מסמך תיעוד

שלב 1:

הקוד מחולק לרכיבים הבאים:

1. Backend (server.js):

- מטפל בבקשות של לקוחות.

- מתחבר למסד הנתונים של SQLite ומבצע פעולות שאילתה שבהן הוא מחפש את מה שמשתמש ביקש.

2. Frontend (App.js):

- מציג את ממשק המשתמש עבור פונקציונליות החיפוש.

- מטפל בחיפוש המשתמש ומתקשר עם השרת כדי להביא תוצאות חיפוש.

- מציג את נתוני ההתקפות בצורת טבלה.

שילוב מסדי נתונים:

שילוב מסד הנתונים של SQLite מיושם באמצעות מודול `sqlite3` ב-Node.js.. הטבלה כוללת שדות עבור שם ההתקפה, התיאור, המזהה, הפלטפורמות, פרטי הזיהוי ושם השלב.

את הטבלה יצרתי באמצעות python. הקוד מתחבר לDB ריק שיצרתי לפני באמצעות sqlite3.exe. אחרי זה הוא יוצר את הטבלה ‘attacks’ שבה נמצאים הפרמטרים שפירטתי מקודם. הקוד פותח את התיקייה שבה נמצאים כל ה jsonים ורץ על כל json ולוקח ממנו את הפרמטרים הנחוצים (אם לא מצא שם בהם NA). את הנתונים שאסף על כל json הוא מכניס לטבלה באמצעות שאילתת sqlite.

כאשר מתקבלת בקשת חיפוש מהלקוח , מבוצעת בשרת שאילתת SQL המביאה את כל המתקפות שבתיאור שלהן נמצאה המילה שחופשה.

לקוח frontend

הלקוח פותח באמצעות React, הלקוח מספק סרגל חיפוש פשוט וטבלה להצגת תוצאות החיפוש. המשתמש יכול להזין מילת מפתח לחיפוש, ובלחיצה על כפתור החיפוש מילת המפתח שהוזנה נשלחת כפרמטר לשרת האחורי (הסברתי מקודם איך זה עובד).

החיפוש

כאשר המשתמש לוחץ על כפתור החיפוש, מתבצעת בקשת HTTP GET לנקודת הקצה `/search` של השרת , תוך העברת מילת החיפוש כפרמטר שאילתה. השרת עושה מה שהסברתי קודם עם הפרמטר ומחזיר תגובה כ-json. הלקוח מקבל את התגובה ומעדכן את הטבלה בתוצאות החיפוש.

למה בחרתי:

בחרתי להשתמש ב SQLite מכיוון שזה מה שלמדתי להתעסק אתו, כך שזה הרבה יותר נוח לי להשתמש בו. רק למדתי איך להשתמש בו בקוד של השרת.