**מטלה 3 – חלק א**

1) היתרון בשימוש ב DoH הוא ש DoH מבוסס על TCP ולכן אני יודע בוודאות שהבקשה שלי התקבלה בזכות הhandshake וגם יש לי בקרה למקרה ונוצר שגיאות במהלך קבלת המידע חזרה בזכות הchecksum.

2) אחד החסרונות של DoH הוא שהיות והוא פרוטוקול יחסית חדש (קיים רק שנתיים) אז כרגע אין תמיכה רבה שלו בהרבה מערכות הפעלה ויקח זמן עד שתהיה לו תמיכה רחבה, לעומת DNS שקיים כבר 20+ שנים ויש לו תמיכה רבה בכל המערכות ההפעלה.

חיסרון שני הוא ששימוש ב DoHמקשה על חברות אבטחת מידע (cybersecurity) לנטר ולנתח את התעבורה על מנת לזהות תוקפים או דברים חשודים. היות והם לא יכולים להסתכל על הheaders שאומרים לאן מישהו מעוניין להגיע מכיוון שהם מוצפנים.

וזה בניגוד לDNS אשר בו הheaders האלה חשופים, גם לתוקפים פוטנציאליים אך גם לחברות אבטחת מידע (cybersecurity) שמאפשר להם לזהות ולמנוע תוקפים פוטנציאליים.

3) DoH כרגע אינו מספק אבטחה מקסימלית מכיוון שלא כולם תומכים בDoH, כלומר האבטחה אינה end-to-end אלא כרגע מספקת אבטחה שמבוססת על hops בין שאילתא לשאילתא (בעיקר בנקודות בהם אאין כרגע תמיכה של DoH).

לכן, עד שתהיה תמיכה מלאה של DoH ועד שימצא פיתרון יותר קבוע, חברות האבטחת מידע יכולות לנטר לפחות בנתיים על התעבורה דרך הweak links בשרשרת התקשורת אשר אינן תומכות בDoH על מנת לנטר על התעבורה נגד תוקפים פוטנציאליים.

4) היתרון של שיטה a היא שיישום של DoH ברמת האפליקציה תוודא שבמידה ויש תמיכה בDoH אז השאילתות שיצאו מרמת האפליקציה בוודאות יהיו מאובטחות.

החיסרון של שיטה a היא שזה דורש ממתכנתים לכתוב אפליקציות שיתמכו בDoH וישתמשו בו בעת שליחת שאילתות DNS, וזה בעייתי כי אז האחריות הזאת נופלת על מספר מצומצם של אנשים.

ובנוסף זה **אולי** עשוי לדרוש טיפה יותר מבחינת מהמערכת לפני שליחת בקשת הDoH.

היתרונות של שיטה b הם ששרת הפרוקסי היא שהשרת יכול לשלוח בקשות DoH עבורי במידה והמערכת שלי אינה תומכת בה ופעולה זאת אינה דורשת יותר משאבים מהמערכת שלי.

ויתרון נוסף הוא שזה מקל על האדם הממוצע להעזר בגורם חיצוני על מנת לאבטח את הבקשות DNS שלו.

החיסרון של שיטה b היא שאם מישהו יושב על הרשת שלי עם packet sniffer אז למרות שהבקשה שלי תצא לאינטרנט מאובטחת, התוקף הפוטנציאלי עדיין יכול לראות לאן אני מנסה לגשת בplain text וזה קצת מביס את הרעיון של DoH, למרות שזה אכן מצמצם את הסיכוי שדבר כזה יקרה.

חיסרון נוסף הוא שזה עדיין לא הכי נגיש לאדם הממוצע ודורש מעט ידע על מנת לדעת להגדיר את השרת פרוקסי בתור הdefault.

היתרון של שיטה c היא שבניגוד לשיטה b, פה, אם מישהו ינסה לראות לאן אני מעונין לגשת, הוא לא יוכל מכיוון שהבקשה יוצאת מהשרת פרוקסי שיושב לי על המחשב בצורה מאובטחת בצורת DoH.

החיסרון העיקרי של שיטה זו הוא שיש צורך להקים את השרת פרוקסי על המחשב והדבר הזה דורש יותר משאבים להקים ולתחזק מאשר שיטה b.

חיסרון הנוסף הוא שעל מנת להקים פרוקסי מקומי דורש ידע יותר מעמיק מאשר קיים אצל האדם הממוצע ולכן רק אנשים שמעוניינים באבטחת מידע או למדו מחשבים יוכלו לבצע דבר כזה.

היתרונות של שיטה d הם שזה לא דורש יותר משאבים מהמחשב על מנת לשלוח בקשות DoH וגם ליותר אנשים יש את היכולת לקחת את האבטחה בידיים ולהשתמש ב DoHעל מנת לשלוח בקשות מאובטחות.

החיסרון הוא שייתכן ויש possible backdoor בplugin שמפר את כל הרעיון של האבטחה של DoH.

חיסרון נוסף אפשרי (שלא הובהר בשאלה האם קיים או לא) אבל אם הplugin שולח אך ורק DoH ואינו חוזר לdefault של DNS במידה אין תמיכה של DoH, אז זה עשוי לגרום לבעיות תקשורת בעת בקשת אתרים.

לדעתי השיטה הטובה ביותר היא שיטה d מכיוון שתוסף מנגיש את השימוש של DoH לכל האנשים, אינו דורש משאבים מהמערכת וגם יש בקרה חיצונית כדוגמת google web store לגבי האופי של התוספים שנמצאים בה.

5) היתרון של שימוש ב DoH לעומת DNS רגיל הוא שDNS עובד על בסיס UDP אשר פשוט מקבל בקשה ושולח את כל המידע ללא שום בקרה שהכל התקבל אבל DoH שעובד על בסיס TCP מאפשר handshake בעת יצירת הקשר, מה שמאפשר שליחת הודעות ACK אשר מהוות ביקורת שכל החבילות עד אותה נקודה התקבלו כמו שצריך ואם לא, אז ישלח הACK האחרון שעד אליו התקבלו כל החבילות.

אם התקבלו 3 duplicate ACK אז TCP יניח שהסגמנט אבד ויבצע שידור מחדש.