**דו"ח PLT**

1. **גרסה א' - iframe**
   1. ניסוי
      1. יצירת קובץ HTML המכיל iframe ותיבת קלט לקבלת URL
      2. יצירת סקריפט JS המודד זמן עד לפונקציית onLoad
      3. חיבור הסקריפט ל-iframe
   2. אוטומציה - סקריפט פייטון
      1. לוקח URL-ים מרשימה
      2. מכניס אותם (אחד אחד) לתוך תיבת הקלט,
      3. מחכה להופעת השינוי שיוצר סקריפט ה-JS.
      4. רושם את כל התוצאות
   3. בעיות
      1. לא מדויק
      2. פרסומות ותוספים יוצרים זמן מת
      3. לא ניתן לביצוע על אתרים מסוימים בעלי משמעות
2. **גרסה ב' - Browser Logs**
   1. ניסוי
      1. תחילת הקלטת תעבורה
      2. קריאה ל-URL
      3. סגירת הקלטת תעבורה
      4. ייצוא לוגים של הבראוזר הקשורים לקריאה זו
      5. מדידת זמנים, גדלים וכל השאר מהלוגים
      6. השוואה עם PCAP
   2. אוטומציה
      1. סקריפט הרצה
         1. לוקח URL-ים מהרשימה
         2. מפעיל TCPDUMP
         3. קורא ל-URL (בבקשת GET)
         4. מחכה לסיום טעינת העמוד
         5. מכבה TCPDUMP
         6. מייצא קובץ לוג מהבראוזר
         7. שולח את קובץ הלוג לפירסור
         8. שולח להשוואה עם PCAP
         9. רושם תוצאות
      2. פארסר
         1. עובר על כל רשומה בלוג
         2. מחלק לסשנים עפ"י רשומות של "הכרזה על כוונה לפתיחת קשר"\*
         3. אוסף דאטה לכל סשן
         4. מוציא פיצ'רים הקשורים לבראוזר
      3. השוואה עם PCAP
         1. בדיקה על פתיחת כל סשן
         2. בדיקת גודל דאטה
         3. בדיקת זמני הגעה
         4. בדיקת סגירת חיבורים
   3. בעיות
      1. לוגים לא מסודרים
      2. רשומות חסרות
      3. רשומות עם מידע חסר
      4. דאטה סט חלקי עם פיצ'רים חסרים
      5. דורש הרבה זיכרון
3. **גרסה 3 - Browser Timing Object**
   1. ניסוי
      1. תחילת הקלטת תעבורה
      2. קריאה ל-URL
      3. סגירת הקלטת תעבורה
      4. ייצוא ערך אובייקט TIMING של ריצה זו
      5. אנליזה מבוססת אגרגציה של קובץ ה-PCAP
      6. השוואה עם אובייקט ה-TIMING
   2. אוטומציה
      1. סקריפט הרצה
         1. לוקח URL-ים מהרשימה
         2. מפעיל TCPDUMP
         3. קורא ל-URL (בבקשת GET)
         4. מחכה לסיום טעינת העמוד
         5. מייצא אובייקט TIMING מהבראוזר
         6. מכבה TCPDUMP
         7. שולח את ה-PCAP לאנליזה (על בסיס אגריגציה)
         8. רושם תוצאות
      2. אגרגציה
         1. מעבר על קובץ ה-PCAP
         2. חיפוש פאקטות DNS או TCP
         3. חילוק לסשנים עפ"י
            1. בקשת DNS - פתיחת חלון
            2. תשובת DNS
            3. פתיחת קשר TCP
            4. מעבר דאטה
            5. סגירת קשר:

סיום מפורש (פאקטת FIN)

סיום משתמע (הרבה RESET-ים ברצף)

סיום מאולץ (סיום חלון "Idling session")

* 1. בעיות
     1. שגיאות בראוזר אוטומטי משפיעות על תוצאות
     2. אחוזים גבוהים של איבוד פאקטות גורמים ל-TO אשר משפיעים על הבראוזר
     3. דורש הרבה זמן ריצה וזיכרון

1. **דאטה-סט**
   * כל ניסוי מתחיל בבחירת זוג סדור של <איבוד פאקטות , עיכוב פאקטות> (loss-delay) והשמתם על הרשת בעזרת הכלי TC
   * מכיל רשומות של ניסויים מעל זוגות loss-delay מ-<0%, 0ms> ועד <10%, 100ms>
   * מורכב מפיצ'רים סטטיסטיים מחושבים על ה-PCAP ופיצ'רים מיוצאים מאובייקט ה-TIMING.
   * כל רשומה מחולקת ל-4 עפ"י גודל חלון זמן לחישוב לייבל ללמידה (0.1, 0.2, 0.5, 1)
     + לייבל הכי טוב נמצא להיות עפ"י חלוקה לשניות מלאות (פחות לייבלים וכו')
   * **ישנו פיצ'ר חסר - אינומרציה של URL עפ"י סוג התוכן**
2. **מכונות לומדות**
   1. השוואת dt
      1. ניסויים עם dt קטן מ-1 הניבו תוצאות מחפירות (דיוק קלסיפיקציה מירבי שהושג - 16.4%)
      2. ניסויים ב-dt שווה ל-1 הניבו תוצאות טובות יותר
   2. השוואת מודלים
      1. מודל "כמות חלונות כלייבל":

* מודל Multi-Label
* כניסה - וקטור של פיצ'רים
* יציאה - ניחוש של כמות חלונות בגודל dt הנדרש להשלמת הסשן
* המודל שהניב את **התוצאות הטובות ביותר היה RF** (על החלק בדאטה-סט בו dt הוא 1 שניות). המודל הצליח להגיע ל**דיוק של 46.7%**.
  + 1. מודל "חלון? כן-לא כלייבל":
* מודל בינארי
* כניסה - וקטור של פיצ'רים + כמות חלונות אפשרית
* יציאה - ניחוש האם כמות החלונות מספיקה להשלמת הסשן, או לא.
* כרגע המודל שוגה בריצתו בנקודה שאינה ידועה ("אומר" שסיים כרגיל אך מחזיר 0 תוצאות)
* צפי תוצאות (בהשוואה למודל הקודם) לא ידוע
  + עדכון: כשעתיים לאחר הרישום הראשוני של מסמך זה המכונה תוקנה וכעת מחזירה אחוזי דיוק של 93.91% שהתקבלו ע"י מודל ה-Light Gradient Boosting

\* הלוגים של הבראוזר אינם מכילים בקשות, אלא רשומות על כוונה לבקש בקשות