

דו"ח מסכם: פרויקט תקשורת נתונים (TCP/IP & Sockets)

חלק 1: אריזה ולכידת נתונים (Encapsulation)

דרך יצירת קובץ ה-CSV

הקובץ group01_http_input.csv נוצר באופן ידני על מנת לDemystify תובנות HTTP בסיסית בין דףדף לשרת. הנתונים הוכנו כך שכל שורה מייצגת הודעה בשכבה האפליקצייתית, הכוללת מזהה הודעה (id), שם הפהוטוקול (HTTP), פורטים, ותוכן ההודעה (לדוגמה: index.html / GET). שיטה זו מאפשרת לנו לשלוט בתוכן המדויק שנשלח ולעקב אחריו בשלבי האריזה.

תיאור והסבר של תהליכי אריזת הנתונים

בתרגיל זה ביצענו סימולציה של מודל TCP/IP דרך קוד המחברת:

1. שכבת האפליקציה (Application Layer): המידע הגלומי נלקח מהקובץ (ההודעה עצמה).
2. שכבת התעבורה (Transport Layer): הקוד בונה Header של TCP הכולל Source Port אקראי ו-Destination Port שהוגדר כ-12345. נעשה שימוש בחישוב checksum כדי להבטיח את תקינות הנתונים.
3. שכבת הרשת (Network Layer): בניית Header של IP (גרסה 4) ה כוללת כתובות ה-IP של המקור והיעד (127.0.0.1) ושדה פרוטוקול המוגדר כ-TCP.
4. שכבת הקישור (Data Link Layer): המנה המובנה נשלחת דרך ממשך ה-Loopback של המערכת.

תיאור תהליכי הלכידה

לכידת הנתונים בוצעה באמצעות תוכנת Wireshark. מכיוון שהתקשרות הבצעה בין תהליכיים פנימיים באותו מחשב (localhost), בחרנו להקשיב לממשך ה-Loopback Adapter Npcap. על מנת לבדוק את הנתונים הרלוונטיות, הפעילנו מסנן (Filter) עבור tcp.port == 12345.

תיאור וניתוח התעבורה ב-Wireshark

בניתוח הנתונים שנלכדו ניתן לראות בבירור את ההיררכיה של שכבות המודול:

- בבדיקה ה-Header IP ניתן לראות שכתובות המקור והיעד הן 127.0.0.1.
- בבדיקה ה-Header TCP ניתן לראות את הפורטים שהגדכנו ואת ה-Sequene Numbers.
- הנתונים נשואות דגל Push (PSH), מה שמעיד על כך שהמידע משכבה האפליקצייתית מועבר מיד לשכבות העליונות. בתוך גוף המנה (Payload) ניתן לראות את הטקסט המדויק שהופיע בקובץ ה-CSV.

חלק 2: יישום רשת (Chat System)

הסבר כללי על המערכת ומבנה הקוד

המערכת מורכבת משרת מרכזי (Server) ולוקוחות קצה (Clients) המתחברים באמצעות TCP Sockets.

- השרת (Server): השרת מאזין בפורט 5555. הוא מנהל מבנה נתונים מסוג Dictionary השומר את שמות המשתמשים ואת ה-Sockets הפעילים שלהם. השרת עשויה שימוש ב-Multithreading, כך שכל לקוח חדש שמלווה בחיבור TCP מקבל תחביבון (Thread) נפרד, מה שמאפשר טיפול בלקוחות רבים בו-זמנית.
- הלקוח (Client): הלקוח מאפשר למשתמש להירשם בשם ייחודי ולשלוח הודעות ליעד ספציפי בפורמט Target:Message. הלקוח מפעיל תחביבון רקע להאזנה קבואה להודעות כניסה, מה שמאפשר תקשורת דו-כיוונית בזמן אמת.

הוראות התקינה והריצה

1. וודא שモתקנת סביבת Python 3 על המחשב.
2. הרצת השרת: יש לפתח טרמינל ולהריץ את הפקודה `python server.py`.
3. הרצת לקוחות: יש לפתח טרמינלים נוספים ולהריץ את הפקודה `python client.py` עבור כל משתמש.
4. שימוש ביצום: בכל לקוח יש להזין שם משתמש, ולאחר מכן ניתן לשלוח הודעות למשתמשים אחרים המוחברים לשרת.

דוגמאות קלט ופלט

- קלט בלקוח א': UserB>Hello from UserA
- פלט בשרת: [NEW CONNECTION] UserA connected.
- פלט בלקוח ב': [UserA : Hello from UserA]

ניתוח תעבורת (עד שכבת הרשת)

בלכידת התעבורת של היישום זיהינו את השלבים הבאים:

1. TCP Handshake: שלוש המנות הראשונות (SYN, SYN-ACK, ACK) המעידות על יצירת חיבור אמין בין הלקוח לשרת בפורט 5555.
2. העברת נתונים: הודעות הציג נראות במנות TCP עם דגלי PSH ו-ACK. בבדיקה תוכן המנה מראה את מחרוזת הטקסט שנשלחה בשכבת האפליקציה.
3. שכבת הרשת: המנות משתמשות בפרוטוקול IP, כאשר שדה ה-Protocol מוגדר כ-6 (TCP), ובתובות המקור/יעד תואמות לכתובות המקומיות של המכוונות.

תיאור שימוש במבנה מלאכותית (AI)

מטרות השימוש

מטרות השימוש

השימוש במבנה מלאכותית נעשה ככלי עזר לשיפור היעילות במספר נקודות:

1. **סיווג בעיצוב מבנה נתונים:** הגדרת מבנה קובץ CSV על מנת שיעמוד בדרישות הטכניות שלuproject.
2. **שימוש לוגיקה בינארית:** עזרה בסינטקס של פונקציית `struct.pack`. לצורך המרת שדות ה-`Headers` לפורמט בינארי (Big-endian) הנדרש ב프וטוקול תקשורת.
3. **ניהול ותיעוד הידע:** נעזרנו בכלי AI לצורך סידור וארגון הממצאים שלנו, המרת הליקטים שהפכנו במהלך העבודה והותבנות מהניטויים לכדי קובץ מסכם, מובנה וברור העונה על דרישות הדיווח.
4. **תמייה טכנית:** קבלת הסברים על הגדרות בתוכנת Wireshark לצורךlibcידת תעבורת Loopback Windows.

דוגמאות פרומפטים

- "How to use Python struct.pack to build a TCP header?"
- ".Create a sample CSV file structure for HTTP application messages
- ".Explain the difference between SYN and PSH flags in Wireshark capture

רשימת תוצרים מצורפים:

1. קובץ הנתונים `group01_http_input.csv`.
2. מחרת `Jupyter` המבילה את קוד האזיה.
3. קבצי המקור של היישום: `client.py` ו-`server.py`.
4. קובץ הגדרת מבנה הנתונים `db_schema.txt`.
5. קבצי libcידת בפורמט `pcap` עבור שני חלקיםuproject.