https://medium.com/@vigowebs/frequently-asked-node-js-interview-questions-and-answers-b74fa1f20678

Milyen biztonsági mechanizmusok érhetők el a Node.js-ben?

A Node.js alkalmazásunkat a következő módszerekkel védhetjük:

Hitelesítés - A hitelesítés az egyik elsődleges biztonsági szakasz, amelyben a felhasználó egyáltalán hozzáfér az alkalmazáshoz. A hitelesítés egy vagy több ellenőrzéssel ellenőrzi a felhasználó személyazonosságát. A Node.js-ben a hitelesítés lehet session-alapú vagy token-alapú. Session-alapú hitelesítés során a felhasználó hitelesítő adatait összehasonlítják a szerveren tárolt felhasználói fiókkal, és sikeres érvényesítés esetén munkamenet indul a felhasználó számára. Amikor a munkamenet lejár, a felhasználónak újra be kell jelentkeznie. Token alapú hitelesítésnél a felhasználó hitelesítő adatait alkalmazzák egy token nevű karakterlánc előállítására, amelyet ezután társítanak a felhasználó szerverhez intézett kéréseihez.

Hibakezelés - Általában a hibaüzenet magyarázatot tartalmaz arra vonatkozóan, hogy mi is hibázott valójában a felhasználó számára az ok megértése érdekében. Ugyanakkor, ha a hiba az alkalmazáskód szintaxisához kapcsolódik, beállítható, hogy a teljes naplótartalmat megjelenítse a kezelőfelületen. Egy tapasztalt hacker számára a naplótartalom rengeteg érzékeny belső információt tárhat fel az alkalmazás kódstruktúrájáról és a szoftverben használt eszközökről.

Kérés validálás - Egy másik szempont, amelyet figyelembe kell venni egy biztonságos Node.js alkalmazás felépítése során, a kérések ellenőrzése vagy más szavakkal a bejövő adatok ellenőrzése az esetleges ellentmondásokra. Úgy tűnhet, hogy az érvénytelen kérelmek nem befolyásolják közvetlenül a Node.js alkalmazás biztonságát, ugyanakkor befolyásolhatják annak teljesítményét és robusztus működését. A bejövő adattípusok és -formátumok érvényesítése, valamint a meghatározott szabályoknak nem megfelelő kérelmek elutasítása további intézkedés lehet a Node.js alkalmazás biztonságának biztosításában.

Node.js biztonsági eszközök és bevált gyakorlatok - Olyan eszközöket használhatunk, mint a helmet (HTTP fejlécek beállításával védi az alkalmazást), a csurf (érvényesíti a tokeneket a beérkező kérésekben és elutasítja az érvényteleneket), a node rate limiter (szabályozza az ismételt kérések arányát).