פרויקט גמר - רשתות

ירין לוי - 213907421
מאיה חייט - 322515669

קישור לgithub :
https://github.com/MayaHayat/NetworkingFinalProject.git

קישור ל Linkedin:

Maya:

Yarin:

תוכן עניינים

חלק יבש **2**

חלק רטוב **4**

מבוא 4

מהלך העבודה 4

הקלטה #1: הודעות רגילות 7

הקלטה #2 : תמונות 8

הקלטה #3 : הודעות קוליות 9

הקלטה #4 : משלבים הכל ביחד 10

הקלטה #5 : הוספת רעש להקלטות 11

הקלטה #6 : הוספת יותר רעש להקלטות ((Discord, Spotify 12

סיכום 13

חלק יבש

כיום בעולם, ישנם יותר מ2 מיליארד משתמשים באפליקציות לשליחת הודעות (IM). כתוצאה מכך ממשלות ותאגידים גדולים מעוניינים לנטר את המשתמשים הללו ברשת על מנת להשפיע על תוכן שהאזרחים צורכים ואף לנטר איומים כלפי הממשלה. המאמר "Practical Traffic Analysis Attacks on Secure Messaging Applications" מדבר על ניתוח תנועת שליחת הודעות באינטרנט ומדגיש את הקלות שבה ניתן לזהות את המשתתפים והמנהלים בקבוצות מסוימות, על אף ההצפנה הנוקשה של אפליקציות אלה (לדוגמא: end to end encryption, end to middle encryption) הכותב אף מציין כי 15 דקות של שיחה בTelegram מספיקות על מנת לזהות את המשתמשים והמנהלים בדיוק של 94% עם שגיאה נמוכה ביותר.

התוקף יכול לנטר את המידע המבוקש על תנועות השיחה ע"י ניתוח המידע, כלומר הוא יאסוף מידע על תזמון שליחת הודעות, אורך ההודעות ומספר ההודעות הנשלחות בחלון זמן מסוים על מנת לזהות את המנהל ומשתתפי הקבוצה (את כתובות הIP שלהם) ובכך יכולה הממשלה או כל גוף אחר המעוניין במידור השיחות לשלוט במידע המועבר בקבוצות.
ישנן 3 דרכים למצוא את האינפורמציה הרצויה של ערוץ:
הראשונה כי כמובן אם השיחה (ערוץ) לא פרטית (public group chat) התוקף יכול להצטרף בתור משתתף ולהקליט את הפרטים הנדרשים (זמן וגודל ההודעות כפי שציינו לעיל).
השניה היא שהתוקף הצטרף לקבוצה ויכול לשלוח הודעות (או שהקבוצה מאפשרת לכולם לפרסם הודעות או שהוא קיבל אישור מיוחד - גישת מנהל), מה שעוזר לו לשלוח כעת הודעות בעצמו ולנטר את הפרטים שהוא מעוניין בהם על ידי שליחה ההודעות בקצב הנדרש ובאורך מסוים שיכול לעזור לו.
האופציה השלישית והאחרונה היא שהתוקף לא מצליח להיכנס לקבוצה (channel) אבל הוא כן הצליח לגלות את הכתובת IP של אחד המשתתפים או המנהלים בקבוצה, התוקף מאזין לתנועה המוצפנת של המשתתף שמצא ומקליט את הפרטים.

התוקף מבצע האזנת סתר (Wiretapping) באמצעות ניטור התעבורה של המידע המוצפן על מנת לזהות את המשתמשים והמנהלים בקבוצה, הדבר מתבצע באמצעות האזנת סתר של תעבורת הרשת של ספקי האינטרנט או הIXP שבהם הוא שולט. למשל, באמצעות חומת האש הגדולה של סין (Great Firewall Of China) לחלופין, היריב יכול להאזין לאנשים ספציפיים (למשל, פעילים חשודים).

שירותי IM מאפשרים למשתמשים שלהם לשלוח סוגים שונים של הודעות, לרוב טקסט, תמונה, וידאו, קובץ אודיו והודעות. הודעות IM מועברות בין משתמשים דרך אחת מצורות התקשורת העיקריות הבאות:

- תקשורת אחד על אחד בין משתמשי IM מכונה "הודעות ישירות". בשירותי IM פופולריים מרכזיים, הודעות ישירות אלו מועברות דרך השרתים של ספקי ה-IM. אלא אם כן ישנה הצפנה מקצה לקצה, לשרתים יש גישה לתוכן התקשורת הללו.
- לאחר מכן, יש לנו "תקשורת קבוצתית פרטית (סגורה)" שבה משתמשים מרובים עוסקים בתקשורת. בתוך קבוצות אלה, כל החברים יכולים לפרסם הודעות ולקרוא את מה שאחרים

פרסמו. לכל קבוצה יש מנהל שיצר אותה ובעל יכולת לנהל משתמשים והודעות. כדי להצטרף לקבוצה סגורה, משתמשים צריכים לקבל הזמנה.

- לבסוף, יש לנו "תקשורת קבוצתית ציבורית (פתוחה)" הידועה גם בשם "ערוצים". צורת תקשורת זו פועלת כשידור, שבו מנהל מערכת אחד או יותר יכולים לפרסם הודעות, וחברים יכולים לקרוא רק פוסטים אלה. הצטרפות לערוצים אינה מצריכה הזמנה; משתמשים יכולים להצטרף באופן חופשי.

מהטבלה ניתן להסיק כי המרכיב המכריע בסוג ההודעות הינו תמונות, המרכיבות 48% מכלל ההודעות, וגודלן הממוצע הוא כ-306.61 בתים. לאחר מכן הודעות טקסט, המרכיבות כ-29.4% מכלל ההודעות וגודלן הממוצע הוא כ-91.33 אלף בתים. ולאחר מכן קבצי וידאו (15.4%), אודיו (5.1%) וקבצים (2.1%).

כותב המאמר מציין כי פיתחו שני אלגוריתמים לניתוח התקפות SIM, כפי שציינו מקודם מטרת ההתקפות היא לזהות את מנהלי הקבוצות והמשתתפים, מה שמאפשר לנו את ההתקפה היא העובדה שאין טשטוש של מאפייני הקבוצה (כלומר אנחנו יודעים את תזמון ההודעות, גודלן והתדירות בה הן נשלחות. האלגוריתם הראשון הוא "event based correlator" שמשתמש בנתונים סטטיסטים של הקבוצות שמתאים משתמשים לערוצים. האלגוריתם השני הוא "shaped based algorithm" שכמו השם שלו מתאים את הצורות של התנועה על מנת להתאים את המשתמשים לערוצים/ קבוצות.

תפקיד התוקף הוא להחליט בין שני מקרים, האם משתמש U נמצא בערוץ C, כלומר הוא פעיל ושולח הודעות בתור משתתף או בתור מנהל או שהמקרה השני שהמשתמש לא קשור לערוץ C. כיוון שהערוץ מוצפן התוקף משתמש הגלאי (event based detector) מעוניין להתאים בין אירועי SIM של משתמש U לאירועים של ערוץ C, כאשר אירוע E יכול להיות הודעת SIM אחת או מספר הודעות שנשלחו בחלון זמן מוגבל. אנחנו מסתכלים על אירוע כאחד מהחמישה: תמונה, סרטון, קובץ, הודעת טקסט או הודעה קולית. כאשר אנחנו מסתכלים על אירוע מסוים אנחנו מקליטים את הגודל S שלו ואת הזמן T בו הוא הופיע. תמונה 8 ניתן לראות קווים קטנים ודקים שנוצרו על ידי הקלטת התראות וכו', כיוון שהם חסרי משמעות הגלאי מתעלם מהם.
תמונה 8 במאמר מציגה את חלוץ האירועים שהתקבלו על ידי שני משתמשים U1 העליון וU2 התחתון. אנחנו רואים שאין התאמה בין תזמון שליחת הסרטון לתנועה שמתבצעת אצל U1 (אין תנועה בזמן שליחת הסרטון) בעוד שיש התאמה מדויקת בין תזמון ההודעות לתנועה של U2 לכן אנחנו יכולים להסיק שU2 אכן קשור לקבוצה C. משהו שחשוב לשים לב אליו זה אורך כל הודעה, כמובן שסרטון יתפוס יותר זמן מאשר תמונה או קובץ טקסט, מידע זה יכול לעזור לנו למנוע טעויות.

חלק רטוב

מבוא

בחלק הרטוב התבקשנו להקליט תקשורת ב- 4 קבוצות IM שונות, בפרויקט שלנו השתמשנו WhatsApp Web וWireshark על מנת להקליט את השיחות כלומר את הפקטות שנשלחו והתקבלו. השתמשנו בפייתון בשביל ליצור גרפים של גודל פקטות כציר ה Y וזמן ההקלטה כציר ה X ובכך נוכל בהמשך להראות בדיוק איזה סוג הודעה התקבלה ומי שלח איזה הודעה ובעזרת הקלטות וניתוח זה יוכל ה"תוקף" לקבל את המידע הנחוץ כלומר בהינתן העובדה שהתוקף כבר נמצא בקבוצה (שהקבוצה הייתה פתוחה ולכן יכול היה להצטרף או שמישהו שכבר היה חבר בקבוצה צירף אותו) יוכל התוקף לזהות מתי נשלחו הודעות ומי נמצא בקבוצה הנוכחית.

חילקנו את העבודה ל 5 הקלטות שונות בכל פעם הקלטנו קבוצה שונה (קבוצת הודעות טקסט, תמונות, הודעות קוליות ושיחה אחת של כולם ביחד וניסינו לזהות מה נשלח בכל הודעה) להקלטה החמישית הוספנו רעש שיצרנו..

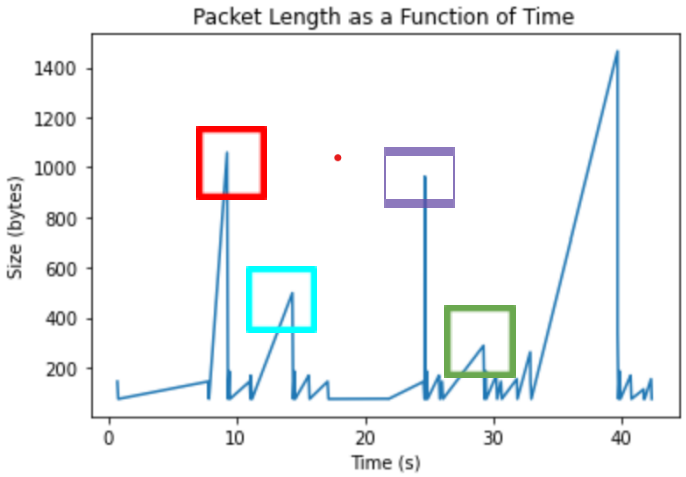
מהלך העבודה

תחילה נשים לב כי :

```
C:\Users\Maya>nslookup web.whatsapp.com
Server:  UnKnown
Address:  2a0d:6fc2:5560:c200::1

Non-authoritative answer:
Name:    mmx-ds.cdn.whatsapp.net
Addresses:  2a03:2880:f276:cd:face:b00c:0:167
          157.240.251.60
Aliases:  web.whatsapp.com
```

כלומר או שהsource או הdestination של הפקטות אמורות להיות הכתובת שמסומנת מעל, כפי שניתן לראות בהקלטה הבאה זה אכן המצב:



ניתן לראות בבירור את הקפיצה בגובה קצת מעל 1,000 סביב ה 10 שניות ומיד אחרי זה קפיצה של 500 כ 5 שניות לאחר מכן ועוד קפיצה בגובה קצת פחות מהקפיצה הראשונה סביבות ה 25 שניות. בתמונה מעל סימנו את הקפיצות והפאקטות המתאימות בכל פעם בצבע שונה.

No.	Time	Source	Destination	Protocol	Length	Info
146	7.877545	2a03:2880:f276:cd:f…	2a0d:6fc2:5560:c200…	TCP	74	443 → 52750 [ACK] Seq=1 Ack=141 Win=1141 Len=0
207	9.292341	2a0d:6fc2:5560:c200…	2a03:2880:f276:cd:f…	TLSv1.2	1059	Application Data
208	9.361159	2a03:2880:f276:cd:f…	2a0d:6fc2:5560:c200…	TCP	74	443 → 52750 [ACK] Seq=1 Ack=1126 Win=1152 Len=0
210	9.516298	2a0d:6fc2:5560:c200…	2a03:2880:f276:cd:f…	TCP	74	52750 → 443 [ACK] Seq=1126 Ack=112 Win=514 Len=0
212	11.046223	2a0d:6fc2:5560:c200…	2a03:2880:f276:cd:f…	TLSv1.2	144	Application Data
213	11.114447	2a03:2880:f276:cd:f…	2a0d:6fc2:5560:c200…	TCP	74	443 → 52750 [ACK] Seq=112 Ack=1196 Win=1152 Len=0
214	11.123377	2a03:2880:f276:cd:f…	2a0d:6fc2:5560:c200…	TLSv1.2	168	Application Data
215	11.126300	2a0d:6fc2:5560:c200…	2a03:2880:f276:cd:f…	TLSv1.2	167	Application Data
221	14.380524	2a0d:6fc2:5560:c200…	2a03:2880:f276:cd:f…	TLSv1.2	499	Application Data
225	14.551144	2a03:2880:f276:cd:f…	2a0d:6fc2:5560:c200…	TLSv1.2	185	Application Data
226	14.599955	2a0d:6fc2:5560:c200…	2a03:2880:f276:cd:f…	TCP	74	52750 → 443 [ACK] Seq=1714 Ack=317 Win=513 Len=0
241	15.654545	2a03:2880:f276:cd:f…	2a0d:6fc2:5560:c200…	TLSv1.2	168	Application Data
242	15.657258	2a0d:6fc2:5560:c200…	2a03:2880:f276:cd:f…	TLSv1.2	167	Application Data
243	15.726272	2a03:2880:f276:cd:f…	2a0d:6fc2:5560:c200…	TCP	74	443 → 52750 [ACK] Seq=411 Ack=1807 Win=1163 Len=0
252	17.129797	2a0d:6fc2:5560:c200…	2a03:2880:f276:cd:f…	TLSv1.2	144	Application Data
253	17.198215	2a03:2880:f276:cd:f…	2a0d:6fc2:5560:c200…	TCP	74	443 → 52750 [ACK] Seq=411 Ack=1877 Win=1163 Len=0
271	21.728191	2a0d:6fc2:5560:c200…	2a03:2880:f276:cd:f…	TCP	75	55177 → 443 [ACK] Seq=1 Ack=1 Win=511 Len=1 [TCP segment of a reassembled PDU]
272	21.811918	2a03:2880:f276:cd:f…	2a0d:6fc2:5560:c200…	TCP	74	443 → 55177 [ACK] Seq=1 Ack=2 Win=278 Len=0
278	24.551597	2a0d:6fc2:5560:c200…	2a03:2880:f276:cd:f…	TLSv1.2	144	Application Data
279	24.619524	2a03:2880:f276:cd:f…	2a0d:6fc2:5560:c200…	TCP	74	443 → 52750 [ACK] Seq=411 Ack=1947 Win=1163 Len=0
280	24.633957	2a0d:6fc2:5560:c200…	2a03:2880:f276:cd:f…	TLSv1.2	963	Application Data

בשביל לבדוק כי ההקלטות שלנו אמינות תזמנו את ההקלטות וכתבנו בהודעות את הזמנים שבהם ההודעות נשלחו כמו הדוגמא הבאה (כמובן שהצילומי מסך אינם ההודעה המלאה כפי שיכולנו לראות מהגרף וצילומי המסך של הפקטות כל הודעה הייתה באורך שונה.

10 mport pandas as pd
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.pyplot as plt

15 import pandas as pd
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.pyplot as plt

25 import pandas as pd
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.pyplot as plt
import pandas as pd

כפי שניתן לראות בצילומי מסך ההודעות שנשלחו הופיעו בדיוק באותו הזמן בגרף כפי שנשלחו לכן נוכל לסמוך על ההקלטות שלנו מכאן הלאה.
כמובן שיכול להראות מוזר שבחרנו הודעות ארוכות בשביל ליצור קפיצות גבוהות בגרף לכן שלחנו גם את ההודעה הבאה
וכפי שניתן לראות יש קפיצה גם סביב ה 30 שניות (מסומן בירוק).

30 הייייי
14:19

לאחר שפתחנו 4 קבוצות וואצטאפ שונות, החלטנו לפני כל הקלטה על סוגי ההודעות שאנחנו מעוניינים לשלוח ותזמונים להודעות. הקלטנו בכל פעם במשך 90 שניות וסיננו לפי התמונה הבאה:

(tcp.port == 443 || tcp.port == 5222) &&(ipv6.src == 2a03:2880:f276:cd:face:b00c:0:167 || ipv6.dst == 2a03:2880:f276:cd:face:b00c:0:167)

No.	Time	Source	Destination	Protocol	Length	Info

אחרי שסיימנו להקליט העברנו את ההקלטה לקובץ csv מהצורה הבאה

	A	B	C	D	E	F	G
1	No.	Time	Source	Destination	Protocol	Length	Info
2	2	0.737911	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	2a03:2880:f276:cd:face:b00c:0:167	TLSv1.2	144	Application Data
3	3	0.807164	2a03:2880:f276:cd:face:b00c:0:167	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	TCP	74	443 > 52750 [ACK] Seq=1 Ack=71 Win=1141 Len=0
4	145	7.809421	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	2a03:2880:f276:cd:face:b00c:0:167	TLSv1.2	144	Application Data
5	146	7.877545	2a03:2880:f276:cd:face:b00c:0:167	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	TCP	74	443 > 52750 [ACK] Seq=1 Ack=141 Win=1141 Len=0
6	207	9.292341	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	2a03:2880:f276:cd:face:b00c:0:167	TLSv1.2	1059	Application Data
7	208	9.361159	2a03:2880:f276:cd:face:b00c:0:167	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	TCP	74	443 > 52750 [ACK] Seq=1 Ack=1126 Win=1152 Len=0
8	209	9.463018	2a03:2880:f276:cd:face:b00c:0:167	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	TLSv1.2	185	Application Data
9	210	9.516298	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	2a03:2880:f276:cd:face:b00c:0:167	TCP	74	52750 > 443 [ACK] Seq=1126 Ack=112 Win=514 Len=0
10	212	11.046223	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	2a03:2880:f276:cd:face:b00c:0:167	TLSv1.2	144	Application Data
11	213	11.114447	2a03:2880:f276:cd:face:b00c:0:167	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	TCP	74	443 > 52750 [ACK] Seq=112 Ack=1196 Win=1152 Len=0
12	214	11.123377	2a03:2880:f276:cd:face:b00c:0:167	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	TLSv1.2	168	Application Data
13	215	11.1263	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	2a03:2880:f276:cd:face:b00c:0:167	TLSv1.2	167	Application Data
14	216	11.237726	2a03:2880:f276:cd:face:b00c:0:167	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	TCP	74	443 > 52750 [ACK] Seq=206 Ack=1289 Win=1152 Len=0
15	221	14.380524	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	2a03:2880:f276:cd:face:b00c:0:167	TLSv1.2	499	Application Data
16	222	14.44844	2a03:2880:f276:cd:face:b00c:0:167	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	TCP	74	443 > 52750 [ACK] Seq=206 Ack=1714 Win=1163 Len=0
17	225	14.551144	2a03:2880:f276:cd:face:b00c:0:167	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	TLSv1.2	185	Application Data
18	226	14.599955	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	2a03:2880:f276:cd:face:b00c:0:167	TCP	74	52750 > 443 [ACK] Seq=1714 Ack=317 Win=513 Len=0
19	241	15.654545	2a03:2880:f276:cd:face:b00c:0:167	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	TLSv1.2	168	Application Data
20	242	15.657258	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	2a03:2880:f276:cd:face:b00c:0:167	TLSv1.2	167	Application Data
21	243	15.726272	2a03:2880:f276:cd:face:b00c:0:167	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	TCP	74	443 > 52750 [ACK] Seq=411 Ack=1807 Win=1163 Len=0
22	252	17.129797	2a0d:6fc2:5560:c200:bdce:1f2e:bc44:2269	2a03:2880:f276:cd:face:b00c:0:167	TLSv1.2	144	Application Data

```
data.describe()
```

	No.	Time	Length
count	64.000000	64.000000	64.000000
mean	274.984375	25.257952	177.281250
std	74.456655	11.863967	237.391477
min	2.000000	0.737911	74.000000
25%	225.750000	14.587752	74.000000
50%	286.500000	25.776032	128.500000
75%	318.250000	32.892591	168.000000
max	377.000000	42.326907	1466.000000

מהטבלה אנחנו יכולים ללמוד שכמו שאמרנו ההקלטה ארכה 42.3 שניות (המקסימום בזמן) האורך הממוצע של הפקטות הן 177 והחציון הוא 128 (כמובן שהSTD גבוה כיוון שישנן הודעות קצרות וארוכות בהקלטה.

ולאחר מכן השתמשנו בקוד שכתבנו מראש בשביל ליצור את הגרפים שראינו מקודם על מנת שנוכל לנתח את ההודעות היוצאות והנכנסות בקבוצות שהתוקף נמצא בהן.

נשים לב כי בהקלטה יש מספר רב של פקטות באורך 74 ביטים (ולפי הטבלה שמעל 25% מהפקטות) שהן מסוג ACK כפי שלמדנו בקורס פקטות מסוג זה אחראיות לשלוח הודעה מהמקבל לשולח כי הוא אכן קיבל את ההודעה לכן נניח בהקלטות אלה שמדובר ברעש. כמובן שפקטות בגודל זה קטנות מדי בשביל להיות הודעה)כיוון שהודעות קטנות (למשל ההודעה ששלחנו בזמן 30) הן סביבות 150 ביטים לפחות.

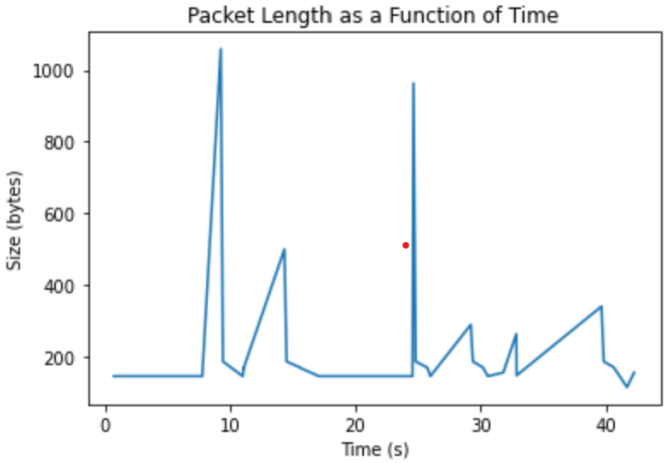
```
filtered_df = data[data['Info'] == 'Application Data']
```

ננסה להוריד את כל השורות המכילות את המילה ACK בעמודת ה Info.

```
filtered_df.describe()
```

	No.	Time	Length
count	32.00000	32.000000	32.000000
mean	269.84375	24.418040	237.031250
std	72.57037	11.637967	215.888036
min	2.00000	0.737911	113.000000
25%	224.00000	14.508489	145.500000
50%	285.50000	25.739643	167.500000
75%	313.50000	32.069422	185.000000
max	376.00000	42.272604	1059.000000

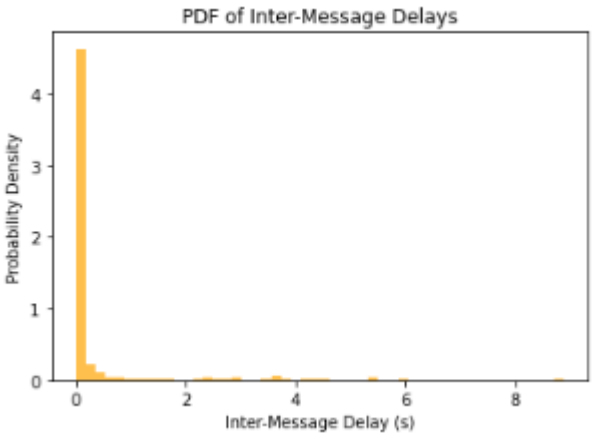
נשים לב שהממוצע וגם החציון עלה והרבעון התחתון יושב כרגע על 145 מספר הרבה יותר הגיוני מאשר 74. נראה איך הגרף נראה עכשיו.

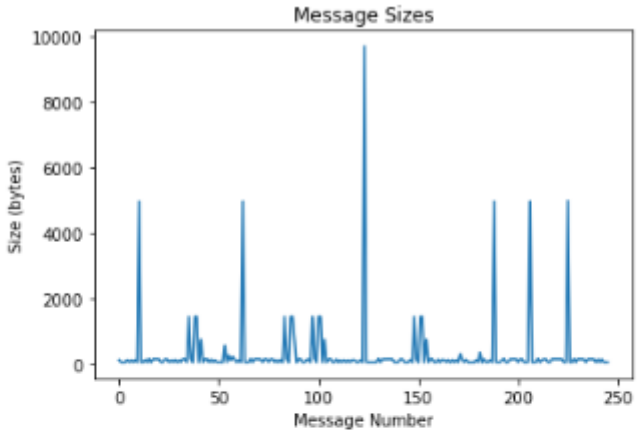


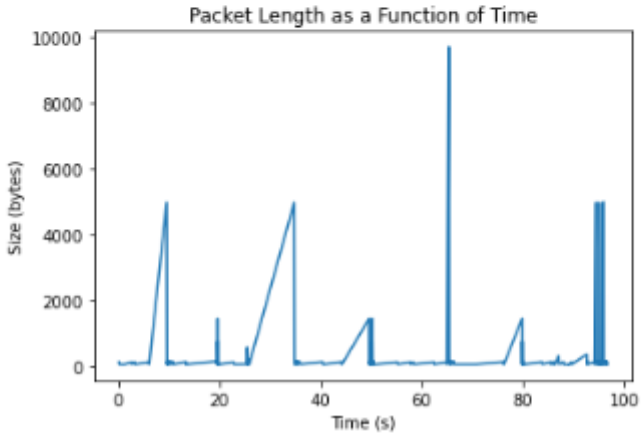
נשים לב שהגרף הרבה יותר נקי עכשיו, הורדנו את כל הקפיצות הקטנות כלומר את הרעש וכעת ניתן לראות בדיות מתי נשלחו איזה הודעות.
במקרה הזה אמנם ההודעות ששלחנו ארוכות יחסית ולכן מספר הביטים גבוה אבל אם אנחנו לא בודקים בהכרח עם הודעות ארוכות כנראה שיהיה חשוב לפלטר את הרעשים.

הקלטה #1: הודעות רגילות

בחלק זה שנינו שלחנו הודעות טקסט באותה קבוצה כאשר ירין שהוא "התוקף" הקליט דרך ה Wireshark את השיחה. ניתן לראות בבירור כי ההודעות ששלח ירין תופסות יותר מקום כלומר הקפיצות הגדולות יותר בגרף. ניתן לראות כי התוקף שלח הודעה ראשונה אחרי 10 שניות וקיבל תשובה מהצד השני כ 20 שניות אחרי שהתחילה ההקלטה, לאחר מכן הוא שלח הודעה שניה באותו הגודל וקיבל תשובה ארוכה יותר (ניתן לראות כי הקו עבה יותר). 65 שניות אחרי שההקלטה התחילה התוקף שלח הודעה ארוכה פי 2 מההודעות ששלח עד כה וקיבל תשובה מהצד השני באותו אורך כמו ההודעה הראשונה שקיבל ובסוף שלח התוקף כ3 הודעות באותו האורך ושם סיים את ההקלטה כי היה בטוח מי נמצא בצד השני של השיחה.



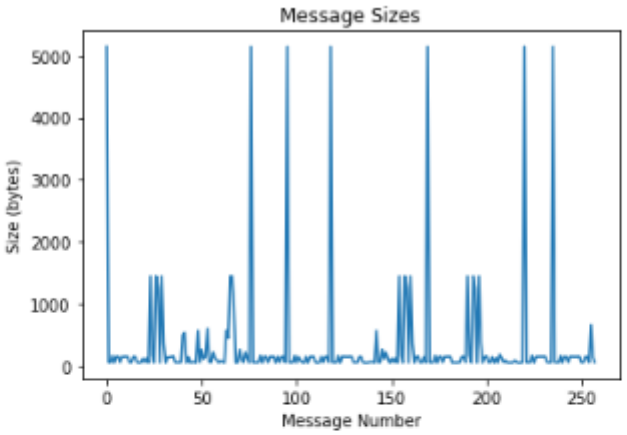


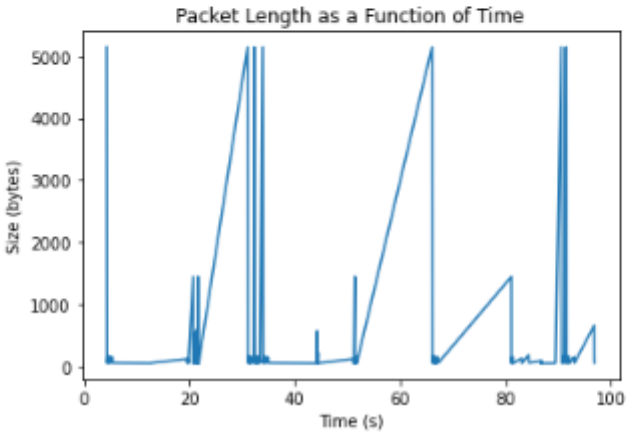


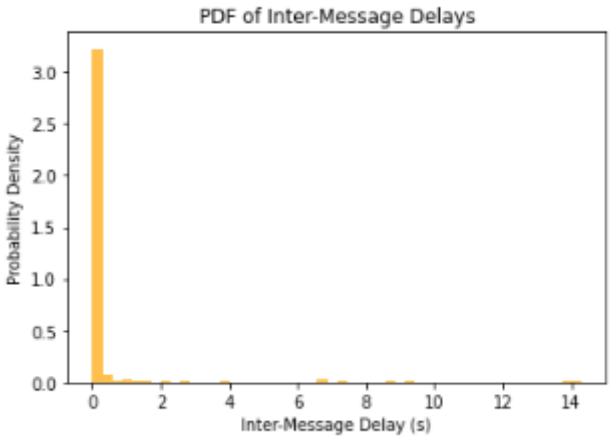
הקלטה #2 : תמונות

ההקלטה הזאת מראה הקלטה של קבוצה שנשלחו בה רק הודעות מסוג תמונה. התוקף בחר תמונה מסוג אחד לשלוח בעוד שהצד השני שלח תמונה שונה. התמונה הראשונה (הפיק הראשון) נשלח בזמן 0 וקיבל תמונה מהצד השני אחרי 20 שניות. לאחר מכן שלח התוקף 3 תמונות בזמן 35 על מנת לוודא כי הוא אכן מדבר עם מי שהוא חושב שהוא מדבר איתו ואכן קיבל מענה 50 שניות לאחר שההקלטה התחילה וענה התוקף 10 שניות אחרי וקיבל תמונה אחרונה מהצד השני ב 80 שניות. התוקף סיים את ההקלטה לאחר ששלח 2 תמונות ב 90 שניות.

נשים לב שהגרף הזה מאוד ברור, ניתן לראות במדויק את הפיקים בכל פעם שנשלחה תמונה, ניתן לראות כמה תמונות נשלחו כל פעם ואיזה צד שלח כל הודעה.

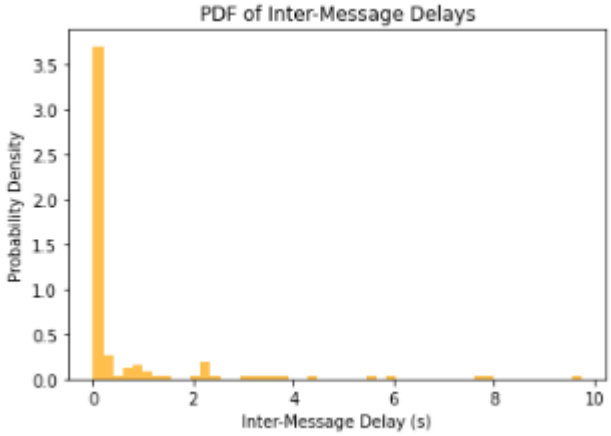


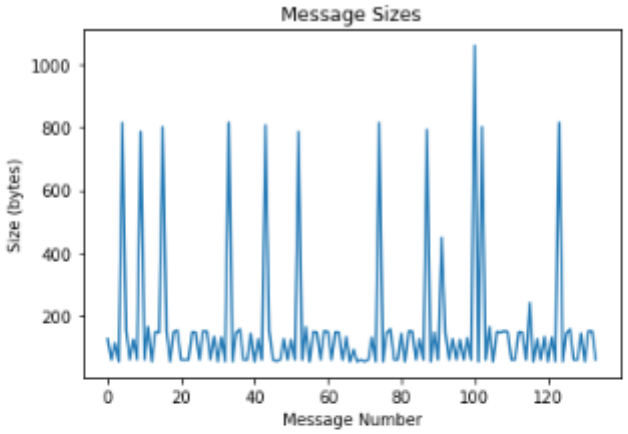


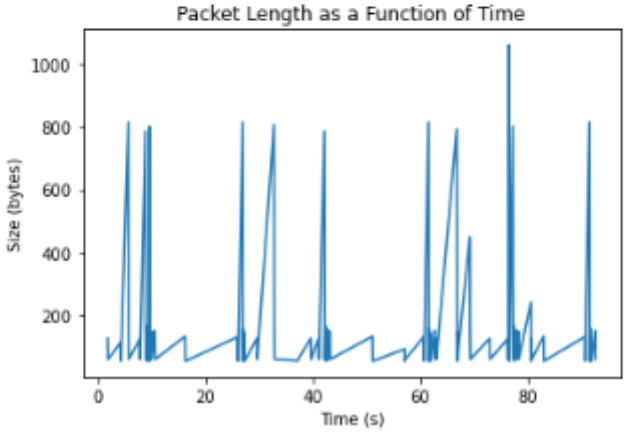


הקלטה #3 : הודעות קוליות

ההקלטה הזו מראה הקלטה של הודעות קוליות בלבד. התוקף שלח הקלטות קוליות כאשר הצד השני מגיב לו. ניתן לראות בהקלטות כדלקמן, התוקף שלח הקלטה קולית לאחר כ-11 שניות, לאחר מכן הוא קיבל הודעה קולית (תגובה) מהצד השני לאחר כ-25 שניות מתחילת ההקלטה. לאחר מכן התוקף שלח עוד הודעה קולית בשנייה ה-40, והתקבלה תגובה מהצד השני בשנייה 60. לבסוף התוקף, שלח הודעה קולית בשנייה 75, קיבל תשובה בערך בשנייה ה 90 וסגר את ההקלטה.

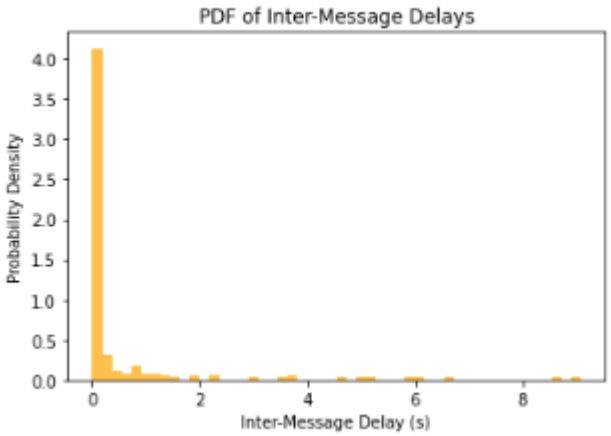


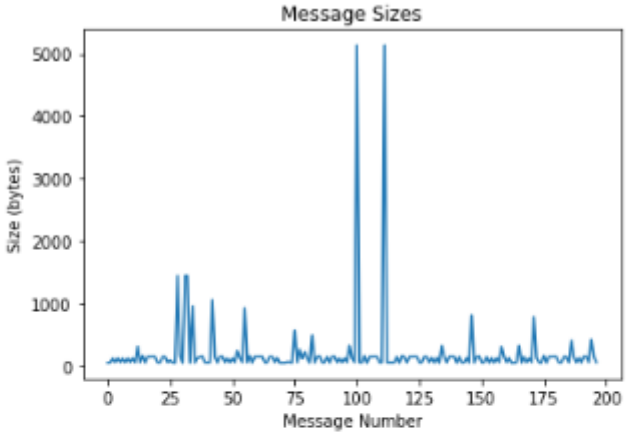


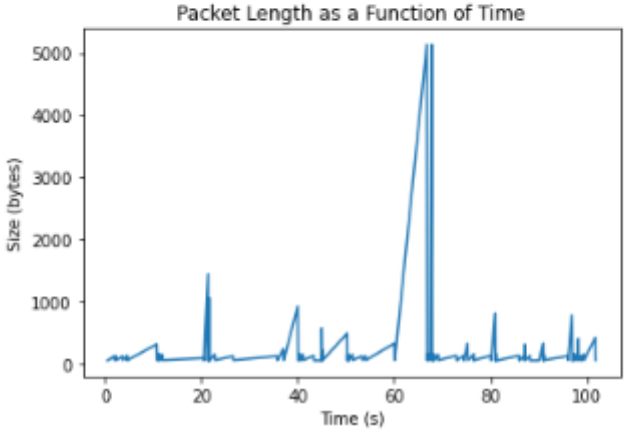


הקלטה #4 : משלבים הכל ביחד

התוקף מתחיל שיחה בקבוצה כ 10 שניות לאחר שהפעיל את ההקלטה ושולח הודעה קצרה, לאחר שמקבל תשובה (תמונה - כפי שניתן לראות לפי העובדה שהפיק גבוהה יותר ב 20 שניות) הוא עונה עם הודעת טקסט ארוכה יותר ב 40 שניות ומקבל תשובה (בטקסט) ב 50 שניות. לאחר מכן התוקף שולח 2 תמונות ברצף לקבוצה ומקבל הודעה קוליות כ 15 שניות לאחר מכן מהצד השני ושולח בחזרה הודעה קוליות באותו האורך ומסיים את השיחה ואת ההקלטה.

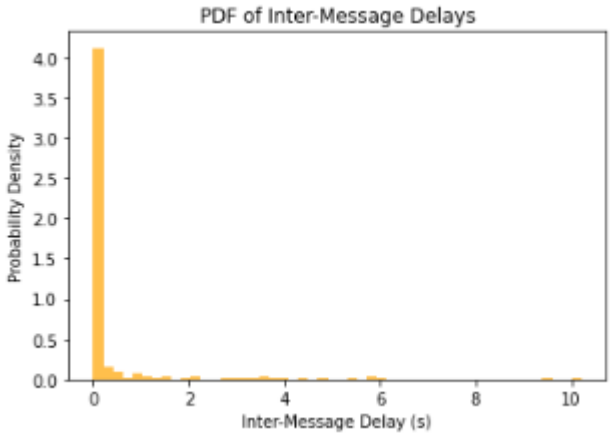


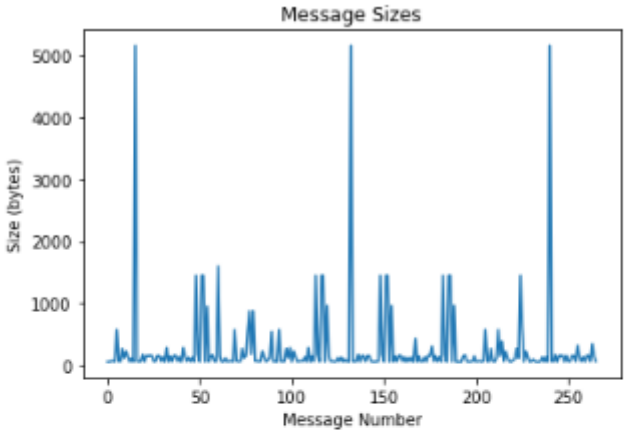


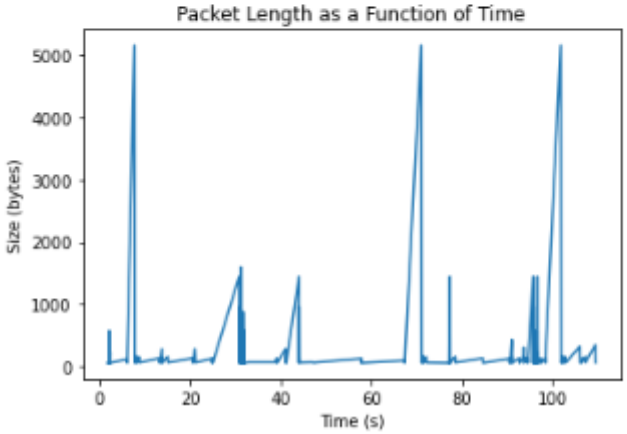


הקלטה #5 : הוספת רעש להקלטות

ההקלטה הבאה מציגה תקשורת בין שני אנשים כאשר יש רעש מקבוצות אחרות והתוקף נמצא במספר קבוצות שונות. התוקף שלח תמונה לאחר 10 שניות בקבוצה אחת מתחילת ההקלטה, והצד השני שלח תמונה חזרה באותה קבוצה בשנייה ה 30. לאחר מכן התוקף שלח תמונה בשנייה ה-70, והתקבלה תגובה בערך השנייה ה-90, לאחר מכן התוקף שלח תמונה בשנייה ה 100. נשים לב כי יש רעש מקבוצות אחרות בין פרקי הזמן הללו, לדוגמא בין השנייה ה 10 לשנייה ה30, נשלחו הודעות טקסט בכמה בקבוצות שונות, כמו כן ניתן לראות כי בין השנייה ה30 לבין השנייה ה70, נשלחו הודעות טקסט ותמונה ולכן ניתן לראות קפיצה בשנייה ה40. כמו כן בקבוצה נוספת אחרת נשלחה תמונה בשניה ה-75 מה שגרם לעוד קפיצה. לבסוף גם ניתן לראות שבשניה ה110 נשלחה בסוף הודעה לפני סגירת ההקלטה.



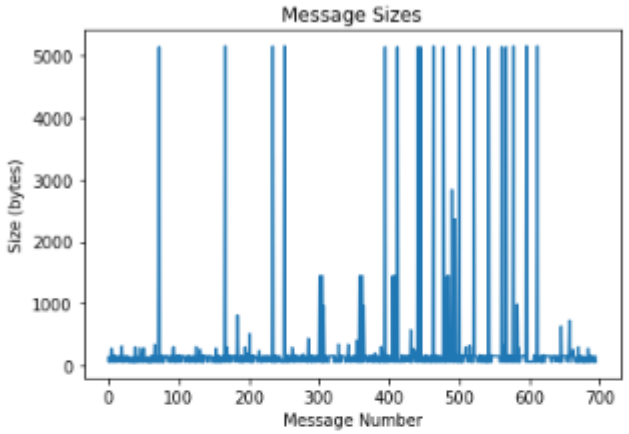


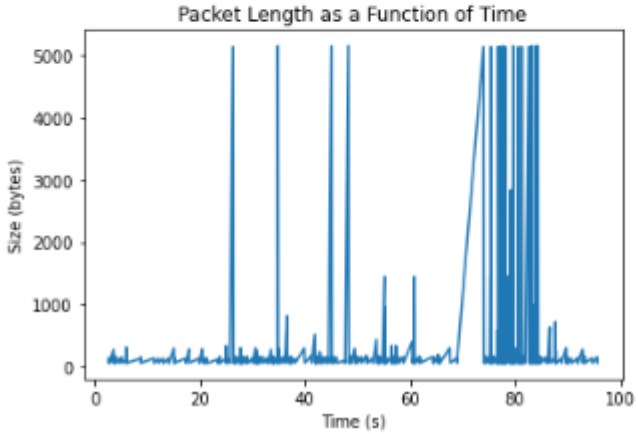


הקלטה #6 : הוספת יותר רעש להקלטות (Discord, Spotify)

בהקלטה הבאה דימינו שיחה של ממש בקבוצה אחת ושלחנו הודעות (גם התוקף) בקבוצות אחרות בזמן ההקלטה, עכשיו אנחנו ננסה להבין איזה הודעה שייכת לכל קבוצה:

קבוצה 1- נשלחה הודעה ולאחר מכן תמונה מהתוקף ב25, התקבלה הודעת טקסט ב 27 ונשלחו כמה הודעות טקסט בקבוצה, לאחר מכן התוקף שלח שתי תמונות סביבות 45 והתקבלה תמונה בזמן 55, התוקף שלח 12 תמונות בזמן 85 ולאחר מכן נשלחו עוד כמה הודעות ונגמרה השיחה.
קבוצה 2 - התקבלה תמונה ב75 ועוד הודעת טקסט שלא ניתן להבחין בה כמה שניות אחרי התמונה.
קבוצה 3 - הודעות בתחילת השיחה, לאחר מכן התוקף שלח תמונה ממש לפני ה 50 שניות וקיבל תמונה ב60 שניות מאז ההתחלה של ההקלטה.





סיכום

כפי שראינו בהקלטות השונות של הקבוצות, ברגע שהתוקף מצליח להיכנס לקבוצה כלשהי הוא יכול בקלות לגלות מי נוכח בקבוצה