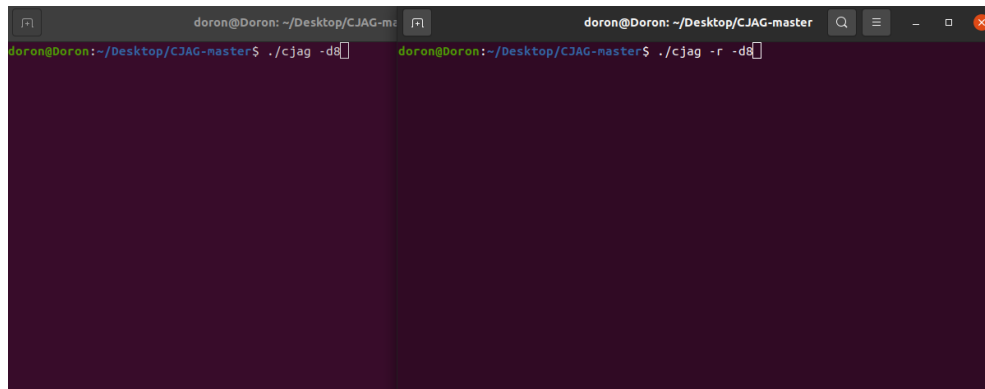
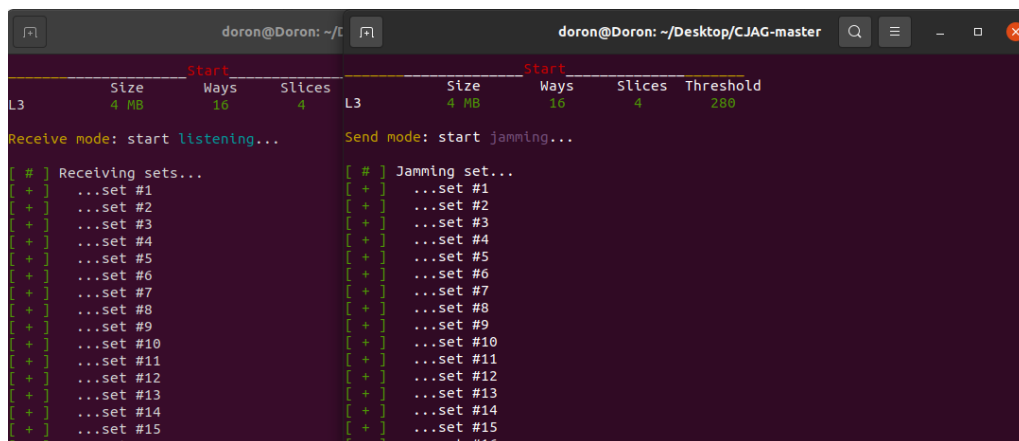


בכדי להריץ את תהליך התקיפה נעקוב אחר הפעולות הבאות:

- נוריד את התיקייה Cache_Covert_Channel (שבתמונות נקראת CJAG-master).
- נפתח את התיקייה ונקמפל את הקבצים על ידי פקודת `make`.
- נבחין כי קובץ הקלט שלנו נמצא גם כן באותה תיקייה אך ניתן לשנות זאת.
- נפתח 2 טרמינלים באותה תיקייה כאשר אחד עבור ה-`receiver` והשני עבור ה-`sender`.
- נריץ את הקוד על ידי הפקודה `./cjag`. כאשר עבור ה-`receiver` נוסיף את הביטוי `-r`, כלומר בחלון של ה-`receiver` נכתוב את הפקודה `./cjag -r`, כאשר בברירת מחדל ירוץ ה-`sender`. נציין כי לאחר שבדקנו את התקיפה במספר מחשבים הבחנו כי מהירות המחשב משפיעה באופן ישיר על ריצה תקינה של ההתקפה ולכן במידה והמחשב קצת איטי יש צורך להוסיף פרמטר נוסף אשר הינו `-d` ולהזין מספר בטווח 10 – 4 (ככל שהמחשב איטי יש צורך במספר גבוה יותר).
- לדוגמה: `./cjag -r -d8`. (עבד ברוב המחשבים אשר ניסינו) כאשר הוספת ביטוי זה צריכה להיות חופפת בשני החלונות אשר פתחנו.



- כעת נשים לב להדפסות אשר מתארות את תהליך בחירת 19 הערוצים אשר באמצעותם נעביר את המידע.



- לאחר סיום בחירת הערוצים תודפס הודעה של אחוז ההצלחה ביצירת הערוצים. נציין כי במידה ותהליך זה איננו הצליח, כלומר אחוזי ההצלחה אינם 100% נרצה לנסות להפעיל ניסיון חדש (יכול לנבוע מרעשים וחוסר התאמות נכונות לחומרת המחשב).
- לאחר יצירת הערוצים תופיע הודעה המודיעה כי תהליך שליחת הקובץ מתחיל

```

doron@doron: ~/Desktop/
doron@doron: ~/Desktop/CJAG-master

sender start send file:
sender get seq: 001
sender send message: 110101010110010100
sender get seq:010
sender send message: 1101111111100100010
sender get seq:011
sender send message: 0001111011010110101
sender get seq:100
sender send message: 0111101011101000100
sender get seq:101
sender send message: 011010111111010011
sender get seq:110
sender send message: 1100010101111100101
sender get seq:011
sender get seq:111
sender send message: 101011111111110010
sender get seq:001
sender send message: 100001010101001011
sender get seq:010
sender send message: 10100101010100110
sender get seq:011
sender send message: 0111111111100110010
sender get seq:100
sender send message: 0010101010101000111

The receiver starts receiving the file:
receiver send seq:001
receiver get seq:001
receiver get message:110101010110010100
receiver send seq:010
receiver get seq:010
receiver get message:1101111111100100010
receiver send seq:011
receiver get seq:011
receiver get message:0001111011010110101
receiver send seq:100
receiver get seq:100
receiver get message:01111010111101000100
receiver send seq:101
receiver get seq:101
receiver get message:011010111111010011
receiver send seq:110
receiver get seq:110
receiver get message:1100010101111100101
receiver send seq:111
receiver get seq:111

```

- לצורך הבנת תהליך השליחה הוספנו מספר הדפסות אשר מופיעות רק במצב של *debug* אשר מוגדר להיות מופעל כברירת מחדל (ניתן לשנות מצב זה בקבצים "*receive.c*", "*send.c*")

```

46 void jag_sendFile(cjag_config_t* config, jag_callback_t cb) {
47     int debug = 1;

```

- בכדי להבחין כי תהליך השליחה הסתיים השתמשנו בשעון *watchdog* אשר נועד לזהות חוסר פעילות מצד השולח ולסיים את התוכנית בהתאם לכך, כאשר הגדרנו את הזמן המביע חוסר פעילות מקסימלי להיות 3 שניות אך ניתן לשנות זאת בהתאם למהירות השליחה (בבדיקות אשר ביצענו מהירות זו הושפעה ממהירות המחשב עליו ביצענו את התקיפה).

```

watchdog_start(&watchdog_settings, 15, receiverTimeout, (void *) output);

```

- לאחר הסיום קובץ הפלט יופיע גם כן באותה תיקיה תחת השם "*output.txt*"

- בנוסף, מכיוון שבמאמר המקורי המפתחים השתמשו בארבעה ביטים אשר מציינים את כמות האפסים בהודעה לצורך התיקון האחרון בשלב ה-*data linker*, ומכיוון שאנו לא מממשים את שלב ה-*data linker*, הוספנו בכל מקרה קובץ פלט בשם "*incorrect messages.txt*" אשר מכיל את כל ההודעות אשר נתפסו כמכילות שגיאה (במאמר הם מציינים כי הם מסמנים הודעות אלו לצורך תיקון סופי).

שם	תאריך שינוי	סוג	גודל
.idea	23/02/2021 13:13	תיקיית קבצים	
.vscode	23/02/2021 13:13	תיקיית קבצים	
cache	23/02/2021 13:13	תיקיית קבצים	
cs	23/02/2021 13:13	תיקיית קבצים	
detection	23/02/2021 13:13	תיקיית קבצים	
jag	23/02/2021 13:13	תיקיית קבצים	
util	23/02/2021 13:13	תיקיית קבצים	
cjag	23/02/2021 14:53	קובץ	146 KB
cjag.c	15/02/2021 20:13	קובץ C	14 KB
cjag.h	25/12/2020 20:30	קובץ H	2 KB
incorrect messages.txt	23/02/2021 14:56	מסמך טקסט	1 KB
input.txt	22/02/2021 20:15	מסמך טקסט	1 KB
Makefile	23/04/2017 19:11	קובץ	1 KB
output.txt	23/02/2021 14:56	מסמך טקסט	1 KB

- נבחין כי אנו מעבירים הודעות בנות 12 ביטים בכל פעם ולכן במידה ותוכן הקובץ איננו מתחלק ב-12 ההודעה האחרונה אשר מועברת תורחב אוטומטית להודעה בת 12 (בחרנו שהרחבה זו תתבצע על ידי אפסים).