

Milyen biztonsági szempontokat kell figyelembe venni egy AI-alapú „Vibe Coding” fejlesztőkörnyezet kialakításánál?

VFP2 KiberVéd - NIS2 Audit felkészítés jogi
állásfoglalás 2026/51

VERZIÓKÖVETÉS

Verziószám	Dátum	Módosította	Módosítások leírása
1.0	2026.09.23	Kálmán és Társa Ügyvédi Iroda	Első kiadott verzió

1. VEZETŐI ÖSSZEFOGLALÓ

Az AI-alapú fejlesztés biztonsági kockázata nem abból fakad, hogy egy modell kódot ír. Abból fakad, hogy az AI mekkora fejlesztési kontextust lát, milyen jogosultsággal kapcsolódik a fejlesztőkörnyezethez, végrehajthat-e műveleteket, és milyen kontrollok választják el a javaslatot a tényleges módosítástól és a kiadástól. Egy böngészőben használt általános chatbot, amelybe a fejlesztő kézzel másol be kódrészletet, egészen más kockázati profilt jelent, mint egy coding agent, amely fájlokat ír, parancsokat futtat, csomagokat telepít és külső eszközöket hív meg. Ugyanaz a modell, sőt ugyanaz a szolgáltató is állhat mögöttük, a kontrollrendszert mégis a képességek határozzák meg.

A kérdést ezért két, egymástól független tengely mentén kell megválaszolni. A szolgáltatási, illetve EIR-tengely azt írja le, hol fut az AI, és merre halad az adat. Az AI-fejlesztőeszköz a Kiberbiztonsági tv. szerinti elektronikus információs rendszer eleme, szolgáltatója közreműködőnek minősül, a vele kötött szerződésnek pedig a kockázatokkal arányos kiberbiztonsági követelményeket kell tartalmaznia. A képesség, illetve autonómia tengely azt írja le, mennyit lát az AI, és mit tud megtenni. Ez dönti el, hogy a használati mód beéri-e egy adatküldési szabállyal és kötelező kódfelülvizsgálattal, vagy sandboxot, kimenő forgalom szűrését, feladathoz kötött hitelesítő adatot és jóváhagyási kaput is igényel.

A gyakorlati válasz négy állításban foglalható össze. Az AI nem kerülheti meg a meglévő biztonságos fejlesztési folyamatot, a generált kód ugyanazokon a review-, teszt-, függőség- és titokvizsgálati kapukon megy át, mint bármely más módosítás. Az agentikus eszköz nem chat, hanem a felhasználó nevében cselekvő szoftveres szereplő, amelyet a legkisebb jogosultság elve szerint kell beállítani, és amely production-, szervezetszintű Git-, cloud admin- vagy hosszú életű secret-jogosultságot nem kaphat automatikusan. Minden külső tartalom, amelyet az agent feldolgoz, legyen az issue, README, log vagy weboldal, nem megbízható bemenet, mert utasításforrássá válhat. Végül az általános chatbot sem kockázatmentes, egyetlen beillesztés is elég ahhoz, hogy forráskód, személyes adat vagy hitelesítő adat kerüljön ki a szervezetből.

A szabályozási keret két szinten működik, és a két szint kiegészíti egymást. Az MI-rendelet az AI-rendszert szabályozza. A coding assistant általános fejlesztési célú használata jellemzően nem nagy kockázatú, a szervezet a legtöbb esetben alkalmazó, az AI-jártasság fejlesztését szolgáló intézkedéseket azonban a rendelet 2025. február 2. óta minden alkalmazótól elvárja. A Kiberbiztonsági tv. a szervezet és az elektronikus információs rendszer biztonságát szabályozza, és az AI-fejlesztőeszközt nem kezeli külön kategóriaként. A meglévő kötelezettségeket, a biztonsági osztályba sorolást, a közreműködői szerződést, a naplózást és az incidensbejelentést kell egy új eszközkörre alkalmazni.

Vezetői döntési szabály. Minél több kontextust lát az AI, minél több műveletet hajthat végre, minél több külső rendszerhez kapcsolódik, és minél kevesebb emberi jóváhagyást igényel, annál szigorúbb technikai és szervezeti kontrollprofilra van szükség.

2. A VIZSGÁLANDÓ KÉRDÉS

Milyen biztonsági szempontokat kell figyelembe venni egy olyan fejlesztőkörnyezet kialakításakor, amelyben a fejlesztő természetes nyelvű utasításokkal, AI segítségével hoz létre kódot, konfigurációt, tesztet, dokumentációt vagy fejlesztési műveletet?

Az iránymutatás a „Vibe Coding” kifejezést munkafogalomként használja. Ez nem jogi kategória. Az AI-val támogatott fejlesztési gyakorlatok széles körét fedi le, az egyszerű kérdés-válasz alapú chathasználattól egészen az eszközöket önállóan kezelő coding agentig. A dokumentum nem egyes termékeket minősít. Célja egy olyan gyártófüggetlen kontrollrendszer kialakítása, amely az eszközök tényleges képességeiből indul ki, és amelyet a Kiberbiztonsági tv. hatálya alá tartozó szervezet (a továbbiakban „Szervezet”) a meglévő megfelelési keretébe illeszthet.

A vizsgálat két tengelyt különít el. A szolgáltatási, illetve EIR-tengely arra a kérdésre válaszol, hogy hol fut az AI, és merre halad az adat, vagyis SaaS, API, on-premise vagy hibrid szolgáltatásról van-e szó, és a Szervezet tud-e egyáltalán a használatról. A képesség, illetve autonómia tengely arra válaszol, hogy mennyit lát az AI, és mit tud megtenni, vagyis csak válaszol, vagy ír, futtat, telepít, külső eszközt hív, esetleg élesít is. Az iránymutatás elsősorban a második tengellyel foglalkozik, az első tengelyen a kapcsolódó MKIK-iránymutatásra támaszkodik, amely a mesterséges intelligencia rendszerek EIR-struktúrába sorolását részletesen tárgyalja.¹

Nem tárgya az állásfoglalásnak az egyes AI-szolgáltatások részletes EIR-besorolása és biztonsági osztályba sorolása, a naplózás mezőszintű követelményei, a generált kód szerzői jogi és licencjogi megítélése, valamint az egyes termékek konkrét beállítási útmutatója.

3. GYAKORLATI JAVASLATOK

3.1 A használati mód azonosítása és a kontrollprofil kiválasztása

A biztonsági tervezés azzal kezdődik, hogy a Szervezet pontosan azonosítja, milyen módon használja az AI-t a fejlesztésben. Javasoljuk, hogy minden use case-t az alábbi öt szint valamelyikébe soroljon, és a szinthez rendelt minimumkontrollt tekintse kiindulópontnak.

- Általános AI chatbot integráció nélkül. A fejlesztő kézzel másolja be a kérdést és a kódrészletet, az AI nem fér hozzá az IDE-hez, a repositoryhoz, a fájlrendszerhez vagy a terminálhoz. Minimumkontroll az adatküldési szabály, a vállalati fiók, a tiltott adattípusok listája, valamint a kötelező review és teszt.
- Kódkiegészítő, illetve inline asszisztens az IDE-ben. A megnyitott fájl és a környező kód alapján tesz javaslatot, önálló shell- vagy Git-műveletet nem végez. Minimumkontroll a kontextuskizárás, a vállalati beállítások és a secure SDLC kapuk.
- Projektkontextust értelmező (workspace-aware) asszisztens. Repository-szintű kontextust indexel, több fájlt értelmez, módosítási javaslatot készít, írni csak kézi

¹ VFP2 KiberVéd, A mesterséges intelligencia (MI) rendszerek besorolása az EIR-struktúrába, szakmai iránymutatás 2026/9

jóváhagyással tud. Minimumkontroll a repository-hatókör, az érzékeny mappák kizárása és a PR-alapú változtatás.

- d) Coding agent. Fájlokat ír, teszteket futtat, shellparancsokat indít, csomagokat telepít, eszközöket hív meg. Minimumkontroll a sandbox, a tool allowlist, a kimenő forgalom szűrése, a feladathoz kötött hitelesítő adat és a jóváhagyási kapu.
- e) CI/CD- vagy eseményvezérelt agent. PR-, issue- vagy pipeline-esemény hatására automatikusan elindul, commitot, review-t, javítást vagy build lépést kezdeményezhet, akár szervezeti szintű jogosultsággal. Minimumkontroll az izolált runner, a minimális token-hatókör, a külső tartalom nem megbízható bemenetként való kezelése és a jóváhagyás minden push és deploy előtt.

A szintek nem a szolgáltatási modellt írják le. Egy általános chatbot működhet publikus SaaS-ként vagy belső, on-premise szolgáltatásként, és egy agent is futtat saját üzemeltetésben, ezért a szolgáltatási, illetve EIR-besorolást külön kell elvégezni. Ha a kontextus szélessége, a műveleti jogosultság, az autonómia foka vagy a külső kapcsolatok köre magas, a use case-t legalább kontrollált agentként kell kezelni.

3.2 Adat- és kontextusvédelem

AI-fejlesztési környezetben a prompt csak az egyik adatküldési csatorna. Egy integrált eszköz megkaphatja a megnyitott fájlokat, a repository részleteit, a terminálkimenetet, a stack trace-t, a ticketeket, a PR-leírásokat és a tool-válaszokat is. A Szervezet ezért először azt térképezze fel, mit lát ténylegesen az AI, majd az alábbi szabályokat vezesse be.

- a) Készüljön rövid, egyértelmű lista arról, milyen forráskód, adat, dokumentáció és terminálkimenet adható át az AI-nak. Publikus és oktatási kód jóváhagyott eszközben szabadon használható. Belső forráskód csak vállalati jóváhagyással rendelkező szolgáltatásba kerülhet, dokumentált feltételekkel. Személyes vagy ügyféladat csak külön engedélyezett use case-ben, megfelelő jogalap és adatfeldolgozói szerződés mellett kerülhet AI-kontextusba. Titok, token, privát kulcs és jelszó semmilyen formában, példaként sem.
- b) Az AI-eszköz saját ignore-, illetve kontextuskizárási mechanizmusát külön be kell állítani. A .gitignore önmagában nem védi az adatokat az AI elől. Alapértelmezés szerint ki kell zárni a .env fájlt és változatait, a tanúsítvány- és kulcsfájlokat, a credential-exportokat, a service-account kulcsokat, a helyi secret store-okat, az SSH-kulcsokat és a production konfigurációkat.
- c) Fejlesztői logok és adatbázis-dumpok átadásakor maszkolt vagy szintetikus adattal kell dolgozni. A fejlesztői AI-session ne örökölje automatikusan a teljes terminálelőzményt, a vágólapot vagy a home könyvtárat.
- d) Ha titok került AI-kontextusba, azt kompromittáltnak kell tekinteni és rotálni kell, függetlenül attól, hogy a szolgáltató mit vállal a megőrzésről.
- e) Az integráció nélküli chatbotnál a technikai kizárás csak korlátozottan működik, ezért ott a szabályzatra, a DLP-re, a böngésző- és tenant-kontrollokra, valamint az oktatásra kerül a hangsúly. A vállalati használat ne magánfiókból történjen, teljes fájl helyett a problémát

bemutató, minimalizált és anonimizált kódrészletet kell átadni, a válasz pedig nem kerülhet közvetlenül production branchre.

3.3 Az agent jogosultságai és a jóváhagyást igénylő műveletek

Az agentikus eszközt a Szervezet ne chatként, hanem a felhasználó nevében cselekvő szoftveres szereplőként kezelje. A jogosultsági profilt a legkisebb jogosultság elve szerint, a projekt érzékenységihez igazítva kell kialakítani.

- a) Az agent fájlrendszer-hozzáférése az aktuális projektre korlátozódik, a home és a credential könyvtárak ki vannak zárva. Érzékeny projektnél ideiglenes munkaterület a megoldás, ahol a gazdagép fájlrendszere csak olvasható vagy nincs csatlakozva.
- b) A shell hozzáférés allowlisten alapul, a veszélyes parancsok tiltottak, privilegizált művelet nem engedélyezett. A hálózati elérés proxy vagy kimenő allowlist mögött működik, amely a jóváhagyott registryt, dokumentációt és API-t engedi, minden más alapértelmezetten tilt.
- c) Az agent Git-jogosultsága a fejlesztői branchre és a PR létrehozására terjed ki, védett branchre közvetlenül nem írhat. Hitelesítő adatként rövid életű, feladathoz kötött, minimális hatókörű tokenet kap, production secret nélkül. Cloud- és CI-hozzáférése legfeljebb olvasási vagy szűk fejlesztői hatókörű.
- d) Emberi jóváhagyás nélkül nem hajtható végre új függőség vagy IDE-bővítmény telepítése, tömeges fájltilrlés, adatbázis-migráció vagy más visszafordíthatatlan parancs, valamint kód vagy bináris letöltése és futtatása külső URL-ről. Ugyanez vonatkozik a CI/CD, a Dockerfile, az IaC, a csomagszkript, a hitelesítési, jogosultságkezelési vagy kriptográfiai kód módosítására, a védett ágra irányuló push, merge vagy release műveletre és minden production- vagy admin-jogosultságot igénylő műveletre.
- e) Az AGENTS.md, a CLAUDE.md, a copilot-instructions.md és a hasonló, tartósan irányító fájlok biztonsági szempontból releváns konfigurációk. Legyen kijelölt tulajdonosuk, csak PR-on és review-val módosuljanak, ne tartalmazzanak titkot, és az agent ne módosíthassa automatikusan a saját irányító fájlját.
- f) Csak biztonsági szempontból felülvizsgált MCP- vagy tool-szerver csatlakozhat. A coding agent ne kapjon irreleváns e-mail-, pénzügyi vagy admin-eszközt, a tool-definíciók változását észlelni kell, és minden tool-hívást naplózni kell az érzékeny argumentumok maszkolásával. Kvóta, időkorlát, valamint költség- és hívásszámkorlát véd a végtelen agenticus ellen.

3.4 A generált kód és a szoftverellátási lánc kapui

Az AI használatával a biztonságos kódolás szabályai nem változnak. A generált kódnak, tesztnek, dokumentációnak és infrastruktúra-változásnak ugyanazokon a változáskezelési és biztonsági kapukon kell átmennie, mint bármely más módosításnak, és a kapukat az AI működése kedvéért nem lehet kikapcsolni.

- a) Az AI által írt kódot ember vizsgálja felül. Az AI saját értékelése és az AI által készített review nem számít független felülvizsgálatnak. Biztonságkritikus területen szakértő reviewer szükséges.

- b) A SAST, a secret-vizsgálat, a függőségvizsgálat, valamint web- és API-komponensnél a DAST ugyanúgy vagy szigorúbban fut az AI által írt kódra, és a kaput az agent nem tudja kikapcsolni. A secret-vizsgálat kiemelten a konfigurációra, a tesztekre, a dokumentációra és a generált példakódra terjed ki.
- c) Az AI által javasolt új csomag nem települhet automatikusan. A Szervezet szervezeti registryt, proxyt vagy allowlistet használ, a lockfile és a verziórögzítés kötelező, a csomag létét, eredetét, karbantartói történetét és sérülékenységi állapotát ellenőrizni kell. A build- vagy install-hookot tartalmazó csomag magasabb kockázati kategóriába tartozik.
- d) A CI/CD workflow, a csomag- és build szkriptek, a Dockerfile és a konténerkonfiguráció, a Terraform és más IaC-kód, valamint a hitelesítési, jogosultságkezelési és kriptográfiai kód biztonságkritikus fájl. Ezekre külön reviewer, védett workflow, a harmadik féltől származó action-ök rögzítése és plan review vonatkozik, az agent pedig nem futtathat production applyt.
- e) Az AI által végzett minden módosítás visszavonható és utólag rekonstruálható, ehhez branch, commit, PR vagy egyenértékű változáskezelési nyomvonal szükséges.

3.5 Szolgáltatói feltételek és a javasolt eljárás

Az AI coding szolgáltatóval kötött szerződés egyben a Kiberbiztonsági tv. szerinti közreműködői szerződés is. A Szervezet a szolgáltató kiválasztásakor legalább az adatfelhasználást és a vállalati kizárás lehetőségét, a megőrzést, az adatlokációt és az alfeldolgozókat, a tenant-izolációt, az identitáskezelést, a kontextuskontrollt, az audit- és agentművelet-napló exportálhatóságát, a modellváltozsról szóló értesítést, a bejelentési határidőkhöz igazított incidensértesítést és a kilépési feltételeket tisztázza. A részletes kérdéssort a 6.3 melléklet tartalmazza.

A bevezetést az alábbi lépésekben javasoljuk.

- a) Hatókör, a use case, a projektkör, az adatosztály, a szolgáltatási és EIR-besorolás, valamint a tulajdonos rögzítése, a kockázati profil jóváhagyása.
- b) Szolgáltató és architektúra, az IAM, a kontextus, a megőrzés, a tool, a sandbox és az adatút ellenőrzése, a szerződéses kikötések rögzítése.
- c) Chat- és asszisztens pilot alacsony kockázatú projekten, kilépési feltétel, hogy nincs szabályzatmegkerülés, a review és a vizsgálatok működnek.
- d) Agent pilot sandboxban, írással, teszttel és korlátozott shell- vagy tool-hozzáféréssel, kilépési feltétel a jóváhagyási, kimenőforgalom-szűrési, secret- és visszaállítási kontrollok tesztelt működése.
- e) Kiterjesztés projektkategóriák és profilok szerint, működő monitoringgal, oktatással, kivételkezeléssel és legalább évenkénti felülvizsgálattal.
- f) Képzés, a fejlesztők célzott AI-jártassági és biztonsági oktatása, amely kiterjed az adatkezelésre, a hallucinált függőségekre, a prompt injectionre, az agent jogosultságaira, a biztonságos review-ra és az incidensbejelentésre, dokumentált tematikával és részvétellel.

4. VÁLASZ ÉS RÉSZLETES INDOKOLÁS

4.1 A vizsgálat kiindulópontjai

Az AI-fejlesztőeszköz és a köré épülő környezet, vagyis az IDE, a repository, a CI/CD-rendszer, a csomag-registry és a tool-szerverek együttese a Kiberbiztonsági tv. hatálya alá tartozó Szervezetnél az elektronikus információs rendszer része. A törvény az elektronikus információs rendszer fogalmát tágan határozza meg.

minden olyan eszköz vagy egymással összekapcsolt vagy kapcsolatban álló eszközök csoportja, amelyek közül egy vagy több valamely program alapján digitális adatok automatizált kezelését végzi, ideértve a kiber-fizikai rendszereket [2024. évi LXIX. törvény 4. § 24. pont b) alpont]

Ebből az következik, hogy az AI-fejlesztőeszköz bevezetése nem önálló megfelelési kérdés, hanem a meglévő kötelezettségek alkalmazása egy új eszközkörre. A védelmi intézkedések körét a biztonsági osztály határozza meg, amelyet a Szervezet a kockázatokkal arányosan állapít meg.

(1) A szervezet elektronikus információs rendszerei, valamint az azokban kezelt adatok, a nyújtott szolgáltatások kockázatokkal arányos védelmének biztosítása érdekében a szervezet az e törvény hatálya alá tartozó, a szervezet rendelkezésében lévő elektronikus információs rendszereit „alap”, „jelentős” vagy „magas” biztonsági osztályba sorolja az érintett elektronikus információs rendszer sértetlensége és rendelkezésre állása, valamint az általa kezelt adat bizalmassága, sértetlensége és rendelkezésre állásának kockázata alapján, szigorodó védelmi előírásokkal.

(4) A szervezet az elektronikus információs rendszer biztonsági osztálya alapján meghatározza és megvalósítja az informatikáért felelős miniszter rendeletében előírt védelmi intézkedéseket az adott elektronikus információs rendszerre vonatkozóan. [2024. évi LXIX. törvény 10. § (1) és (4) bekezdés]

A kockázatarányosság elve a Vibe Coding környezetben négy tényezőn keresztül érvényesíthető. A kontextus szélessége azt mutatja, hogy egy rövid kódrészletet, egy projektmapát vagy a teljes monorepót látja-e az AI a ticketekkel, logokkal és secret-közeli fájlokkal együtt. A műveleti jogosultság azt, hogy csak javaslatot ad, vagy admin shellt, cloud-, CI/CD- és production-hozzáférést kap. Az autonómia azt, hogy minden lépést ember indít és fogad el, vagy eseményvezérelt és tartós működésről van szó. A külső kapcsolatok pedig azt, hogy az AI csak jóváhagyott registryt és dokumentációt ér el, vagy tetszőleges webhelyet, MCP-szerveret és külső API-t. Ez a modell nem jogszabályi pontozás, hanem a tervezést és a jóváhagyást segítő eszköz. Ha bármelyik tényező magas, a use case legalább kontrollált agent profilt igényel.

Az OWASP Secure Coding with AI útmutatója ugyanezt a logikát követi, amikor külön foglalkozik a hallucinált függőségekkel, az indirekt prompt injectionnel, az MCP- és tool-kockázatokkal, az agentek sandboxolásával, a rules fájlokkal, a kontextusszivárgással és a

CI/CD-agentek kockázataival.² Az ENISA ehhez az életciklus- és ellátásilánc-szemléletet teszi hozzá, mert a fejlesztőkörnyezet egyszerre függhet a modellszolgáltatótól, az IDE-bővítménytől, az agent-keretrendszertől, a tool-szerverektől, a csomag-registrytól és a CI/CD-rendszertől.³

4.2 Hol fut az AI, és merre halad az adat? A szolgáltatási és EIR-tengely

4.2.1 A szolgáltató mint közreműködő

Az AI coding szolgáltatás nyújtója, legyen az SaaS-modell, API, IDE-bővítmény vagy agent-platform szolgáltatója, a Szervezet elektronikus információs rendszerével kapcsolatos tevékenységet lát el, ezért a Kiberbiztonsági tv. szerinti közreműködőnek minősül. A törvény ebben az esetben a szervezet vezetőjére telepíti a felelősséget, és a kiberbiztonsági követelmények szerződéses rögzítését várja el.

(5) A szervezet vezetője az elektronikus információs rendszer védelmének biztosítása érdekében

d) ha a szervezet közreműködőt vesz igénybe az elektronikus információs rendszer létrehozása, üzemeltetése, auditálása, karbantartása, javítása, illetve a kiberbiztonsági incidensek kezelése során, vagy a szervezet elektronikus információs rendszerével kapcsolatos adatkezelési, adatfeldolgozási tevékenység ellátásához, gondoskodik arról, hogy a közreműködő által az elektronikus információs rendszerrel kapcsolatosan ellátott tevékenységgel összefüggésben szükséges kiberbiztonsági követelmények az e törvényben foglaltaknak megfelelően szerződéses kötelemként teljesüljenek; [2024. évi LXIX. törvény 6. § (5) bekezdés d) pont]

A szerződéses kötelelem tartalmát a gyakorlatban a 3.5 pontban és a 6.3 mellékletben felsorolt kérdések jelölik ki. Ezek közül az adatfelhasználás és a megőrzés a legfontosabb. Ha a szolgáltató a promptot, a kódot vagy a tool-kimenetet saját modellfejlesztésre használja, a Szervezet forráskódja és üzleti információja kikerül a Szervezet ellenőrzése alól, személyes adat esetén pedig a szolgáltató már nem adatfeldolgozóként jár el. A vállalati kizárás lehetősége, a rövid és beállítható megőrzés, valamint az agentművelet-napló exportálhatósága ezért nem kényelmi funkció, hanem a közreműködői szerződés kötelező eleme. Ugyanez igaz az incidensértesítésre, amelynek határideje a Szervezet saját, a 4.4.1 pontban tárgyalt bejelentési határidőihöz igazodik. A 7/2024. (VI. 24.) MK rendelet 2. mellékletének a beszerzésre és az ellátási lánc biztonságára vonatkozó védelmi intézkedései ugyanezt a követelményrendszert írják le, a biztonsági osztálytól függő mélységben.⁴

Magas érzékenységgű kódnál a Szervezet mérlegelje a privát, on-premise vagy izolált megoldást. A konkrét EIR- és biztonsági osztályba sorolást ilyenkor is a kapcsolódó MKIK-iránymutatás

² OWASP Cheat Sheet Series, Secure Coding with AI Cheat Sheet (cheatsheetseries.owasp.org)

³ ENISA, Artificial Intelligence Cybersecurity Challenges (2020); ENISA, Cybersecurity of AI and Standardisation (2023)

⁴ 7/2024. (VI. 24.) MK rendelet 2. melléklet, a beszerzésre és az ellátási lánc biztonságára vonatkozó védelmi intézkedések, a biztonsági osztálytól függő kötelezettséggel

szerint kell elvégezni, amely a SaaS, API és on-premise modellek besorolását részletesen tárgyalja.⁵

4.2.2 Személyes adat a promptban és a kontextusban

Hibakereséskor egy adatbázis-dump, egy log vagy egy stack trace könnyen tartalmaz ügyféladatokat, felhasználónevet, e-mail-címet vagy azonosítót. Ha ez AI-kontextusba kerül, az adattovábbítás és az adatfeldolgozó igénybevétele a GDPR hatálya alá esik, függetlenül attól, hogy a fejlesztő szándékosan adta-e át az adatot. A GDPR az adatfeldolgozó kiválasztását feltételhez köti.

*(1) Ha az adatkezelést az adatkezelő nevében más végzi, az adatkezelő kizárólag olyan adatfeldolgozókat vehet igénybe, akik vagy amelyek megfelelő garanciákat nyújtanak az adatkezelés e rendelet követelményeinek való megfelelését és az érintettek jogainak védelmét biztosító, megfelelő technikai és szervezési intézkedések végrehajtására.
[GDPR 28. cikk (1) bekezdés]*

Ebből két gyakorlati következmény adódik. Személyes adat csak külön engedélyezett use case-ben, jogalappal és a GDPR 28. cikk (3) bekezdése szerinti adatfeldolgozói szerződéssel kerülhet AI-kontextusba, a szolgáltatói adatfeldolgozási megállapodás pedig a szolgáltatói értékelés része. Alapesetben pedig maszkolt vagy szintetikus adattal kell dolgozni, ami egyben a GDPR 32. cikke szerinti, a kockázathoz igazodó biztonsági intézkedés is.⁶ A szolgáltató által ígért törlés vagy rövid megőrzés nem helyettesíti a megelőzést, mert az adat a kiadás pillanatában már elhagyta a Szervezet ellenőrzési körét.

4.2.3 Shadow AI a fejlesztésben

A magánfiókos vagy nem jóváhagyott AI-használat azért különösen veszélyes a fejlesztésben, mert a fejlesztő a Szervezet minden biztonsági kontrollját megkerülve adhat át forráskódot vagy üzleti információt. Ilyenkor nincs közreműködői szerződés, nincs adatfeldolgozói megállapodás, nincs kizárási beállítás és nincs napló. A Szervezet ezért a jóváhagyott eszköz- és use case-listát tegye a normál fejlesztési folyamat részévé, a magánfiókos használatot pedig érzékeny kóddal tiltsa meg. A shadow AI EIR-vonatkozásait és az IT-eszközök magánhasználatát a kapcsolódó MKIK-iránymutatások tárgyalják.⁷

4.3 Mennyit lát az AI, és mit tud megtenni? A képesség és autonómia tengely

4.3.1 Az általános chatbot sem kockázatmentes

Az integráció nélküli chatbot a legalacsonyabb autonómiaszint. Az AI nem fér hozzá közvetlenül a repositoryhoz, a fájlrendszerhez vagy a terminálhoz, a végrehajtási kockázat ezért alacsony. Az adatkiadás kockázata azonban nem az, mert itt a felhasználó maga dönti el, mit másol be, és a technikai kontextuskizárás nem működik. Ennél a használati módnál ezért az adatkezelésre és az

⁵ VFP2 KiberVéd, szakmai iránymutatás 2026/9, a SaaS, API és on-premise modellek besorolásáról szóló rész

⁶ GDPR 28. cikk (3) bekezdés és 32. cikk (1) bekezdés

⁷ VFP2 KiberVéd, A munkáltató által biztosított IT-eszközök magánhasználatát, szakmai iránymutatás 2026/5; VFP2 KiberVéd, szakmai iránymutatás 2026/9, a shadow AI-ról szóló rész

emberi ellenőrzésre kell helyezni a hangsúlyt. A másik kockázat az ellenőrizetlen kód átvétele. A chatbot válasza hibás vagy sérülékeny lehet, elavult függőséget javasolhat, ezért előbb fejlesztői review-n és teszten kell átmennie.

4.3.2 A kontextus mint adatkiadási csatorna

Az integrált eszközöknél a kockázat súlypontja áthelyeződik. Az IDE-asszisztens és különösen a workspace-aware asszisztens nem csak azt látja, amit a fejlesztő beír, hanem azt is, amit a projekt tartalmaz. Az OWASP külön kiemeli, hogy az AI-eszközök saját kontextuskizárási mechanizmust használnak, és a .gitignore önmagában nem védi az adatokat az AI elől.⁸ A tipikus hiba, hogy a .env fájl, egy service-account kulcs vagy egy production konfiguráció bekerül az indexelt kontextusba, a fejlesztő pedig nem is tudja, hogy az eszköz elküldte. A kontextuskizárás ezért a kontrollprofil része, amelyet a Szervezet központilag állít be és tesztel. A kikerült titkot rotálni kell, mert a szolgáltató megőrzési és tanítási gyakorlatát a Szervezet nem tudja ellenőrizni.

4.3.3 Az agent mint szoftveres szereplő

A coding agent minőségileg más, mint az asszisztens. Fájlokat ír, teszteket futtat, shellparancsokat indít, csomagokat telepít, eszközöket hív meg, és mindezt a felhasználó nevében teszi. Az OWASP ezért azt javasolja, hogy az agentek sandboxban fussanak, fájlrendszer- és hálózati hozzáférésük korlátozott legyen, és csak rövid életű, feladatszintű hitelesítő adatot kapjanak.⁹ Az agentbiztonsági útmutató három további követelményt emel ki. A nagy hatású döntéseknél emberi felügyeletet kell fenntartani, az agent memóriájában tárolt érzékeny adatokat védeni kell, a végtelen ciklusok ellen pedig költség- és erőforráskorlát szükséges.¹⁰

A legkisebb jogosultság elve itt konkrét kérdéssé válik. Ha az agent a fejlesztő teljes Git-, cloud- és shelljogosultságával fut, akkor egy hibás értelmezés vagy egy sikeres prompt injection ugyanazzal a jogosultsággal okoz kárt. A jogosultsági profil ezért nem a fejlesztő, hanem a feladat jogosultsága. Ez a 3.3 pontban leírt profilok alapja, és ez indokolja a jóváhagyást igénylő műveletek listáját is. A lista nem a hatékonyságot korlátozza, hanem azokat a pontokat jelöli ki, ahol a hiba visszafordíthatatlan, vagy a jogosultság magasabb, mint amit a feladat igényel.

4.3.4 Indirekt prompt injection, irányító fájlok és tool-integráció

Agentikus környezetben a megszokott fejlesztői bemenetek utasításforrássá válhatnak. Egy issue rejtett utasítással arra veheti rá az agentet, hogy más fájlokat módosítson vagy adatot küldjön ki. Egy klónozott repository README fájlja az agentnek szóló utasítást rejthet, egy célzottan összeállított hibaüzenet tool-hívásra ösztönözheti, egy webes keresés eredménye pedig manipulálhatja az agentet. Az OWASP ezt indirekt prompt injectionnek nevezi, és azt javasolja, hogy az agent által feldolgozott minden külső tartalom nem megbízható bemenetként kezeljünk.¹¹ A védekezés három rétegű. A külső tartalom külön megbízhatósági zónába tartozik, és csak a szükséges fájlok kerülnek kontextusba. A logtartalmat adatként, nem parancsként kell

⁸ OWASP, Secure Coding with AI Cheat Sheet, a kontextusszivárgásról és a kontextuskizárásról szóló rész

⁹ OWASP, Secure Coding with AI Cheat Sheet, az agentek sandboxolásáról szóló rész

¹⁰ OWASP Cheat Sheet Series, AI Agent Security Cheat Sheet (cheatsheetseries.owasp.org)

¹¹ OWASP, Secure Coding with AI Cheat Sheet, az indirekt prompt injectionről szóló rész

kezeln, a tool-hívás jóváhagyáshoz kötött. Az agent kimenetét, vagyis a diffet, kötelező felülvizsgálni.

A tartósan irányító fájlok külön figyelmet érdemelnek. Az AGENTS.md, a CLAUDE.md és a hasonló rules fájlok minden sessionben befolyásolják az agent viselkedését, ezért egy rejtett vagy rosszindulatú módosításuk tartós eltérést jelent. Ezeket biztonsági szempontból releváns konfigurációként kell kezelni, kijelölt tulajdonossal, kötelező review-val, titok nélkül, az agent önmódosításának tiltásával és integritás-ellenőrzéssel.¹²

Az MCP és a hozzá hasonló tool-protokollok műveleti képességekkel ruházzák fel a nyelvi modellt. Az OWASP MCP-biztonsági útmutatója rámutat, hogy maguk a tool-definíciók is a kontextus részét képezik, így prompt injectiont hordozhatnak, a túl széles jogkörű vagy rosszindulatú tool-szerver pedig ellátási lánc- és jogosultsági kockázatot jelent. Az útmutató ezért a tool-definíciók rögzítését, a leírásuk utólagos megváltozásának figyelését és minden tool-hívás teljes paraméterezéssel történő naplózását javasolja.¹³ A Szervezet ebből a jóváhagyott tool-nyilvántartás, a minimális eszközkészlet, az argumentumellenőrzés, a hálózati izoláció és a naplózás követelményét vezeti le.

4.3.5 Függőségek, ellátási lánc és biztonságkritikus fájlok

Az OWASP két külön kockázatként kezeli a nem létező, hallucinált csomagneveket és az elavult, ismert sérülékenységet tartalmazó függőségverziókat.¹⁴ A hallucinált csomagnév azért veszélyes, mert a támadók figyelik ezeket a neveket, és maguk regisztrálják őket a nyilvános registryben. Ha az agent automatikusan telepíti a javasolt csomagot, a rosszindulatú kód az install-hookon keresztül már a fejlesztő gépén fut, és látja mindazt, amit az agent lát. A védekezés a szervezeti registry vagy proxy, a lockfile és a verziórögzítés, a csomag létének, eredetének és karbantartói történetének ellenőrzése, valamint az, hogy új csomag felülvizsgálat nélkül nem települhet.

A biztonságkritikus fájlok külön kaput igényelnek, mert a rajtuk végzett módosítás magasabb jogosultsággal fut, mint a fejlesztő. A CI/CD workflow automatizáltan, secret-hozzáféréssel fut. A csomag- és build szkriptek telepítéskor vagy buildkor kódot futtatnak. A Dockerfile és a konténerkonfiguráció a futtatási környezetet és az ellátási láncot határozza meg. A Terraform és más IaC-kód valós infrastruktúrát és jogosultságot módosít. A hitelesítési, jogosultságkezelési és kriptográfiai kód közvetlen biztonsági kontrollt valósít meg. Ezekre a Szervezet külön reviewert, védett workflow-t, a külső action-ök rögzítését és plan review-t ír elő, az agent pedig nem futtathat production applyt. A NIST SSDF a biztonságos fejlesztési gyakorlatokat az SDLC részeként kezeli, az SP 800-218A pedig AI-specifikus kiegészítésekkel segíti az AI-t fejlesztő vagy beszerző szervezeteket.¹⁵

4.3.6 A secure SDLC nem kerülhető meg. Tesztek és az önigazolás kockázata

A biztonságos AI-támogatott fejlesztés alapelve, hogy az AI nem kerülheti meg a meglévő fejlesztési folyamatot. A hatókört és az elfogadási feltételeket ember rögzíti, a kontextus

¹² OWASP, Secure Coding with AI Cheat Sheet, a rules fájlokról szóló rész

¹³ OWASP Cheat Sheet Series, MCP Security Cheat Sheet (cheatsheetseries.owasp.org)

¹⁴ OWASP, Secure Coding with AI Cheat Sheet, a hallucinált és az elavult függőségekről szóló rész

¹⁵ NIST SP 800-218, Secure Software Development Framework (SSDF) Version 1.1 (2022); NIST SP 800-218A, Secure Software Development Practices for Generative AI and Dual-Use Foundation Models, An SSDF Community Profile (2024)

összeállításánál az adatbesorolás és a kontextuskizárás érvényesül, a kód létrehozását a sandbox és az agent-policy keretezi, az automatikus ellenőrzés kapuit nem lehet megkerülni, az elfogadásért az emberi reviewer felel, a merge és a release pedig védett branchhez és jóváhagyáshoz kötött. Productionben az autonóm AI alapértelmezetten nem adhat ki parancsot és nem indíthat deployt.

Külön kockázat az önigazolás. Az agent által írt tesztek hasznosak, de önmagukban nem bizonyítják, hogy az ugyanezen agent által generált biztonságkritikus kód helyes. Ha ugyanaz a forrás írja a kódot és a tesztet, a teszt könnyen csak a saját feltevéseit igazolja vissza. Az OWASP ezért a hitelesítés, a jogosultságkezelés, a bemenetellenőrzés és a kriptográfia területén független fél általi ellenőrzést javasol.¹⁶ Ugyanez az oka annak, hogy az AI által készített kódfeülvizsgálat nem számít független review-nak. A hamis biztonságérzet ellen a független review, a biztonsági teszt és a fejlesztők AI-jártassága véd.

4.4 A szabályozási keret két szintje

4.4.1 A Kiberbiztonsági tv. kötelezettségeinek alkalmazása az AI-fejlesztőeszközre

A Kiberbiztonsági tv. hatálya alá tartozó Szervezetnél a fenti kontrollok jelentős része nem választás, hanem a szervezet vezetőjét terhelő törvényi kötelezettség új eszközkörre alkalmazva. A törvény a nyomkövethetőséget és az incidensreagálást a vezető feladatai közé sorolja.

(5) A szervezet vezetője az elektronikus információs rendszer védelmének biztosítása érdekében

c) gondoskodik az elektronikus információs rendszer eseményeinek nyomkövethetőségéről;

e) az elektronikus információs rendszert érintő kiberfenyegetés, kiberbiztonsági incidensközeli helyzet vagy kiberbiztonsági incidens bekövetkezésekor minden szükséges és rendelkezésére álló erőforrás felhasználásával gondoskodik a gyors és hatékony reagálásról, az illetékes kiberbiztonsági incidenskezelő központnak való bejelentésről, a kiberbiztonsági incidensek kezeléséről, valamint a helyreállításról;

[2024. évi LXIX. törvény 6. § (5) bekezdés c) és e) pont]

A nyomkövethetőség az AI-fejlesztési folyamatban azt jelenti, hogy a Szervezet nyomon követési rendszere, ahol ez technikailag megoldható, megkapja az AI-specifikus eseményeket. Ide tartozik az AI-t használó felhasználó vagy szolgáltatásfiók, a használt szolgáltatás és modell, az AI számára elérhetővé tett repository-környezet, az agent által módosított fájlok köre a kapcsolódó branch, commit vagy PR megjelölésével, valamint a végrehajtott shell- és tool-műveletek az érzékeny argumentumok maszkolásával. Ide tartoznak továbbá a bekerült függőség-, build-, CI- és IaC-változások, a jóváhagyási döntések a jóváhagyók személyével, valamint a blokkolt műveletek és a szabályzatsértések. A naplómezők pontos körét, a megőrzést, az integritást és a SIEM-követelményeket a 7/2024. (VI. 24.) MK rendelet 2. mellékletének naplózási védelmi intézkedései és a kapcsolódó MKIK naplózási iránymutatás rendezik.¹⁷

¹⁶ OWASP, Secure Coding with AI Cheat Sheet, a biztonságkritikus tesztek független ellenőrzéséről szóló rész

Az incidensreagálásnál a Szervezetnek tudnia kell, hogy az AI-specifikus események incidenssé válhatnak. A törvény fogalma tág.

olyan esemény, amely veszélyezteti az elektronikus információs rendszereken tárolt, továbbított vagy kezelt adatok vagy az e rendszerek által kínált vagy azokon keresztül elérhető szolgáltatások rendelkezésre állását, sértetlenségét vagy bizalmasságát [2024. évi LXIX. törvény 4. § 46. pont]

Ebbe a körbe tartozik, ha titok vagy token került AI-kontextusba, ha az agent váratlan parancsot vagy hálózati hívást végzett, ha rosszindulatú vagy gyanús függőség került be, ha egy rules fájl kompromittálódott, ha shadow AI-használat történt érzékeny kóddal, vagy ha az AI módosított egy CI/CD vagy IaC biztonsági kontrollt. Minden ilyen eseménynél a Szervezet incidenskezelési eljárása szerint kell eldönteni, hogy a törvény 4. § 40. pontja szerinti jelentős kiberbiztonsági incidensről van-e szó. Ha igen, a Kiberbiztonsági tv. 66. §-a szerinti bejelentés keretében a Szervezet a tudomásszerzéstől számított 24 órán belül első bejelentést, 72 órán belül eseménybejelentést, az eseménybejelentést követő egy hónapon belül pedig zárójelentést nyújt be az incidenskezelő központnak.¹⁸ Ezek a határidők indokolják, hogy a szolgáltatói szerződés incidensértesítési kikötése ezekhez igazodjon. Ugyanezért kell az elsődleges technikai reakciókat, vagyis a titok azonnali rotációját, az agent leállítását és a sandbox izolálását, a build és a release leállítását, az ismert jó verzió visszaállítását, valamint a pipeline és a deploy befagyasztását előre rögzített runbookban tartani.

4.4.2 Az MI-rendelet szerepkörei és az AI-jártasság

Az MI-rendelet nem a Vibe Coding szabályzata, és önmagában a coding assistant használata nem határozza meg a rendelet szerinti kockázati kategóriát. A kötelezettségek az AI-rendszerhez és a Szervezet abban betöltött szerepéhez kötődnek. A legtöbb Vibe Coding helyzetben a Szervezet alkalmazó, vagyis kész coding assistantot vagy agentet használ a saját felelősségére.

olyan természetes vagy jogi személy, hatóság, ügynökség vagy egyéb szerv, amely a felügyelete alá tartozó MI-rendszert használja, kivéve, ha az MI-rendszert személyes, nem szakmai tevékenység keretében használják [MI-rendelet 3. cikk 4. pont]

Szolgáltatóvá a Szervezet akkor válik, ha saját AI-rendszert, például belső coding agentet vagy egy modellre épülő saját fejlesztőeszközt fejleszt és a saját neve alatt helyez üzembe, illetve ha egy nagy kockázatú rendszert jelentősen módosít.¹⁹ A generált szoftver rendeltetése külön kérdés. Ha az elkészült alkalmazás a III. melléklet szerinti területre esik, maga a végtermék is lehet nagy kockázatú AI-rendszer, akkor is, ha a fejlesztéshez használt eszköz nem az.²⁰ A use case adatlapon ezért a szerepkört és a kockázati besorolást is rögzíteni kell.

¹⁷ 7/2024. (VI. 24.) MK rendelet 2. melléklet, naplózási és elszámoltathatósági védelmi intézkedések; VFP2 KiberVéd, A naplózás fontossága és mélysége a NIS2-megfelelés szempontjából, jogi állásfoglalás 2026/9

¹⁸ 2024. évi LXIX. törvény 66. §; 418/2024. (XII. 23.) Korm. rendelet 77. § (1) bekezdés 1., 3. és 5. pont

¹⁹ MI-rendelet 3. cikk 3. pont és 25. cikk (1) bekezdés

²⁰ MI-rendelet 6. cikk és III. melléklet

Az AI-jártasság a rendelet minden alkalmazóra kiterjedő kötelezettsége. A 4. cikk 2025. február 2. óta alkalmazandó, szövegét a Digital Omnibus on AI rendelet 2026. július 27-től újraszövegezte. Eszerint a szolgáltatóknak és az alkalmazóknak intézkedéseket kell tenniük a személyzetük és az AI-t a nevükben használó más személyek AI-jártasságának fejlesztésére, figyelembe véve a technikai tudást, a tapasztalatot, a képzettséget és a használati környezetet. A Szervezetnek nem egyénenként meghatározott jártassági szintet kell garantálnia, hanem a bevezetett rendszerekhez és szerepkörökhöz illeszkedő, dokumentálható intézkedéseket kell felmutatnia.²¹ Vibe Coding környezetben ezért a fejlesztői oktatás ne álljon meg a promptolási technikáknál, hanem terjedjen ki az adatkezelésre, a hallucinált függőségekre, a prompt injectionre, az agent jogosultságaira, a biztonságos review-ra és az incidensbejelentésre, a képzés megtörténtét és tematikáját pedig dokumentálni kell.

A nagy kockázatú rendszerekre vonatkozó 9., 14. és 15. cikk kockázatkezelést, emberi felügyeletet, valamint pontosságot, robusztusságot és kiberbiztonságot ír elő. Ezek a rendelkezések nem vonatkoznak minden coding assistant használatra, alkalmazásukat a Digital Omnibus a III. melléklet szerinti rendszereknél 2027. december 2-ára, az I. melléklet szerintiéknél 2028. augusztus 2-ára halasztotta.²² Tervezési elvként azonban jól hasznosíthatók, mert éppen azt mondják ki, amit ez az iránymutatás a kontrollprofilokkal megvalósít. Ha egy use case mégis nagy kockázatú rendszert érint, a 26. cikk szerinti alkalmazói kötelezettségek, köztük az emberi felügyelet és az automatikusan keletkező naplók legalább hat hónapos megőrzése, valamint a 73. cikk szerinti súlyos incidens bejelentése a Kiberbiztonsági tv. szerinti bejelentéssel párhuzamosan áll fenn.²³ Az általános célú AI-modell szolgáltatójának 53. cikk szerinti műszaki dokumentációja, felhasználási feltételei és szerzői jogi szabályzata a szolgáltatói értékelés bemenete.²⁴

4.4.3 Mit kell felmutatni egy auditon?

Az AI coding eszköz bevezetéséről szóló döntés akkor állja meg a helyét egy auditon, ha a Szervezet mindkét szabályozási szinten bizonyítékot tud felmutatni. A jóváhagyott use case-lista tulajdonossal, kontrollprofillal, MI-rendelet szerinti szerepkörrel és kockázati besorolással igazolja, hogy minden vállalati AI coding használat azonosított. A kontextuskonfiguráció, az agent-policy, a sandbox-konfiguráció és a token-hatókör a kontextusvédelmet és a legkisebb jogosultságot, a branch-szabály, a CODEOWNER-bejegyzés és a jóváhagyási történet az emberi review-t, a CI pipeline riportja, a registry, a lockfile és a tool-nyilvántartás a technikai kapukat dokumentálja. A közreműködői kikötéseket tartalmazó szerződés és az adatfeldolgozói megállapodás a szolgáltatói kontrollt, a képzési program és a jelenléti ív az AI-jártasságot, a runbook az incidensfelkészültséget, az agent- és admin-eseményexport pedig a nyomon követés átadását igazolja.

A kontrollok tehát nem AI-specifikus különkövetelmények, hanem a meglévő kiberbiztonsági és adatvédelmi kötelezettségek leképezései egy új eszközkörre. Ez egyben a válasz arra a vezetői

²¹ MI-rendelet 4. cikk, az (EU) 2026/1744 rendelettel (Digital Omnibus on AI, hatályos 2026. július 27-től) megállapított szöveg; az eredeti szöveg 2025. február 2-től volt alkalmazandó

²² MI-rendelet 9., 14. és 15. cikk; az alkalmazás halasztásáról az (EU) 2026/1744 rendelet rendelkezik

²³ MI-rendelet 26. cikk (1), (2) és (6) bekezdés, valamint 73. cikk

²⁴ MI-rendelet 53. cikk (1) bekezdés, alkalmazandó 2025. augusztus 2-től

kérdésre is, hogy miért nem elegendő a fejlesztők tájékoztatása. A tájékoztatás a kontextuskiadás kockázatát csökkenti, a végrehajtási és az ellátásilánc-kockázatot azonban csak a technikai kontrollprofil kezeli.

5. FELHASZNÁLT FORRÁSOK

- 2024. évi LXIX. törvény Magyarország kiberbiztonságáról (Kiberbiztonsági tv.); 7/2024. (VI. 24.) MK rendelet a biztonsági osztályba sorolás követelményeiről, valamint az egyes biztonsági osztályok esetében alkalmazandó konkrét védelmi intézkedésekről; 418/2024. (XII. 23.) Korm. rendelet a Magyarország kiberbiztonságáról szóló törvény végrehajtásáról
- Az Európai Parlament és a Tanács (EU) 2024/1689 rendelete (MI-rendelet), különösen a 3., 4., 6., 9., 14., 15., 25., 26., 53. és 73. cikk, valamint a III. melléklet; az Európai Parlament és a Tanács (EU) 2026/1744 rendelete (Digital Omnibus on AI); az Európai Parlament és a Tanács (EU) 2016/679 rendelete (GDPR), különösen a 28. és a 32. cikk
- VFP2 KiberVéd, A mesterséges intelligencia (MI) rendszerek besorolása az EIR-struktúrába, szakmai iránymutatás 2026/9; VFP2 KiberVéd, A naplózás fontossága és mélysége a NIS2-megfelelés szempontjából, jogi állásfoglalás 2026/9; VFP2 KiberVéd, A munkáltató által biztosított IT-eszközök magánhasználat, szakmai iránymutatás 2026/5
- OWASP Cheat Sheet Series, Secure Coding with AI, AI Agent Security és MCP Security (cheatsheetseries.owasp.org); NIST SP 800-218 Secure Software Development Framework (SSDF) 1.1 és NIST SP 800-218A SSDF Community Profile for Generative AI and Dual-Use Foundation Models (csrc.nist.gov); ENISA, Artificial Intelligence Cybersecurity Challenges és Cybersecurity of AI and Standardisation (enisa.europa.eu)

6. MELLÉKLETEK

6.1 AI coding use case engedélyezési adatlap

Mező	Kitöltendő tartalom
Use case neve és üzleti célja	
Tulajdonos, érintett repository vagy projekt	
Adat és forráskód érzékenysége	Alacsony / Közepes / Magas / Kritikus
AI-használati mód	Chatbot / Kódkiegészítő / Workspace-aware asszisztens / Agent / CI-agent
Szolgáltatási modell, EIR-besorolás és biztonsági osztály	SaaS / API / on-premise / hibrid, a kapcsolódó MKIK-iránymutatás szerint
MI-rendelet szerinti szerepkör és kockázati besorolás	Alkalmazó / Szolgáltató, nem nagy kockázatú / nagy kockázatú / vizsgálendő, indoklással
Az AI által olvasható és írható hatókör	
Shell- és tool-jogosultság, külső hálózati elérés	Nincs / Jóváhagyással / Allowlist
Használt credential típusa, production-hozzáférés	Tiltott / Kivételesen engedélyezett, indoklással
Kötelező biztonsági kapuk, reviewer vagy CODEOWNER	
Kontextuskizárási lista	
Szolgáltatói értékelés és szerződés hivatkozása	
Kockázati profil	Chat / Asszisztens / Kontrollált agent / Szigorított agent / Nem engedélyezhető
Jóváhagyó és a felülvizsgálat dátuma	

6.2 Go / No-Go ellenőrzőlista bevezetés előtt

Ellenőrzési pont	Állapot
Az eszköz jóváhagyott, és vállalati admin-kontroll alatt áll.	☐ Igen ☐ Nem
Az EIR- és szolgáltatási besorolás a kapcsolódó iránymutatás szerint megtörtént.	☐ Igen ☐ Nem
Az MI-rendelet szerinti szerepkör és a kockázati besorolás rögzítve van.	☐ Igen ☐ Nem
A szolgáltatói szerződés tartalmazza a kiberbiztonsági és adatkezelési kikötéseket.	☐ Igen ☐ Nem
Dokumentált, hogy milyen adat és kód kerülhet AI-kontextusba.	☐ Igen ☐ Nem
A titkok és az érzékeny fájlok kizárása technikailag be van állítva.	☐ Igen ☐ Nem
Az agent repository-, fájl-, shell-, hálózati és tool-jogosultsága minimális, production credential nincs a környezetében.	☐ Igen ☐ Nem
A védett branch, a PR és az emberi review kötelező.	☐ Igen ☐ Nem

Ellenőrzési pont	Állapot
A SAST-, secret- és függőségvizsgálat működik, a kritikus build-, CI- és IaC-fájlokra külön review van.	☐ Igen ☐ Nem
Az AI-specifikus incidensreakció és a secret-rotációs eljárás ismert.	☐ Igen ☐ Nem
A fejlesztők AI-jártassági és biztonsági oktatást kaptak, és ez dokumentált.	☐ Igen ☐ Nem

6.3 Szolgáltatói kérdőív rövidített változata

- Milyen szervezeti adatokat küld az IDE, a bővítmény vagy az agent a modellhez, és a szolgáltató használja-e ezeket modellfejlesztésre?
- Milyen megőrzési, törlési és vállalati kizárási lehetőségek állnak rendelkezésre?
- Milyen SSO-, MFA-, SCIM- és szerepköralapú jogosultságkezelési kontrollok érhetők el, és hogyan működik a központi kiléptetés?
- Kizárhatók-e érzékeny mappák és fájlok a projektindexelésből, és szabályozható-e a tool-hozzáférés?
- Milyen audit-, session- és agentművelet-események exportálhatók a szervezeti monitoringhoz?
- Milyen harmadik féltől származó modellek, bővítmények, toolok és alfeldolgozók vesznek részt, és hol történik a feldolgozás?
- Milyen változáskezelési és értesítési folyamat működik, ha a modell, a tool vagy az adatkezelési feltételek lényegesen megváltoznak?
- Hogyan működik az incidensértesítés, és igazodik-e a Szervezet 24, illetve 72 órás bejelentési határidőihöz?
- Vállalja-e a szolgáltató a Kiberbiztonsági tv. 6. § (5) bekezdés d) pontja szerinti kiberbiztonsági követelmények szerződéses rögzítését és az adatfeldolgozói megállapodás megkötését?