עדכון טבלאות ניתוב** – בדיקת טבלאות הניתוב בנתבים באמצעות traceroute ולנסות לעדכן אותן מחדש.
  בדיקת תקינות הכבלים – בדיקה שהכבלים תקינים ולהחליף במידת הצורך, שינוי ערוץ Wi-Fi (Wi-Fi Channel) בנתב כדי להפחית הפרעות.
- **איתור מגבלת MTU -** אם מאתרים מגבלת MTU ברשת ניתן לשלוח את החבילה בחלקים קטנים יותר.

גורמים בשכבת התעבורה:

- **חלון הגלישה (Window Size) קטן מדי** – גורם להעברת נתונים איטית ולעיתים גם להפסקות תעבורה.
- **מנגנון בקרת זרימה (TCP Flow Control) -** אם המקבל מאותת שהוא לא יכול לקבל עוד נתונים, השולח עלול להאט או להפיל חיבורים .
- **הרבה שליחות חוזרות -** כאשר TCP מזהה אובדן חבילות, הוא שולח מחדש, אך יותר מדי שליחות חוזרות יכולות לגרום לעומס נוסף ולאובדן נוסף.
- **Timeouts ושגיאות ACK -** אם הACK לא מגיע, TCP עשוי להאט או להתנתק.

פתרונות:

- **הגדלת חלון הגלישה -** כדי לאפשר שליחה של יותר חבילות לפני המתנה לאישור.
- **שיפור הנתונים של המקבל** – שדרוג של הנתונים של המקבל אם מתאפשר או דחיסה של הנתונים כדי להקטין את המידע שמעבירים.
- **בדיקת איכות החיבור -** בדיקה של איכות החיבור עם ping וtraceroutו ולהשתמש בתיעדוף תעבורה בנתב כדי להבטיח שחיבורים חשובים יקבלו עדיפות. בנוסף, אם משתמשים ברשת אלחוטית, כדאי לשקול חיבור ישיר בכבל כדי להפחית שגיאות.
- **הגדלת timeout של TCP -** ניתן להגדיל את הזמן לפני שהחיבור יתנתק כדי לתת לו יותר זמן לשלוח מחדש חבילות.

חלק 2- ניתוח מאמרים

מאמר 1: FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition

מאמר זה מציג שיטה חדשה לסיווג סוגים שונים של תעבורת אינטרנט, כולל תעבורה המוצפנת באמצעות VPNאו Tor. המחברים מציעים להמיר את הנתונים הבסיסיים של זרימת המידע (גודל החבילות וזמני ההגעה) לתמונה דו-ממדית הנקראת FlowPic. לאחר מכן הם מיישמים רשת עצבית קונבולוציונית (CNN)כדי לזהות את סוג הזרימה, כגון גלישה, צ'אט, וידאו, שיחות קול באינטרנט(VoIP) , העברת קבצים, או אפילו יישום ספציפי כמו Skype או YouTube.

התרומה העיקרית

הרעיון המרכזי הוא שעל-ידי יצירת FlowPics, משימת הסיווג הופכת לדומה לזיהוי תמונות, שהוכיח את עצמו כחזק מאוד בתחומים רבים. במקום להסתמך על מאפיינים שנבחרו ידנית או לקרוא את התוכן עצמו (שלעתים קרובות מוצפן), המאמר מראה כיצד להשתמש ב CNN-כדי לזהות קטגוריות תעבורה. זה עובד אפילו עבור מקטעים חד-כיווניים קצרים של זרימה (לדוגמה, בלוקים של 60 שניות) ונמנע מהצורך לאחסן כמויות גדולות של מידע או לפענח תעבורה מוצפנת.

מאפייני התעבורה שנעשה בהם שימוש וחדשנות

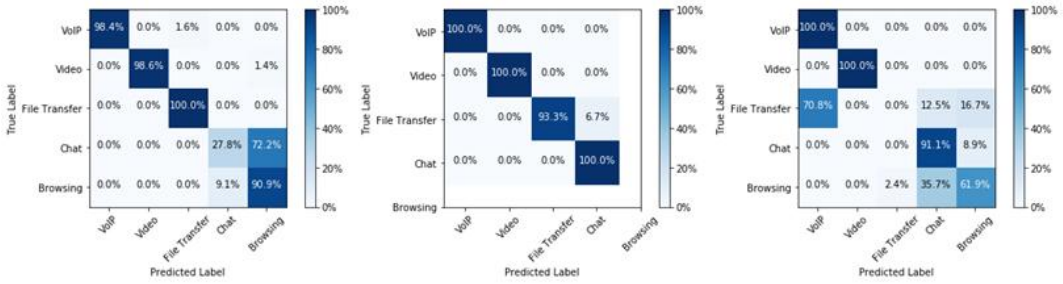
הם לוקחים כל זרימה ומציינים את הגודל (עד 1500 בייטים) ואת זמן ההגעה המדויק של כל חבילה. לאחר מכן הם ממפים את גדלי החבילות לפי זמן לרשת של 1500 על 1500. כל תא ברשת סופר כמה חבילות הופיעו עבור גודל וזמן זה. רשת זו היא ה ,FlowPic-שניתן להזין ל .CNN-לא נעשה שימוש בנתוני התוכן. הצעד החדשני כאן הוא הצגת הנתונים כ"תמונה" שמשקפת דפוסי זמן וגודל בצורה חזותית, מה שמקל על ה CNN-לחלץ דפוסים משמעותיים.

תוצאות עיקריות

הם בודקים את השיטה שלהם על מסדי נתונים מאוניברסיטת New Brunswick (מסד הנתונים ISCX VPN-nonVPN ומסד הנתונים ISCX Tor-nonTor) הכוללים חמש קטגוריות עיקריות ,VoIP :וידאו, צ'אט, גלישה והעברת קבצים, בשלושה תנאים: תעבורה לא מוצפנת(non-VPN) , תעבורה מוצפנת תעבורה מוצפנת בעזרת VPN ותעבורה מוצפנת בעזרת Tor.

הם משיגים דיוק גבוה מאוד במספר משימות. לדוגמה, כאשר מבחינים בין חמש קטגוריות תעבורה בתעבורת ,VPN הם מגיעים לדיוק של כ-98.4%. עם תעבורה שאינה דרך VPN, הדיוק הוא כ-Tor85% . מורכב יותר, אך הם עדיין משיגים כ-68% עבור המשימה בת חמש הקטגוריות. הם גם עושים ניסוי מחלקה אחת מול כולם (למשל, זיהוי האם התעבורה היא VoIP או לא) ומקבלים תוצאות גבוהות של עד 99% במקרים רבים. בדיקה נוספת מתמקדת בזיהוי יישומים ספציפיים, כגון Skype לעומת YouTube לעומת Netflix לעומת יישומי VoIP או וידאו אחרים, והמודל מסווג אותם נכון בדיוק של כ-99.7%.

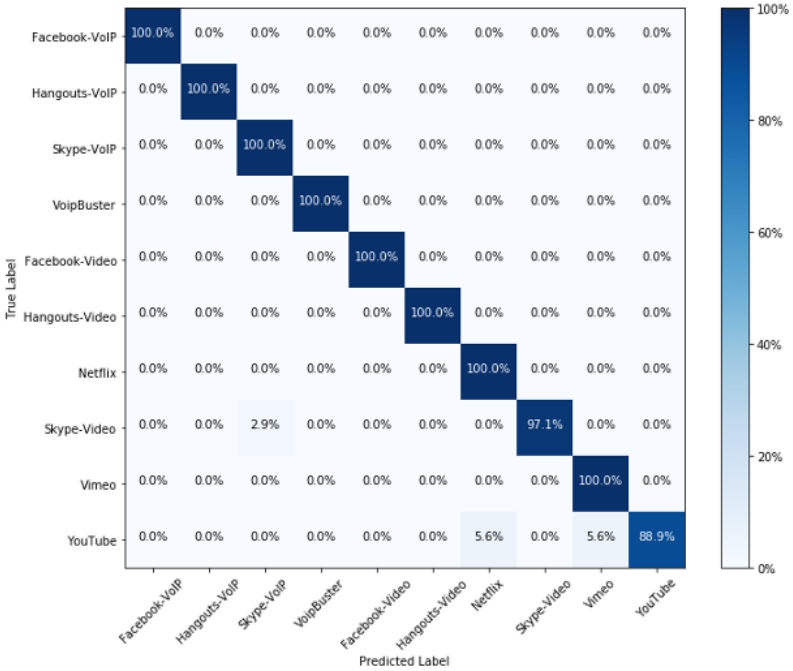
להלן איור (במקור איור 4 במאמר) המציג את מטריצות הבלבול לסיווג חמש הקטגוריות(Voip, וידאו, צ'אט, גלישה, העברת קבצים) בשלושת התנאים(non-VPN, VPN, Tor) . כל שורה היא הקטגוריה האמיתית, וכל עמודה היא מה שה CNN-ניבא:



(a) Non-VPN (b) VPN (c) Tor

אנו רואים שרוב הקטגוריות מזוהות בקלות, אם כי הבחנה בין צ'אט לגלישה היא לפעמים מסובכת בתנאי non-VPNותעבורת Tor מאתגרת יותר באופן כללי כי Tor מצפין ומאגד זרימות בצורה ייחודית.

להלן איור נוסף (במקור איור 5 במאמר), המציג את מטריצת הבלבול לזיהוי עשרה יישומים שונים של Voip או הזרמת וידאו (לדוגמא,Google Hangouts, Skype, YouTube) כמעט כל יישום מזוהה עם רמת דיוק ורגישות גבוהה מאוד:



תובנות מתוצאות אלה

1. המרת נתוני זמן וגודל לתמונות מאפשרת ל CNN-ללמוד באופן אוטומטי מה חשוב, במקום לבחור ידנית מאפיינים מספריים או להסתכל על תוכן המידע.

2. גם אם יישום מסוים (כמו וידאו בפייסבוק) לא היה בשימוש במהלך האימון, ה CNN-לעתים קרובות מזהה שהיישום החדש שייך לקטגוריה הנכונה (למשל, וידאו). משמעות הדבר היא שהמודל קולט התנהגויות ליבה של סוג תעבורה, במקום רק לשנן חתימות של יישומים ספציפיים.

3. בהשוואה בין תעבורה לא מוצפנת, VPN ותעבורת Tor, תעבורה שאינה מוצפנת קלה יחסית לסיווג, VPN עדיין מטופל היטב על ידי השיטה, ו Tor נשאר הקשה ביותר בשל האופן שבTor מאגד חבילות. למרות זאת, התוצאות עדיין טובות ומראות פוטנציאל לשימוש בעולם האמיתי.

4. ארכיטקטורת ה CNN-שנעשה בה שימוש (רשת מסוג LeNet-5) נשארה זהה בכל הניסויים. המחברים לא כיוונו את העיצוב לכל תרחיש, כך שיש פוטנציאל לשפר את הארכיטקטורה עוד יותר.

מסקנה

באופן כללי, המאמר מראה שעל-ידי שימוש בייצוג FlowPic בתוספת רשת עצבית קונבולוציונית, ניתן לסווג תעבורה מוצפנת ולא מוצפנת לקטגוריות או יישומים ספציפיים בדיוק גבוה מאוד. גישה זו דורשת רק גדלים וזמנים של חבילות, מה שהופך אותה לעמידה להצפנה ויעילה מבחינת אחסון נתונים. ההצלחה בזיהוי יישומים לא מוכרים או חדשים מצביעה על כך שהשיטה לוכדת את הדפוס האמיתי של כל קטגוריה, במקום להתאים יתר על המידה לדוגמאות מסוימות.

מאמר 2: Early Traffic Classification With Encrypted ClientHello: A Multi-Country Study

מאמר זה מציג שיטה חדשה לסיווג זרמי תעבורת אינטרנט שונים בשלב מוקדם מאוד, גם כאשר הם משתמשים בטכניקות הצפנה מתקדמות המסתירות מטא-דאטה חשוב בלחיצת היד הראשונית. המחברים מתמקדים בהתפתחות חדשה הנקראת Encrypted ClientHello (ECH) ב TLS 1.3, כאשר הודעת ClientHello (כולל ציון שם השרת) יכולה להיות מוצפנת. בגלל ECH, אסטרטגיות סיווג תעבורה ישנות רבות המסתמכות על צפייה בSNI הלא מוצפן אינן עובדות היטב יותר. כדי לפתור זאת, המחברים מתכננים מסווג היברידי המשלב שדות לחיצת יד TLS שנותרו לא מוצפנים עם סטטיסטיקות זרימה בסיסיות, כל זאת לפני שהשרת שולח נתוני יישום בפועל. הם גם אוספים מערך נתוני תעבורה גדול ממדינות רבות כדי לבדוק ביסודיות את הגישה שלהם.

התרומה העיקרית

1. הם מציגים מסווג תעבורה היברידי חדש מבוסס Random Forest (hRFTC) המזהה תעבורה על ידי שילוב:

- שדות שנותרו לא מוצפנים מלחיצת היד של TLS (לדוגמא, פרמטרים של cipher-suite אשר לא ניתן להסתיר אותם במלואם על ידי ECH).
- סטטיסטיקות מבוססות שטף על גודל החבילות והתזמונים לפני שנתוני שרת האפליקציה מתחילים.

2. הם בונים מערך נתוני תעבורת TLS גדול וממדינות מרובות עם יותר מ-600,000 זרימות מצפון אמריקה, אירופה ואסיה. מערך נתונים זה כולל 19 סוגי זרימות, הכוללים וידאו ממאגר זמני, סרטונים קצרים, הזרמת מוזיקה ותעבורת אינטרנט פופולרית מגוונת. מערך הנתונים זמין לציבור כדי לעזור לחוקרים אחרים להעריך את השיטות שלהם.

3. הם בודקים עד כמה שיטות סיווג מתקדמות יכולות להתמודד עם ECH, ומראים שמסווגים המסתמכים אך ורק על פרטי לחיצת יד TLS עלולים לפעול באופן גרוע (מכיוון ש ECH-מצפין את רובם). לעומת זאת, גישת הhRFTC שלהם משפרת משמעותית את דיוק הסיווג בתרחיש מאתגר זה.

מאפייני התעבורה שנעשה בהם שימוש (וחדשנות)

המסווג שלהם משתמש בשני סוגי מאפיינים:

1. **שדות TLS מבוססי חבילות (לא מוצפנים)** גם עם ECH, חלק מהשדות של ClientHello ו-ServerHello ההתחלתיים חייבים להישאר לא מוצפנים, כגון מידע בסיסי על חבילת ההצפנה, גרסת QUIC, או סמנים אקראיים להרחבה לצורך תאימות לאחור. המחברים אוספים את הבייטים שנשארו ומסדרים אותם ל"תוכן מורכב מחדש" באורך קבוע, מה שעוזר לאלגוריתם Random Forestלפרש נתוני לחיצת יד TLS בצורה מובנית.

2. **סטטיסטיקות חבילות מבוססות זרימה** לפני שמגיעה החבילה הראשונה בכיוון ההורדה המכילה נתוני יישום בפועל, הם מודדים כמה חבילות מופיעות, גודלן, תדירות הגעתן ובאיזה כיוון (העלאה או הורדה). אלה כוללים:

- מספר החבילות בכל כיוון.
- גודלי חבילות ממוצעים, מינימליים ומקסימליים בהעלאה ובהורדה.
- מדדי זמן בין חבילות.

- היסטוגרמה קטנה של גדלי חבילות בטווחים שונים.

החדשנות היא שילוב פרמטרי TLS חלקיים אלה עם סטטיסטיקות זרימה לטווח קצר. רוב הגישות בעבר השתמשו רק באחד או בשני, או שהם דרשו יותר חבילות ממה שמעשי לצרכי זמן אמת. גישה היברידית חדשה זו יכולה לעבוד עם מעט חבילות ועם לחיצות יד מוצפנות ECH.

תוצאות עיקריות ותובנות

המאמר מראה שהסתמכות אך ורק על נתוני TLS לא מוצפנים יכולה להיות לא אמינה עבור תעבורתECH , במיוחד מכיוון שאתרים מרובים יכולים לשתף פרמטרי TLS דומים. המחברים מדגימים:

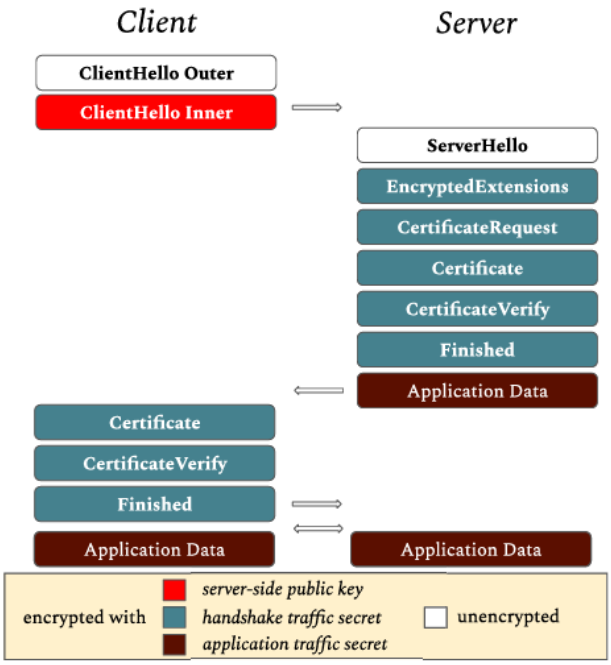
- **ביצועים נמוכים של שיטות מבוססות TLS בלבד תחת ECH** - עם אימוץ של חבילות הצפנה זהות או הגדרות לחיצת יד על ידי שירותים רבים, שיטות המסתכלות רק על שדות TLS יכולות לרדת לציון F של ~38% בתת-קבוצות קשות. זה מאשר ש ECHיכול לסכל סיווג אם אנו מסתמכים רק על בייטים של לחיצת יד שנותרו.

- **ביצועי המסווג ההיברידי** מסווג התעבורה היברידי (hRFTC) Random Forest שלהם משיג ציון F של עד ~94.6% על מערך הנתונים המלא, קפיצה גדולה מעל שיטות קיימות. הוא נשאר מהיר להפעלה ועמיד גם כאשר רק חלק קטן מהתעבורה משמש לאימון.

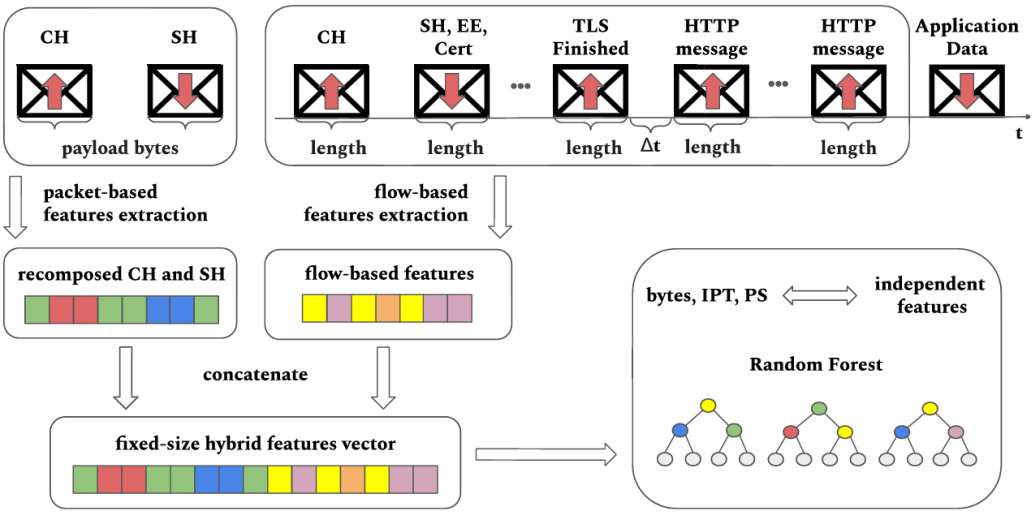
- **השפעת מאפיינים מבוססי זרימה** המחברים מודדים את חשיבות המאפיינים ב-Random Forestשלהם כדי לראות אילו היבטים חשובים ביותר. התוצאות מראות שהשילוב של סטטיסטיקות מבוססות גודל, פערי תזמון ושדות כותרת TLS מסוימים שנותרו הוא הדרך החזקה ביותר לסווג תעבורה בשלב מוקדם.

- **וריאציות גיאוגרפיות** בדיקות מראות שהתעבורה יכולה להיראות שונה בהתאם למיקום המשתמש, לעתים קרובות בגלל האופן שבו רשתות משלוח תוכן מפרקות חבילות או בגלל גדלי מקטעים מקסימליים שונים. מסווג שאומן באזור אחד עשוי לפעול באופן גרוע באזור חדש לגמרי. לכן, אם ספק רשת רוצה דיוק גבוה בכל העולם, ייתכן שהמודל יצטרך אימון מקומי או כוונון עדין.

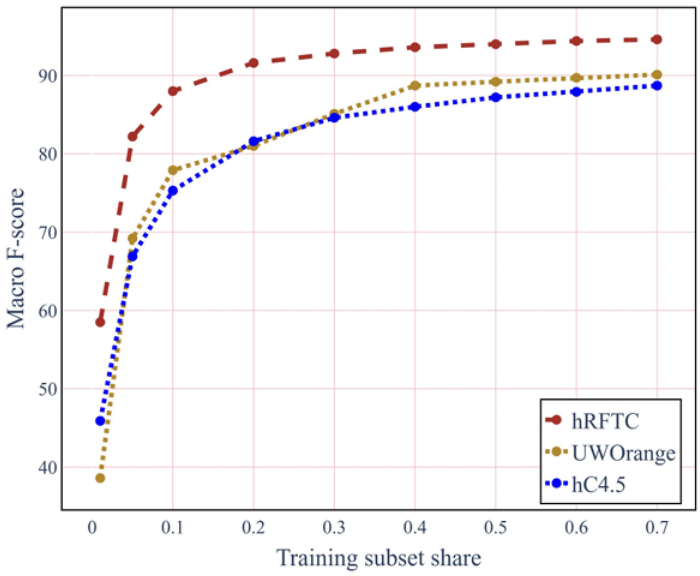
החוקרים מדגימים את לחיצת היד של TLS 1.3 ביחד עם הודעת ClientHello מוצפנת (ECH), ומדגימים כיצד חלק מהודעת ה ClietHello עוברת הצפנה (ClientHello Inner בתרשים), תוך החבאה של ה SNI, בעוד שרק חלק קטן מההודעה נשאר חשוף:



הם גם מציגים כיצד המסווג ההיברידי שלהם משתמש בשילוב של מאפיינים מבוססי-חבילות (שדות TLS שנותרו) ומבוססי-זרימה (גדלי חבילות, זמני הגעה):



המחברים מציגים כיצד גודל נתוני האימון משפיע על הביצועים. ככל שהאימון מופחת, דיוק הסיווג יורד, אך הגישה ההיברידית שלהם עדיין נשארת יעילה.



מסקנה

על ידי הצפנת הודעת הClientHello ב TLS 1.3 רוב המטא-דאטה הסטנדרטי המשמש לסיווג תעבורה מוקדם מוסתר. עם זאת, מאמר זה מראה שחלק מפרטי לחיצת היד בהכרח נשארים חשופים, ועל ידי הוספת סטטיסטיקות זרימה לטווח קצר, עדיין ניתן לסווג תעבורה בדיוק גבוה לפני הגעת נתוני היישום. מסווג Random Forest ההיברידי החדש שלהם עולה בביצועיו על שיטות רבות ידועות במערך נתונים גדול ורב-מדינתי, תוך הדגשת האופן שבו הבדלים מבוססי מיקום יכולים להשפיע על ההתנהגות המוקדמת של התעבורה. עבודה זו מרמזת שהסתרה מלאה של סוג היישום ממכשירי רשת קשה יותר מהצפוי, אך גם שסיווג אינו פשוט יותר כמו קריאת SNI או מספר שדות .TLS ככל שסטנדרטי ההצפנה מתפתחים, שילוב של שדות לחיצת יד חלקיים עם סטטיסטיקות זרימה בסיסיות צפוי להישאר הדרך היעילה ביותר ליישם סיווג תעבורה מוקדם.

מאמר 3: Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application

מאמר זה חוקר כיצד תוקף שיכול רק לצפות באופן פסיבי בתעבורת HTTPS עדיין יכול לזהות איזו מערכת הפעלה, דפדפן ואפליקציה המשתמש מפעיל, גם אם התעבורה מוצפנת באמצעותTLS/SSL . המחברים בונים מאגר נתונים גדול של למעלה מ-20,000 מפגשים ומראים שעל ידי בחינת מגוון מאפיינים כגון גדלי חבילות, זמני ביניים להגעה, ופרטי לחיצת ידיים של SSL הם יכולים לסווג במדויק את הצירוף <מערכת הפעלה, דפדפן, אפליקציה>.

תרומה עיקרית

הם מדגימים שניתן לזהות את כל שלושת הרכיבים (מערכת הפעלה, דפדפן ואפליקציה) בתעבורת HTTPS של המשתמש. מערכת ההפעלה יכולה להיות Windows, Ubuntu וmacOS . הדפדפן יכול להיות Chrome, Firefox, Safari, Internet Explorer או Non-Browser (עבור אפליקציות שולחן עבודה מסוימות). והאפליקציה כוללת פלטפורמות כמו Facebook, YouTube, Twitter, Dropbox ועוד.

המחברים אוספים מאגר נתונים של יותר מ-20,000 מפגשים מתויגים. מאגר נתונים זה לוכד תרחישים שונים: פעולות משתמש פעילות, תעבורת רקע ממערכות הפעלה ודפדפנים, וסוגים שונים של תנאי רשת (קווי/אלחוטי, זמנים שונים ביום).

הם מציגים סט של מאפיינים חדשים, מעבר למדדים הטיפוסיים ברמת החבילה או מבוססי זמן המתמקדים ספציפית בשדות לחיצת ידיים של SSL ובדפוסים "מתפרצים (bursty) " המוצגים לעתים קרובות על ידי תעבורת דפדפן. מאפיינים חדשים אלה, בשילוב עם סט סטנדרטי של מאפייני בסיס, משפרים משמעותית את דיוק הסיווג.

מאגר נתונים ומיצוי מאפיינים

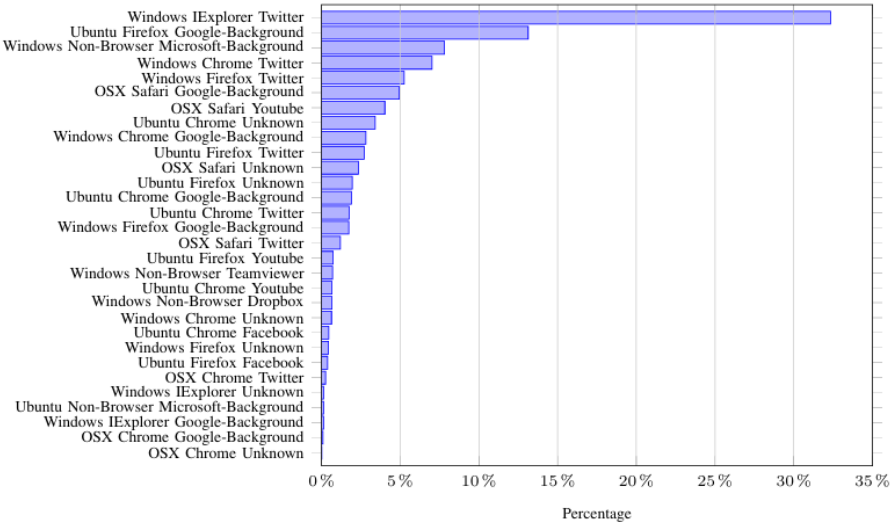
- מאגר הנתונים נאסף באמצעות כלי האוטומציה של האינטרנט Selenium בתוספת אינטראקציות ידניות. התעבורה נלכדה בפורט 443, והמפגשים הוגדרו על ידי החמישייה הבאה: <פרוטוקול IP, מקור IP, יעד, פורט מקור, פורט יעד>.
- התווית של כל מפגש היא הצירוף <מערכת הפעלה, דפדפן, אפליקציה>. בסך הכל, ישנם 30 צירופים אפשריים, כגון "Windows, Chrome, YouTube" או

"Ubuntu, Firefox, Twitter".

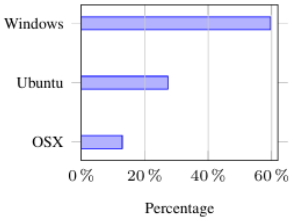
- שני סוגים של מאפיינים מחולצים:

1. **מאפייני בסיס:** אלה נמצאים בשימוש נרחב במשימות סיווג תעבורה אחרות (למשל, סך החבילות, בייטים בכל כיוון, זמני ביניים להגעה, ומדדים סטטיסטיים של גדלי חבילות).
2. **מאפיינים חדשים:** המחברים מנתחים פרטי לחיצת ידיים של SSL (שיטות הצפנה, מזהי מפגש וכו'), פרמטרים של TCP (גודל חלון, גורמי סילום), ומאפיינים של תעבורת דפדפן "מתפרצת", שבה בקשות מגיעות בשיאים קצרים המופרדים על ידי תקופות של חוסר פעילות.

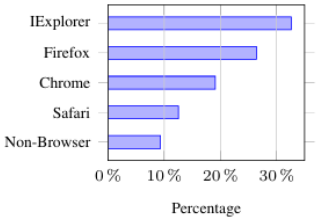
הם ממחישים את התפלגות מאגר הנתונים, המראה כמה מפגשים יש לכל מערכת הפעלה, דפדפן ואפליקציה:



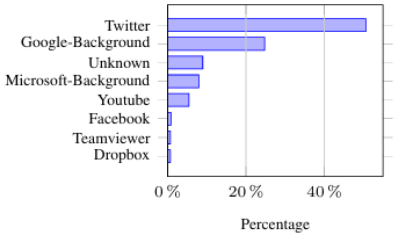
(a) Dataset labels (tuple) statistics



(b) Dataset OS statistics

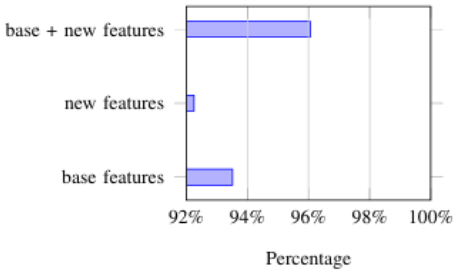


(c) Dataset browser statistics

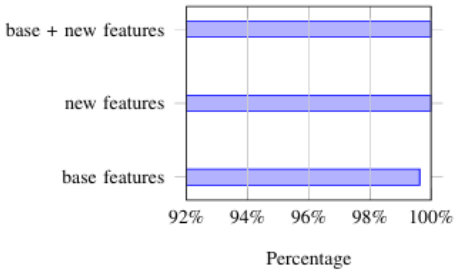


(d) Dataset application statistics

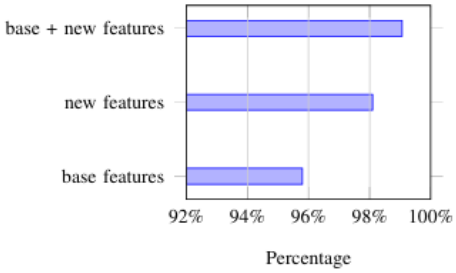
הם משווים את דיוק הסיווג בעת שימוש במאפייני בסיס בלבד, מאפיינים חדשים בלבד, וסט מאפיינים משולב.



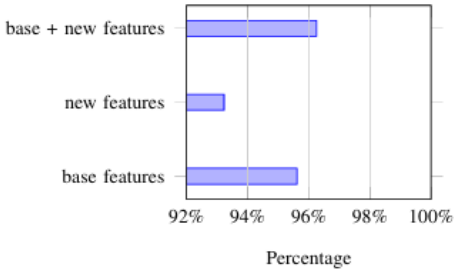
(a) Tuple Accuracy Results



(b) OS Accuracy Results

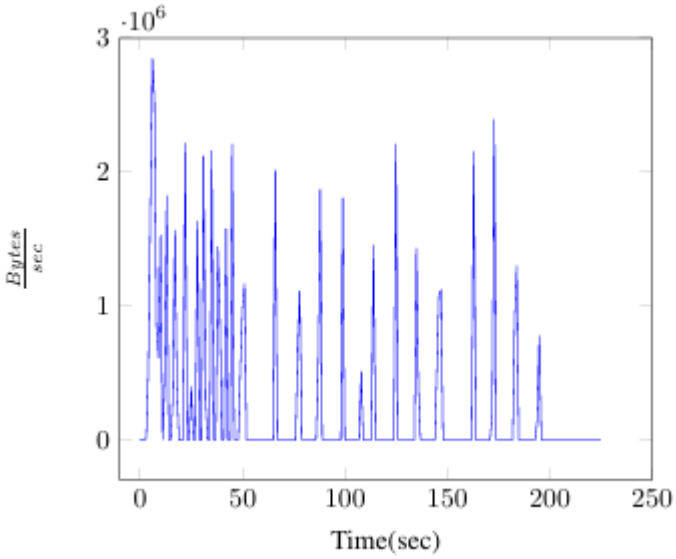


(c) Browser Accuracy Results



(d) Application Accuracy Results

הם גם מספקים דוגמה להתנהגות המתפרצת של תעבורת דפדפן, שבה בייטים משודרים בשיאים חדים לאורך זמן.



תוצאות עיקריות

- דיוק הצירוף <מערכת הפעלה, דפדפן, אפליקציה> על ידי הזנת סט המאפיינים המשולב למכונת תמיכה וקטורית (SVM) עם ליבת RBF הם משיגים דיוק של כ-96.06%. זה מראה שגם כאשר תוכן הTLS מוצפן לחלוטין, דפוסי המטא של התעבורה עדיין חושפים פרטים על מערכת ההפעלה, הדפדפן והאפליקציה.

- השפעת המאפיינים החדשים המאפיינים החדשים מבוססי SSL והתפרצויות שהציגו המחברים משפרים את הדיוק באופן ניכר. שימוש במאפייני בסיס בלבד מניב דיוק של כ-93.5%, שעולה ליותר מ-96% כאשר המאפיינים החדשים נכללים.

- תובנות ממטריצות בלבול ברוב המקרים, המסווג מבחין בין דפדפנים וסוגי מערכות הפעלה כמעט באופן מושלם. מקור השגיאה העיקרי נובע מזרימות מתויגות "לא ידוע", שעשויות לייצג תעבורה מחוץ לסט האפליקציות הידועות. אפילו כך, הטעויות הן מעטות יחסית.

מסקנה

באמצעות תצפית שיטתית על התעבורה ומיצוי מאפיינים, תוקף יכול לזהות במדויק את מערכת ההפעלה, הדפדפן והאפליקציה של המשתמש, אפילו כשתוכן ה HTTPS-מוצפן. עבודה זו מעלה חששות לפרטיות על ידי הדגשת כיצד מטא-נתונים, כגון אורכי חבילות ומאפייני לחיצת ידיים, יכולים לשמש לזיהוי טביעת אצבע של מכשירי המשתמשים והתנהגויותיהם. מחקרים עתידיים עשויים להרחיב את הטכניקות הללו כדי לכסות זיהוי של מערכת הפעלה/דפדפן במובייל או להבדיל בין פעולות משתמש (למשל, פרסום לעומת קריאת תוכן) בתוך אפליקציה בודדת.

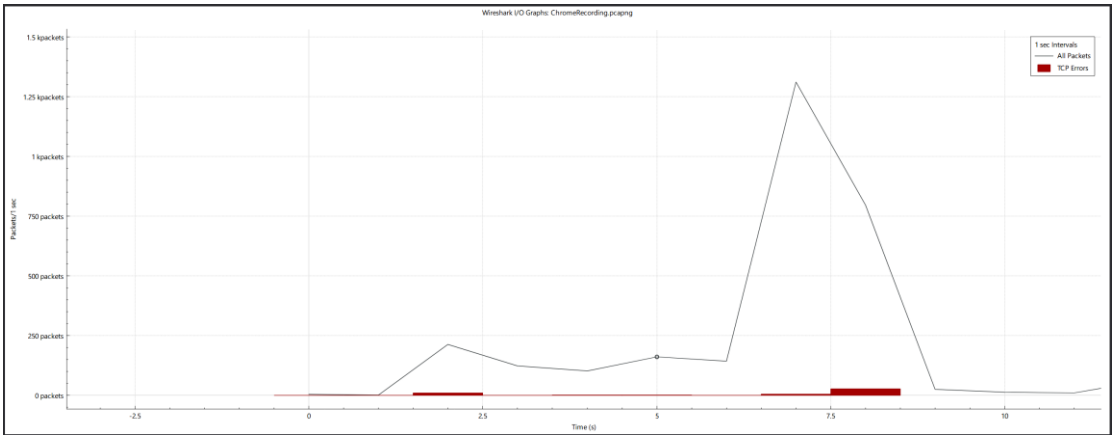
חלק 3- ניתוח גרפים

תוכנות שמהן הוקלטה התנועה: Chrome, Firefox, Discord, Spotify, YouTube.

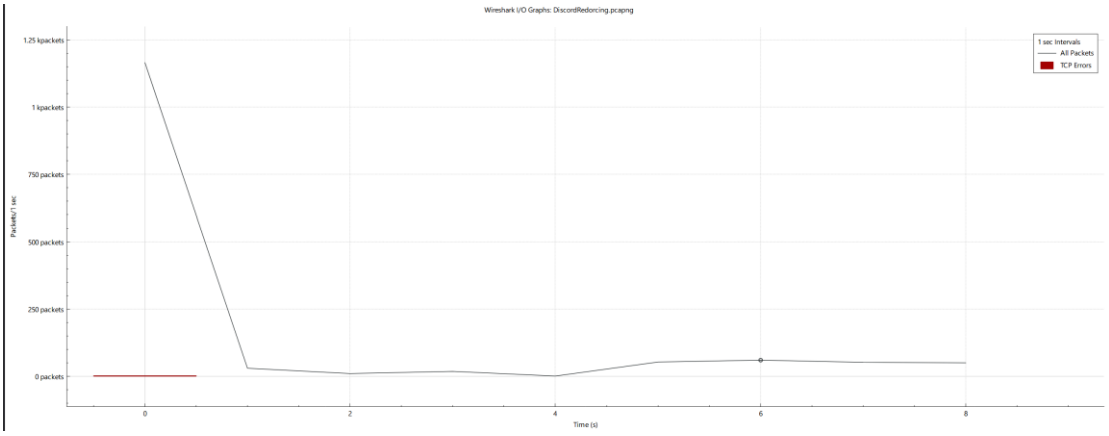
להלן גרפים מויירשארק ומהקוד שלנו שיצרנו עבור תכונות מסויימות בתנועה באינטרנט:

מספר פקטות לאורך זמן:

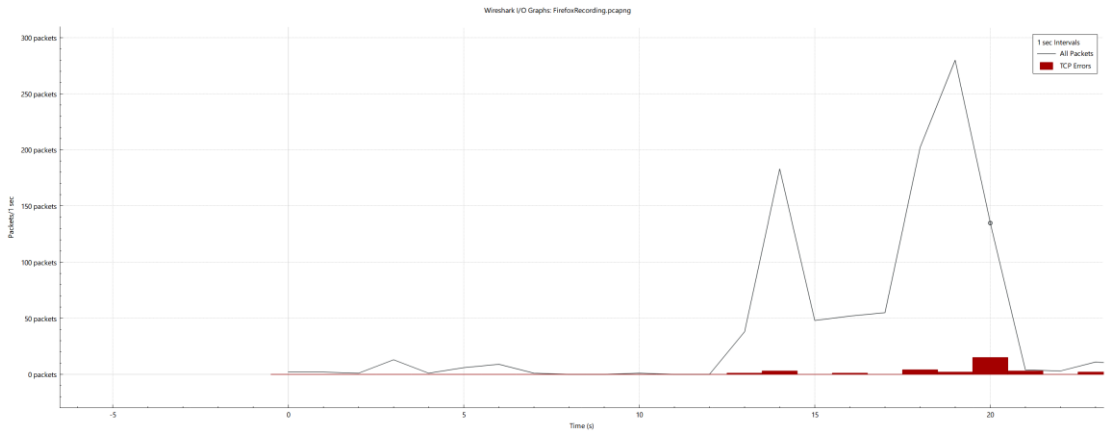
:Chrome



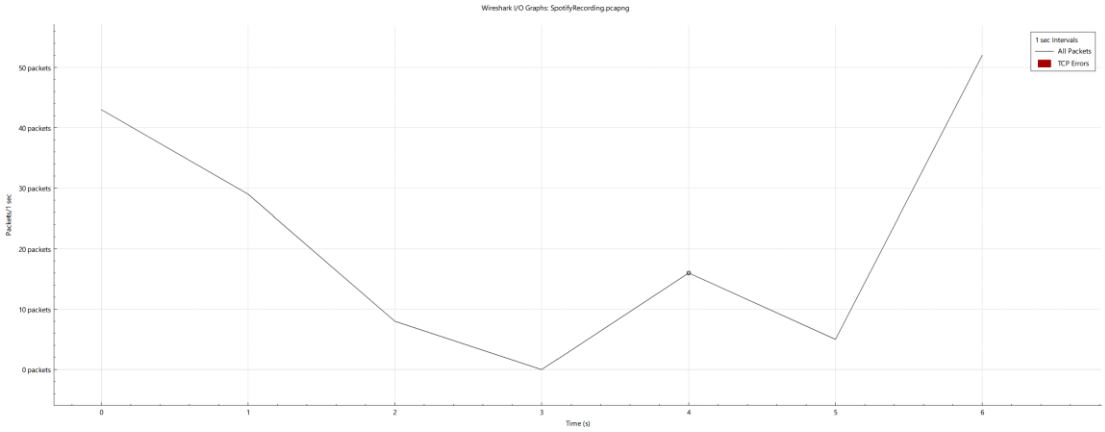
:Discord



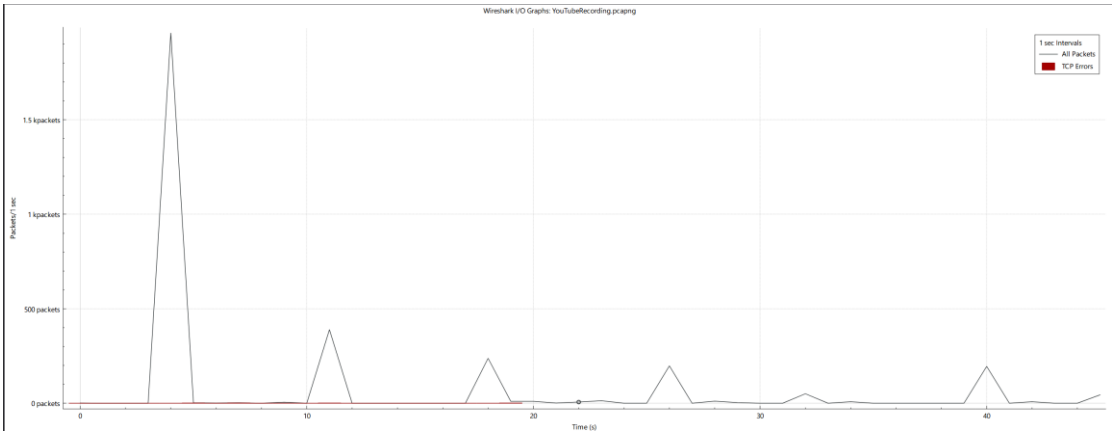
:Firefox



:Spotify

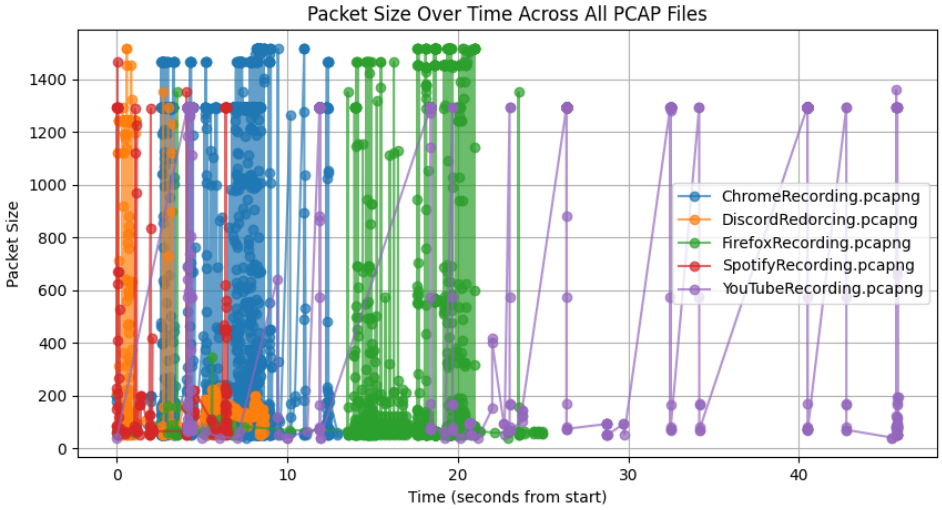


:YouTube



אצל הדפדפנים Firefox ו Chrome התנועה היא אחידה עם קפיצות גדולות, Discord מקבל המון פקטות בהתחלה אבל לאורך השיחה הכמות יורדת ונשארת אחידה, בYouTube יש קפיצה אחת גדולה בהתחלה ועוד המון קטנות בהמשך, וSpotify התנועה עולה ויורדת בהדרגה.

גודל פקטות לאורך זמן:



הסבר:

Chrome: משתמש בהרבה גדלים של פקטות, חלקן גדולות מאוד וחלקן קטנות מאוד. גודל הפקטות עולה על זה של האפליקציות האחרות לרוב.

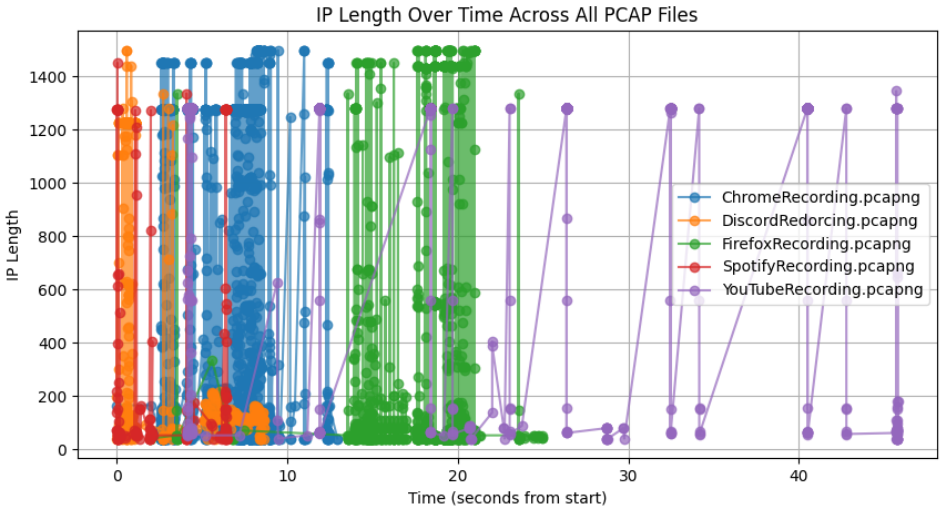
Discord: משתמש בפקטות גדולות מאוד בהתחלה אבל הגודל שלהן יורד מאוד מהר. רוב גדלי הפקטות לא עולים על אלה של האפליקציות האחרות.

Firefox: גרף גודל הפקטות נראה דומה לזה של Chrome, רק שהוא משתמש בפקטות גדולות יותר לעיתים תכופות יותר.

Spotify: משתמש לרוב בפקטות בטווח קבוע, חוץ מבהתחלה. גודל הפקטות לרוב קטן מזה של האפליקציות האחרות.

YouTube: משתמש בפקטות בטווח קבוע. גודל הפקטות לא עולה על זה של Chrome וFirefox בשיאן.

אורך פקטת IP (הערה: הגרף נראה כמעט זהה לגרף גודל הפקטות משום שאורך פקטת הIP כמעט זהה לגודל כל הפקטה):



הסבר:

Chrome: משתמש בהרבה גדלים של פקטות, חלקן גדולות מאוד וחלקן קטנות מאוד. גודל הפקטות עולה על זה של האפליקציות האחרות לרוב.

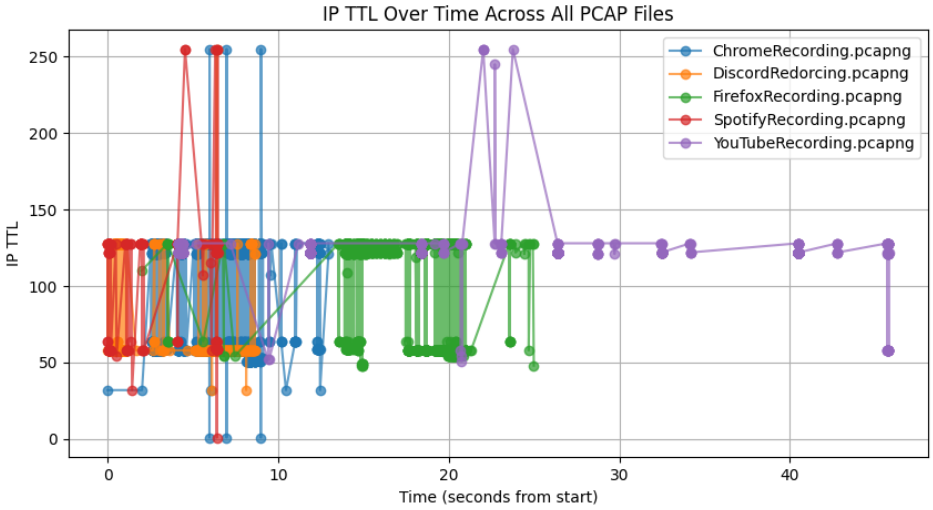
Discord: משתמש בפקטות גדולות מאוד בהתחלה אבל הגודל שלהן יורד מאוד מהר. רוב גדלי הפקטות לא עולים על אלה של האפליקציות האחרות.

Firefox: גרף גודל הפקטות נראה דומה לזה של Chrome, רק שהוא משתמש בפקטות גדולות יותר לעיתים תכופות יותר.

Spotify: משתמש לרוב בפקטות בטווח קבוע, חוץ מבהתחלה. גודל הפקטות לרוב קטן מזה של האפליקציות האחרות.

YouTube: משתמש בפקטות בטווח קבוע. גודל הפקטות לא עולה על זה של Chrome וFirefox בשיאן.

IP TTL:



הסבר:

Chrome: רוב הפקטות נעות בין IP TTL בטווח של בערך 125 עד לקצת יותר מ50, שזה הטווח שבו רוב הפקטות מכל האפליקציות נמצאות. יש מעט שמגיעות ל-יותר מ250, מעט שמגיעות לפחות מ50 ומעט שמגיעות לאפס.

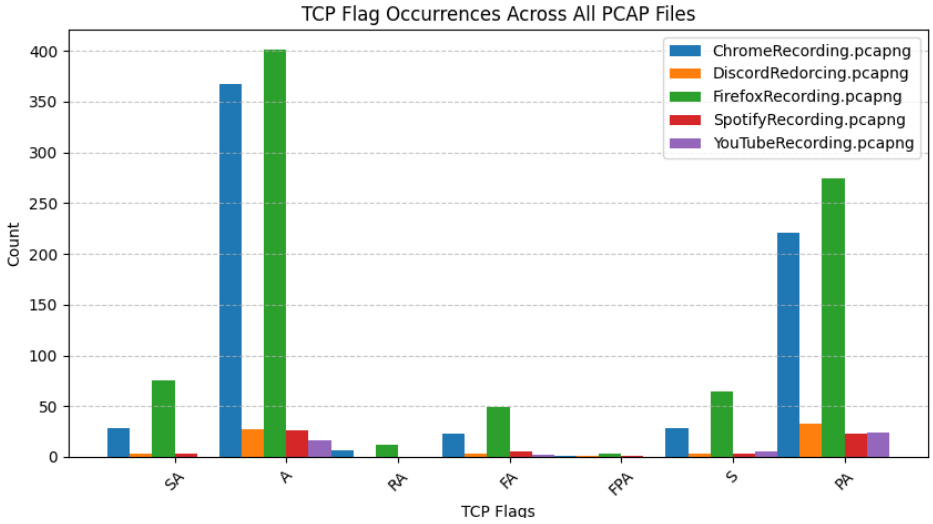
Discord: רוב הפקטות נעות בין IP TTL בטווח של בערך 125 עד לקצת יותר מ50, מעטות מגיעות למתחת ל50.

Firefox: רוב הפקטות נעות בין IP TTL בטווח של בערך 125 עד לקצת יותר מ50, שזה הטווח שבו רוב הפקטות מכל האפליקציות נמצאות. מעטות יורדות מתחת ל50.

Spotify: רוב הפקטות נעות בין IP TTL בטווח של בערך 125 עד לקצת יותר מ50, שזה הטווח שבו רוב הפקטות מכל האפליקציות נמצאות. יש מעט שמגיעות ל-יותר מ250, מעט שמגיעות לפחות מ50 ומעט שמגיעות לאפס.

YouTube: משתמש בפקטות בטווח קבוע. גודל הפקטות לא עולה על זה של Chrome וFirefox בשיאן.

דגלי TCP:



הסבר:

Chrome: פקטות הTCP שChrome משתמש בהן משתמשות הכי הרבה בדגלי A וPA, ורק מעט מהן משתמשות בדגלי SA, RA, FA וS.

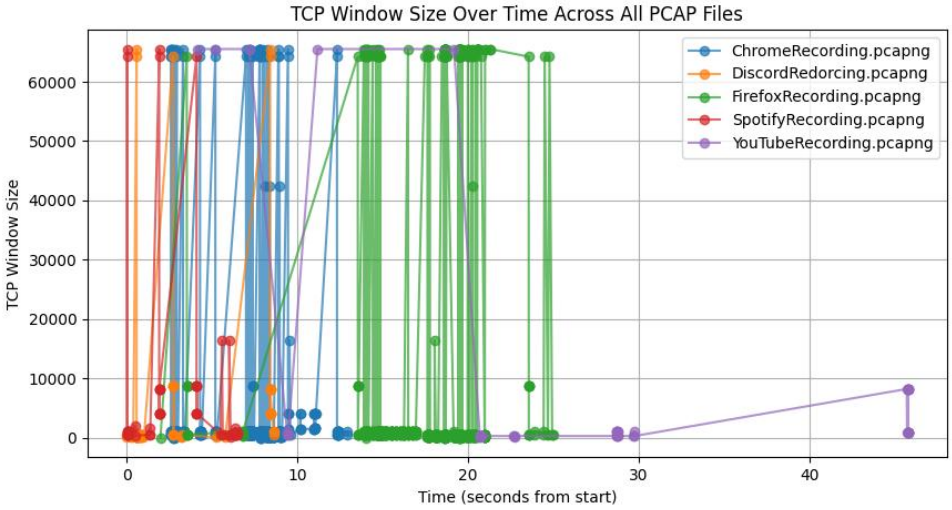
Discord: לא משתמש בהרבה פקטות TCP יחסית לFirefox וChrome. הדגלים הכי נפוצים בפקטות הTCP שלו הם A וPA כאשר SA, FA וS נמצאים רק בבודדות מהפקטות שלו.

Firefox: גרף דגלי הTCP נראה דומה לזה של Chrome, רק שהוא משתמש ביותר פקטות ממנו.

Spotify: גרף דגלי הTCP נראה דומה לזה של Discord, רק שיש אצלו פחות גדלי PA וA, אבל מעט יותר דגלי FA.

YouTube: הדגלים הכי נפוצים בגרף של YouTube הם A וPA.

גודל חלון TCP:



הסבר:

Chrome: גודל חלון הTCP של Chrome מתחלק בין קרוב מאוד לבערך 65000 או לקרוב ל0.

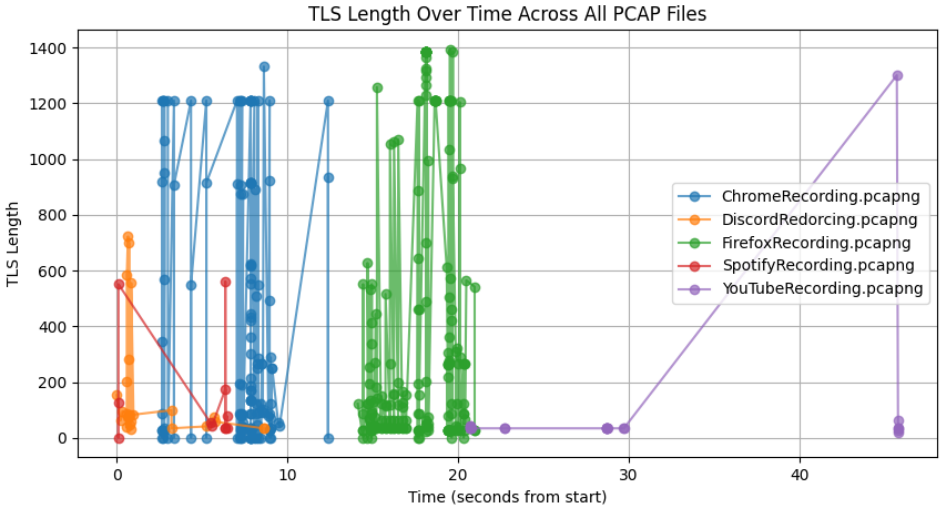
Discord: הגרף של Discord דומה לשל Chrome אבל יש פחות פקטות בקצה העליון של הגרף ויותר פקטות שקרובות לגודל חלון TCP של 10000.

Firefox: גרף גודל חלון הTCP נראה דומה לזה של Chrome אבל הוא משתמש בפקטות על חלונות TCP גדולים יותר לעיתים תכופות יותר. נראה שהוא משתמש ביותר פקטות TCP בכללי, משום שיש גם יותר פקטות TCP שקרובות ל0.

Spotify: משתמש לרוב בפקטות בטווח קבוע, חוץ מבהתחלה. גודל הפקטות לרוב קטן מזה של האפליקציות האחרות.

YouTube: משתמש בפקטות בטווח קבוע. גודל הפקטות לא עולה על זה של Chrome וFirefox בשיאן.

גודל חלון TLS:



הסבר:

Chrome: רוב פקטות הTLS מרוכזות בטווח שבין קצת יותר מ1200 ל0.

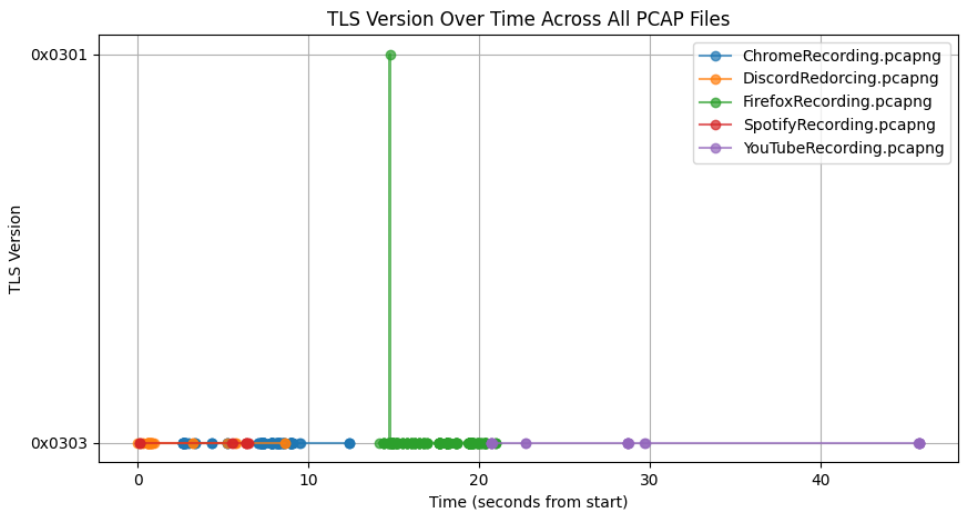
Discord: הפקטות מרוכזות בטווח של קצת פחות מ800 לקצת יותר מ0.

Firefox: יש יותר פקטות TLS ורובן בטווח של Chrome, למרות שיש גם מעט שעברו את הטווח והגיעו כמעט ל1400.

Spotify: משתמש בפקטות בגודל שבין קצת פחות מ600 ל0.

YouTube: רוב פקטות הTLS שYouTube משתמש בהן קטנות מאוד, חוץ מאחד שעוברת את גבול מ1200.

גרסת TLS:



הסבר:

רוב הפקטות מרוב האפליקציות משתמשות בTLS גרסת 0x0303 חוץ מפקטה אחת של Firefox שמשתמשת בגרסת 0x0301.

.4

אפשרות 1: ההאקר יודע את גודל הפקטות, זמן הפקטות וFlow ID:

במקרה כזה, ההאקר יוכל לנחש את האפליקציות שבהן משתמש הנתקף בכך שהוא יבודד את הזרמים ויסתכל על התנועה של הפקטות ועל הנתונים שלהן לאורך זמן. נניח, אם יש הרבה מאוד פקטות בהרבה גדלים שונים בזמן יחסית קצר וחלקן מאוד גדלות וחלקן מאוד קטנות, אפשר להניח שמדובר בדפדפן כמו Firefox. באמצעות המידע הזה, ההאקר יכול לא רק לדעת לאילו אתרים ו/או אפליקציות המשתמש נכנס אלה גם מתי ולכמה זמן, וכך ללמוד את הרגליו.

הדרכים למגר את ההתקפה הזאת הן:

-שימוש בVPN על מנת להסוות את הIP של המחשב.

-שימוש בדפדפנים שמתמקדים בפרטיות כמו TOR (The Onion Router) כדי להסוות את התנועה באינטרנט.

-Traffic Padding: הוספת פקטות נוספות שלא קשורות לתנועה על מנת להסוות את התנועה כדי שלא תיראה כאילו היא שייכת לאתר מסויים.

אפשרות 2: ההאקר יודע רק את הגודל ואת הזמן של הפקטה.

במקרה זה ההאקר יכול לנחש את האפליקציות שבהן המשתמש משתמש בכך שהוא מסתכל על פרצי התנועה, שכן לאפליקציות מסוימות יש פרצי תנועה שמאפיינים אותן. (לדוגמא, בגרף של YouTube בחלק 3 ניתן לראות שהתנועה שמאפיינת אותו היא קוצנית, חלקה לרוב עם פרצים. לעומת דפדפן שהתנועה שלו היא המון פקטות מגדלים שונים וסוגים שונים לפרקי זמן קטנים.) כמובן שזה יהיה יותר קשה לפענח בדיוק באילו אפליקציות/אתרים המותקף משתמש בלי הtuple שמכיל את כתובות הIP והפורטים.

הדרכים למגר את ההתקפה הזאת הן:

-שימוש בTOR כדי להסוות את התנועה באינטרנט.

-Traffic Padding: הוספת פקטות נוספות שלא קשורות לתנועה על מנת להסוות את התנועה כדי שלא תיראה כאילו היא שייכת לאתר מסויים.

ביביליוגרפיה:

https://matplotlib.org/2.0.2/Matplotlib.pdf

https://medium.com/a-bit-off/scapy-ways-of-reading-pcaps-1367a05e98a8