Final Project HTTP Server + DHCP + DNS

נכתב ע"י : דור ינאי, יבגני איוונוב

שרתים:

- . הוא פרוטוקול תקשורת המשמש להקצאה של כתובות IP ייחודיות למחשבים ברשת מקומית dhcp.py (1 (יש הסברים על הקוד בתוך הקוד)
- שאם IP שאם לא אז הוא בודק ע"י dns.py (2 ממצא בו אז הוא שולח אותו ישירות ואם לא אז הוא בודק ע"י cACHE אשר שולחת בקשת DNS אשר שולחת בקשת socket.gethostbyname הפקודה מכניסה אותו ל CACHE. תפקיד השרת הינו לבדוק ולהחזיר את האייפי שממופה לכל כתובת דומיין שהמשתמש ייתן.(הסבר על הקוד נמצא בתוך הקוד).
- response שרת אשר פונים אליו בבקשת GET שרת אשר פונים אליו בבקשח Redirectserver.py (3 מסויים, יחזיר 302 וכתובת חדשה עם המיקום העדכני של הקובץ, כי הקובץ אינו נמצא אצלו.
 - השרת שבפועל מחזיק את הקבצים שהקליינט רוצה להוריד באמצעות TCP_Fileserver.py (4 פרוטוקול TCP.
 - השרת שבפועל מחזיק את הקבצים שהקליינט רוצה להוריד באמצעות RUDP_Fileserver.py (5 משופר, שאנו קוראים לו RUDP.

:קליינטים

client.py (1 – 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:

sudo python3 FILENAME.py להריץ את כל קבצי הקוד מהטרמינל (חלונות נפרדים) עם הפקודה onetwork interface. לשים לב שכאשר מריצים את שרת ה

: יופיעו אפשרויות באופן הבא 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.

. ואז כל השאר יבוצע אוטומטית network interface שתמש יצטרך להכניס את – DHCP

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 Q = - - ×

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:
```

:וטרמינל WIRESHARK

שם של הקובץ dns.pcap: DNS



שם של הקובץ DHCP:

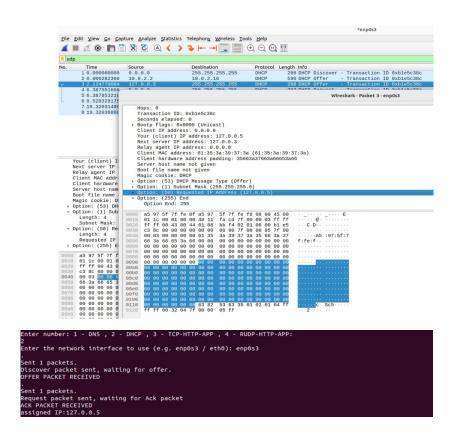
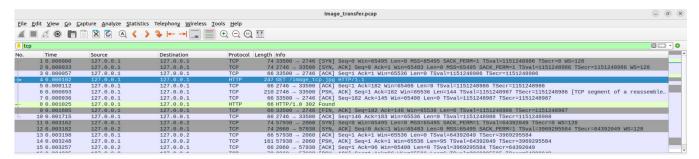


image tcp transfar.pcap && שם של הקובץ:TCP-Redirect-HTTP txt tcp transfer.pcap

<u>תמונה:</u>



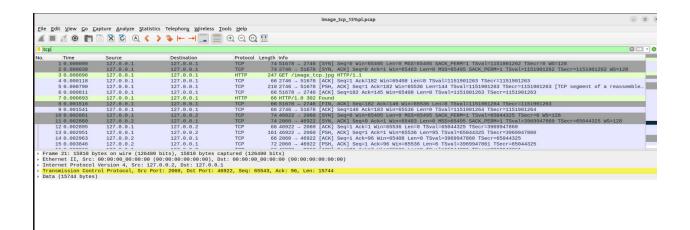
redirecta שזה response 302 מה שבירוק זה

```
Frame 8: bb bytes on wire (528 bits), bb bytes captured (528 bits)
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00), Dst: 00:00:00_00:00:00 (0)
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
```

```
Wireshark · Packet 14 · image_transfer.pcap
> Frame 14: 161 bytes on wire (1288 bits), 161 bytes captured (1288 bits)
> Ethernet II, Src: 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
                     = Congestion Window Reduced (CWR). Not set
      00 00 00 00 00 00 00
                              00 00 00 00 08 00 45 00
0010
      00 93 ca 93 40 00 40 06
                              71 ce 7f 00 00 01 7f 00
                                                         00 02 e2 52 08 0c 25 29
                              58 08 91 0d ff ee 80 18
                                                         · · · R · · %) X · · · · · ·
     02 00 fe 88 00 00 01 01
                              08 0a 03 d6 8b 71 ec 96
0030
                              4c 45 20 68 74 74 70 3a
     a4 e0 47 45 54 5f 46 49
                                                         .. GET FI LE http:
     2f 2f 31 32 37 2e 30 2e
                              30 2e 32 3a 32 30 36 30
                                                         //127.0. 0.2:2060
      2f 69 6d 61 67 65 5f 74
                              63 70 2e 6a 70 67 20 48
                                                         /image_t cp.jpg H
                                                         TTP/1.1 ·· Host:
0070
     54 54 50 2f 31 2e 31 20
                              0d 0a 48 6f 73 74 3a 20
0080
     31 32 37 2e 30 2e 30 2e
                              32 20 0d 0a 43 6f 6e 6e
                                                         127.0.0. 2 -- Conn
     65 63 74 69 6f 6e 3a 20 6f 70 65 6e 20 0d 0a 0d
                                                         ection: open -
00a0
     0a
```

בקשת הקובץ מהשרת השני.

image_tcp_15%pl.pcap: 15% איבוד פאקטות



אם נרד בהקלטה נראה את השליחה מחדש.

1) מנה לפחות 4 הבדלים בין פרוטוקול TCP לפרוטוקול 1

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 SYN 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 Gateaway Protocol זהו פרוטוקול ניתוב אשר בעזרתו מאפשרים לנתבים BGP(Border Gateaway Protocol) לתקשר ביניהם. פרוטוקול זה אחראי על מציאת הדרך היעילה ביותר להעברת נתונים בין הרשתות.
 - ברוטוקול זה שונה מ OSPF בכך שBGP זהו פרוטוקול אשר משומש לניתוב בין רשתות OSPF בכות ושונות, ו OSPF זהו פרוטוקול אשר משומש 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.

				1		
Application	Port src	Port Des	Ip Src 💌	Ip Des 🔼	Mac Src 🐣	Mac Des <u></u>
DNS	53	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	67	127.0.0.1	127.0.0.1	dhcp mac	client mac
DHCP-Client	67	68	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	depends what port is open for the client ATM	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	depends what port is open for the client ATM	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	depends what port is open for the client ATM	127.0.0.3	127.0.0.1	0	0,

5) ההבדלים העיקריים בין ARP ל DNS

א)DNS ממפה דומיינים לכתובות IP ו ARP ממפה כתובות IP לכתובות DNS (ב) ממפה דומיינים לכתובות ARP ו ARP עובד על שכבת הקו(DATA LINK). ב) DNS משומש בDNS ובחבי האינטרנט