cloudfield

comma_0

Bezpečná AI

Podpora služeb moderního státu

Vilém Markovič

Komplexní přístup k zabezpečení AI aplikací

cloudfield

comma_0

Kdo má zodpovědnost za AI?

Za AI aplikaci v organizaci obvykle nenese odpovědnost jeden útvar, ale je to rozdělené mezi několik rolí. U dobře řízené firmy (hlavně pokud řešíte compliance typu NIS2, governance, bezpečnost, auditovatelnost) to bývá takto:

1. **Business owner / Product owner (hlavní byznysová odpovědnost)**
2. **IT / Engineering / Enterprise Architecture**
3. **Security / CISO organizace**
4. **Legal / Compliance / DPO**
5. **Risk / Internal Audit**
6. **AI Governance Board / Steering Committee**

cloudfield

comma_0

AI aplikace = Chain of Security

AI služba je jen JEDNA komponenta celkového řešení

Bezpečnostní vrstvy stacku:

IDE vývoj • Frontend • API Gateway • Business Logic

AI Service • Data Storage • Network

Identity & Access • Monitoring

⚠ **Zabezpečená AI + nezabezpečená infrastruktura = FAIL**

cloudfield

comma_0

EU AI Act — High-Risk Systémy

Finanční sektor | Státní správa - GDPR kategorie | Kritická infrastruktura

Povinné kontroly:

✓ Evidence use case & risk assessment

✓ Evidence modelů (verze, data, metriky)

✓ Klasifikace dat (GDPR kategorie)

✓ Input validation (anti-injection)

✓ Output filtering (PII, toxic)

✓ Audit trail (min. 6 měsíců)

✓ Human oversight

🚨 Sankce: 7% obratu nebo €35M

cloudfield

comma_0

Pre-Deployment Checklist

- ☐ Klasifikace dle EU AI Act
- ☐ Risk assessment dokumentace
- ☐ API Security: OAuth 2.0 + mTLS
- ☐ Input/Output validation aktivní
- ☐ Model integrity checks
- ☐ Data retention (6+ měsíců)
- ☐ Monitoring & alerting
- ☐ Encryption (rest + transit)
- ☐ RBAC + Least Privilege
- ☐ Supply chain assessment
- ☐ Compliance verification

cloudfield

comma_0

OWASP LLM Top 10

1. Prompt Injection — manipulace vstupy
2. Sensitive Information Disclosure - LLM vyzradí data
3. Supply Chain Vulnerabilities – kompromitovaný model z repo
4. Data and Model Poisoning - útočník otráví data nebo knowledge base
5. Model Failure —aplikace slepě věří výstupu modelu
6. Excessive Agency— model má příliš široká oprávnění
7. System Prompt Leakage— únik interních instrukcí
8. Vector and Embedding Weaknesses — útok na retrieval vrstvu
9. Misinformation / Hallucination Abuse — model generuje nepravdy a aplikace je bere jako fakt
10. **Un**bounded Consumption – LLM spotřebovává nekontrolovaně zdroje

cloudfield

comma_0

Maximalizuj benefit ve vývoji | Minimalizuj risk v produkci

Security Best Practices

OpenAI: Private EP • Content Filter • Managed ID

Blob: CMK • Versioning • Soft delete

Key Vault: Purge protection • RBAC • HSM

APIM: Rate limit • OAuth • IP filter

Policy: Compliance • Auto-remediation

Defender: CSPM • Vuln scan • Threats

Monitor: Alerts • Logs • Retention

Entra: Conditional Access • MFA • PIM

✗ Kritická Selhání

Hardcoded API keys → leak

Veřejné API bez auth → zneužití

Bez validation → prompt injection

Model bez integrity → backdoor

Nedostatečné logy → žádný audit

Přímý production → žádný staging

Vendor bez diligence → supply chain

Ignorování AI Act → pokuty 7%

cloudfield

comma_0

4 Fáze: Volnost → Spolehlivost

PRE-POC: Volný agent v sandboxu

Maximální volnost • Rychlá iterace

POC: Agent kompiluje workflow

Navrhuje řešení • Lidský přezkum

TEST: LLM

Deterministika

PRODUKCE: Pouze kód

LLM (pro use case) • Plná auditovatelnost

Stupně volnosti monotónně klesají

cloudfield

comma_0

Příklad: Jira Agents

✗ Varianta A — Vendor-managed:

Agent v production • Černá skříňka

Žádný sandbox • Exit = přepis

→ Verdikt: 0/5 → REJECT

✓ Varianta B — Vlastní platforma:

Pre-schválené workflows

Sandbox • Full traces • Reverzibilita

→ Verdikt: 5/5 → ADOPT

cloudfield

comma_0

O čem to celé je?

Chief Product Officer z Cisco**, AI (RSAC 2026), Jeetu Patela**

„Před nasazením AI agentů je potřeba mít zero-trust arch. a neprůstřelnou governance a monitoring, jinak nebude jejich použití nikdy bezpečné.“

AI Agenti:

„velmi inteligentní teenageři bez strachu z následků“

cloudfield

comma_0

Nedopusťme

cloudfield

comma_0

Point of No Return

Pokud AI nelze odstranit:

✗ Není to triviální

✗ Černá skříňka

✗ Vendor lock-in

✗ 100× pomalejší fallback

⚠ **Governance DNES = prevence lock-in ZÍTRA**

"AI už nemaluje obrázky. Řídí přístup k barvám."

cloudfield

comma_0

Q&A

Děkuji za pozornost

Vilém Markovič
vilem.markovic@comma0.io

"AI už nemaluje obrázky. Řídí přístup k barvám."

cloudfield

comma_0

Backup slides

cloudfield

comma_0

AI coding a past vibecodingu

Nástroje, které máme k dispozici:

Claude Code · GitHub Copilot · Codex · Cursor · ChatGPT · Gemini · ...

Realita roku 2026

AI vygeneruje fungující kód za vteřiny. Zrychlení 5×, 10×, někde 100×.

Vibecoding — past, ne workflow

Vývojář klikne **Accept** a doufá, že to funguje.

AI chrlí kód, který vypadá dobře a nedělá, co má.

Bez review, testů a porozumění je to gambling, ne vývoj.

cloudfield

comma_0

Vývojář vede — řízený proces

U nás v Cloudfieldu — cf-powers

Plugin do Claude Code. Skills, které vedou agenta strukturovaným vývojem.

Workflow: **/analyse** → **/write-plan** → **execute**

Cross-check agenti · BA + Dev + Security + Performance = 4× kontrola

TDD + verifikace · testy před implementací, verification-before-completion

Claude Code kóduje / Codex reviewuje · dva nezávislé modely

Zkušený vývojář je NENAHRADITELNÝ

Řídí proces · přebírá zodpovědnost · MUSÍ rozumět dodanému kódu

Extrémní zrychlení = extrémní zodpovědnost.

cloudfield

comma_0

AI doporučuje i závislosti

"Použij tenhle balíček." — řekne AI agent. A co když?

Zastaralý · verze pět let stará, nezáplatované CVE

Nevhodná licence · GPL ve vašem proprietárním produktu

Malicious · kompromitovaný maintainer, supply chain útok

Typosquat · **liteIlm** s velkým I místo malého L vs **litellm**

Co se reálně dělo v 2025–2026:

· Kampaň TeamPCP — kompromis LiteLLM → Trivy → Checkmarx → npm

· Shai-Hulud worm — npm worm replikující se přes stovky balíčků

· chalk & debug phishing — knihovny se stovkami milionů stažení týdně

AI doporučí. Vy nainstalujete. Stačí jeden krok.

cloudfield

comma_0

Shieldoo Gate — naše odpověď

Scan-on-demand package proxy

Balíček projde, až projde scanem. Žádné změny v klientech kromě URL.

Ekosystémy: PyPI · npm · Docker · NuGet · Maven · RubyGems · Go

7 detekčních vrstev:

GuardDog · Trivy · OSV · Typosquat · Version Diff · Reputation · AI

Open-source · Apache 2.0 · self-hosted

github.com/cloudfieldcz/shieldoo-gate

A to nejdůležitější:

Sami jsme ho postavili s AI — ALE v řízeném procesu (cf-powers).

7 dní práce. Production-ready. Funguje to. A dále rozvíjíme ..