Czat, agent, model lokalny i chmurowy: cztery różne modele zagrożeń

Poziom: srednio **Dotyczy: bezpieczeństwo AI, modele lokalne, agenci**

Czas: ~25 min

O bezpieczeństwie AI mówi się zwykle jednym tchem, a to dwa różne pytania. Pierwsze brzmi, dokąd płyną nasze dane, i na nie odpowiada wybór między modelem lokalnym a chmurowym. Drugie brzmi, co może się wydarzyć na skutek przeczytanego tekstu, i na nie odpowiada wybór między czatem a agentem. Ta strona rozdziela oba pytania i pokazuje, że lokalność nie zastępuje ograniczania uprawnień.

> **Umiejscowienie w serii.** Strona otwiera rozdział o bezpieczeństwie i opiera się na wiedzy ze stron Różnica pomiędzy czatem a agentem oraz Modele lokalne: AI na własnym sprzęcie. Zasady dla danych i kont opisuje Polityka użycia AI w firmie: wzór dla Zielonej, a trzy nogi ryzyka agenta strona Śmiertelna trójka: dane, obca treść i wyjście na zewnątrz.

Z ŻYCIA WZIĘTE

Pan Zbigniew wraca z urlopu z nowym laptopem i słyszy, że mamy w firmie model lokalny na `gpu01`. Pyta więc, czy w takim razie można do niego wklejać wszystko, łącznie z hasłami do systemów, bo „nic nie wychodzi z firmy”. Odpowiadamy, że dane faktycznie zostają u nas, ale hasło wklejone do czegokolwiek, co zapisuje historię, jest już poza kontrolą, i że to inna sprawa niż to, co model może zrobić z przeczytanym tekstem. Pan Zbigniew pyta wtedy o drugi temat: czy asystent mógłby sam czytać jego pocztę i odpisywać, i tu zaczyna się rozmowa o agentach.

Skróty użyte na tej stronie (2)

Skrót	Po angielsku	Po polsku i co znaczy

AI	Artificial Intelligence	sztuczna inteligencja: programy, które naśladują część ludzkiego rozumowania i rozmawiają w języku naturalnym
MCP	Model Context Protocol	protokół kontekstu modelu: otwarty standard podłączania narzędzi i danych do modeli AI

Leksykon: nazwy i pojęcia z tej strony (21)

Nazwa	Co to jest
agent	Model z dostępem do narzędzi, który w pętli planuje, wywołuje narzędzia, czyta wyniki i decyduje o kolejnym kroku. Tym różni się od czatu, że jego działania mogą zmieniać stan systemów.
czat	Rozmowa z modelem, w której on tylko odpowiada tekstem. Sam niczego nie uruchamia ani nie zmienia, dopóki nie damy mu narzędzi.
Docker i Docker Compose	Docker uruchamia programy w odizolowanych kontenerach. Compose opisuje kilka kontenerów jednym plikiem.
Gitea	Własny serwer repozytoriów Git z API. Jego serwer MCP (gitea-mcp) podłączamy do agenta w trybie tylko do odczytu.
guardrail	Mechanizm poza modelem, który ogranicza, co model lub agent może zrobić: filtr, lista dozwolonych narzędzi, limit kroków, zgoda człowieka. Prośba w prompcie guardrailem nie jest.
klasa danych	Podział informacji na publiczne, wewnętrzne, poufne i zastrzeżone. Od klasy zależy, który model wolno wywołać.
najmniejsze uprawnienia	Zasada, że dajemy tylko te uprawnienia, których zadanie wymaga, i tylko na czas zadania.
offline	Praca bez połączenia z internetem: model, głos i rozpoznawanie mowy działają na naszym komputerze. Test offline sprawdzamy, zanim cokolwiek podłączymy do sieci.
Ollama	Program do pobierania i uruchamiania modeli językowych na własnym komputerze. Wystawia API HTTP na porcie 11434, domyślnie bez uwierzytelniania.
Open WebUI	Interfejs czatu w przeglądarce, który łączy się z Ollamą i innymi serwerami modeli. Może logować użytkowników i dawać zespołowi wspólny dostęp.
piaskownica	Odgrodzone środowisko (kontener, osobne konto, maszyna wirtualna), w którym agent może coś zepsuć, nie dotykając reszty systemu.
prompt i prompt systemowy	Prompt to tekst, którym dajemy modelowi polecenie. Prompt systemowy ustawia twórca aplikacji, a nie użytkownik, i opisuje rolę oraz zasady modelu; jest to prośba, a nie zabezpieczenie.
prompt injection	Atak, w którym treść danych (strona, mail, plik, wynik narzędzia) podszywa się pod polecenie dla modelu. Model nie odróżnia pewnie instrukcji od danych, więc obrona nie może polegać wyłącznie na prośbie w prompcie.
sekret i poświadczenie	Informacja, która otwiera dostęp: hasło, klucz API, token. Trzymamy ją w magazynie sekretów, a nie w czacie, pliku konfiguracyjnym ani repozytorium.

Nazwa	Co to jest
serwer MCP	Program, który udostępnia agentowi narzędzia i dane w jednolity sposób, według protokołu MCP. Działa lokalnie jako podproces albo zdalnie przez HTTP.
Simon Willison	Programista i autor bloga o narzędziach AI. 16 czerwca 2025 roku opisał „śmiertelną trójkę” agentów.
śmiertelna trójka	Nazwa Simona Willisona (czerwiec 2025) dla trzech zdolności, które razem otwierają drogę do kradzieży danych: dostępu do danych prywatnych, kontaktu z niezaufaną treścią i możliwości wysyłania danych na zewnątrz. Zabezpieczenie polega na odjęciu jednej z nich.
token	Słowo o dwóch znaczeniach. W modelu to kawałek tekstu (słowo, jego część albo znak), na które model dzieli wszystko, co czyta i pisze; od liczby tokenów zależą szybkość i koszt. W dostępie to ciąg znaków, który zastępuje hasło i otwiera drzwi do usługi.
tylko do odczytu	Tryb, w którym narzędzie może czytać, ale nie może zmieniać. Każde nowe podłączenie zaczynamy od niego.
uwierzytelnianie i autoryzacja	Uwierzytelnianie sprawdza, kim jesteś. Autoryzacja sprawdza, co ci wolno.
Zielona	Fikcyjna firma, w której dzieją się nasze przykłady; opisuje ją strona „Świat Zielonej”. Tomek jest młodszym administratorem, Ola programistką, pani Halina główną księgową.

Dwa pytania, które łatwo pomylić

Pierwsze pytanie dotyczy drogi danych. Model chmurowy dostaje wszystko, co do niego wysyłamy, na serwery dostawcy, i tam obowiązują jego zasady przechowywania. Model lokalny dostaje to samo na naszym sprzęcie i dane nie opuszczają naszej sieci. Odpowiedź zależy więc od miejsca, w którym działa model.

Drugie pytanie dotyczy skutków. Czat zamienia tekst na tekst, więc najgorsze, co może się zdarzyć, to zła odpowiedź albo ujawnienie tego, co sami wkleiliśmy. Agent zamienia tekst na działania: czyta pliki, woła interfejsy, uruchamia polecenia i zmienia dane. Odpowiedź zależy więc od tego, jakie narzędzia ma model, a nie od tego, gdzie stoi.

Cztery ćwiartki

Dwie osie dają cztery kombinacje, z których każda ma własne ryzyka i własne zabezpieczenia. Poniższa tabela zbiera to, co ważne, a kolejne sekcje rozwijają po jednej ćwiartce.

	Czat (tekst na tekst)	Agent (tekst na działania)

Model w chmurze	dane wychodzą z każdą wiadomością; ryzyka: wklejone sekrety i dane osobowe, historia rozmów, zrzuty ekranu	dane wychodzą w całym kontekście, także z plików i wyników narzędzi; ryzyka: obca treść z poleceniem, nadmierne uprawnienia, wyciek przez narzędzia
Model lokalny	dane zostają na naszym sprzęcie; ryzyka: usługa dostępna w sieci bez uwierzytelnienia, modele z nieznanych źródeł, brak aktualizacji	dane zostają na naszym sprzęcie; ryzyka: obca treść z poleceniem, nadmierne uprawnienia, usługi lokalne bez uwierzytelnienia

Czat w chmurze

Ryzyko leży głównie po stronie człowieka: to on decyduje, co wkleić. Dlatego tu działa polityka użycia: klasy danych, konta firmowe zamiast prywatnych i reguła, że hasła, klucze i tokeny nie trafiają do żadnego narzędzia. Opowieść o Tomku, Token w czacie: opowieść o jednym wklejeniu, pokazuje, co kosztuje jedno wklejenie.

Czat lokalny

Dane nie wychodzą, więc polityka dopuszcza tu klasę poufną. Pojawia się za to ryzyko techniczne, którego w chmurze nie ma: usługa, która obsługuje model, musi być dobrze zamknięta. Serwer modeli nasłuchujący w sieci bez uwierzytelnienia jest otwartą furtką do mocy obliczeniowej, a często i do plików, które z nim połączono. Dlatego usługę modeli trzymamy na adresie lokalnym, tak jak Ollama w naszym asystencie domowym, która nasłuchuje wyłącznie na `127.0.0.1`, a dostęp z innych komputerów dajemy przez serwer pośredniczący z logowaniem, jak Open WebUI w Zielonej.

Agent w chmurze

Agent w chmurze widzi więcej niż czat, bo do modelu trafiają nie tylko nasze wiadomości, ale i wszystko, co przeczytają jego narzędzia: pliki projektu, wyniki poleceń, strony z bazy wiedzy. To dane, których nie wkleiliśmy świadomie. Ryzyko drugiego rodzaju to obca treść: dokument albo strona, w której ktoś ukrył polecenie dla modelu. Obie rzeczy ogranicza ta sama zasada, czyli najmniejsze uprawnienia: konto serwisowe, tokeny o minimalnej roli, piaskownica i zakaz nieograniczonej powłoki.

Agent lokalny

To ćwiartka, do której zmierza nasz asystent domowy, gdy od fazy dziewiątej dostaje pierwsze narzędzia. Dane nie wychodzą do dostawcy, ale cała reszta ryzyk agenta zostaje, bo prompt injection działa tak samo na modelu, który stoi na `gpu01`, jak na tym, który stoi w chmurze. Nie ma

wiarygodnego sposobu, żeby model odróżniał polecenia od danych, więc nie zakładamy, że model lokalny jest odporniejszy na cudze polecenia.

> **Lokalność odpowiada na „dokąd płyną dane”, nie na „co agent może zrobić”.** Agent lokalny z dostępem do powłoki, poczty i plików jest dokładnie tak samo niebezpieczny jak agent w chmurze z tymi samymi uprawnieniami. Różnica dotyczy tego, kto widzi dane, a nie tego, kto może je zmienić.

Kto jest przeciwnikiem

W każdej ćwiartce przeciwnikiem jest ktoś inny. Poniższa tabela zbiera cztery postaci, z którymi liczymy się najczęściej, i mówi, w której ćwiartce są groźne.

Przeciwnik	Czat w chmurze	Agent w chmurze	Czat lokalny	Agent lokalny
dostawca usługi, który przechowuje rozmowy	tak	tak	nie	nie
autor obcej treści z ukrytym poleceniem	rzadko, tylko przez wklejony tekst	tak	rzadko, tylko przez wklejony tekst	tak
ktoś z sieci, kto trafia na źle zamkniętą usługę	nie	tylko serwery narzędzi	tak	tak
my sami: wklejone sekrety, zbyt szerokie uprawnienia	tak	tak	tak	tak

Ostatni wiersz jest taki sam w każdej kolumnie i to jest ważna obserwacja. Najczęstsze incydenty biorą się z pomyłek własnych, a nie z działań napastników, więc zabezpieczenia, które chronią nas przed samymi sobą, mają największy wpływ.

Co z tego wynika dla Zielonej

Polityka użycia AI dzieli dane na cztery klasy i do każdej przypisuje narzędzia. Z podziału na ćwiartki wynikają dodatkowe zasady, które dopisujemy do polityki, kiedy w grę wchodzą agenci. Poniższa tabela jest przykładem decyzji dla czterech zadań.

Zadanie	Dane	Ćwiartka	Dlaczego

streszczenie umowy z odbiorcą	poufne	czat lokalny	dane zostają w firmie, a czat nie ma narzędzi
przegląd zgłoszeń w Gitea	wewnętrzne	agent lokalny albo chmurowy, tylko odczyt	odczyt przez konto serwisowe, bez narzędzi wychodzących
pomoc przy kodzie własnego zespołu	wewnętrzne	agent w chmurze w piaskownicy	kod bez tajemnic, ale agent ma ograniczoną powłokę
odpowiadanie na maile z zewnątrz	poufne	żadna, bez zatwierdzenia człowieka	obca treść i możliwość wysłania odpowiedzi to dwie nogi ryzyka

Mity i sprostowania

Mit	Sprostowanie
Model lokalny jest bezpieczny, bo działa offline.	Offline chroni dane przed dostawcą, ale nie przed poleceniem ukrytym w pliku, który agent sam otworzy. Połączenie z siecią nie jest warunkiem prompt injection.
Agent w chmurze wysyła cały nasz dysk.	Do modelu trafia to, co agent przeczyta narzędziem i co mu pokażemy. Zakres wyznaczają uprawnienia, więc to je projektujemy z największą starannością.
Czat nie jest ryzykowny, bo nic nie robi.	Czat z wtyczkami, przeglądaniem stron albo podłączonymi usługami jest już agentem. Granica między nimi jest płynna i przebiega tam, gdzie pojawia się narzędzie.
Model wie, że nie wolno, bo napisaliśmy to w prompcie systemowym.	Zakaz w prompcie jest prośbą, a nie barierą. Barierą jest to, że model nie ma narzędzia albo ma je bez uprawnień.

Ćwiczenia

1. Zespół instaluje serwer modeli na laptopie programisty i uruchamia go tak, żeby słuchał na wszystkich adresach, „żeby kolega mógł się podłączyć”. Co jest nie tak i co robimy zamiast tego?

Odpowiedź

Usługa bez uwierzytelnienia dostępna w sieci pozwala każdemu, kto ją znajdzie, korzystać z modelu i z tego, do czego jest podłączona. Zostawiamy nasłuch na adresie lokalnym, a udostępnienie robimy przez serwer pośredniczący z logowaniem, tak jak w Zielonej: Open WebUI na `docker01` z kontami z katalogu użytkowników.

2. Kasia chce, żeby czat w chmurze „zajrzał do jej poczty i napisał odpowiedzi”. Do której ćwiartki to należy i co zmieniamy?

Odpowiedź

Do agenta w chmurze. Poczta zawiera dane prywatne i obce treści, a wysyłanie odpowiedzi to droga na zewnątrz, czyli komplet trzech nóg ryzyka. Zamiast pełnej automatyzacji proponujemy agenta, który czyta pocztę i przygotowuje szkice, a wysyła je człowiek po przeczytaniu.

3. Dlaczego w ostatnim wierszu tabeli przeciwników jest „tak” we wszystkich kolumnach?

Odpowiedź

Bo własne pomyłki, takie jak wklejenie sekretu albo nadanie zbyt szerokich uprawnień, zdarzają się niezależnie od miejsca uruchomienia modelu i rodzaju narzędzia. Dlatego polityka użycia, konta serwisowe i tokeny o minimalnej roli dotyczą każdej ćwiartki.

Źródła i dalsze lektury

Przypisy

[1] Trzy nogi ryzyka agenta: Simon Willison, „The lethal trifecta for AI agents” (16 czerwca 2025).

[2] Zagrożenia po stronie serwerów MCP, w tym przejęcie lokalnego serwera: MCP, Security Best Practices.

Lektury

Obszerniej o ryzykach w infrastrukturze mówi Bezpieczeństwo AI w infrastrukturze: modele, agenci i MCP w sieci firmy, o ograniczaniu uprawnień agenta Projektowanie guardraili: jak ograniczyć, co model i agent mogą zrobić, a o samej różnicy czat i agent Różnica pomiędzy czatem a agentem.

Nota metodyczna: czego nie wiemy

Podział na cztery ćwiartki jest naszym uproszczeniem, a nie standardem. Granica między czatem a agentem jest płynna, a ocena „rzadko” i „tak” w tabeli przeciwników wyraża nasz osąd, nie pomiar.

Nie porównywaliśmy odporności różnych modeli na prompt injection, więc nie twierdzimy, że któryś jest bezpieczniejszy. Opis domyślnego zachowania usług modelowych zależy od oprogramowania i jego wersji, dlatego adres nasłuchu każdej usługi sprawdzamy w jej konfiguracji, zamiast polegać na przekonaniu.

Revision #3
Created 2026-10-20 08:59:08 UTC by Tomasz Konieczny
Updated 2026-10-20 09:05:16 UTC by Tomasz Konieczny