Český rozhlas

Pravidla pro užívání umělé inteligence v Českém rozhlasu S/2024/001/GŘ//0
Příloha č. 3

Pravidla využití AI pro generování zdrojových kódů k programům a aplikacím

Čl. 1 Úvod

1. Tato příloha č. 3 směrnice AI specifikuje pravidla pro využívání nástrojů AI pro generování zdrojových kódů k programům a aplikacím v ČRo (dále jako „zdrojový kód“). Generování zdrojových kódů zahrnuje i úpravy, opravy či jiné dílčí činnosti na zdrojovém kódu, ke kterým programátor AI využije.

2. Pro způsoby užití uvedené v těchto pravidlech platí, že u nich byla posouzena rizika a dopady v souladu se směrnicí Pravidla pro užívání umělé inteligence v Českém rozhlasu.

Čl. 2 Výklad pojmů pro účely směrnice AI

1. Zdrojový kód je základní formou programového kódu, je čitelný pro programátora a zapsaný v určitém programovacím jazyce (např. Python, C#).

2. Komplexní zdrojový kód se vyznačuje svou rozsáhlostí a strukturou, která znemožňuje rychlou kontrolu. Zdrojový kód může být závislý na dalších softwarových komponentách, jako jsou například externí knihovny nebo volání služeb API. Komplexní zdrojový kód, který se na takové komponenty odkazuje, musí být analyzován zvláště detailně s ohledem na možná rizika a to především proto, že není přehledný.

3. Po překladu v překladači nebo interpretu z něj vznikne kompilovaný (bajtkód, binární nebo strojový) kód, který již není pro člověka čitelný.

Čl. 3 Pravidla pro generování zdrojových kódů

1. Český rozhlas neurčuje konkrétní nástroje, které smí programátoři využít pro generování zdrojových kódů. Programátoři smí ke generování zdrojových kódů využít nástroje AI dostupné na trhu bez omezení.

2. Každý programátor, který generuje zdrojové kódy pomocí AI, musí být řádně proškolen v oblasti pravidel užití AI v ČRo.

3. V souladu s principem osobní odpovědnosti nese za vygenerovaný zdrojový kód plnou odpovědnost osoba, která jej pomocí AI vygenerovala stejně jako by jej vytvářela sama. Programátor je povinen provést kontrolu správnosti vygenerovaného kódu z hlediska jeho funkčnosti, testovatelnosti a bezpečnosti například metodou tzv. unit testování, tedy ověření funkčnosti a korektnosti dílčích částí kódu. Zvýšená kontrola bezpečnosti kódu je nutná na všech výstupech, především však tam, kde má kód přesah do veřejně dostupných služeb ČRo.

4. Charakter vstupních dat může ovlivnit chování kódu. Vstupní data je nutné vždy validovat a očistit od citlivých informací (například osobní údaje, přístupové údaje, hesla, přístupové

tokeny)., Pomocí jejich manipulace by v opačném případě mohlo dojít k nesprávné činnosti kódu, nebo ovlivnění bezpečnosti (například SQL útok).

5. Pokud je zdrojový kód určen pro práci s daty, které mohou mít charakter chráněných dat (např. osobní údaje), je třeba před nasazením takového kódu do provozu nejprve ověřit, že budou dodrženy všechny požadavky platných právních předpisů (např. z hlediska GDPR a předání osobních údajů třetím osobám či do třetích zemí). Zvýšené riziko a tedy i zvýšené opatrnosti pak je třeba dbát zejména v případě generování komplexních zdrojových kódů, které nejsou přehledné.

6. Vygenerovaný zdrojový kód je třeba udržovat aktuální vzhledem ke změnám v navazujících softwarových systémech a sítích. Pravidelně je třeba kontrolovat kód a aktualizovat všechny závislosti, pokud existují. Je třeba sledovat vznik možných slabých míst a zavádět opatření pro minimalizaci rizik.

7. Programátor má povinnost dokumentovat všechny verze kódu včetně bezpečnostních opatření a postupů. Dokumentace usnadní sdílení informací v týmu a bude užitečná při návrhu budoucích vylepšení. V těchto případech je nezbytné používat nástroje, které dokumentaci vytváří automaticky (např. Git, pokud je to v daném konkrétním případě možné) nebo způsob tvorby kódu zapsat do běžné dokumentace (prompt, seznam proměnných, seznam výsledných verzí apod.). Vedoucí oddělení, kde se AI používá pro generování zdrojových kódů, odpovídá za nastavení systému dokumentace při generování programových kódů.