איום 1	ניצול שטח גדול מזיכרון השרת ע"מ לנטרל את פעולתו
רכיבים מושפעים	שליחת קובץ לשרת, ושמירת הקובץ בזיכרון מקומי בשרת
סוג החולשה	Denial-of-service attack
הסבר	הפרוטוקול מאפשר ללקוחות שליחת של קבצים, וכחלק מהפרוטוקול השרת שומר את הקובץ הנשלח כקובץ מקומי. ע"פ הפרוטוקול יש 4 בתים לתיאור גודל תוכן הקובץ כלומר ניתן לשלוח כול פעם קובץ בגודל עד 4 גיגה. משתמש זדוני יכול לשלוח כמות בלתי מוגבלת (תאורטית) של קבצים (עד 4 גיגה לכול קובץ) בשמות שונים ובכך להעמיס את זיכרון השרת.
תוצאה	העמסת זיכרון השרת ע"י קבצים של לקוח או לקוחת, דבר זה יכול לגרום להאטה בקצב פעולת השרת ולבסוף לנטרול פעולת השרת.
דרישות	אפשרות לשליחת קבצים לשרת, דבר זה ניתן כחלק מהפרוטוקול.
השפעה	האטה בפעולת השרת, או גרימה לקריסתו.
פתרון מוצע	להגביל את מספר נתוני הקבצים שנשמרים בשרת לכול לקוח, והגבלה כללית של מספר הקבצים הנשמרים בשרת לכול הלקוחות יחד, כאשר מגיעים לחסם ההגבלה ניתן למחוק קבצים ישנים (ניתן לדעת זאת ע"י בדיקת last seen של כול לקוח) או להתריע ללקוח שעבר את מכסת הקבצים המותרת וחלק מקבציו ימחקו.

איום 2	עקיפת מנגנוני ההזדהות בפרוטוקול והתחזות ללקוח (או לשרת)
רכיבים מושפעים	פעולות הקשורות בהחלפת מפתחות ובשליחת קבצים
סוג החולשה	Man in the middle
הסבר	כאשר משתמש זדוני מאזין לתווך התקשורת בין הלקוח והשרת, הוא יכול לגנוב מידע שנשלח בצורה לא מוצפנת, לדוגמה client id שנשלח לא מוצפן כחלק מה header בכול בקשה (פרט לרישום לקוח חדש) לשרת או תגובה מהשרת. ע"י מידע זה ניתן להשיג מידע על בקשות ששולח הלקוח לשרת או על התגובות מהשרת ללקוח הספציפי שעליו הושג המידע, המשתמש הזדוני יכול גם להתחזות ללקוח או להתחזות לשרת.
תוצאה	המשתמש הזדוני יכול להזדהות בתור הלקוח בכל אחת מהפעולות הנ"ל שכוללות הזדהות, הוא יכול לשלוח קבצים לשרת בשם הלקוח (וכך לדרוס קבצים קיימים), לאמת אותם או למחוק אותם בשמו, ולהחליף את מפתחות ההצפנה בצורה שתמנע מהלקוח שימוש בשירות. מעבר לכך, במידה והמשתמש הזדוני מצליח להתחזות לשרת, הוא יוכל לתעל את התעבורה דרכו, לדלות ולהחליף מפתחות הצפנה, שיסייעו לו לפענח את המידע המוצפן בפרוטוקול.
דרישות	אפשרות להאזין לתווך התקשורת בין הלקוח והשרת.
השפעה	התחזות ללקוחות שונים, שיבוש פרוטוקול התקשורת בעת ביצוע רצף פעולות. באופן פוטנציאלי – אף עקיפה מוחלטת של תהליך ההצפנה בפרוטוקול והשגת גישה לקובץ מוצפן בהעברתו.
פתרון מוצע	הצפנה מוחלטת של כל התעבורה כולה בין הלקוח לבין השרת, כולל כל שדות הפרוטוקול – כדוגמת TLS/SSL. הצפנה זו תמנע השגה של כל פרט פגיע פוטנציאלי אודות הלקוח, ובכך תמנע כל התחזות אליו או השגה של פרטים בנוגע לפעולות שהוא מבצע או הקבצים שהוא מעביר. שימוש בפרוטוקול TLS עם ביסוס Chain of Trust יצמצם את פוטנציאל ההתחזות לשרת על ידי משתמש זדוני.

|
הפרוטוקול תקשורת שרת - לקוח שמוצע מאפשר ללקוחות להעביר קבצים באופן מוצפן מהלקוח אל השרת.
תהליך התקשורת ע"פ הפרוטוקול המוצע:
1.הלקוח יוצר קשר עם השרת, שולח בקשה לרישום. במקרה והלקוח כבר רשום הו שולח בקשה להתחברות מחדש, אם הבקשה הצליחה הוא ישר עובר לפעולת שליחת הקובץ, אחרת נרשם מחדש.
2. השרת בתגובה שומר את נתוני הלקוח במסד נתונים ומייצר עבור הלקוח מזהה יחודי, לאחר מכן שולח הודעת אישור הכוללת את המזהה היחודי.
3. הלקוח שולח מפתח ציבורי לשרת.
4. השרת יוצר מפתח AES מצפין אותו בעזרת המפתח הציבורי שקיבל מהלקוח ושולח ללקוח.
5.הלקוח שולח קובץ מוצפן ע"י מפתח הAESשקיבל. ומחשב checksum לקובץ (הלא מוצפן).
6.השרת מפענח את הקובץ ומחשב checksum
7. הלקוח בודק האם תוצאות חישוב checksum זהים אצלו ואצל השרת. ושולח עדכון לשרת
8. אם הCRC תקין - השרת שומר את הקובץ ושולח אישור ללקוח שהקובץ נקלט בהצלחה.
9.אם הCRC לא תקין - הלקוח מנסה לשלוח שוב את הקובץ (עד 4 פעמים).