

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

חלק 1: אריזה ולכידת מנות (Encapsulation)

דרך יצירת קובץ ה-CSV

הקובץ group01_http_input.csv נוצר באופן ידני על מנת לדמות תעבורת HTTP בסיסית בין דפדפן לשרת. הנתונים הוכנו כך שכל שורה מייצגת הודעה בשכבת האפליקציה, הכוללת מזהה הודעה (msg_id), שם הפרוטוקול (HTTP), פורטים, ותוכן ההודעה (לדוגמה: GET /index.html). שיטה זו אפשרה לנו לשלוט בתוכן המדויק שנשלח ולעקוב אחריו בשלבי האריזה.

תיאור והסבר של תהליך אריזת המנות

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

1. שכבת האפליקציה (Application Layer): המידע הגולמי 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) הכוללת את כתובות ה-IP של המקור והיעד (127.0.0.1) ושדה פרוטוקול המוגדר כ-TCP.
4. שכבת הקישור (Data Link Layer): המנה המוכנה נשלחת דרך ממשק ה-Loopback של המערכת.

תיאור תהליך הלכידה

לכידת המנות בוצעה באמצעות תוכנת Wireshark. מכיוון שהתקשורת התבצעה בין תהליכים פנימיים באותו מחשב (Localhost), בחרנו להקשיב לממשק ה-Npcap Loopback Adapter. על מנת לבדוד את המנות הרלוונטיות, הפעלנו מסנן (Filter) עבור tcp.port == 12345.

תיאור וניתוח התעבורה ב-Wireshark

בניתוח המנות שנלכדו ניתן לראות בבירור את ההיררכיה של שכבות המודל:

- בבדיקת ה-IP Header ניתן לראות שכתובות המקור והיעד הן 127.0.0.1.
- בבדיקת ה-TCP Header ניתן לראות את הפורטים שהגדרנו ואת ה-Sequence Numbers.
- המנות נושאות דגל (PSH (Push), מה שמעיד על כך שהמידע משכבת האפליקציה 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) ולקוחות קצה (Clients) המתקשרים 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 עושה שימוש ב-Multithreading, כך שכל לקוח חדש שמלווה בחיבור TCP מקבל תהליכון (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 תהליכון רקע להאזנה קבועה להודעות נכנסות, מה שמאפשר תקשורת דו-כיוונית בזמן אמת.

הוראות התקנה והרצה

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

דוגמאות קלט ופלט

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

ניתוח תעבורה (עד שכבת הרשת)

בליכידת התעבורה 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 הצ'אט נראות במנות TCP עם דגלי PSH ו-ACK. בחינת תוכן המנה מראה את מחרוזת הטקסט שנשלחה בשכבת האפליקציה.
3. שכבת הרשת: המנות משתמשות בפרוטוקול IP, כאשר שדה ה-Protocol מוגדר כ-6 (TCP), וכתובות המקור/יעד תואמות לכתובות המקומיות של המכונות.

תיאור שימוש בבינה מלאכותית (AI))

מטרות השימוש

מטרות השימוש

השימוש בבינה מלאכותית נעשה ככלי עזר לשיפור היעילות במספר נקודות:

1. **סיוע בעיצוב מבנה נתונים:** הגדרת מבנה קובץ ה-CSV על מנת שיעמוד בדרישות הטכניות של הפרויקט.
2. **מימוש לוגיקה בינארית:** עזרה בסינטקס של פונקציות struct.pack לצורך המרת שדות ה-Headers לפורמט בינארי (Big-endian) הנדרש בפרוטוקולי תקשורת.
3. **ניהול ותיעוד הידע:** נעזרנו בכלי ה-AI לצורך סידור וארגון הממצאים שלנו, המרת הלקחים שהפקנו במהלך העבודה והתובנות מהניסויים לכדי קובץ מסכם, מובנה וברור העונה על דרישות הדיווח.
4. **תמיכה טכנית:** קבלת הסברים על הגדרות בתוכנת Wireshark לצורך לכידת תעבורת Loopback במערכת ההפעלה 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. מחברת Jupyter המכילה את קוד האריזה.
3. קבצי המקור של היישום: server.py ו-client.py.
4. קובץ הגדרת מבנה הנתונים db_schema.txt.
5. קבצי לכידה בפורמט pcap עבור שני חלקי הפרויקט.