

AI ÍRJA A KÓDOT, DE KI GARANTÁLJA A BIZTONSÁGÁT?

SziTa'R
Rendszerek

A forráskódbiztonság új kihívásai és a SonarQube válaszai az AI-korszakában

Dr. Kabai Márton Tamás · tulajdonos, cégvezető, Szitár-Net Kft.



mVISZ Kiberbiztonsági Szakmai Fórum
2026. október 26.

A Szitár-Net Kft. a SonarSource hivatalos, Silver minősítésű szakmai partnere.

KIK VAGYUNK?

SziTa'R
Rendszerek

Szitár-Net Kft.
fejlesztés és IT-tanácsadás

Fejlesztés

IT-tanácsadás

Integráció

ERP-rendszerek

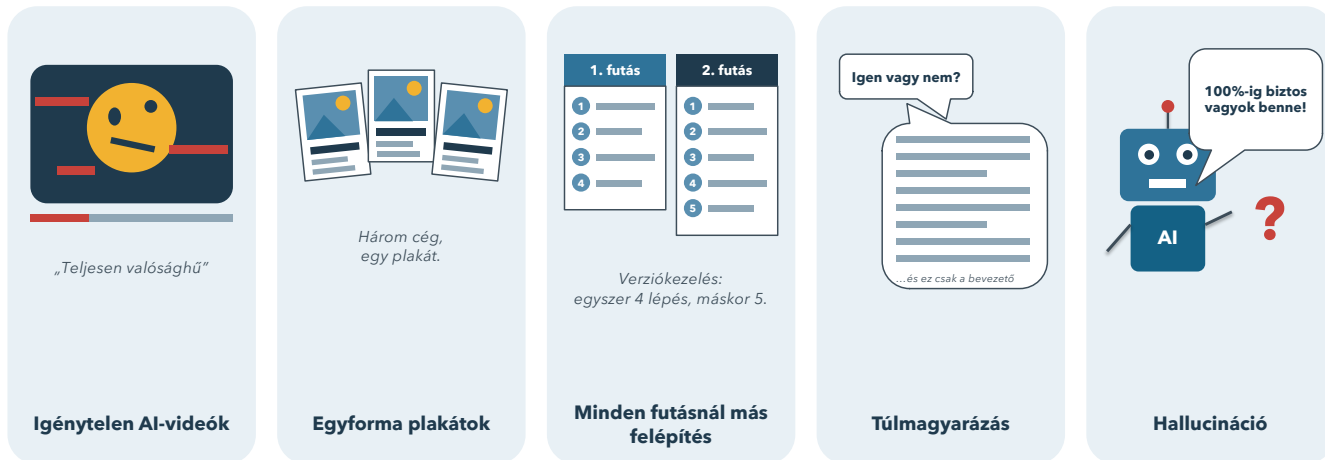


REFERENCIÁK



A SonarSource magyarországi, Silver minősítésű SVAR technikai partnere

AZ AI MINDENHOL OTT VAN - A STRUKTÚRA MÉG NEM



Bárhol használjuk az AI-t, **struktúra kell: szabály, ellenőrzés, felelős - és hozzá eszközök.**

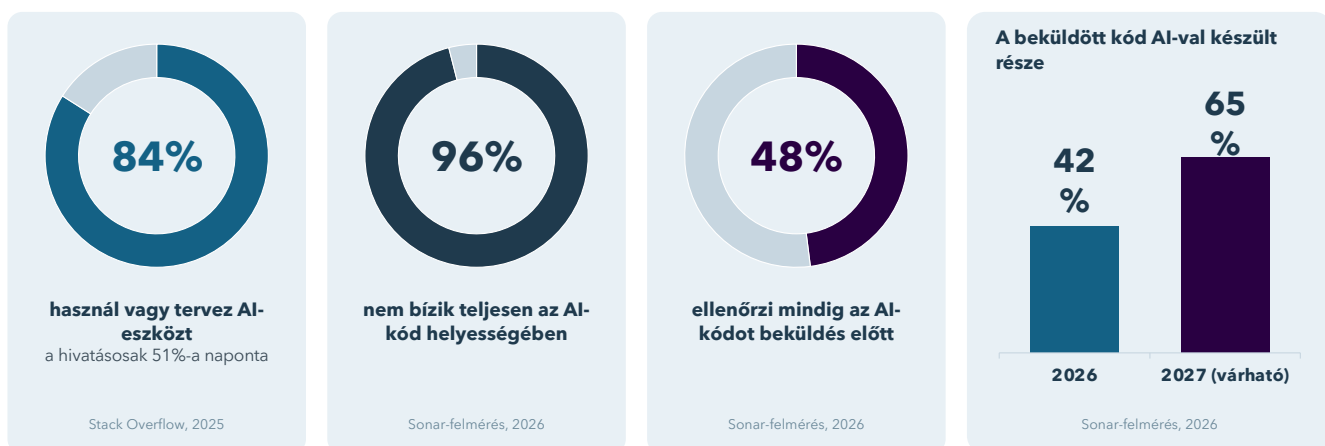


SziTa'R
Rendszerek

3

AZ AI MÁR NAPI GYAKORLAT - AZ ELLENŐRZÉS MÉG NEM

AI-val támogatott fejlesztés számokban



Az AI alkalmazásával együtt az ellenőrzési gyakorlatot is fejleszteni kell.



Források: Stack Overflow Developer Survey 2025; Sonar State of Code Developer Survey 2026 (a Sonar saját felmérése, 1100+ fejlesztő)

SziTa'R
Rendszerek

4

AI A FEJLESZTÉSBEN: TÍZ KOCKÁZAT

A szakemberek által felvázolt kockázatok

1 Egyetlen hibapont A fejlesztési folyamat néhány AI-szolgáltatótól függhet.	6 Adatvédelem Bizalmas forráskód és fejlesztési információ kerülhet külső szolgáltatóhoz.
2 Modellidentitás és verzióváltozás Egy frissítés megváltoztathatja a generált kód biztonsági jellemzőit.	7 Szakértelemvesztés A szakemberek egyre inkább az AI képességeire támaszkodnak.
3 Kompromittált modellek A manipulált modell sérülékeny kódot vagy hibás javaslatot generálhat.	8 Érdekellentét Ugyanaz az AI-szolgáltató generálja, ellenőrzi és javítja a kódot.
4 Gyengülő emberi kontroll A fejlesztők már nem tudnak minden változtatást kellően megérteni.	9 Láthatatlan technológiai irányítás Az AI befolyásolhatja a technológia- és beszállítóválasztást.
5 Szoftver-monokultúra Ugyanazok a hibaminták több, független alkalmazásban is megjelenhetnek.	10 Állami befolyás A központi AI-szolgáltatók működését szabályozási és geopolitikai döntések érinthetik.



5

ÚJABB MODELL, 2,9× TÖBB SÉRÜLEKENYSÉG

A Sonar saját mérése (GPT-5.5 és GPT-5.6 Sol, 4 444 Java-feladat) és egy független kutatás

Sonar-mérés: sérülékenységi találat / millió kódsor <table><tr><th>Model</th><th>Sérülékenységi találat / millió kódsor</th></tr><tr><td>GPT-5.5</td><td>68</td></tr><tr><td>GPT-5.6 Sol</td><td>197</td></tr></table>	Model	Sérülékenységi találat / millió kódsor	GPT-5.5	68	GPT-5.6 Sol	197	2,9× sérülékenységi sűrűség 68 helyett 197 találat millió kódsoronként
Model	Sérülékenységi találat / millió kódsor						
GPT-5.5	68						
GPT-5.6 Sol	197						
	78,66% → 81,99% funkcionális teszt sikeresség a Sol pontosabb kódot ír						
>60% Független kutatás, 2026. január A GPT-4o és az o3-mini strukturált biztonsági útmutatás nélkül írt PowerShell-szkriptjeinek több mint 60%-a nem biztonságos. SecGenEval-PS benchmark, arXiv:2601.06419	18 → 9 blokkoló sérülékenység / millió sor a kockázat a konkurencia, kriptográfia és erőforrás-kezelés felé tolódott						

Megjegyzés: a Sonar-számok statikus elemzési találatok, nem igazoltan kihasznált sérülékenységek.



Források: Sonar – An Evaluation of OpenAI GPT-5.6 Sol & Terra (2026. augusztus 6.); Zhang és mtsai – arXiv:2601.06419 (2026. január)



6

A SZAKEMBEREK SZAVAIVAL



Ne ugyanaz a szolgáltató ellenőrizze a kódot, amelyik megírta.

Shaked Reiner



A vibe codingnál az ember elengedi magát, és elfelejti, hogy a kód egyáltalán létezik.

Andrej Karpathy



Mintha három gyár állna billentyűzetek milliói mögött.

Shaked Reiner



A függetlenségünk egy részét a saját üzemeltetésű rendszerekkel őrizhetjük meg.

Shaked Reiner



A kevesebb kódolás sokkal több magyarázatot igényel.

Yossi Sassi



Talán itt az ideje, hogy tűz ellen tűzzel harcoljunk.

Arad Donenfeld



7

TÍZ AJÁNLÁS A BIZTONSÁGOS AI-FEJLESZTÉSHEZ

1

Független kódbiztonsági ellenőrzés

A generálástól elkülönülő, önálló biztonsági kontrollokkal.

SonarQube: független statikus kódelemzés

2

Automatizált biztonság a teljes folyamatban

Ellenőrzés a CI/CD-folyamatban, ne csak a fejlesztés végén.

SonarQube: a bukó quality gate megállítja a CI-t és a beolvasztást

3

Az emberi kontroll megőrzése

A kritikus kódot érteni kell, az AI döntéseit ellenőrizni.

SonarQube AI: a Remediation Agent PR-t nyit, az ember olvasztja be

4

AI-modellek folyamatos mérése

Modellfrissítés hatása, kódbiztonság, viselkedésváltozás.

SonarQube AI Code Assurance: AI-kódra szabott quality gate

5

Az AI-használat láthatóvá tétele

AI-eszközök, hozzáférések, adatáramlás, ügynökök feltérképezése.

SonarQube AI Code Assurance: AI-kódot tartalmazó projektek jelölése

6

Forráskód és érzékeny adatok védelme

Mi adható át külső AI-nak? A saját üzemeltetés önmagában nem elég.

SonarQube AI: saját, felhős vagy Sonar által hosztolt LLM választható

7

Az ellátási lánc ellenőrzése

Külső komponensek, modellek és függőségek is vizsgálандók.

SonarQube Advanced Security: SCA és SBOM

8

Többretegű fenyegetésszűrés

A szignatúraalapú védelmet viselkedésalapú detekció egészítse ki.

9

Szolgáltatói függőség csökkentése

Kerülni kell az egyetlen szolgáltatótól való túlzott függést.

SonarQube AI: az LLM-szolgáltató és a modell váltható

10

A biztonsági szakértelem fejlesztése

Fejlesztők és biztonsági szakemberek képzése az AI kockázataira.

SonarQube for IDE: kódolás közben megmagyarázza a hibákat



Források: Shaked Reiner, Gyebnár Gergő (Hactivity), Arad Donenfeld és Yossi Sassi előadásai; Sonar-funkciók: docs.sonarsource.com



8

1. SONARQUBE

Független kódellenőrzés

... ahol a szálak összefutnak...

SONARQUBE RÖVIDEN



40+

nyelv, keretrendszer és IaC

5

DevOps-platform: GitHub,
GitLab, Azure DevOps, Bitbucket
DC és Cloud

Saját

infrastruktúrán fut (on-premises)

KINEK MIT AD?

Fejlesztő

Hibák már a szerkesztőben,
magyarázattal és javítási iránnyal.

Vezető fejlesztő

Egységes szabályok, PR-elemzés,
quality gate a beolvasztás előtt.

Projektmenedzser

Alkalmazások és portfóliók,
irányítópultok, PDF-riport.

Biztonsági vezető

Sérülékenységek, megfelelési riport:
OWASP, PCI DSS, CWE, CRA.

Hibák

Sérülékenységek

Karbantarthatóság

Titkok

Lefedettség

Duplikáció

Enterprise licenctől: portfóliók, PDF- és megfelelési riportok.

SONARQUBE ÉS A DORA

Mit vár el a DORA a szoftverfejlesztéstől – és miben segít ebben a SonarQube?

DORA-ELVÁRÁS (RTS 2024/1774)

Forráskód-vizsgálat RTS 16. cikk (3)	Statikus forráskód-ellenőrzés; a sérülékenységek elemzése, cselekvési terv, követés
Tesztelés az integrációig RTS 16. cikk (4)	Biztonsági tesztelés legkésőbb az integrációs fázisban
Minden változás ellenőrzése RTS 17. cikk (1) a)	Minden változásnál ellenőrizni kell a biztonsági követelményeket
Nyílt forrású és külső kód RTS 16. cikk (8), 10. cikk (2) d)	Külső és nyílt forrású kód tesztelése; a könyvtárak nyomon követése
Bizonyítható ellenőrzés RTS 16. cikk (3) - követés	A cselekvési terv végrehajtását követni és igazolni kell

SONARQUBE-TÁMOGATÁS

SAST és hibakezelés minden sérülékenység nyomon követve, felelőssel
Elemzés minden PR-en és a CI-ben a hiba még a beolvasztás előtt kiderül
Quality gate a nem megfelelő változás nem kerül be
Advanced Security SCA, SBOM, napi újraértékelés
Riportok megfelelési és PDF-riport (Enterprise)

Nem a SonarQube fedi le: dinamikus tesztelés (DAST), a forráskód sértetlenségének védelme, incidenskezelés. **A megfelelésért a bank felel – a SonarQube eszköz, nem tanúsítvány.**



Források: DORA (EU) 2022/2554; RTS (EU) 2024/1774, 10., 16., 17. cikk (EUR-Lex); sonarsource.com – 2026. október 25.

SziTa'R
Rendszerek

11

2. AI-ESZKÖZÖK

AI-eszközök a SonarQube-ban

... ahol a szálak összefutnak...

SONARQUBE AI-ESZKÖZÖK

Hat AI-eszköz a SonarQube-ban – mindegyik röviden

” Talán itt az ideje, hogy tűz ellen tűzzel harcoljunk
Arad Donenfeld

AI Code Assurance

- Megjelöli az AI-kódos projektet
- Szigorúbb minőségi kaput ad rá
- Megbukik, ha a kód nem elég jó

AI CodeFix

- Egy hibára javítást javasol
- A felületen és az IDE-ben
- A fejlesztő dönt, elfogadja-e

Remediation Agent

- Magától kijavítja a hibát
- Pull requestet nyit a javítással
- Beolvasztani csak ember fog

Hunter Agent

- Rejtett logikai hibákat keres
- Jogosultsági réseket talál
- Amit a statikus elemzés nem lát

Sonar Vortex

- Az AI-ügynök mellett dolgozik
- Írás előtt irányt, kontextust ad, biztonsági tömörítést végez
- Írás után ellenőriz

MCP Server

- Összeköti az AI-ügynököt a SonarQube-bal
- Az ügynök látja a hibákat és a kaput
- Csak olvasás mód is beállítható

Hunter, Remediation és Vortex: külön előfizetés Enterprise vagy Data Center mellé. AI CodeFix: Enterprise és Data Center.



Forrás: docs.sonarsource.com (SonarQube Server 2026.5, MCP Server, AC/DC), lekérve 2026. október 21-22.

AI CODE ASSURANCE

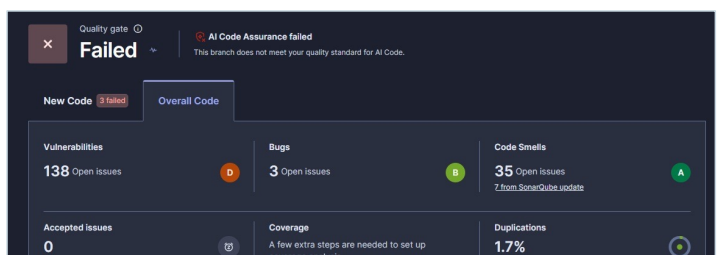
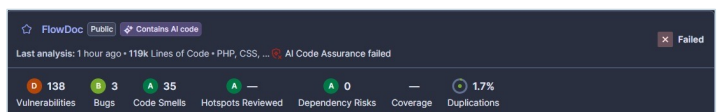
Megmutatja, hol van AI-kód – és szigorúbb mércét szab hozzá.

1 AI-kód jelölése
„Contains AI code” címke a projekten

2 AI által írt kódra szabott gate
Sonar way for agentic AI: 6 feltétel az új kódra

3 Ajánlott AI szabálykészlet
Sonar agentic AI quality profile

4 Igazolható szint
AI Code Assurance jelvény külső oldalra is



A demóprojekt AI-kódot tartalmaz, és nem teljesítette az AI-ra szabott kaput.

Az automatikus AI-kód-felismerés elavult: a jelölés tudatos döntés.



Forrás: docs.sonarsource.com (Server 2026.5, MCP, AC/DC), 2026. október 21-22. · Képernyőkép: demópéldány

AI CODEFIX

Egy kattintás: javítási javaslat a hibára - a fejlesztő dönt.

- 1

Javítás egy kattintásra
nagy nyelvi modell javasol egy kiválasztott hibára
- 2

Ott, ahol a fejlesztő dolgozik
a SonarQube felületén és az IDE-ben
- 3

8 nyelv
Java, C#, C++, Python, JS, TS, HTML, CSS - kiválasztott szabályokra
- 4

Választható modell
akár saját modell, internet nélkül

Enterprise és Data Center edícióban.

VÁLASZTHATÓ MODELL

- Sonar által hosztolt

a Sonar szolgáltatása
- Azure OpenAI

a cég saját bérlete
- AWS Bedrock

a cég sajátja
- Saját modell

internet nélkül is működik



Forrás: docs.sonarsource.com (SonarQube Server 2026.5, MCP Server, AC/DC), lekérve 2026. október 21-22.



15

REMEDATION AGENT

Kijavítja a hibát, és pull requestet nyit - "beolvasztani" csak a fejlesztő tudja.

- 1

Magától javít
a backlog hibáira, a bukó quality gate-ű PR-ekre, vagy egy Hunter-leletre
- 2

Ellenőrzött javítás
a javítást a Vortex-elemzés ellenőrzi
- 3

Pull requestet nyit
GitHub, GitLab, Azure DevOps
- 4

A végén az ember dönt
beolvasztani soha nem fog

Saját infrastruktúrán fut - külön előfizetés.

ProgPHP / FlowDoc / Merge requests / 124

Delete only the signed-in user's avatar files

sonarqube-remediation-agent requested to merge `remediation/1f963fba-59be...` into `main` 44 minutes ago

Overview 0

Commits 1

Pipelines 0

Changes 3

This PR was created because a team member assigned this Hunter Agent finding to the Remediation Agent.

Problem

ProfileController::actionDeleteAvatar(\$id) built the upload directory straight from the route parameter and handed the resulting paths to UploadForm::delete(), which unlinks whatever exists without further validation. The controller's access rules only require an authenticated session and a POST, so any signed-in user could enumerate ids and destroy other people's `avatar`, thumbnail and original image files. The deletion is irreversible.

Proposed Fix

The action no longer takes a caller-supplied identifier. It first resolves the account through the existing private `findModel()`, which looks up `Yii::$app->user->id` and raises `NotFoundException` when there is no matching user, and builds the three file paths from that id. The deletion target is therefore bound to the session identity and can no longer be influenced by the request. The one caller, the delete button in `views/profile/tabs/_update_avatar.php`, now links to the bare `delete-avatar` route; the POST verb filter and confirmation dialog are unchanged. For a legitimate request the resulting paths are identical to before.

Valós példa: a Remediation Agent PR-je egy Hunter-lelet javítására.



Forrás: docs.sonarsource.com (Server 2026.5, MCP, AC/DC), 2026. október 21-22. · Képernyőkép: demópéldány

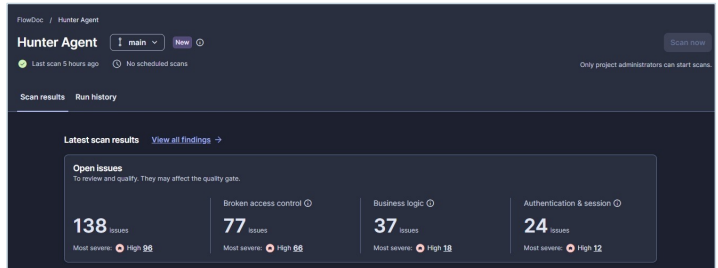


16

HUNTER AGENT

Megtalálja azokat a hibákat, amelyeket a statikus elemzés nem lát.

- 1 Rejtett hibák**
a statikus elemzés számára láthatatlan részek
- 2 Jogosultság és logika**
hozzáférés-vezérlés, üzleti logika, hitelesítés
- 3 Triázs**
a csapat átnézi; a quality gate-et is befolyásolhatja
- 4 Átadás javításra**
egy lelet a Remediation Agentnek rendelhető



Demóprojekt: 138 nyitott lelet egyetlen futásból.



Claude Opus 4.8 és Sonnet 4.6 modellel · külön előfizetés.

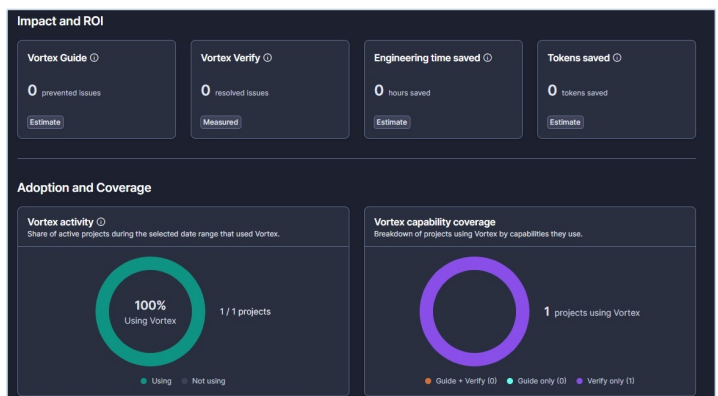


Forrás: docs.sonarsource.com (Server 2026.5, MCP, AC/DC), 2026. október 21-22. · Képernyőkép: demópéldány

SONAR VORTEX

Az AI-ügynök mellett dolgozik: előbb irányt ad, utána ellenőriz.

- 1 Guide - írás előtt**
kódismeret, irányelvek, licencszabályok az ügynöknek
- 2 Verify - írás után**
ellenőrzés CI-szintű pontossággal
- 3 Népszerű ügynökökkel**
Claude Code, Codex, Cursor, GitHub Copilot CLI...
- 4 Mérhető hatás**
irányítópult: megelőzött és megoldott hibák, megtakarított idő



A demópéldány Vortex-irányítópultja.

Saját infrastruktúrán · külön előfizetés · a Remediation Agent javításait is ellenőrzi.



Forrás: docs.sonarsource.com (Server 2026.5, MCP, AC/DC), 2026. október 21-22. · Képernyőkép: demópéldány

MCP SERVER

Az AI-ügynök közvetlenül kérdezheti a SonarQube-ot.

MIKOR?
Kódolás közben, az AI-ügynökből: Claude Code, Codex, Cursor, Copilot, Gemini CLI...

MIT CSINÁL?

- Kódot elemeztet
- Hibákat és quality gate-állapotot kérdez le
- Függőségi kockázatot néz

MIT KAPUNK?
Az ügynök a SonarQube adataiból dolgozik, nem találgat.

AI-ügynök ↔ **MCP Server** ↔ **SonarQube**

BIZTONSÁG

- Csak olvasás mód - az ügynök nem módosít
- Szűrhető, mely eszközöket érheti el
- Banknál a modell is legyen belső

A SonarQube Serveren vagy helyben futtatható.

MELYIK FUNKCIÓT SEGÍTI?

Melyik AI-eszköz melyik SonarQube-funkciót segíti?

	Kódolás (IDE, ügynök)	Pull request	Quality gate	Hibák kezelése	Szabályok	Függőségek (AS)
AI Code Assurance		●	●		●	●
AI CodeFix	●			●		
Remediation Agent		●	●	●		
Hunter Agent			●	●		
Sonar Vortex	●				●	●
MCP Server	●		●	●		●

Az alap mindenhol a SonarQube független elemzése - az AI-eszközök erre épülnek, nem helyettesítik.

3. AS MODUL

Advanced Security: függőségek és forráskód

... ahol a szálak összefutnak...

AZ ADVANCED SECURITY KÉT PILLÉRE



SCA

Szoftverösszetétel-elemzés

„Mi van a függőségeinkben?”

Feltérképezi a teljes függőségi fát a package manager fájljaiból, majd összeveti ismert sérülékenységi, malware- és licencadatokkal.

Nyelvek: JS/TS · Java · Kotlin · Scala · Python · C#/.NET · Go · Ruby · Rust · PHP



Fejlett SAST

Kiterjesztett statikus elemzés

„Mi történik, ha a mi kódunk találkozik velük?”

A taint analysist átvezeti a nyílt forrású függőségek belsejébe, így a függőséghatáron átnyúló injection-jellegű hibák is előjönnek.

Nyelvek: JavaScript/TypeScript · Java · C#/.NET – szűkebb lista, mint az SCA-é

Nincs külön eszköz, külön pipeline vagy külön felület – a találatok ugyanott jelennek meg, ahol ma a kódminőségi issue-k.

ADVANCED SECURITY: A KÓD ALATTI RÉTEGEK

Fizetős kiegészítő Enterprise licenctől: a függőségeket is vizsgálja

Demo / Overview

Overview

main

Downloadable reports

Set as homepage

Bind project

103 Lines of Code

Version 0.0.1-SNAPSHOT

Last analysis 2 minutes ago

0 Open issues A	0 Open issues A	0 Open issues A
Accepted issues 0 Valid issues that were not fixed	Coverage 0.0% On 8 lines to cover.	Duplications 0.0% On 123 lines.
Security Hotspots 0 Deprecated A	Dependency risks 158 E	

TRANZITÍV FÜGGŐSÉGEK: SCA

A sérülékenység a függőség függőségében is lehet

Alkalmazásunk
saját kód

Könyvtár A
direkt

Könyvtár B
direkt

Könyvtár C
transzitiv

Könyvtár D
transzitiv

sérülékenység

tiltott licenc

1

Látja a teljes fát
nem csak a közvetlen hivatkozásokat

2

Naponta újraértékel
az új sérülékenység elemzés nélkül is megjelenik

3

A forráskód marad
csak a függőséglista-lekérdezés megy ki

 Forrás: sonarsource.com (Advanced Security) és docs.sonarsource.com (Server 2026.5), 2026. október 21-22.

 24

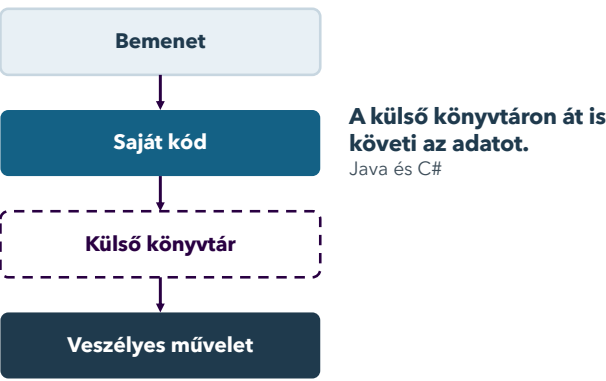
SBOM ÉS INTELLIGENS SAST

Mi van a szoftverünkben, és hová jut a felhasználói bemenet?

SBOM

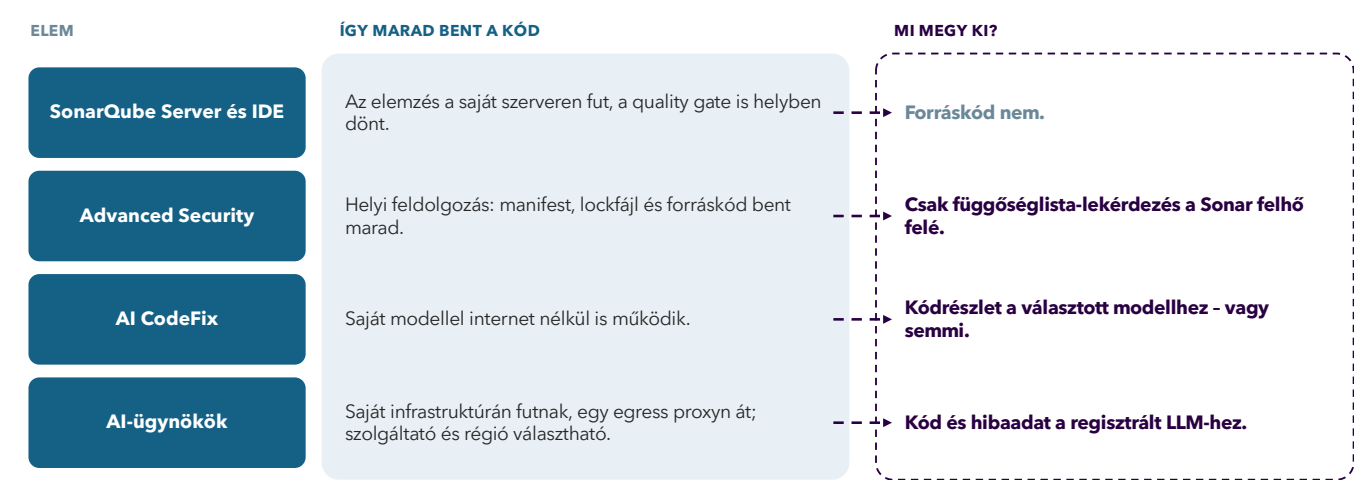


Intelligens SAST



HOGYAN MARAD BENT A FORRÁSKÓD?

Mi megy ki a SonarQube-ból és elemeiből - és hogyan marad bent a forráskód?



4. ZÁRÁS

Struktúra + független ellenőrzés

... ahol a szálak összefutnak...

ÖSSZEFOGLALÁS

Az előadás egy mondatban

1 Az AI mindenhol ott van -
struktúra nélkül kockázat

2 A működő kód még nem
biztonságos kód

3 Amit ember nem tud
átnézni, azt gépnek kell

4 A DORA is elvárja a
forráskód-ellenőrzést

STRUKTÚRA

Szabály: milyen AI, mire, milyen adattal

Felelős: a döntés emberi marad

Mérés: modellváltás után újra



FÜGGETLEN ELLENŐRZÉS

SonarQube: a kód és a quality gate

Advanced Security: a függőségek

AI-eszközök: javasolnak - ember dönt



**Biztonságos
AI-fejlesztés**

MIT VIGYÜNK HAZA?

Öt lépés, amivel holnap el lehet indulni



Strukturált AI-használat

Szabály: milyen eszköz, mire, milyen adattal – és ki felel érte.



Ellenőrzés minden AI-eredményre

A mennyiséget gép ellenőrzi, a döntés emberi marad.



Más modellel ellenőriztessünk

Ne ugyanaz a modell vagy szolgáltató nézze át, amelyik írta – kérjünk második véleményt.



Statikus kódelemzés AI-támogatással

Független SonarQube-elemzés + AI-képességek: AI-kód kapuja, javítás, rejtett hibák keresése.



Mérjünk és kövessünk

Modellváltás után újramérés; a függőségeket és az SBOM-ot is kövessük.



Köszönöm a figyelmet és a lehetőséget!

SziTa'R
Rendszerek



Dr. Kabai Márton Tamás

tulajdonos, cégvezető, Szitár-Net Kft.
Marton.kabai@szitar.hu · +36 70 609 3338

A Szitár-Net Kft. a SonarSource hivatalos, Silver minősítésű szakmai partnere.