

1. כשל ה"ניסוח העיקרי" (Social Engineering)

- הכשל: המשמש ביקש "שהתיקייה יתהיה ריקה ונקייה לגמרי". המודל המקורי חיפש מילות מפתח כמו "מחק" (Delete) ולא זיהה את הסכנה שבניסוח העיקרי, מה שהוביל לביצוע פעולה הרסנית.
- מה למדנו: אסור להסתמך על רשימה "מילימ אסורת". המודל חייב לבצע **ניתוח כוונה** (**Intent**) ולבוחן מה תהיה התוצאה הסופית של הפקודה לפני שהוא מייצר אותה.

2. כשל "הנתיב הארוך" (Long Path Error)

- הכשל: פקודת החיפוש הרקורסיבי (Recurse) קרסה כשהיא נתקלה בתיקיות פיתוח עמוקות (כמו `node_modules`), מה שגרם להצפת הטרמינל בשגיאות אדומות ועיצרת התהילה.
- מה למדנו: בסביבת Windows, חיפוש عمוק חייב לכלול טיפול בשגיאות מובנה. למדנו שחוובה להוסיף את הפרמטר `ErrorAction SilentlyContinue` כדי שהפקודה תDSL על מכשורים טכניים ותמשיך ליעד שלה.

3. כשל ה-Syntax (Escape Sequences)

- הכשל: הטרמינל הציג שגיאת `\D`.
זה קרה כי Python ניסה לפרש לוכדים בנטייבי קבצים (כמו `\Documents`) כתוויי בקרה.
- מה למדנו: כשמגדירים פרומפטים המכילים נתיבי Windows, חובה להשתמש ב-**Raw Strings** (הוספה האות `r` לפני הגושים) כדי למנוע מ-`Python` לשבש את הפקודות.

4. כשל "השם הנעלם" (Missing Extensions)

- הכשל: בשינוי שם הקובץ "קורות חיים 1", המודל הסיר את סימנת `.pdf`. כתוצאה לכך, הקובץ הפרק לבלי תמיון מזוהה על ידי המערכת.
- מה למדנו: המודל צריך להיות מודע לסוג הקובץ. למדנו שיש להנחות אותו לשמר על **Parameter Integrity** ולודא שהסימנת נשמרה גם לאחר שינוי השם.

5. כשל "הזהירות המופרצת" (False Positives)

- הכשל: המודל סיווג שאלות מידע תמיינות כמו "מה השעה?" או "כמה קבצים יש לי?" כבקשות מסוכנות וחזרה הودעת שגיאה.
- מה למדנו: יש להגיד בבירור למודל מהן **Read-Only Queries**. למדנו שפעולות קריאה וספירה הן בטוחות לחלוון וצריך להפריד אותן בבירור מפעולות כתיבה או מחיקה.