

Final Project

HTTP Server + DHCP + DNS

נכתב ע"י : דור ינאי, יבגני איוונוב

שרתים:

(1) `dhcp.py` – הוא פרוטוקול תקשורת המשמש להקצאה של כתובות IP ייחודיות למחשבים ברשת מקומית .
(יש הסברים על הקוד בתוך הקוד)

(2) `dns.py` – מחזיק `CACHE` שאם IP נמצא בו אז הוא שולח אותו ישירות ואם לא אז הוא בודק ע"י הפקודה `socket.gethostbyname` אשר שולחת בקשת DNS לגוגל כדי להשיג את האייפי, ואז מכניסה אותו ל `CACHE`. תפקיד השרת הינו לבדוק ולהחזיר את האייפי שממופה לכל כתובת דומיין שהמשתמש ייתן.(הסבר על הקוד נמצא בתוך הקוד).

(3) `Redirectserver.py` – שרת אשר פונים אליו בבקשת GET להוריד קובץ מסויים, יחזיר `response` 302 וכתובת חדשה עם המיקום העדכני של הקובץ, כי הקובץ אינו נמצא אצלו.

(4) `TCP_Fileserver.py` - השרת שבפועל מחזיק את הקבצים שהקליינט רוצה להוריד באמצעות פרוטוקול TCP.

(5) `RUDP_Fileserver.py` - השרת שבפועל מחזיק את הקבצים שהקליינט רוצה להוריד באמצעות פרוטוקול UDP משופר, שאנו קוראים לו RUDP.

קליינטים:

(1) `client.py` – הקליינט של הפרוייקט, מכיל בתוכו פונקציות אשר מתפעלות את כל השרתים.
כמו שליחת GET לשרת `redirect`, בקשה להורדת הקבצים מהמיקום החדש ובקשות DHCP ו DNS.

איך להריץ את הפרויקט:

להריץ את כל קבצי הקוד מהטרמינל (חלונות נפרדים) עם הפקודה `sudo python3 FILENAME.py` לשם לב שכאשר מריצים את שרת ה `dhcp` צריך להקליד את `network interfacen` .
לאחר מכן בחלון שהורץ ה `client.py` יופיעו אפשרויות באופן הבא :

```
dor@dor-VirtualBox: ~/Documents/NetComs/FinalProject-HTTPServerRD/HTTP Project
dor@dor-VirtualBox:~/Documents/NetComs/FinalProject-HTTPServerRD/HTTP Project$ sudo python3 client.py
[sudo] password for dor:
Enter number: 1 - DNS , 2 - DHCP , 3 - TCP-HTTP-APP , 4 - RUDP-HTTP-APP:
```

לאחר מכן אם נלחץ DNS המשתמש התבקש לשים דומיין ואז יקבל אייפי.
DHCP – משתמש יצטרך להכניס את `network interfacen` ואז כל השאר יבוצע אוטומטית.
ואם נבחר את האפשרויות של HTTP אז נתבקש לבחור איזה קובץ להוריד, תמונה או קובץ טקסט

```
dor@dor-VirtualBox: ~/Documents/NetComs/FinalProject-HTTPServerRD/HTTP Project
dor@dor-VirtualBox:~/Documents/NetComs/FinalProject-HTTPServerRD/HTTP Project$ sudo python3 client.py
[sudo] password for dor:
Enter number: 1 - DNS , 2 - DHCP , 3 - TCP-HTTP-APP , 4 - RUDP-HTTP-APP:
3
choose what to download
Enter number: 1 - Image, 2 - Text file, Any other input will QUIT:
```

WIRESHARK וטרמינל:

DNS: שם של הקובץ dns.pcap

The image displays two side-by-side screenshots. The left screenshot shows the Wireshark interface with a packet list on the left and a packet details pane on the right. The selected packet is a DNS query (Frame 14) from 10.0.2.15 to 208.67.222.222. The details pane shows the query for 'www.amazon.com' with transaction ID 0xdc89. The right screenshot shows a terminal window with the command 'python3 client.py' and its output, which includes the domain 'www.amazon.com' and IP '65.9.104.215'.

DHCP: שם של הקובץ dhcp.pcap

The image displays two side-by-side screenshots. The left screenshot shows the Wireshark interface with a packet list on the left and a packet details pane on the right. The selected packet is a DHCP Offer (Frame 3) from 10.0.2.3 to 127.0.0.5. The details pane shows the offer for IP address 127.0.0.5 with transaction ID 0xb1e5c38c. The right screenshot shows a terminal window with the command 'python3 client.py' and its output, which includes the IP address '127.0.0.5' and the message 'assigned IP:127.0.0.5'.

image tcp transfer.pcap & TCP-Redirect-HTTP שם של הקובץ

txt tcp transfer.pcap

תמונה:

Image_transfer.pcap						
tcp						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	74	33500 → 2746 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=1151248986 TSecr=0 WS=128
2	0.000033	127.0.0.1	127.0.0.1	TCP	74	2746 → 33500 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=1151248986 TSecr=1151248986 WS=128
3	0.000057	127.0.0.1	127.0.0.1	TCP	66	33500 → 2746 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1151248986 TSecr=1151248986
4	0.000102	127.0.0.1	127.0.0.1	HTTP	210	2746 → 33500 [PSH, ACK] Seq=1 Ack=182 Win=65536 Len=144 TSval=1151248987 TSecr=1151248987 [TCP segment of a reassembled...
5	0.000112	127.0.0.1	127.0.0.1	TCP	66	2746 → 33500 [ACK] Seq=1 Ack=182 Win=65488 Len=0 TSval=1151248986 TSecr=1151248986
6	0.000893	127.0.0.1	127.0.0.1	TCP	210	2746 → 33500 [PSH, ACK] Seq=1 Ack=182 Win=65536 Len=144 TSval=1151248987 TSecr=1151248987 [TCP segment of a reassembled...
7	0.000936	127.0.0.1	127.0.0.1	TCP	66	33500 → 2746 [ACK] Seq=182 Ack=145 Win=65488 Len=0 TSval=1151248987 TSecr=1151248987
8	0.001025	127.0.0.1	127.0.0.1	HTTP	66	HTTP/1.0 302 Found
9	0.001085	127.0.0.1	127.0.0.1	TCP	66	33500 → 2746 [FIN, ACK] Seq=182 Ack=146 Win=65536 Len=0 TSval=1151248988 TSecr=1151248987
10	0.001715	127.0.0.1	127.0.0.1	TCP	66	2746 → 33500 [ACK] Seq=146 Ack=183 Win=65536 Len=0 TSval=1151248988 TSecr=1151248988
11	0.003162	127.0.0.1	127.0.0.2	TCP	74	57938 → 2060 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=64392049 TSecr=0 WS=128
12	0.003182	127.0.0.1	127.0.0.2	TCP	74	2060 → 57938 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3969295584 TSecr=64392049 WS=128
13	0.003198	127.0.0.1	127.0.0.2	TCP	66	57938 → 2060 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=64392049 TSecr=3969295584
14	0.003248	127.0.0.1	127.0.0.2	TCP	161	57938 → 2060 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=95 TSval=64392049 TSecr=3969295584
15	0.003257	127.0.0.2	127.0.0.1	TCP	66	2060 → 57938 [ACK] Seq=1 Ack=96 Win=65488 Len=0 TSval=3969295584 TSecr=64392049

מה שברוק זה 302 response שזה redirect

```
Frame 8: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 2746, Dst Port: 33500, Seq: 145, Ack: 182
[2 Reassembled TCP Segments (144 bytes): #6(144), #8(0)]
Hypertext Transfer Protocol
  HTTP/1.0 302 Found\r\n
  Server: SimpleHTTP/0.6 Python/3.11.2\r\n
  Date: Sun, 26 Mar 2023 17:47:45 GMT\r\n
  Location: http://127.0.0.2:2060/image_tcp.jpg\r\n
  \r\n
```

Wireshark · Packet 14 · image_transfer.pcap

```
Frame 14: 161 bytes on wire (1288 bits), 161 bytes captured (1288 bits) on interface 0
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.2
Transmission Control Protocol, Src Port: 57938, Dst Port: 2060, Seq: 1, Ack: 1, Len: 95
  Source Port: 57938
  Destination Port: 2060
  [Stream index: 1]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 95]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 623466504
  [Next Sequence Number: 96 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 2433613806
  1000 .... = Header Length: 32 bytes (8)
  Flags: 0x018 (PSH, ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    0 .... = Congestion Window Reduced (CWR): Not set
    0000 00 00 00 00 00 00 00 00 00 00 08 00 45 00 .....E-
    0010 00 93 ca 93 40 00 40 06 71 ce 7f 00 00 01 7f 00 ...@. q .....
    0020 00 02 e2 52 08 0c 25 29 58 08 91 0d ff ee 80 18 ...R.% X .....
    0030 02 00 fe 88 00 00 01 01 08 0a 03 d6 8b 71 ec 96 .....q...
    0040 a4 e0 47 45 54 5f 46 49 4c 45 20 68 74 74 70 3a ..GET_FI LE http:
    0050 2f 2f 31 32 37 2e 30 2e 30 2e 32 3a 32 30 36 30 //127.0. 0.2:2060
    0060 2f 69 6d 61 67 65 5f 74 63 70 2e 6a 70 67 20 48 /image_t cp.jpg H
    0070 54 54 50 2f 31 2e 31 20 0d 0a 48 6f 73 74 3a 20 TTP/1.1 ..Host:
    0080 31 32 37 2e 30 2e 30 2e 32 20 0d 0a 43 6f 6e 6e 127.0.0. 2 ..Conn
    0090 65 63 74 69 6f 6e 3a 20 6f 70 65 6e 20 0d 0a 0d ection: open ...
    00a0 0a
```

בקשת הקובץ מהשרת השני.

איבוד פאקטות 15%pl.pcap: image_tcp_15%

Image_tcp_15%pl.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	74	51678 → 2746 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=1151901262 TSecr=0 WS=128
2	0.000030	127.0.0.1	127.0.0.1	TCP	74	2746 → 51678 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=1151901262 TSecr=1151901262 WS=128
3	0.000096	127.0.0.1	127.0.0.1	HTTP	247	GET /image_tcp.jpg HTTP/1.1
4	0.000118	127.0.0.1	127.0.0.1	TCP	66	2746 → 51678 [ACK] Seq=1 Ack=182 Win=65488 Len=0 TSval=1151901263 TSecr=1151901263
5	0.000790	127.0.0.1	127.0.0.1	TCP	210	2746 → 51678 [PSH, ACK] Seq=1 Ack=182 Win=65536 Len=144 TSval=1151901263 TSecr=1151901263 [TCP segment of a reassembled...
6	0.000811	127.0.0.1	127.0.0.1	TCP	66	51678 → 2746 [ACK] Seq=182 Ack=145 Win=65488 Len=0 TSval=1151901263 TSecr=1151901263
7	0.000893	127.0.0.1	127.0.0.1	HTTP	66	HTTP/1.0 302 Found
8	0.001516	127.0.0.1	127.0.0.1	TCP	66	51678 → 2746 [FIN, ACK] Seq=182 Ack=146 Win=65536 Len=0 TSval=1151901264 TSecr=1151901263
9	0.001541	127.0.0.1	127.0.0.1	TCP	66	2746 → 51678 [ACK] Seq=146 Ack=183 Win=65536 Len=0 TSval=1151901264 TSecr=1151901264
10	0.002861	127.0.0.1	127.0.0.2	TCP	74	46922 → 2868 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=65044325 TSecr=0 WS=128
11	0.002889	127.0.0.2	127.0.0.1	TCP	74	2868 → 46922 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3969947860 TSecr=65044325 WS=128
12	0.002895	127.0.0.1	127.0.0.2	TCP	66	46922 → 2868 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=65044325 TSecr=3969947860
13	0.002951	127.0.0.1	127.0.0.2	TCP	161	46922 → 2868 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=95 TSval=65044325 TSecr=3969947860
14	0.002963	127.0.0.2	127.0.0.1	TCP	66	2868 → 46922 [ACK] Seq=1 Ack=96 Win=65488 Len=0 TSval=3969947860 TSecr=65044325
15	0.003846	127.0.0.2	127.0.0.1	TCP	72	2868 → 46922 [PSH, ACK] Seq=1 Ack=96 Win=65536 Len=6 TSval=3969947861 TSecr=65044325

Frame 21: 15810 bytes on wire (126480 bits), 15810 bytes captured (126480 bits) on interface 0

Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)

Internet Protocol Version 4, Src: 127.0.0.2, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 2868, Dst Port: 46922, Seq: 65543, Ack: 96, Len: 15744

Data (15744 bytes)

אם נרד בהקלטה נראה את השליחה מחדש.

(1) מנה לפחות 4 הבדלים בין פרוטוקול TCP לפרוטוקול QUIC.

פתרון:

(א) יצירת חיבור: TCP דורש לחיצת יד תלת כיוונית כדי ליצור חיבור בין שתי נקודות קצה. תהליך זה כולל הודעת SYN שנשלחה על ידי נקודת הקצה היזומה, הודעת ACK בתגובה על ידי נקודת הקצה המקבלת, והודעת SYN-ACK סופית על ידי נקודת הקצה היוזמת כדי לאשר את החיבור. לעומת זאת, QUIC משתמש בלחיצת יד של זמן טיול אחד (RTT) כדי ליצור חיבור, מה שמפחית את ההשהיה והתקורה של יצירת חיבור.

(ב) אמינות: TCP הוא פרוטוקול אמין המבטיח שכל מנות הנתונים מועברות למקלט בסדר שנשלחו. אם חבילה אובדת, TCP ישדר אותה מחדש עד שהיא תתקבל. QUIC מציע גם אמינות, אך הוא משתמש במנגנון אחר. הוא מקבץ מספר מנות ליחידה אחת הנקראת מרחב מספרי מנות, ומשדר מחדש רק את היחידות שלא קיבלו אישור על ידי המקלט.

(ג) בקרת גודש: TCP משתמש באלגוריתם בקרת גודש כדי לנהל את כמות הנתונים הנשלחים דרך הרשת. זה מתחיל עם מספר קטן של מנות ומגדיל את המספר בהדרגה עד שהוא מזהה גודש, ואז מקטין את מספר החבילות. QUIC משתמש גם באלגוריתם בקרת גודש, אך הוא פועל ברמה גבוהה יותר מ-TCP, תוך התחשבות בגודש של מספר זרמים וחיבורים.

(ד) אבטחה: QUIC כולל הצפנה כחלק מהפרוטוקול שלו, בעוד ש-TCP לא. QUIC משתמש בפרוטוקול Transport Layer Security (TLS) כדי להצפין את כל הנתונים במעבר, מה שעוזר למנוע גישה לא מורשית או האזנה לתעבורת רשת. TCP יכול להיות מוצפן עם TLS, אבל זה דורש תצורה והגדרה נוספת.

(2) מנה לפחות 2 הבדלים עיקריים בין CUBIC ל-VEGAS

פתרון:

(א) גישה לבקרת גודש: Vegas ו-Cubic משתמשות בגישות שונות כדי להתמודד עם עומס ברשת. Vegas מודדת את זמן ההליכה הלוך ושוב (RTT) של מנות נתונים ומתאים את קצב השליחה בהתאם כדי למנוע עומס. לעומת זאת, Cubic משתמשת במודל של גודל חלון TCP כדי להעריך את רוחב הפס הזמין ולהתאים את קצב השליחה בהתאם.

(ב) התנהגות בתנאי רשת שונים: גם Vegas ו-Cubic מתנהגות אחרת בתנאי רשת שונים. Vegas מתאימה יותר לרשתות עם רוחב פס נמוך ועיכוב נמוך, בעוד ש-Cubic מיועד לרשתות מהירות ורוחב פס גבוה. וגאס מגיבה במהירות לשינויים בתנאי הרשת, בעוד ש-Cubic איטית יותר בתגובה אך יכולה להתמודד עם רשתות מוצרים גדולים יותר עם עיכוב רוחב פס בצורה יעילה יותר.

(3) הסבר מהו פרוטוקול BGP, במה הוא שונה מ-OSPF והאם הוא עובד על פי מסלולים קצרים? פתרון:

(א) BGP (Border Gateway Protocol) זהו פרוטוקול ניתוב אשר בעזרתו מאפשרים לנתבים לתקשר ביניהם. פרוטוקול זה אחראי על מציאת הדרך היעילה ביותר להעברת נתונים בין הרשתות.

(ב) פרוטוקול זה שונה מ-OSPF בכך ש-BGP זהו פרוטוקול אשר משמש לניתוב בין רשתות רבות ושונות, ו-OSPF זהו פרוטוקול אשר משמש לניתוב בין רשת יחידה.

(ג) פרוטוקול BGP יכול להשתמש במסלולים קצרים, הוא לא תמיד ייקח את המסלול הקצר ביותר כי לא תמיד זה הכי יעיל, אבל הוא בהחלט יכול לקחת את מסלול זה אם הוא היעיל ביותר.

Application	Port src	Port Des	Ip Src	Ip Des	Mac Src	Mac Des
DNS		depends what port is open for the client ATM	127.0.0.1	127.0.0.1	0	0
DNS-Client	depends what port is open for the client ATM	53	127.0.0.1	127.0.0.1	0	0
DHCP		68	127.0.0.1	127.0.0.1	dhcp mac	client mac
DHCP-Client		67	127.0.0.1	127.0.0.1	client mac	dhcp mac
HTTP-Client-To-Redirect	depends what port is open for the client ATM	2746	127.0.0.1	127.0.0.1	0	0
Redirectserver-to-client		2746	127.0.0.2	127.0.0.1	0	0
HTTP-Client-To-TCPFileserver	depends what port is open for the client ATM	2060	127.0.0.1	127.0.0.2	0	0
TCP-Fileserver-To-Client		2060	127.0.0.2	127.0.0.1	0	0
HTTP-Client-To-RUDPFileserver	depends what port is open for the client ATM	3247	127.0.0.1	127.0.0.3	0	0
RUDP-FileServer-To-Client		3247	127.0.0.3	127.0.0.1	0	0

(4) ההבדלים העיקריים בין ARP ל-DNS הם:

(א) DNS ממפה דומיינים לכתובות IP ו-ARP ממפה כתובות IP לכתובות MAC.
 (ב) DNS עובד על שכבת האפליקציה, ו-ARP עובד על שכבת הקו (DATA LINK).
 (ג) ARP משמש ב-DNS ו-LAN ברחבי האינטרנט

(5)