

**מעבדה מספר 3**

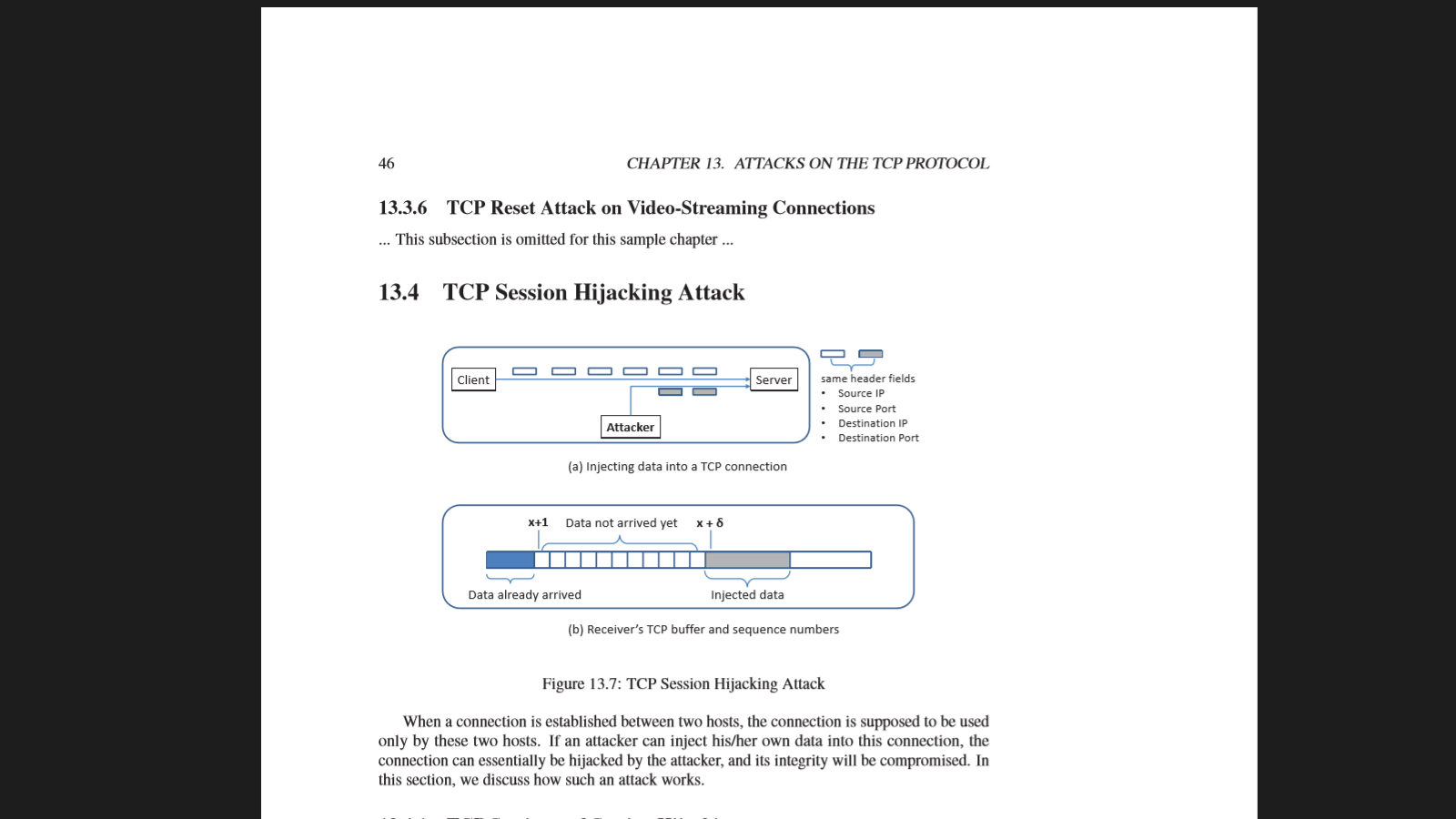
**TCP Session Hijacking Attack**

**TCP Session Hijacking Attack**

לאחר סיום פרוטוקול לחיצת היד המשולשת נוצר קשר TCP בין שני מחשבים. מנקודה זו והלאה, שני הצדדים יוכלו לשלוח ולקבל מידע בינהם. קשר זה שנוצר, אמור להיות משומש רק על ידי שני הצדדים שפתחו אותו. אם תוקף חיצוני יוכל להתחבר גם הוא להעברת התקשורת בקשר, כלומר לקבל את הפאקטות הנשלחות ולשלוח פאקטות בעצמו, אזי נאמר כי הקשר "נפרץ" ואמינותו נפגעה.

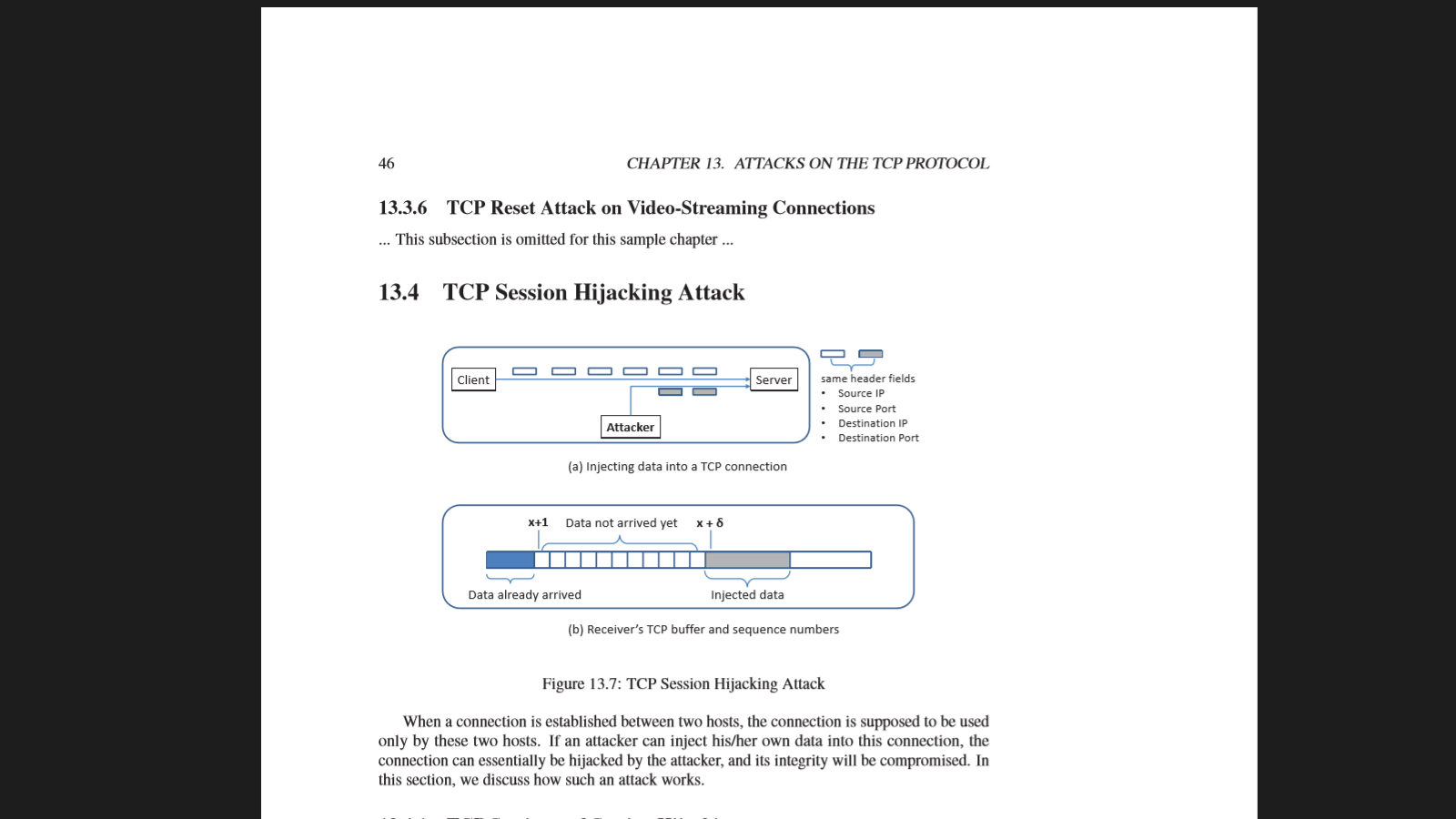
מאחר ומחשב יכול להיות חלק מכמה קשרי TCP, בעת קבלת הודעה עליו לדעת לאיזה קשר היא שייכת. לצורך כך, משתמש המחשב בארבעה שדות: source TCP address, source port, destination IP address, destinaion port. נהוג לקרוא לארבעה שדות אלו ה"חתימה" של קשר ה – TCP.

כמו שראינו בעבר, ניתן בקלות יחסית לזייף פאקטת TCP עם חתימה של קשר קיים ולשולחה, כך שהצד המקבל לא יוכל להבדיל באם נשלחה פאקטה מהצד השני או מתוקף חיצוני. בצורה זו יכול התוקף "לדחוף" הודעות לקשר, לבצע פעולות בשם הצדדים ולאסוף מידע רגיש.

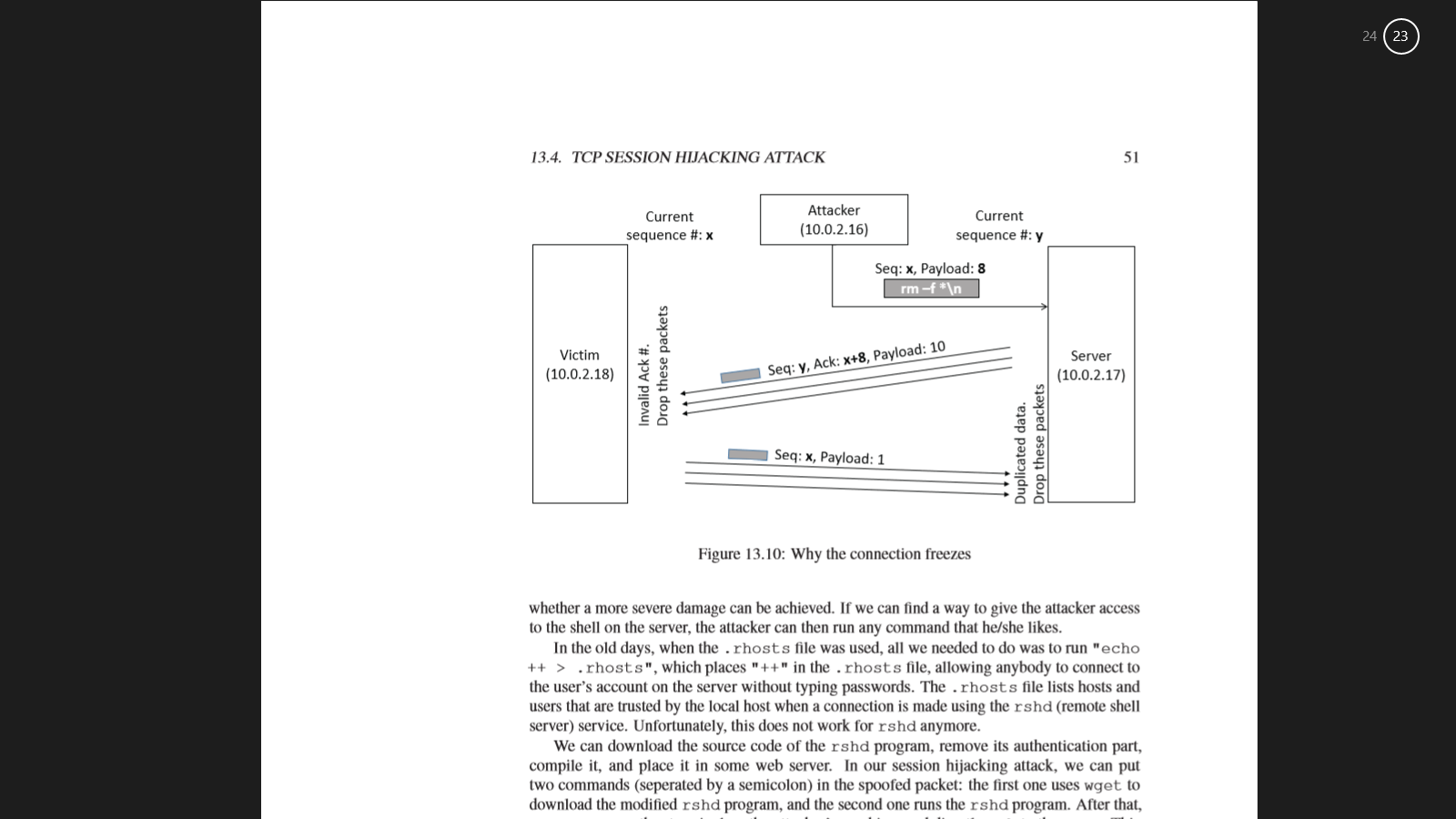


על מנת לעשות זאת, מלבד חתימת הקשר, התוקף צריך בנוסף לזייף ולשמור על עקביות ה – seqence number – המספר הסידורי בקשר. בתקשורת TCP אנו מתייחסים לדאטה כאל מחרוזת – string, כך שלכל בייט בדאטה ישנו מספר סידורי בלעדי, המאפיין את מיקומו במחרוזת. בפאקטת הTCP ישנו שדה בן 32 ביט המכיל את מיקום הבייט הראשון במידע הנשלח. בעת קבלת הפאקטה, המחשב שומר את המידע בבאפר המסודר על פי המספרים הסידוריים ובצורה זו, גם אם סדר קבלת ההודעות שונה, סדר המידע הנשלח נשמר. לכן, לצורך העניין, אם המחשב קיבל מידע עד הבייט ה- x, הוא מצפה לקבל את הבייט ה- x+1 בהודעה הבאה. אם המספר הסידורי המזוייף יהיה x + d, אזי ההודעה תחשב כהודעה שהגיעה מחוץ לסדר והמידע ישמר בבאפר, במידה וישנו מקום, בעמדה ה – x+d, כאשר נשארו לפניו d-1 ביטים ריקים עבור המידע המתחיל במקום ה – x+1. במקרה זה, המידע המזוייף לא יעבור לשכבת האפקליקציה עד להשלמת המידע החסר בהודעה ולכן לא יהיה לו כל אפקט. אם הערך של d גדול מדי, המידע עלול להיזרק אל מעבר לגבולות הבאפר.

לעיתים, עקב הקושי בדיוק במספר הסידורי, הוא מזוייף מראש עם מרווח טעות קטן, כך שהפאקטה המזויפת תשמר בבאפר ואף על פי שלא תיושם מיד, לאחר שליחת כמה הודעות בין הצדדים ומילוא המרווח עד המידע המזוייף, תשלח ההודעה המזויפת לשכבת האפליקציה ותוצא לפועל.



אם כן, על מנת לפרוץ לקשר TCP, עלינו לזייף נכונה את חתימת הקשר ובנוסף את המספר הסידורי של ההודעות. באשר הצלחנו בכך, נוכל לשלוח, לבצע פעולות ולקבל מידע רגיש, כלומר "לפרוץ" את קשר ה-TCP.



**שיטות מנע**

Session hijacking attack מתאפשרת עקב ידיעת חתימת קשר ה – TCP ועל ידי אפשרות לזייף מספר סידורי מתאים. על כן ינשן כמה גישות מנע:

1. שימוש במפתח קשר ראנדומאלי ומשתנה. בצורה זו קשה יותר לזייף פאקטה שתחשב קבילה.
2. יצירת חתימת קשר חדשה, לאחר הקמת הקשר, על ידי שני הצדדים. בצורה זו התוקף לא ידע לזהות את הקשר אותו הוא רוצה לתקוף.
3. הצפנה ווידוא של המידע ברמת ה- IP , כלומר הרמה המכילה את מספרי הכתובות, הפורטים והמספרים הסידוריים. קיימים פרוטוקולי בטיחות רשת - IPsec המנהלים שיטת מנע זו.

**שאלות תיאורטיות**

1. כתבו את הגדרת קשר "נפרץ".
2. הגדר את המושג "חתימה" של קשר TCP ומדוע משתמשים בו.
3. מה עלינו לעשות על מנת לבצע Session hijacking attack?
4. הסבר מהי החשיבות בבחירת המספר הסידורי של הפאקטה הנשלחת וכיצד מתמודדים עם בחירת המספר הסידורי המתאים.
5. הסבר בקצרה את שיטות המנע כנגד Session hijacking attack.

**שאלות פרקטיות**

בשאלות הבאות תקבלו 2 קבצים Sessionhunter.py ו Session\_hijack.py

כאשר Sessionhunter הינו קובץ ההפעלה של המתקפה עליכם להריץ רק אותו.

הפעלה:

1. הריצו את[interface] Sessionhunter.py
2. הריצו את Server ו client
3. כתבו help בSessionhunter ותקבלו את כל הפקודות שניתן לכתוב לו.
4. בצעו הסנפה ואז אעבירו מסר מהclient ל server
5. כתבו ls בSessionhunter ותקבלו את רשימת הIP שהוא הצליח לקלוט
6. כעת כתבו hijack [n] כאשר n הוא המספר שרואים בlist conections (ls)
7. כתבו מספר פעמים בין הclient לserver עד אשר אתם רואים שהצלחתם לראות מה עבר בינהם.
8. לימדו את כל הפקודות שיש בSessionhunter

כל פקודה חדשה שאתם מוסיפים הוסיפו אותה גם בhelp

1. הוסיפו מצב שרושמים פקודה לא נכונה יודפס למסך : not an option!
2. כאשר מבצעים hijack [n] נפתח החלון שדרכו אנו מאזינים לאחד מהIP's , הוסיפו פקודה לביצוע hijack\_list שמקבלת מספר רב של פורטים לדוגמא

hijack 0 1 2 3

והוא יפתח חלונות האזנה לכל הconections שאנו מבקשים.

1. הוסיפו לקוד שכאשר מבצעים hijack\_all הוא יריץ את כל הIP's שנמצאים לו בconnections list מבלי לתת לו את מספרי הconections.
2. הוסיפו את הפקודה empty אשר מרוקנת את כל הconections .
3. הוסיפו את הפקודה sniff [ip] אשר מקבלת IP ורק לו מאזינה. השתמשו בפונקציה sniff\_contection\_filter
4. שנו את הקובץ session\_hijack שהוא ימשיך לקרוא את הקשר בין הclient לserver