

דוח מסכם: פרויקט תקשורת מחשבים - חלק 1

מגישות: נטע גולזאד 213487200, עדן זרביאן 212055024

https://github.com/edenzarbian/networking_project

פרוטוקול HTTP:

פרוטוקול HTTP הוא פרוטוקול תקשורת המשמש להעברת מידע בשכבת האפליקציה (Application Layer) במודל ה-OSI בין לקוח (Client) לבין שרת (Server).

- **הלקוח (Client)** - מייצג יישום קצה, כגון דפדפן או סקריפט (לדוגמה: קובץ Python), אשר שולח בקשה לשרת.
- **השרת (Server)** - מאזין לבקשות, מעבד אותן ומחזיר תשובה הכוללת קוד סטטוס (לדוגמה 404 Not Found) וכן את המידע המבוקש.

הפרוטוקול פועל במודל בקשה תגובה (Request Response), שבו הלקוח יוזם את הבקשה והשרת מחזיר תגובה מתאימה. בפרויקט זה נעשה שימוש בפרוטוקול HTTP לצורך הדגמת תעבורת רשת פשוטה וברורה, שניתן לנתח בקלות באמצעות הכלי Wireshark.

HTTP פועל מעל פרוטוקול TCP בשכבת התעבורה, אשר אחראי על יצירת חיבור אמין בין הלקוח לשרת, על שמירת סדר הנתונים ועל שלמותם.

קובץ CSV:

כחלק מהפרויקט יצרנו קובץ נתונים בפורמט CSV בשם group212055024_fixed, אשר משמש כקלט ליצירת תעבורת הרשת.

הקובץ נוצר באופן ידני, במטרה לדמות הודעות בשכבת האפליקציה בצורה מבוקרת וברורה.

קובץ ה-CSV מייצג הודעות HTTP בשכבת האפליקציה לפני תהליך האריזה לשכבות הנמוכות יותר של מודל TCP/IP. כל שורה בקובץ מדמה הודעת HTTP אחת הנשלחת מהלקוח אל השרת.

מבנה הקובץ כולל את השדות הבאים: msg_id, app_protocol, src_app, dst_app, message timestamp

יצירה ידנית של הקובץ אפשרה לנו שליטה מלאה בתוכן ההודעות, בסדר שליחתן ובזמני השליחה, והקלה על ניתוח התעבורה שנוצרה בהמשך באמצעות Wireshark.

תהליך ה- Encapsulation והרצת המחברת Jupyter:

בשלב זה השתמשנו במחברת raw_tcp_ip_notebook.ipynb, על מנת לדמות את תהליך אריזת הנתונים (Encapsulation) כפי שהוא מתבצע במודל TCP/IP.

הקוד במחברת אינו שולח הודעות רשת אמיתיות, אלא מדמה את התהליך הלוגי של העברת הודעה בין שכבות הרשת, בדומה לאופן שבו מערכת ההפעלה מטפלת בנתונים בעת שליחתם.

תהליך ה- Encapsulation מתבצע באמצעות הוספת כותרות (Headers) בכל שכבה. כותרות אלו מכילות מידע טכני הנדרש לצורך ניתוב, זיהוי וטיפול נכון בהודעה על ידי המחשבים ברשת.

תהליך זה מדמה את אופן פעולת ה- Client היוצר את התעבורה, תעבורה שאותה ניתן לראות ולנתח לאחר מכן באמצעות הכלי Wireshark.

פירוט תהליך האריזה לפי שכבות:

שכבת היישום (Application Layer) - בשלב זה הקוד קורא את שדה ה- message מתוך קובץ ה- CSV ומשתמש בו כבסיס להודעה. לצורך הדגמה, ההודעה מיוצגת כ- Hello Packet או כהודעת HTTP פשוטה.

שכבת התעבורה (Transport Layer - TCP) - בשלב זה מתווסף TCP Header, הכולל בין היתר פורט מקור (Source Port), פורט יעד (Destination Port) ודגלים רלוונטיים. מידע זה מאפשר ניהול תקשורת אמינה בין הלקוח לשרת.

שכבת הרשת (Network Layer - IP) - בשלב זה מתווסף IP Header, המכיל את כתובות ה- IP של המקור והיעד. בפרויקט נעשה שימוש בכתובת Loopback (127.0.0.1), המדמה תקשורת מקומית בתוך אותו מחשב.

בסיום תהליך ה- Encapsulation ההודעה הארוזה נשלחת כסימולציית תעבורת רשת אותה ניתן ללכוד ולנתח באמצעות Wireshark.

תהליך הלכידה ב-Wireshark:

כדי ללכוד את התעבורה שנוצרה על ידי הסקריפט, הפעלנו את תוכנת Wireshark ונבחרנו ממשק Adapter for loopback traffic capture המאפשר לכידה של תעבורת Loopback, כלומר תקשורת מקומית המתבצעת בתוך אותו מחשב.

לאחר בחירת הממשק הגדרנו מסנן תצוגה: tcp.port == 12345, מסנן זה שימש להצגת התעבורה הרלוונטית לפרויקט בלבד.

לאחר מכן הרצנו את מחברת ה- Jupyter, והתעבורה נלכדה בזמן אמת. בסיום ההרצה שמרנו את הלכידה בקובץ פורמט pcap. לצורך ניתוח.

ניתוח תעבורה ב-Wireshark:

- התנועה שנדגמה ב-Wireshark משקפת תרחיש של "מבוי סתום" בתקשורת ה-TCP בין שני תהליכים מקומיים:

562	5.487409	127.0.0.1	127.0.0.1	TCP	44	12345 → 28402 [RST] Seq=1 Win=0 Len=0
563	5.591926	127.0.0.1	127.0.0.1	TCP	59	[TCP Retransmission] 28402 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=15
564	5.592108	127.0.0.1	127.0.0.1	TCP	44	12345 → 28402 [RST] Seq=1 Win=0 Len=0
565	5.699480	127.0.0.1	127.0.0.1	TCP	68	[TCP Retransmission] 28402 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=24
566	5.699633	127.0.0.1	127.0.0.1	TCP	44	12345 → 28402 [RST] Seq=1 Win=0 Len=0
567	5.801465	127.0.0.1	127.0.0.1	TCP	59	[TCP Retransmission] 28402 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=15
568	5.801512	127.0.0.1	127.0.0.1	TCP	44	12345 → 28402 [RST] Seq=1 Win=0 Len=0
569	5.904812	127.0.0.1	127.0.0.1	TCP	64	[TCP Retransmission] 28402 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=20
570	5.904913	127.0.0.1	127.0.0.1	TCP	44	12345 → 28402 [RST] Seq=1 Win=0 Len=0
571	6.007550	127.0.0.1	127.0.0.1	TCP	59	[TCP Retransmission] 28402 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=15
572	6.007638	127.0.0.1	127.0.0.1	TCP	44	12345 → 28402 [RST] Seq=1 Win=0 Len=0

מה רואים בתמונה?

- **הניסיון:** כתובת המקור (כתובת 127.0.0.1) מנסה לשלוח מידע מפורט 28402 לפורט 12345.
- **התוכן:** השולח מנסה להעביר פיסות מידע קטנות (בגודל 15, 20 או 24 בתים) ומשתמש בדגל **PSH** כדי לבקש מהצד השני לטפל במידע מיד.
- **החסימה (RST):** בכל פעם שנשלח מידע, הצד המקבל (פורט 12345) עונה מיד עם הודעת **RST**, המשמעות של RST היא שהחיבור סגור.
- **שליחה מחדש (Retransmission):** בגלל שהצד השולח לא מוותר, הוא מנסה לשלוח את אותו המידע שוב ושוב (מופיע כ-**TCP Retransmission**), ונדחה שוב ושוב.

למה זה קורה?

- אין לנו שרת ולכן אין תהליך שמאזין לפורט 12345, ולכן חוזרות הודעות RST.
- הלקוח לא מקבל סגירה מסודרת ולכן מנסה לבצע Retransmission, שליחת המידע שוב.

- **הודעת אישור** מהיעד אל הלקוח על חיבור מוצלח והעברת הודעות תקינה:

0000	02 00 00 00 45 00 00 37	00 01 00 00 40 06 7c beE..7....@.. .
0010	7f 00 00 01 7f 00 00 01	6e f2 30 39 00 00 00 00	...n...00
0020	00 00 00 00 50 18 20 00	3c d7 00 00 48 54 54 50	...P..<...HTTP
0030	2f 31 2e 31 20 32 30 30	20 4f 4b	/1.1 200 OK

- ניתוח הנתונים של החבילה:

```
Protocol: TCP (6)
Header Checksum: 0xcbe [validation disabled]
[Header checksum status: Unverified]
Source Address: 127.0.0.1
Destination Address: 127.0.0.1
[Stream index: 1]
Transmission Control Protocol, Src Port: 28402, Dst Port: 12345, Seq: 1, Ack: 1, Len: 15
Source Port: 28402
Destination Port: 12345
[Stream index: 7]
[Stream Packet Number: 11]
> [Conversation completeness: Incomplete (40)]
[TCP Segment Len: 15]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 0
[Next Sequence Number: 16 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment Number (raw): 0
0101 ... = Header Length: 20 bytes (5)
> Flags: 0x018 (PSH, ACK)
Window: 8192
[Calculated window size: 8192]
[Window size scaling factor: -1 (unknown)]
Checksum: 0x3cd7 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
> [Timestamps]
> [SEQ/ACK analysis]
[Client Continuous Streams: 0]
```

מה רואים בתמונה?

1. זהות השולח והמקבל

- **כתובות ה-IP:** גם כתובת המקור וגם כתובת היעד הן 127.0.0.1. זה אומר שהתקשורת מתבצעת בתוך המחשב.
- **פורט מקור (Source Port):** 28402 זהו הפורט שמתחיל את ההתקשרות.
- **פורט יעד (Destination Port):** 12345 זהו הפורט אליו נשלח המידע.

2. מצב התקשורת

- **Protocol:** מציינ את סוג הפרוטוקול בו הודעות עוברות - TCP
- **Flags:**
 - ◆ **ACK:** מאשר את קבלת המידע קודם.
 - ◆ **Push (PSH):** אומר למערכת להעביר את המידע ישר לאפליקציה ולא לחכות שהבאפר יתמלא, מאחר וזה צאט אנו רוצים שהעברת המידע תהיה מהירה ככל הניתן.

3. נתוני הרצף והמטען

- **Acknowledgment Number:** הוא 1, מה שמעיד על כך שהצד השולח מחכה לקבל את הביטוי הראשון מהצד השני.
- **TCP Payload:** גודל המידע שנשלח בחבילה הזו הוא 15 בתים.

