**Exercise 2 – TCP Attacks**

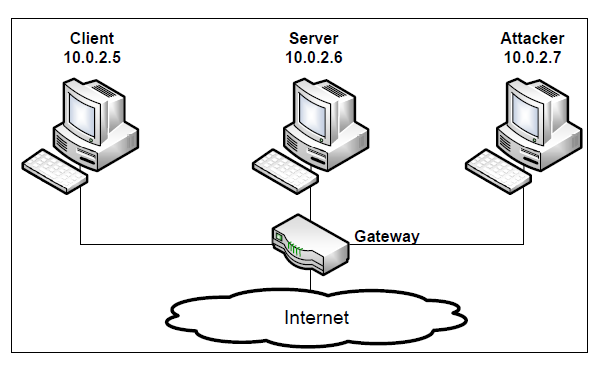
**מגישים:**

דוד סיידון

אופיר גן

**Lab Environment:**

כל המכונות הווירטואליות מוגדרות עם כרטיס רשת בתצורת NAT על גבי סביבת עבודה של vMware Workstation.



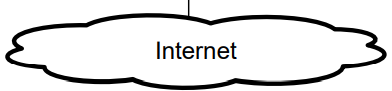
**Laptop**

**10.0.2.1**

**HUB**

**Laptop**

**10.0.2.1**



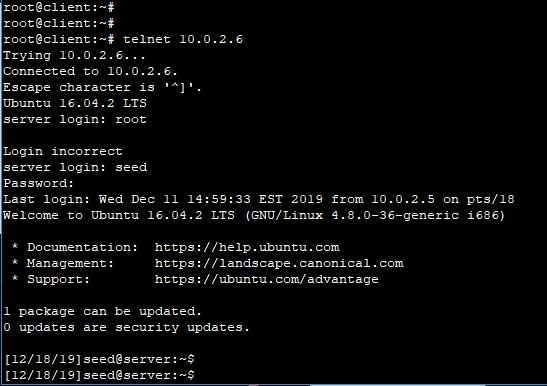
**Task 3.1: SYN Flooding Attack**

SYN cookie שומר על המחשב מפני התקפת SYN Flooding בכך שלא מתבצעת הקצאה של משאב עבור בקשת חיבור של לקוח לפני קבלת Ack ממחשב הלקוח. כך שבמצב SYN\_RECV לא נשמר משאב עבור הלקוח.

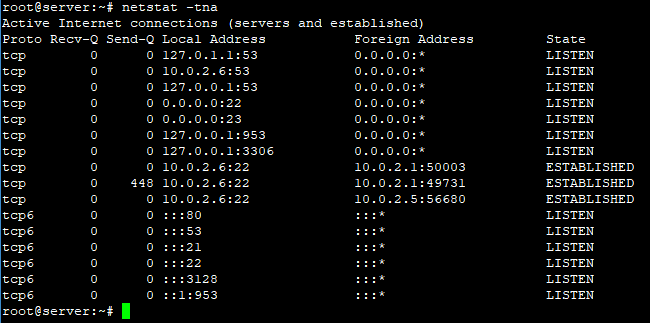
לעומת זאת, כאשר המנגנון כבוי והחיבור במצב SYN\_RECV כן מוקצה משאב לזמן קצוב לטובת החיבור, וברגע שמוקצים כל המשאבים לא ניתן להתחבר לשרת עד שמתפנים משאבים.

במהלך התקיפה התוקף לא מפסיק לייצר בקשות חיבור מול השרת ע"י שליחת SYN ובכך גורם להקצאת כל המשאבים של כרטיס הרשת של השרת לטובתו, ובכך אין אפשרות לשרת להקצות משאבים ללקוחות ולאפשר להם להתחבר.

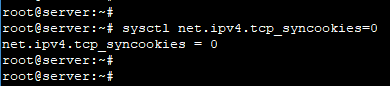
מחשב הלקוח מצליח להתחבר לשרת לפני התקיפה:



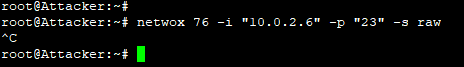
מצב פורטים בשרת לפני התקיפה:



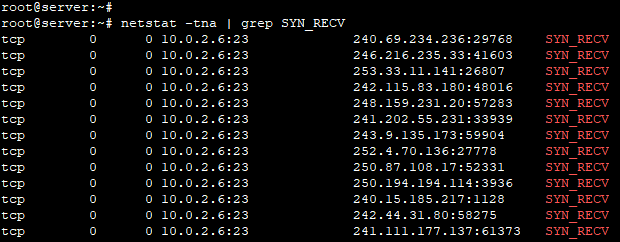
ביטול מנגנון הגנה בפני SYN Flooding בשרת:



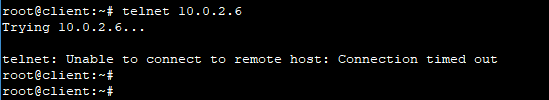
המחשב התוקף מבצע את התקיפה:



מצב שרת לאחר התקיפה:



מחשב לקוח מנסה להתחבר לשרת בזמן התקיפה (אך ללא הצלחה):



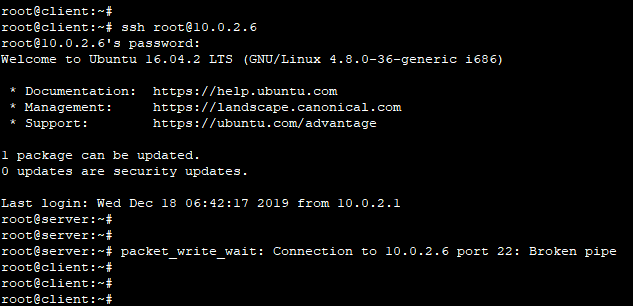
לסיכום:

הציפייה הייתה שהשרת יתמלא בבקשות של syn, יקצה משא ויעבור למצב ביניים לפני יצירת session מלא (כלומר – syn received). ברגע שהשרת יתמלא בבקשות, נסה להתחבר וניתן לראות שניסיון ההתחברות נכשל.

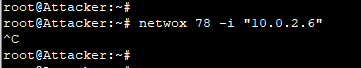
**Task 3.2: TCP RST Attacks on telnet and ssh Connections**

מחשב הלקוח מתחבר לשרת ב-SSH/TELNET. התקיפה מבוצעת על ידי שליחת פקטת TCP עם דגל RST דולק וsequence number מתאים אשר מדמה שהפקטה הנ"ל נשלחה מהלקוח לשרת ובעצם זאת הפקטה שלנו. השרת יקבל מאתנו בקשה לסגירת החיבור ומה שיקרה בפועל, החיבור בין השרת ללקוח ייסגר.

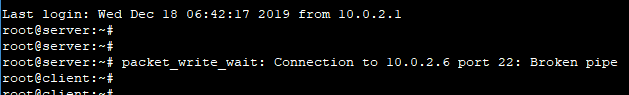
מצב מחשב לקוח: מחובר לשרת ב SSH



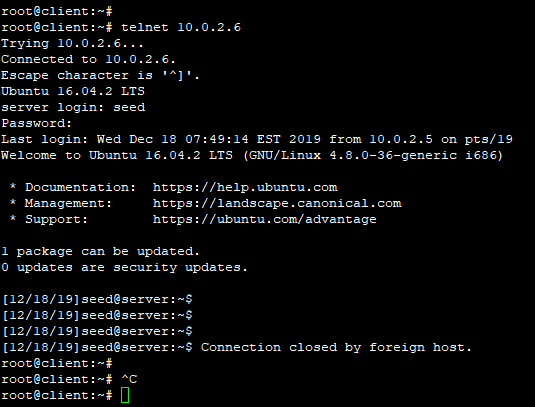
מחשב תוקף מבצע את התקיפה:



מחשב לקוח מנותק מהשרת:

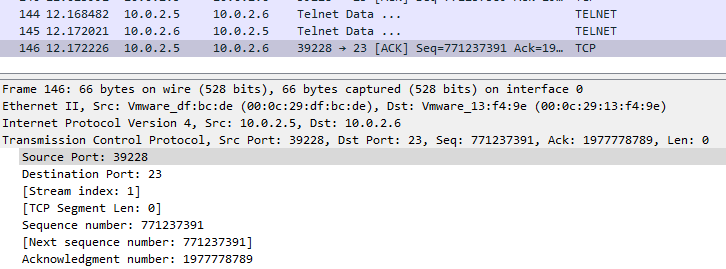


מחשב לקוח בביצוע ההתקפה בחיבור Telnet:

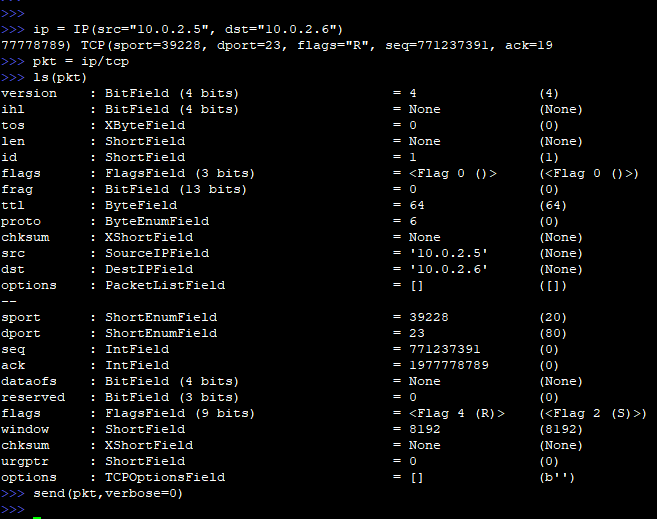


ביצוע ההתקפה ע"י Scapy:

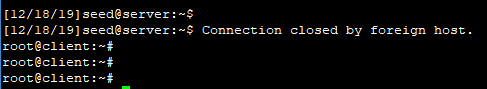
מציאת נתונים רלוונטיים בעזרת Wireshark



ביצוע התקיפה:



מצב הלקוח לאחר התקיפה:



לסיכום:

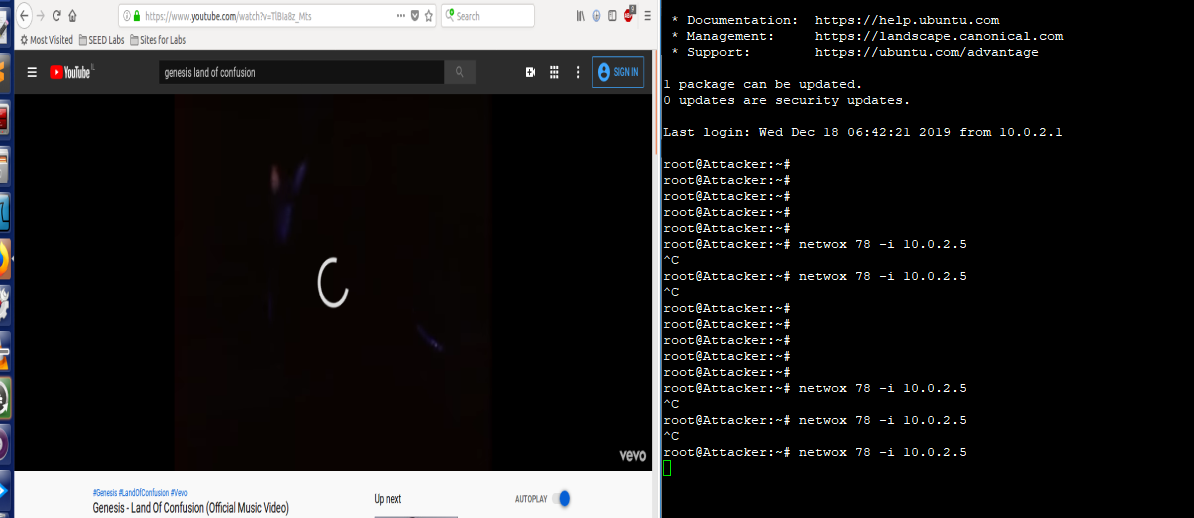
ניתן לראות שלפי התקציר שנרשם בתחילה המשימה, הפעולות שביצענו עבדו בהצלחה בעזרת הנתונים שאספנו וההתקפה הצליחה כפי שציפינו.

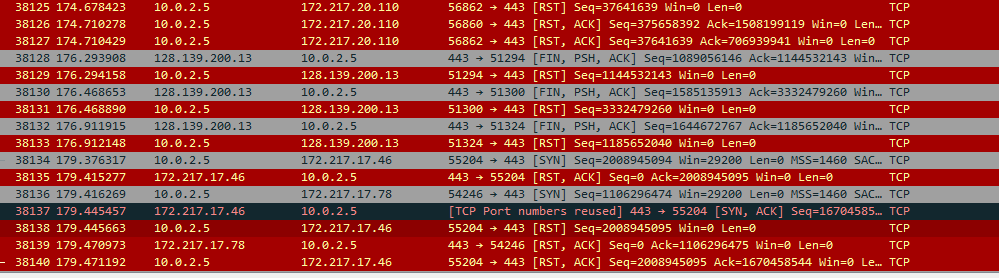
**Task 3.3: TCP RST Attacks on Video Streaming Applications**

הלקוח יתחבר בעזרת הדפדפן ביוטיוב ויצפה בסרטון. על ידי netwox 78 נבצע RST לכל פקטה שהלקוח שולח, מה שיביא לכך שהחיבור בין הלקוח לבין הסרטון ביוטיוב יתנתק ובעקבות זאת הסרטון הנפה בידי הלקוח יתקע.

בצד ימין מבוצעת התקיפה ע"י Netwox

ניתן לראות כי החיבור לשרת מופסק והסרטון לא מצליח להיטען.





לסיכום:

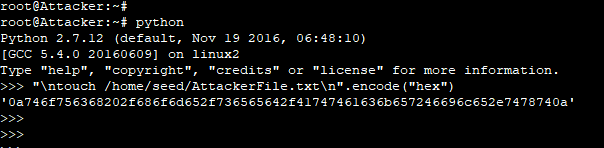
התקיפה עבדה בדיוק לפי איך שציפינו אותה.

ביצוע הריסט לכל פקטה עבר בהצלחה והסרטון שנצפה בידי הלקוח הפסיק להיטען ונעצר.

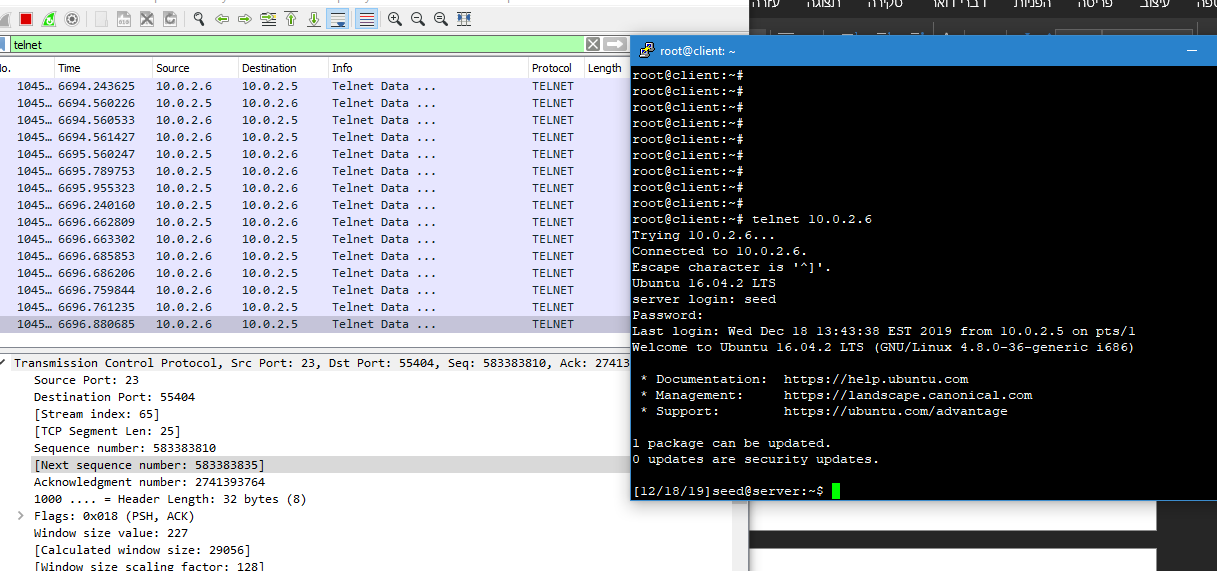
**Task 3.4: TCP Session Hijacking**

התהליך יתבצע בכך שהמחשב התוקף ייצר פקטה של TCP המכילה פקודה שתבוצע בצד השרת והפקטה תשלח עם פרטים המזהים אותה כזאת שנשלחה מהלקוח אל השרת שמחובר אליו בTELNET. את כל הפרטים הדרושים אנו נייצג בעזרת תוכנת WIRESHARK ונצבע את התקיפה בעזרת תכונה NETWOX 40 שמייצר פקטת TCP לפי הפרטים שהושגו מהפקטה.

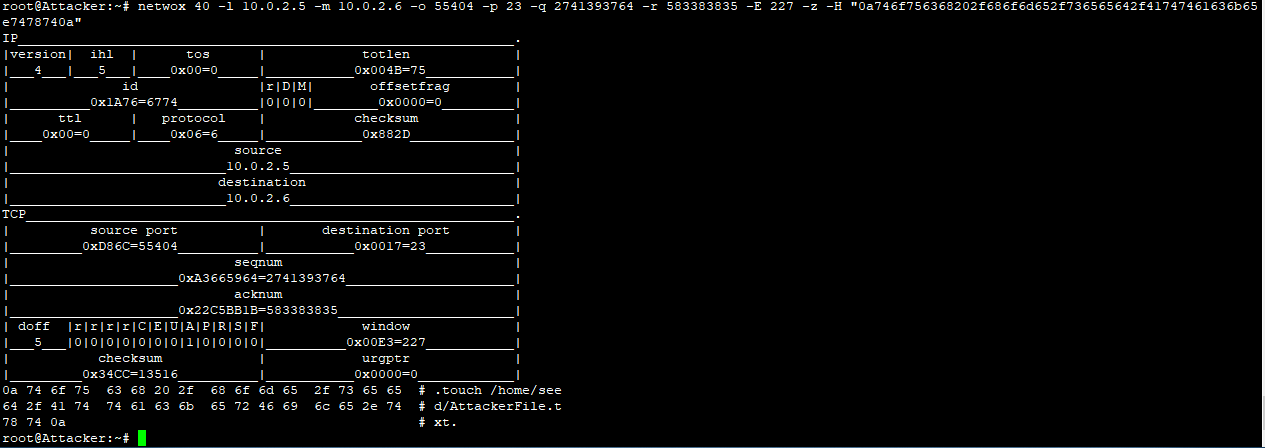
המרת הפקודה:

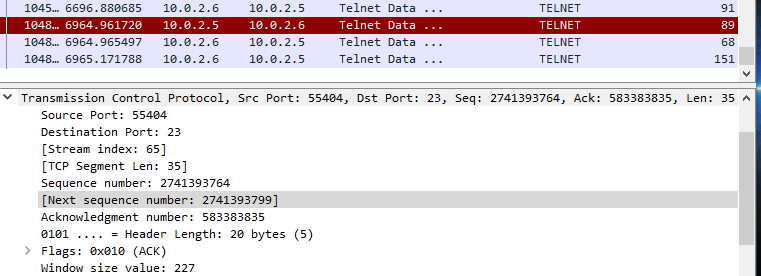


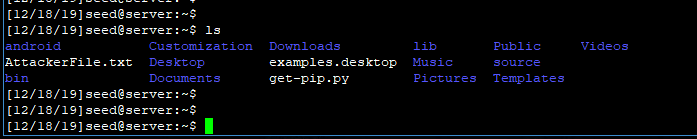
השגת המידע הדרוש בעזרת Wireshark:



הרצת הפקודה מהמחשב התוקף:





הקובץ שנוצר בשרת ע"י המחשב התוקף:

לסיכום:

לאחר מספר נסיונות, התקיפה הצליחה. בנסיונות הדרושים לא הוכנסו הנתונים הנכונים שהיו צריכים להיות בפקטה ולכן השרת התעלם מהבקשה של התוקף והתקיפה לא עבדה.