Networks - Assignment 4

By:
Daniel Appel - 207386699
Yuval Ben Simhon - 318916335

חלק א

1. **הציגו יתרון אחד לשימוש ב-DoH והסבירו אותו (כמובן, מעבר לעובדה שהוא מאובטח ומוצפן)**

היתרון: DoH משתמש בTCP

הסבר:

לעומת הDNS שנשלח ע"י UDP, הDoH נשלח על גבי TCP , כפי שלמדנו הTCP אמנם איטי יותר אבל המידע שמתקבל יהיה אמין יותר, ויש פחות סיכוי שמידע כלשהו יאבד בדרך.

בנוסף לכך (ולעומת זאת), עקב העובדה שה DoH משתמש ב port 443, ישנה חלוקה בעומסים בין ה HTTP הרגיל לבין ה DoH, ולכן ניתן לתפעל יותר requests בו זמנית על אותו שרת.

2. **הציגו והסבירו על שני חסרונות לשימוש בשיטת DoH לעומת DNS הרגיל.**

שני חסרונות:

1. איטי יותר -

א.TCP: לעומת הDNS שנשלח ע"י UDP, הDoH נשלח על גבי TCP וכידוע פרוטוקול הTCP איטי יותר מפרוטוקול הUDP.

ב. הצפנה: השימוש בהצפנה ב DoH מאטה את העברת המידע.
קראנו של- DoH לוקח הכי הרבה זמן לטעון כל עמוד מבין כל שאר פרוטוקולי הצפנת ה-DNS(למרות שההבדל בין זמני טעינה מוצפנים ללא מוצפנים הוא די קלוש).

2. הצפנת יתר-

'יעילות וחוסר יעילות'- הצפנת (DoH) DNS מפרה 'חוקי המעקב' שמדינה זכאית להם.
למשל, באנגליה רצו להשתמש בDNS כדי לוודא את הגיל של הצופים בתוכן שמיועד למבוגרים, אבל השימוש ב DoH יכול לגרום למעקב להיות בלתי אפשרי.
לכן בכמה מדינות נאסר השימוש בDoH .

3. בחרו אחד מהחסרונות משאלה (2), הציעו דרך למתן\לעקוף\לפתור חיסרון זה והסבירו אותה.

פתרון אפשרי לבעיית ההצפנת יתר הוא שה DNS SERVERS של המדינה ידרשו מהמשתמש IDENTIFIER מוצפן כלשהו על מנת לאפשר לגורמים מדיניים את המעקב הדרוש, אך מאפשרים בכל זאת את האבטחת מידע ש DoH מציע ומאפשר.
כנראה, שהוספה כזו תגרום להאטה בהתהליך כולו, אך זו דרך אפשרית לפתרון הבעיה.

4. **ישנן 4 דרכים בהן ניתן לשלב את שיטת ה-DoH באינטרנט שלנו:**

1. **מימוש DoH ברמת האפליקציות (לדוגמא: לעדכן את קוד הדפדפן כך שישלחו שאילתות דרך HTTPS)**
2. **מימוש DoH ברמת שרת proxy* ברשת (מהמחשב לשרת נשלח לפורט 53 והלאה, כבר 443)**
3. **מימוש DoH ברמת שרת proxy מקומי (על המכונה רץ שרת proxy)**
4. **התקנת plugin המממש DoH ברמת הגדרות המחשב ("מעכשיו, אתה שולח רק DoH")**

כתבו השוואה בין כל ארבעת השיטות, בהשוואתכם הראו יתרונות וחסרונות לכל שיטה והציגו מהי, לדעתכם, השיטה המועדפת מבין הארבעה. כלומר, הציגו את השיטה בה, לדעתכם, היתרונות הגדולים ביותר לעומת החסרונות הקטנים ביותר.

מימוש DoH	רמת האפליקציות	רמת שרת proxy ברשת	רמת שרת proxy מקומי	התקנת plugin
מימוש ראשוני	חסרון (יחסי) - נצטרך לממש בצורה יחסית ייחודית לכל דפדפן בנפרד. דבר זה עלול להימשך זמן רב ולא כל דפדפן יעשה זאת. לעומת זאת, כמות המימושים (הדפדפנים) היא לא גדולה מאוד, לכן יכול להתבצע.	יתרון - מימוש יחיד לשרת יחיד (או קבוצה של שרתים עם אותו מימוש), ידרוש משאבים מינימליים של מתפעלי השרת.	יתרון - מימוש יחיד שמתפרסם לרוב המחשבים	יתרון - מימוש יחיד שמתפרסם לרוב המחשבים
עדכון	חסרון (יחסי)- כאשר המימוש יהיה ברמת האפליקציות - למשל עדכון קוד הדפדפן- נצטרך לעבור על כל דפדפן ,לגשת לקוד שלו ולשנות את הקוד	יתרון - כל עדכון יצטרך להתבצע מבחינה כמותי כמספר השרתים, ומכיוון שה source code היה כמעט זהה, העדכון יהיה כמעט יחיד.	חסרון-כאשר המימוש יהיה עי התקנת plugin על המחשב, ונרצה לבצע עדכון, נצטרך לעבור על כל מחשב עליו מימשנו את הDoH ולשנות את	חסרון-כאשר המימוש יהיה עי התקנת plugin על המחשב, ונרצה לבצע עדכון, נצטרך לעבור על כל מחשב עליו מימשנו את הDoH ולשנות את

	בהתאם.דבר זה יכול לקחת זמן רב ודורש הרבה משאבים. לעומת זאת, מות המימושים (הדפדפנים) היא לא גדולה מאוד, לכן יכול להתבצע.		הקוד בהתאם.דבר זה יכול לקחת זמן רב ודורש הרבה משאבים.	הקוד בהתאם.דבר זה יכול לקחת זמן רב ודורש הרבה משאבים.
תפיסת זיכרון	יתרון- המימוש יהיה בשרת של החברה של הדפדפן ולכן גם הזיכרון שיוקצה למימוש יהיה שם ולא מהמחשב.	יתרון-המימוש יהיה על שרת ברשת ולכן הקצאת הזיכרון תיהיה מהרשת	חסרון-כאשר המימוש יהיה על המחשב, הזיכרון שיוקצה לכך יהיה מהמחשב.	חסרון-כאשר המימוש יהיה על המחשב, הזיכרון שיוקצה לכך יהיה מהמחשב.
זמן ריצה של כל request	יתרון- תוסף ה DoH מותקן על הדפדפן יגרור זמן ריצה מהיר של השאילתא.	חסרון-זמן ריצה של שאילתא יצטרך לעבור מסלול ארוך יותר - לעבור מהרשת שלנו לשרת proxy ברש אחרת והוא יעביר הלאה את השאילתא, יקבל תשובה ויחזיר בחזרה אלינו.	חסרון-זמן ריצה של שאילתא יצטרך לעבור מסלול ארוך יותר - לעבור מהרשת שלנו לשרת proxy (אומנם באותה רשת-כלומר יותר מהיר משרת proxy ברשת אבל עדיין איטי יחסית לשאר המימושים המקומיים) והוא יעביר הלאה את השאילתא, יקבל תשובה ויחזיר בחזרה אלינו.	יתרון- תוסף ה DoH מותקן על המחשב יגרור זמן ריצה מהיר של השאילתא.

מבחינת המימוש הטוב ביותר- נתייחס לשלושה צדדים:

1. לקוח
2. מפתח
3. גם לקוח וגם מפתח

מבחינת לקוח - המימוש הטוב ביותר עבורינו יהיה המימוש ברמת האפליקציות- הן מבחינת נוחות -התקנה ועדכון יעשו ללא התערבות הלקוח (או התערבות מינימלית), והן מבחינת זמן ריצה מהיר.

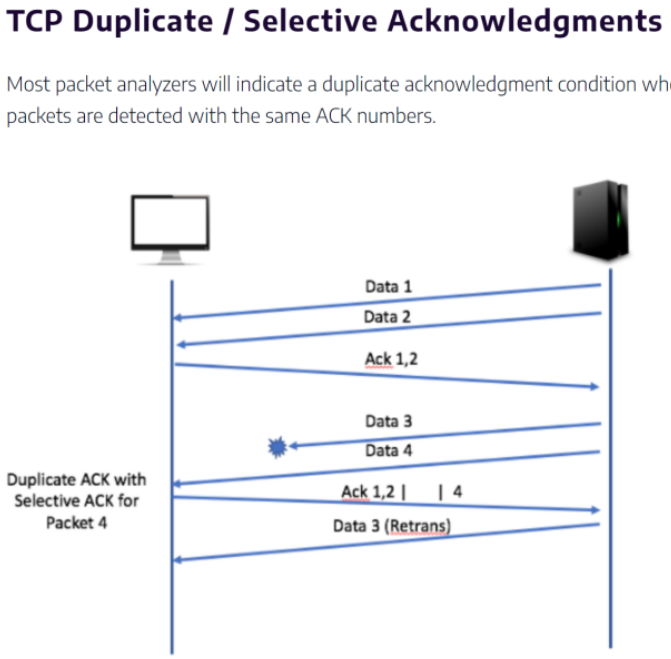
מבחינת מפתח- המימוש הטוב ביותר עבורינו יהיה proxy ברשת- המימוש הראשוני יהיה אחיד וכנ"ל לגבי העדכון, והזיכרון יוקצה מהרשת .

שילוב: לדעתינו, המימוש האופטימאלי מבחינת הלקוח והמפתח ביחד הוא שילוב בין המימוש ברמת האפליקציה למימוש ב proxy ברשת. כל דפדפן שיכול להרשות לעצמו את המשאבים לשינוי הגישה ירוויח מהמימוש ברמת האפליקציה, וכך הלקוחות יהנו מהגלישה המהירה. דפדפנים שלא יכולים להרשות לעצם את המשאבים, יעדיפו להשתמש ב proxy ברשת שיאפשר גלישה קלה (מבחינת מאמצים) ללקוחות, אך שכל request יקח יותר זמן להתבצע.

5. **נניח שאנו ברשת שקיים בה איבוד פקטות (packet loss) באחוז לא ידוע ואנו רוצים לטעון דף שצריך 25 שאילתות כדי לבקש את כל המשאבים שבו. הציגו יתרון ברור שיש ל-DoH לעומת Do53. (רמז: מנגנון הקיים ב-TCP)**

DO53 עובד עם פרוטוקול UDP לעומת DoH שעובד עם פרוטוקול TCP.
אחד המנגנונים הקיימים ב TCP (<u>ולא ב UDP</u>) הוא מנגנון Duplicate/Selective Acknowledgments - שלפיו כאשר מחשב שולח packet הוא מקבל תגובה ייחודית בחזרה שהתקבל ה- packet הספציפית הזה, ולכן כאשר תהיה packet שלא הגיע, לא נקבל תשובה בחזרה עבור ה-packet הזה, וכך אפשר לדעת במקרה שלנו למשל על איבוד פאקטות - ע"י כך שנקבל \ לא נקבל את כל ה-25 תגובות.

תמונה להמחשה:



TCP Duplicate / Selective Acknowledgments

חלק ב

	mean QUBIQ	mean RENO
0%	0.000062	0.0001092
10%	0.0000744	0.0001286
15%	0.0002394	0.0002744
20%	0.0001	0.0001848
25%	0.0003148	0.0000996
30%	0.0000848	0.0003944

פקודות ההרצה לשינוי כמות איבוד פאקטות:

```
appeldaniel@ubuntu: ~
Reading package lists... Done
Building dependency tree
Reading state information... Done
iproute2 is already the newest version (5.5.0-1ubuntu1).
iproute2 set to manually installed.
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 40 not upgraded.
appeldaniel@ubuntu:~$ sudo tc qdisk add dev lo root netem loss 10%
Object "qdisk" is unknown, try "tc help".
appeldaniel@ubuntu:~$ sudo tc qdisc add dev lo root netem loss 10%
appeldaniel@ubuntu:~$ sudo tc qdisc change dev lo root netem loss 15%
appeldaniel@ubuntu:~$ sudo tc qdisc del dev lo root netem
appeldaniel@ubuntu:~$ sudo tc qdisc add dev lo root netem loss 10%
appeldaniel@ubuntu:~$ sudo tc qdisc change dev lo root netem loss 15%
appeldaniel@ubuntu:~$ sudo tc qdisc change dev lo root netem loss 20%
appeldaniel@ubuntu:~$ sudo tc qdisc change dev lo root netem loss 25%
appeldaniel@ubuntu:~$ sudo tc qdisc change dev lo root netem loss 30%
appeldaniel@ubuntu:~$ sudo tc qdisc del dev lo root netem
[sudo] password for appeldaniel:
appeldaniel@ubuntu:~$
```

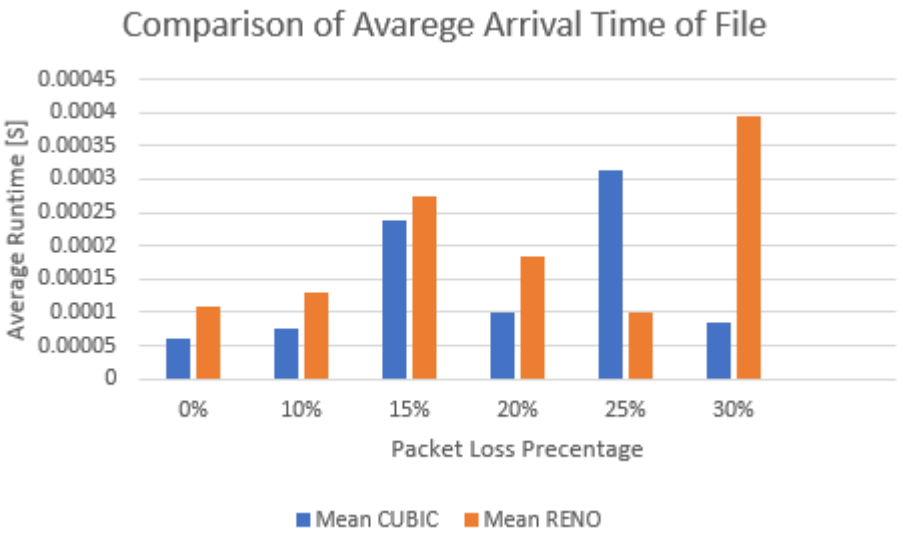
דוגמאט להרצת הServer:

```
Run: untitled
/home/appeldaniel/CLionProjects/untitled/cmake-build-debug/untitled
Bind() success
Waiting for incoming TCP-connections...
A new client connection accepted
File Opened
File Closed
message was successfully sent .
File 2 sent
A new client connection accepted
File Opened
File Closed
message was successfully sent .
File 2 sent
A new client connection accepted
File Opened
File Closed
message was successfully sent .
File 2 sent
A new client connection accepted
```

דוגמאט להרצת הClient:

```
untitled1
/home/appeldaniel/CLionProjects/untitled1/cmake-build-debug/untitled1
connected to server
file successfully received
Time taken: 0.000087
Process finished with exit code 0
```

השוואה בין זמני הגעת הקבצים:



מסקנות:

היינו מצפים שעם גדילת אחוזי איבוד הפאקטות, זמני ההגעה של הממוצעים של הקובץ היו גדלים, אך במקרה שלנו זה לא קרה.

אנו יכולים לשער כי הסיבה לכך היא שהמחשבים שעליהם הרצנו את התוכניות היו קרובים מאוד אחד לשני (באותה רשת פנימית), וגם כי הקובץ היה לא מספיק גדול כדי לתפוס את השינויים.

אנו מצפים שאם נגדיל את הקובץ, נוכל לראות את השינויים בצורה בולטת הרבה יותר.

כמו כן, בבדיקה זאת לא ראינו הבדל משמעותי בין שני השיטות ל- Congestion Control מבחינת זמני הריצה, לכן אין אנו יכולים להעיד על יעילות של שיטה אחת מעבר לשנייה (ניתן לראות שזמני הגעת הקבצים ב-Reno היו ארוכים יותר אך השינוי בינו לבין ה- Cubic היה מינורי).

Source links:
https://www.digitalwhisper.co.il/files/Zines/0x7F/DW127-4-DNSOverHTTPS.pdf

https://blog.apnic.net/2018/10/12/doh-dns-over-https-explained/

https://resources.infosecinstitute.com/topic/dns-over-https-doh/

https://securityboulevard.com/2022/05/dns-over-https-facts-you-should-know/#:~:text=This%20way%2C%20DNS%20over%20HTTPS%20(DoH)%20shields%20the%20users,web%20browsers%20is%20totally%20encrypted

https://accedian.com/blog/network-packet-loss-retransmissions-and-duplicate-acknowledgements/