

HTTPS/TLS 1.2+

כל תקשורת בין הדפדפן לשרת חייבת להיות מוצפנת דרך HTTPS עם פרוטוקול TLS 1.2 או גבוה יותר, וזה SSL Certificate צריך להיות בתוקף ובדוק כל 3 חודשים.

Database encryption (AES-256)

כל המידע בבסיס הנתונים (שמות, אימיילים, חשבונות בנק, עסקאות) חייב להיות מוצפן עם AES-256 כדי שגם אם מישהו גונב את הקובץ, הוא לא יוכל לקרוא את הנתונים.

Backup encryption (AES-256)

כל גיבוי של בסיס הנתונים (יומי, שבועי, חודשי) צריך להיות מוצפן עם AES-256 כדי שהנתונים יהיו בטוחים גם בעת אחסון בענן.

Secret Manager (credentials)

כל הסודות (API keys, database passwords, OAuth secrets) צריכים להיות מאוחסנים בתוך Secret Manager (AWS/Google Cloud) ולא בקוד או בקובץ env. כדי שלא יתפזרו.

(TOTP / Biometric) 2FA

כל משתמש חייב להפעיל Two-Factor Authentication דרך Google Authenticator או Biometric כדי שגם אם סיסמה נגנבה, עדיין לא יוכלו להיכנס.

Audit logs (+24 חודשים, בלתי ניתנים לשינוי)

כל פעולה (login, view, edit, delete) צריכה להיות מתועדת בנפרד עם תאריך, שעה, מי עשה, מה עשה, ו- IP address בלי יכולת למחוק או לשנות תחת db אחר.

Session timeout

משתמש שלא עשה שום דבר במשך זמן מוגדר (5 דקות אדמין, 15 דקות משתמש) צריך להיות מנותק אוטומטית, כדי שאם אחד שכח את המחשב בפתוח, מישהו אחר לא יוכל להשתמש בו.

Restore Test (monthly)

כל חודש צריך לבחור גיבוי אקראי, לשחזר אותו לסביבת בדיקה (לא לפרודקשן) ולבדוק שכל הנתונים שם ושהאתר עובד - כדי לוודא שבחירום אפשר בעצם להחזיר הכל.

CORS configuration

הגדר שדומיין מסוים בלבד (yourapp.com) יכול לגשת לAPI, CORS אסור לחלוטין כדי שאתרים זדוניים לא יוכלו לשלוח בקשות בשם המשתמש. (לא יודע מה זה CORS אחקור את זה)

Rate limiting

אם משתמש משדר הרבה בקשות בזמן קצר (כמו ניסיון brute force) השרת צריך לחסום אותו זמנית - כדי להגן מפני התקפות של סיסמאות או 2FA tokens, קריסות וכדומה.

SQL Injection prevention

אף פעם לא לשלב נתונים של משתמש ישירות בשאילתת - SQL תמיד להשתמש ב parameterized queries - כדי שמישהו לא יוכל להזריק קוד SQL מזיק דרך input field

XSS prevention

כל הקלט של המשתמשים צריך להיות escaped לפני הצגה בעמוד (או להשתמש ב template engine) כדי שמישהו לא יוכל להזריק JavaScript זדוני שיגנוב מידע של משתמשים אחרים.

הסבר של איך עושים זאת:

במקום להכניס לדף ישירות טקסט כמו:

```
<script>alert('xss')</script>
```

לתוך שדה שבו מוסיפים מידע, עושים לו escaping כלומר ממירים את התווים המיוחדים לייצוג בטוח, למשל:

```
&lt;script&gt;alert(&#39;xss&#39;)&lt;/script&gt;
```

בדפדפן זה יוצג כטקסט רגיל על המסך, ולא ירוץ כקוד JavaScript בפועל זה אומר:

<	הופך ל-	<	•
>	הופך ל-	>	•
"	הופך ל-	"	•
'	הופך ל-	'	•
&	הופך ל-	&	•

ככה ה-input של המשתמש **נשאר תוכן** ולא הופך לקוד שרץ בתוך הדף.

CSRF protection

כל form בעמוד צריך להיות עם token נסתר (CSRF token) ששונה לכל משתמש - כדי שאתר זדוני לא יוכל לשדל משתמש שלנו לעשות פעולה בלי שהוא יודע.
