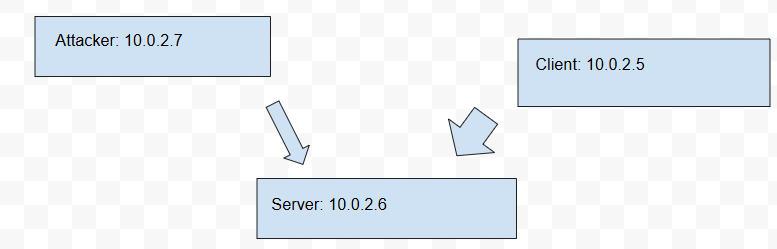
**חלק 1 – syn flood**

**קבוצה מספר 2**

**יוגב כהן 315824508**

**שי שלו 211783519**

**מור משה 323915363**

****

**במשימה זו, נבצע התקפת TCP Flood במטרה להציף את השרת בבקשות SYN רבות, דבר שיגרום לו למנוע חיבורים חדשים ולהפוך אותו לבלתי זמין. ההתקפה מנצלת את שלב החיבור ב-TCP, שבו השרת שולח חבילות SYN-ACK מבלי להשלים את החיבור. התוצאה הצפויה היא האטה או קריסת השרת עקב עומס משאבים.**

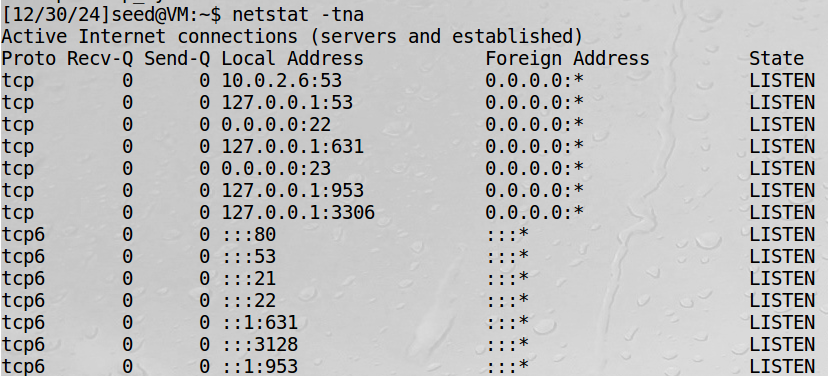
יצרנו 3 מכונות שונות -attacker, client and server והגדרנו להם כתובות IP

Attacker – 10.0.2.7

Server – 10.0.2.6

Client – 10.0.2.5

ביטלנו את ה-cookies עבור התקיפה:

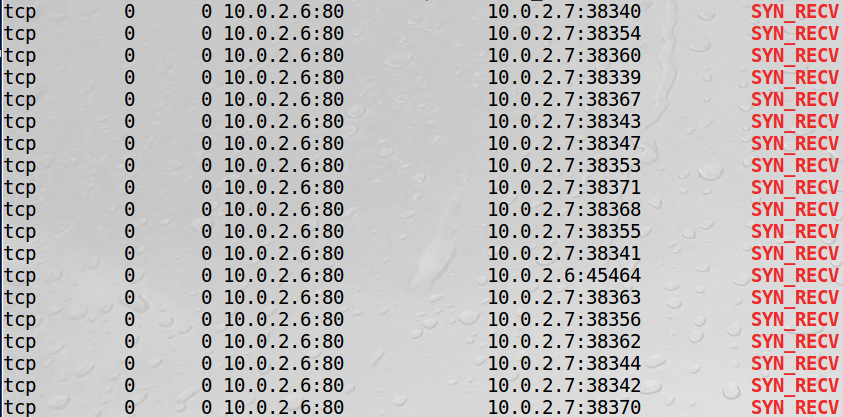
נבדוק שכל הפורטים שמאזינים:

תמונה שמכילה טקסט, תפריט, מספר, צילום מסך

התיאור נוצר באופן אוטומטינבצע התחברות של ה-client לשרת ונבדוק שהחיבור אכן בוצע:

נבצע את התקיפה מה- attackerלשרת:

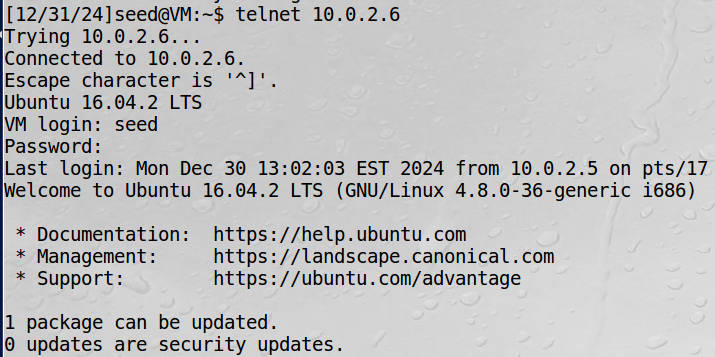


נבדוק את פורט 80: 

ננסה לעשות חיבור מה- client לשרת:

נחזיר את ה- cookies:



נבצע את התקיפה שוב כשיש cookies:

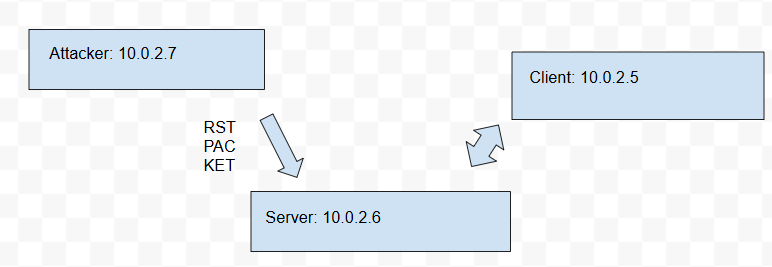
סיכום:

במשימה זו הצלחנו לבצע התקפת TCP Flood על השרת, שהפך לא זמין בעקבות הצפתו בבקשות SYN. ההצלחה הוכחה באמצעות ניטור משאבי השרת, לוגים מהרשת, ובדיקות מצד הלקוח שהראו כשל בחיבורים. גילינו שהשרת פגיע להתקפות מסוג זה בהיעדר מנגנוני הגנה. התוצאות תאמו את התיאוריה, שמסבירה כיצד ניצול יתר של משאבי השרת מוביל לשיבוש פעילותו.

TCP RST Attacks on telnet and ssh Connections - חלק 2

מבוא:

במשימה זו נבצע התקפת TCP RST (Reset) במטרה לשבש חיבורי SSH ו-Telnet פעילים. התקפה זו מנצלת חולשה בתהליך ניהול החיבורים של TCP, שבו ניתן לשלוח חבילת RST מזויפת כדי לנתק חיבורים קיימים. מטרת המשימה היא להבין את ההשפעה של התקפות אלו על חיבורי פרוטוקולים מבוססי TCP המשמשים לניהול מרוחק.



השתמשנו בtelnet כדי ליצור חיבור בין הclient לserver:



חיפשנו בwireshark את החיבור המתאים:

תמונה שמכילה טקסט, חשמל, צילום מסך, תוכנה

התיאור נוצר באופן אוטומטי

בנינו קובץ פייתון עם הפרטים עבור RESET ATTACK:

תמונה שמכילה טקסט, צילום מסך, תוכנה

התיאור נוצר באופן אוטומטי

שמנו את הפקודה הבאה:



ההודעה הבאה מראה שאכן אנו שולחים את הקובץ.

ד

ואכן ניתן לראות שהחיבור נסגר.



הפעם ניצור חיבור SSH על פורט 22 תמונה שמכילה טקסט, צילום מסך, גופן, מכתב

התיאור נוצר באופן אוטומטי

נחפש בwireshark את החיבור עם הdestination port המתאים

תמונה שמכילה טקסט, צילום מסך, תוכנה, סמל מחשב

התיאור נוצר באופן אוטומטי

עכשיו נשלח שוב את ה RESET :



לאחר מכן ניתן לראות שאכן החיבור נותק :

סיכום:

במשימה זו הצלחנו לבצע התקפת TCP RST על חיבורי SSH ו-Telnet פעילים, מה שגרם לניתוק החיבורים באופן מידי. ההצלחה נבדקה באמצעות ניטור השרת והלקוח, שהראו ניתוק החיבור ברגע קבלת חבילת RST מזויפת. גילינו שהתקפה זו אפקטיבית במיוחד אם התוקף מסוגל לזהות את פרטי החיבור (כמו מספרי הפורטים והסדרה). התוצאה תאמה את התיאוריה, המראה כיצד TCP מתמודד עם בקשות RST כחלק מתכנון הפרוטוקול. האתגרים כללו זיהוי הפרמטרים המדויקים ליצירת חבילת ה-RST, אותם פתרנו באמצעות כלי ניתוח תעבורה לזיהוי פרטי החיבורים.

חלק 3 - TCP RST Attacks on Video Streaming Applications

במשימה זו נבצע התקפת TCP RST על שירותי הזרמת וידאו (Video Streaming Services) במטרה לנתק חיבורים פעילים בין הלקוח לשרת. התקפה זו מנצלת את מנגנון ה-Reset של TCP, שבו ניתן לשלוח חבילת RST מזויפת כדי להפסיק חיבורים פעילים. מטרת המשימה היא להבין כיצד התקפה זו משפיעה על חוויית המשתמש בשירותי סטרימינג ועל רציפות הזרמת הנתונים.

בחירת סרטון יוטיוב והפעלה בצד הCLIENT לוודא שאכן הכל עובד כשורה:

תמונה שמכילה טקסט, פני אדם, צילום מסך, אדם

התיאור נוצר באופן אוטומטי

השתמשנו בפקודת הnetwox 78 עבור הפסקת הסרטון :



וכפי שניתן לראות הסרטון אצל הקליינט נעצר :

תמונה שמכילה טקסט, צילום מסך, תוכנה, סמל מחשב

התיאור נוצר באופן אוטומטי

סיכום:

במשימה זו, ביצענו התקפת TCP RST במטרה להפסיק את חיבור ה-TCP בין הקורבן לשרת הווידאו, במקרה זה YouTube. ההתקפה התבצעה באמצעות כלי Netwox, תוך שליחה של חבילת RST לכתובת ה-IP של הקורבן. כתוצאה מכך, חיבור ה-TCP נותק והדפדפן הציג את השגיאה "Secure Connection Failed".

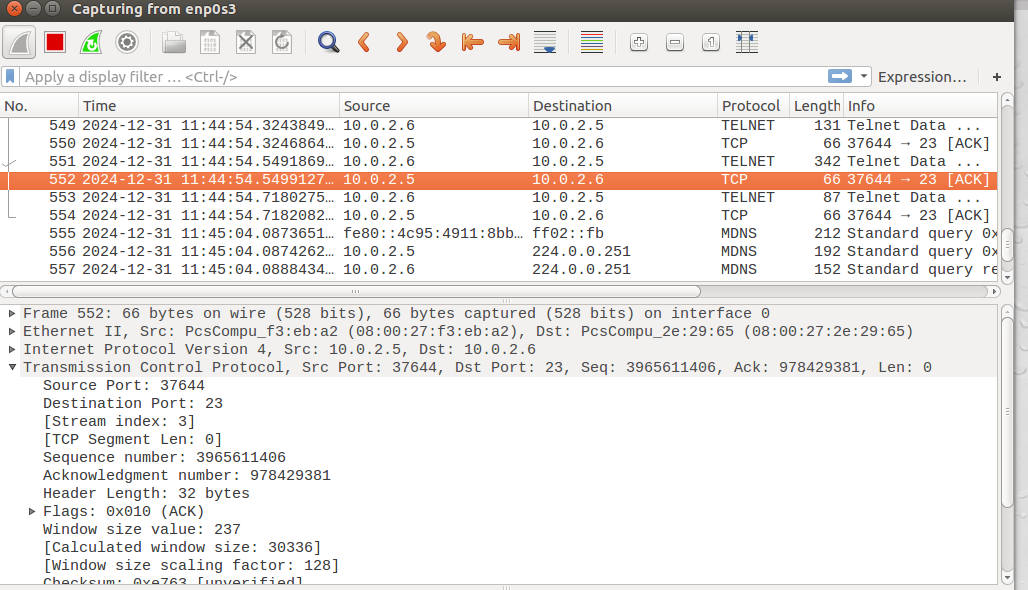
חלק 4 -TCP Session Hijacking

מבוא:

**תמונה שמכילה טקסט, צילום מסך, גופן, תרשים

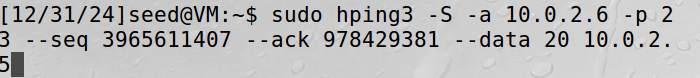
התיאור נוצר באופן אוטומטי**

במשימה זו נבצע התקפת TCP Session Hijacking, שמטרתה להשתלט על חיבור TCP פעיל בין לקוח לשרת. ההתקפה מנצלת חולשה בתהליך ניהול הסשן של TCP, שבו התוקף מזהה ומזייף חבילות נתונים כדי להיכנס לחיבור ולשלוח פקודות בשם המשתמש החוקי. המשימה נועדה להבין את השפעת ההתקפה ואת הדרכים שבהן ניתן להתגונן מפניה.

ננסה לחפש את החיבור TCP האחרון:

ננסה לחפש את החיבור TCP האחרון בין הקליינט לסרבר:

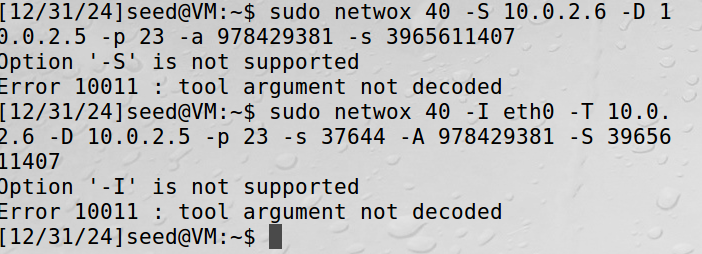
כעת כשיש לנו את מספר הACK וגם את הSEQ נוכל ליצור פקטה זדונית מהקליינט לסרבר ( בשם הקליינט):

ניסינו בהתחלה בעזרת HPING3:

ללא הצלחה.

ננסה בעזרת netwox:

שוב ללא הצלחה.



סיכום

במשימה זו ניסינו לבצע התקפת TCP Session Hijacking על חיבור פעיל, אך לא הצלחנו להשיג שליטה על החיבור. זיהינו את פרטי החיבור הפעיל, כולל מספרי הפורטים ורצף החבילות, אך ניסיונות השתלטות באמצעות הכלים HPing3 ו-Netwox לא צלחו. למרות הכישלון, הבנו את המורכבות הכרוכה בזיהוי וניחוש רצף המספרים המדויק שנדרש להשתלטות על הסשן.

המשימה חידדה את ההבנה בנוגע לאתגרים הטכניים בהתקפה מסוג זה, כמו הצורך בתזמון מדויק ומעקב מדויק אחר החבילות. התוצאות הראו שפרוטוקול TCP אמנם פגיע תאורטית, אך יש צורך במיומנות גבוהה ובתנאים מסוימים כדי לנצל פגיעות זו בפועל.