ENGINE V4 · ŘÍZENO AGENTEM

LITHIUM ARGON

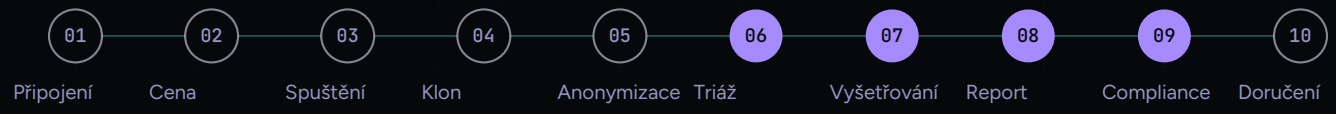
Celý ARGON flow a AI SAST engine V4 — architektura, pipeline a technické specifikace

// CELÝ PRŮBĚH ARGON, KROK ZA KROKEM

Deset přesně definovaných kroků. Šest je orchestrace. Čtyři je AI agent, který vyšetřuje váš kód.

Od přístupu k repozitáři až po report připravený na audit prochází každý sken stejným desetikrokovým flow — plně automatizovaně, bez zásahu vývojáře. Šest kroků je procesních a orchestračních: připojení k repozitáři, stanovení ceny, spuštění skenu, izolovaný klon, anonymizace zdrojového kódu a doručení výsledku. Kolem nich stojí samotný AI SAST engine V4 — interně osmifázový pipeline, nejprve deterministický, pak agentní, podrobně na straně 2.

END-TO-END WORKFLOW — 10 KROKŮ



○ orchestrace · procesní krok, žádný krok enginu

● AI SAST engine V4 · kroky 06–09, podrobně na straně 2

VŠECH 10 KROKŮ, STRUČNĚ

01 · Připojení
Přístup pouze pro čtení k libovolnému Git hostingu — GitHub, GitLab, Bitbucket, nebo self-hosted instance.

02 · Cena
Deterministický průchod file filtrem přes syrový repozitář stanoví cenu ještě před jakýmkoliv AI voláním nebo klonem.

03 · Spuštění
Naplánováno vámi, zadáno manuálně, nebo spuštěno programově přes ARGON API.

04 · Klon
Izolované zpracovatelské prostředí, žádný přístup k externí síti, zdrojový kód okamžitě smazán po analýze.

05 · Anonymizace
Tři sekvenční průchody odstraní síťová data, tajemství a vlastní tokeny — podrobně na straně 2.

06 · Triáž
File filtr, AI mapa architektury a deterministický OSV.dev scan závislostí/CVE se zkombinují do priority. Detail enginu na straně 2.

07 · Vyšetřování
AI agent prochází kód vlastními nástroji, hledá zranitelnosti a pak každého kandidáta znovu ověří, než ho potvrdí. Detail enginu na straně 2.

08 · Report
Sestaví manažerské shrnutí, technické shrnutí, detailní technický report a strukturovaný JSON. Detail enginu na straně 2.

09 · Compliance
Potvrzené nálezy deterministicky namapovány na klauzule NIS2, CRA, ISO 27001 a TISAX, plus SBOM export. Detail enginu na straně 2.

10 · Doručení
Reporty odeslány e-mailem okamžitě; každý běh navíc zapsán do verzovaného, nezměnitelného auditního repozitáře.

Pokračování na straně 2 — kompletní osmifázový vnitřek V4 enginu.

KROK 05 · ANONYMIZACE ZDROJOVÉHO KÓDU — PŘED SPUŠTĚNÍM ENGINU

1 · SÍŤ

IP a MAC adresy nahrazeny konzistentními zástupnými hodnotami.

2 · TAJEMSTVÍ

API klíče, tokeny a connection stringy automaticky detekovány a nahrazeny.

3 · MANUÁLNÍ TOKENY

Řetězce definované projektem — názvy, hostitelé, identifikátory — nahrazeny jako poslední.

Probíhá dřív, než se kód dostane k V4 enginu níže nebo opustí izolovaný klon.

KROKY 06-09 · V4 ENGINE — OSM VNITŘNÍCH FÁZÍ

AI architekt sestaví kompletní mapu vaší kódové základny, poté ji AI vyšetřovatel aktivně prozkoumá vlastními nástroji, čte kód a sleduje logiku napříč moduly — na rozdíl od SAST založeného na pravidlech, který jen porovnává známé vzory s pevnou sadou pravidel. Každý kandidát je pak nezávisle prověřen kvůli falešným poplachům, závažnosti a zneužitelnosti, než se dostane do reportu.

01

FILTR

Deterministický tvrdý filtr plus AI klasifikační průchod. Vylučuje binárky, generované soubory, závislostní stromy, ne-kódová aktiva.

`MODEL: Haiku 4.5`

02

ARCHITEKT

Zmapuje moduly, vstupní body, datové toky a vysoce rizikové oblasti z filtrovaného kódu.

`MODEL: Haiku 4.5 + Sonnet 5`

03

DEPENDENCY SCAN

Každá závislost zkontrolována proti databázi CVE OSV.dev; sestaven kompletní SBOM. Ověřitelný fakt, ne hypotéza.

04

TRIÁŽ

Zkombinuje mapu architektury a nálezy ze scanu závislostí do priority pro vyšetřování. Stejný vstup, stejné pořadí, pokaždé.

05

VYŠETŘOVÁNÍ

Agentní smyčka s nástroji čte, prochází a označuje kandidátní nálezy napříč kódovou základnou — ne pevný seznam pravidel.

`MODEL: Opus 5`

06

POTVRZENÍ

Odmítá falešné poplachy a kontroluje mitigace jinde v kódu; hodnotí závažnost, zneužitelnost, dopad a opravu.

`MODEL: Sonnet 5`

07

REPORT

Sestaví manažerské shrnutí, technické shrnutí, detailní technický report a strukturovaný JSON.

08

COMPLIANCE

Deterministicky namapuje potvrzené nálezy na klauzule NIS2, CRA, ISO 27001 a TISAX; exportuje SBOM.

Modely se pro každý krok konfigurují v registru AI modelů ARGON a upgradují nezávisle na tomto dokumentu — výše jsou aktuální výchozí hodnoty. Kroky bez tagu modelu neběží na žádné AI: deterministický kód, stejný vstup, stejný výstup, vždy.

Pokračování na straně 3 — co obsahuje nález a formát compliance evidence reportu.

CO ZÍSKÁTE

CO OBSAHUJE KAŽDÝ NÁLEZ

Závažnost (critical · high · medium · low · info), hodnocení zneužitelnosti (potvrzeno zneužitelné · pravděpodobně zneužitelné · teoretické · nezneužitelné) — samostatná osa od závažnosti — dopad a doporučenou opravu. Reporty se přizpůsobují čtenáři: srozumitelný jazyk pro vedení, plný technický detail pro techniky — přesný soubor, řádek a opravu.

CO SE Z ANALÝZY VYLUČUJE

Binárky, mediální soubory, build výstupy, závislostní stromy, lockfiles a testovací fixtures jsou vyloučeny ještě před začátkem analýzy. Druhý AI průchod pak klasifikuje, co zbylo — bezpečnostně relevantní konfigurace (web.xml, application.yml, JSP, connection stringy) zůstává vždy v rozsahu.

COMPLIANCE EVIDENCE REPORTS

Pro každý povolený rámec — NIS2, CRA, ISO 27001, TISAX — samostatný, čistě deterministický evidenční report: potvrzené nálezy namapované na konkrétní klauzule rámce, rozklad podle závažnosti a zneužitelnosti, harmonogram povinností a sazby pokut za nedodržení. Dodáváno jako JSON, Markdown a branded PDF, plus SBOM export ve formátech SPDX a CycloneDX. Ne certifikát shody — technický důkazní materiál pro váš compliance tým.

Pokračování na straně 4 — kompletní tabulka technických specifikací.

TECHNICKÉ SPECIFIKACE

Velikost repozitáře	Až 1 GB efektivního zdrojového kódu (po filtraci)
Rychlost analýzy	Závisí na velikosti repozitáře a komplexitě kódu — malý repozitář (do 4 MB) dokončen zhruba za 3 minuty
Spouštění	Naplánovaně (denně / týdně / měsíčně / čtvrtletně / ročně), zadáno manuálně, nebo na vyžádání přes ARGON API
Rozsah testování	Plně odvozeno agentem — žádný předem definovaný katalog testů ke konfiguraci nebo údržbě
Vyšetřovací nástroje	Agent sám čte soubory, čte rozsahy řádků, prohledává kód, dohledává info o závislostech, vypisuje adresáře a označuje nálezy
Manifesty závislostí	npm, Python (pip / Poetry / Pipenv), PHP Composer, .NET, Go a Ruby lockfiles parsovány na přesné verze — Maven/Gradle a .csproj zatím nepodporovány
Pokrytí zranitelností	Zaměřeno na OWASP, CWE a známé CVE prostřednictvím kontrol verzí závislostí, mimo jiné uznávané standardy
Compliance rámce	NIS2, CRA, ISO 27001, TISAX — deterministické mapování klauzulí, žádná AI v tomto kroku; JSON, Markdown, PDF a SPDX/CycloneDX SBOM pro každý rámec
Reportování pokrytí	Podíl relevantních souborů, které byly skutečně prověřeny, součást každého reportu
Formáty reportů	Manažerské shrnutí · technické shrnutí · detailní technický report · strukturovaný JSON — každý report dostupný i jako branded PDF
Auditní stopa	Každý dokončený běh zapsán do vyhrazeného, verzovaného Git repozitáře
Anonymizace	Tři sekvenční průchody dokončeny ještě před zahájením AI analýzy
Prostředí spuštění	Izolovaný klon bez přístupu k externí síti; zdrojový kód okamžitě smazán po analýze
Nasazení	Plně spravovaná, kontejnerizovaná platforma — nic se neinstaluje na vaší infrastruktuře