



ОТЧЕТ ОТ ТЕСТ ЗА ПРОНИКВАНЕ В AI / LLM · СЪСТАВЕН ПРИМЕР

Съставен примерен Тест за проникване в AI / LLM

Изготвен за [Клиент] от PXLSecurity LTD.

Това е примерен документ, не предаван на клиент. Той е съставен от формата на няколко анонимизирани ангажимента на PXLSec срещу вградени генеративни AI функционалности, AI агенти с достъп до инструменти и услугите зад тях. Всяка констатация е преписана със синтетични, илюстративни данни върху резервирани примерни имена, адреси и клиентски организации. Той не описва нито един действителен клиент и в него не присъства действителен клиент, доставчик, продукт, хост, човек или данни за достъп.

НОМЕР НА ОТЧЕТА

PXL-2026-0207

ВЕРСИЯ

1.0 (окончателна)

ДАТА НА ОТЧЕТА

[Дати на ангажимента]

ПЕРИОД НА ТЕСТВАНЕ

[Договорени дати]

КЛАСИФИКАЦИЯ

Конфиденциално — примерен документ

ИЗГОТВЕН ОТ

PXLSecurity LTD., София, България

PXLSecurity LTD. · бул. „Васил Левски“ 12, 1142 София, България · VAT BG203239331 · hello@pxlsec.com · независими от 2014 г.

НАЧАЛНИ СТРАНИЦИ

Контрол на документа

Тази страница отразява какво представлява документът, който го държи и как се е променял. В този примерен документ всяко поле, идентифициращо клиента, е оставено като очевиден заместител.

Заглавие на документа	Съставен примерен тест за проникване в AI / LLM — отчет
Номер на отчета	PXL-2026-0207 · версия 1.0 (окончателна) · заменя 0.9 (проект)
Клиент	[Клиент]
Дати на ангажимента	[Договорени дати] · отчет с дата [Дати на ангажимента]
Автор	Старши консултант по сигурността, PXLSecurity LTD.
Осигуряване на качеството	Втори старши консултант, който не е извършвал тестването: технически преглед и независимо възпроизвеждане на всяка констатация
Класификация	Конфиденциално. Този примерен документ не съдържа клиентски данни.

История на редакциите

Версия	Дата	Автор	Промяна
0.1	[Договорени дати]	Водещ консултант	Първоначален проект, издаден в края на активното тестване
0.9	[Договорени дати]	Проверяващ качеството	Технически преглед; всяка констатация възпроизведена независимо
1.0	[Дати на ангажимента]	Водещ консултант	Окончателно издание до списъка за разпространение по-долу

Разпространение

Име	Роля	Организация	Копие
[Лице за контакт на клиента]	Поръчител на ангажимента	[Клиент]	Пълен отчет
[Лице за контакт на клиента]	Продуктов отговорник за AI	[Клиент]	Пълен отчет
Водещ консултант	Автор	PXLSecurity LTD.	Пълен отчет
[Лице за контакт на доставчика на платформата]	Реакция на доставчика по сигурността	[Доставчик]	Само съответните констатации, по указание на клиента

Конфиденциалност. Действителен отчет от този вид описва, с подробност, която позволява възпроизвеждане, как да бъдат достигнати данни за достъп, конфигурация и данни на други клиентски организации в жива AI функционалност. С него трябва да се работи като с един от най-чувствителните документи, които организацията притежава. Този конкретен документ е примерен и не съдържа клиентски данни, така че нищо от това не се отнася за него — той е публикуван умишлено.

НАЧАЛНИ СТРАНИЦИ

Съдържание

Ръководството може да прочете раздели 1 до 4 и да действа; инженерите могат да работят от раздел 5 нататък, без да четат нищо друго.

	РАЗДЕЛ	СТР.
1	Резюме за ръководството Какво беше тествано, основната картина на риска и трите неща, които да поправите първо — написано за нетехнически читател.	4
2	Обхват на ангажимента Цели, вид на оценката, акаунти, период, препратки, изключения и допусканията, от които зависи резултатът.	6
3	Методология Петте етапа, шестте области, които един AI тест покрива, как се използват инструментите и правилата за безопасност, при които тестваме.	8
4	Оценки на критичността Скалата и как CVSS информира оценката, без да я замества.	11
5	Обобщение на констатациите Всяка констатация с нейния идентификатор, критичност и статус, в реда на отстраняване.	12
6	Подробни констатации По един раздел за всяка констатация: описание, доказателство, възпроизвеждане, въздействие и конкретна поправка.	13
7	План за отстраняване Поправките, групирани на вълни, с отговорник и последователност.	29
8	Повторен тест и приключване Какво покрива повторният тест и какво заявява писмото за приключване.	31
A	Приложение А — Достъп, акаунти и инструменти Какво поискахме, защо всяка точка имаше значение и използваните инструменти.	32
B	Приложение Б — Извън обхвата и какво този тест не е Какво беше изключено и съседната работа, която този ангажимент не извършва.	35
C	Приложение В — Препратки и какво представляват Рамките, на които се опира, описани точно, без да се подразбира каквато и да е сертификация.	36

D Приложение Г — Как да четете този отчет

Кой кой раздел да прочете и как да действа по него.

37

Бележка какво представлява този примерен документ. Констатациите F-01 до F-09 са взети от формата на няколко различни анонимизирани ангажмента и представени тук, сякаш идват от един, за да може потенциален клиент да види целия обхват на работата с едно прочитане. Всеки запис, клиентска организация, идентификатор, акаунт и оценка е синтетичен и никъде не е назован доставчик, платформа или продукт — умишлено, за да описва документът проблема, а не инструментите на този месец.

РАЗДЕЛ 1

Резюме за ръководството

PXLSecurity LTD. беше нает от [Клиент] да извърши ръчен тест за проникване на AI функционалност и услугите зад нея. Този раздел е написан за нетехнически читател. Той казва какво беше тествано, колко сериозен е резултатът и какво да се направи първо.

Какво беше тествано

Старши инженери тестваха AI функционалността на ръка през договорения период, работейки на нивото на заявките, които софтуерът действително прави, а не през екраните, които потребителят вижда. Съставният обхват зад този примерен документ покриваше генеративен асистент, вграден в бизнес приложение, агента, който извиква вътрешни инструменти от името на потребител, услугата, която съхранява конфигурацията на промптите и данните за достъп на процеса, хранилището с разговорите, документите и векторните представяния, и обикновения уеб и API слой, на който цялото това е публикувано. Тестването беше с удостоверяване, с умишлено обикновени акаунти в две клиентски организации, така че резултатът отразява какво би могъл да достигне влязъл служител, а не какво вижда анонимен посетител.

Заглавието

Оценката установи осем констатации и едно положително наблюдение: две High, три Medium, три Low и една Informational. Общата оценка на риска за ангажмента е High. Моделът се държи добре през цялото време. Проблемите стоят пред него.

9 вписвания	2 High	3 Medium	3 Low	High обща оценка
----------------	-----------	-------------	----------	---------------------

Констатации по критичност



■ High 2 ■ Medium 3 ■ Low 3 ■ Informational 1

Числата са за вписванията, повдигнати в този обобщен примерен документ. Критичността е оценка по бизнес риск, обяснена в раздел 4. Вписването Informational е положително наблюдение, а не дефект.

Къде попаднаха констатациите

Административен слой	Подсказки, жизнен цикъл на модела, зареждане на данни, данни за достъп	3 констатации
Изоляция на AI данните и клиентската организация	Информационни представяния, резюмета	1 констатация
Граница на агента и инструментите	Наличната власт действа агентът	1 констатация
Самият модел	Поведение, откази и изход	0 констатации

РЕЗЮМЕ ЗА РЪКОВОДСТВОТО · ПРОДЪЛЖЕНИЕ

Какво означава и какво да направите първо

Сериозните констатации имат една обща тема и тя не е моделът. AI функционалност вътре в бизнес приложение не е просто модел; тя е малка разпределена система със собствена административна повърхност. Нещо съхранява конфигурацията на промптите, която моделът изпълнява. Нещо създава модели, стартира обучения и ги изтрива. Нещо задейства задачите, които изтеглят бизнес данни в корпуса. И нещо държи данните за достъп, които процесът използва, за да достигне хранилището и системите около него. Всичко това беше достижимо от обикновен бизнес акаунт.

Тази подредба има търговско значение, защото е обратна на това, където е вниманието на индустрията. Да бъде убеден един модел да се държи зле купува на нападателя влияние върху изхода на една сесия, в рамките на това, което моделът вече е имал право да прави. Достъп за запис до административната повърхност купува данните за достъп на процеса, инструкциите, които всеки потребител получава, и избора какво функцията чете и къде се записват данните ѝ. Оценка, която беше погледнала само чат полето, щеше да отчете, че асистентът се държи добре, което беше вярно, и щеше да пропусне всяка констатация по-горе.

Трите останали констатации се намират в обикновения уеб и API слой, на който са публикувани AI крайните точки — обработка на грешки, ограничаване на честотата и политика за различен произход. Нито една от деветте не изискваше новаторска атака. Всяка зависи от решение за оторизация или конфигурация, което е било разумно, когато е било взето, и никога не е било преразгледано — което е добрата новина: по-голямата част от работата е налагане и конфигурация, а не разработка.

Трите неща, които да поправите първо

- 1 Подменете днес данните за достъп до хранилището на процеса и проверете кой ги е ползвал. Те бяха четими от всеки акаунт, който може да влезе в клиентската организация, а записът, който ги съдържаше, беше записваем — което означава, че нападател е могъл и да избере къде отиват данните на функцията. Подмяната не изисква промяна в кода и издание. (F-01)

- 2 Поставете една проверка само за администратори върху цялата административна повърхност на AI, налагана от сървъра. Конфигурацията на подсказките, данните за достъп, жизненият цикъл на модела и зареждането на данни в момента се охраняват от обхват, който обикновените служители притежават. Това е единна проверка на правата, а не списък от кръпки по крайни точки — а услуга на същата платформа вече го прави правилно, така че има работещ пример за копиране. (F-01, F-03, F-05)
- 3 Накарайте хранилището на AI данните да проверява собственост при всяка операция. Потребител на една клиентска организация можеше да чете разговорите, документите, векторните представяния и резюметата на друга клиентска организация, докато операцията за изтриване върху същия обект беше правилно отказана. Проверката съществува; трябва да бъде прилагана последователно, а не написана отново. (F-02)

Какво следва. Вашите инженери работят от раздел 6, който дава на всяка констатация път за възпроизвеждане, доказателството зад нея и конкретна поправка; раздел 7 групира тези поправки на вълни с отговорник за всяка. Когато сте готови, повторният тест проверява всяка поправка и преиздаваме отчета с писмо за приключване. Там, където констатацията принадлежи на доставчик на платформа, а не на вас, раздел 6 записва операцията и отговора за всеки опит, така че да може да бъде предадена както е.

Моделът се държи добре. Проблемът беше пред него — в това кой можеше да пише подсказките, да държи данните за достъп и да избира какво функцията чете.

РЕЗЮМЕ ЗА РЪКОВОДСТВОТО · НАКРАТКО

РАЗДЕЛ 2

Обхват на ангажимента

Констатация може да съществува само за нещо, което е било в обхвата, достижимо и тествано. Всичко извън тези граници е нетествано, а не сигурно.

Клиент	[Клиент]
Цели	Съставен обхват за този примерен документ: генеративен асистент, вграден в бизнес приложение; слой за оркестрация на агента и инструментите, които той може да извиква; административната услуга на AI, съдържаща конфигурацията на промптите, жизнения цикъл на модела, задействането на зареждане на данни и записите с данни за достъп на процеса; хранилището на AI данните, съдържащо разговори, документи, векторни представяния и генерирани артефакти; и уеб и API повърхността, на която те са публикувани. В действителен ангажимент всяка цел е изброена поединично.
Вид на оценката	Grey box, с удостоверяване, тествано на нивото на заявките, а не през административния интерфейс.
Използвани акаунти	Два обикновени неадминистраторски акаунта в един и същ клиентска организация и роля, за да може да бъде тествана изолацията между потребители; един административен акаунт, за да може административната повърхност да бъде изброена както е предвидено и после опитана отдолу; и един обикновен акаунт във втора клиентска организация.

Среда	Непроизводствен клиентска организация, отразяващ същия модел на оторизация, с ясно обозначено синтетично съдържание в корпуса. Операции, които изразходват ресурси или променят състоянието, бяха задействани само с писмено разрешение.
Период на тестване	[Договорени дати] · работно време, с посочени лица за връзка на разположение от двете страни.
Препратки	OWASP Top 10 for LLM Applications; OWASP Agentic Security Initiative; MITRE ATLAS; NIST AI RMF; OWASP API Security Top 10; OWASP WSTG; CVSS v3.1. Какво е и какво не е всяко от тези неща, е в Приложение В.
Изключения	Отказ на услуга и тестване под натиск; самата хоствана услуга за модела на трета страна, която не е на клиента да разрешава; red teaming на безопасността на модела; оценка на модела; социално инженерство; и всяка система, която не е назована в подписания обхват. Вижте Приложение Б.
Екип	Старши консултанти на PXLSec, с независимо осигуряване на качеството от консултант, който не е извършвал тестването.

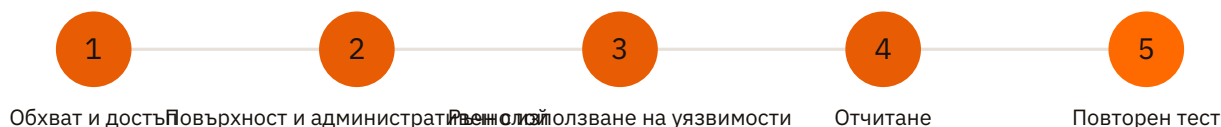
Допускания и ограничения

- 1 Резултатът е ограничен във времето. AI функционалностите се променят бързо: версия на модел, промяна в подсказка или нов инструмент, добавен след периода на тестване, са извън това, което беше тествано.
- 2 Обхватът е граница, не заключение. Изключените системи не бяха тествани.
- 3 Липсата на констатация не е доказателство за липса. Никоя оценка с ограничена продължителност не може да докаже, че не е останала уязвимост, а поведението на модела не е изчерпателно изброимо.
- 4 Предоставената информация беше приета както е дадена. Документацията, описите на инструментите и данните за достъп бяха използвани както са предадени, без независим одит.
- 5 Наличността и разходите бяха защитени. Тестове, носещи риск от прекъсване или съществен разход, бяха пропуснати, ограничени или изпълнени срещу тестова среда, така че някои пътища бяха потвърдени чрез наблюдение, а не чрез изпълнение.

РАЗДЕЛ 3

Методология

Тестването следваше стандартния ни подход от пет етапа, приложен към AI функционалност. Всеки етап произвежда определен резултат и ангажиментът не продължава напред, докато този резултат не съществува и не е бил съгласуван.



1 · Обхват, модел на заплахите и достъп

Преди да бъде изпратен каквото и да е трафик, уговаряме писмено какво е в обхвата, какво е изключено, кога се провежда тестването и на кого да се обадим, ако нещо се случува. За AI ангажимент това отива по-далеч от обичайното: дали тестерите могат да задействат зареждане на данни или обучение, какво функцията има право да прави от името на потребител и кои акаунти и клиентски организации съществуват. Писмено изявление какво функцията има право да прави е елементът, който най-много променя отчета, защото без него тестерът може да опише какво прави системата, но не и къде тя се разминава с това, което е било предвидено.

Произвежда: подписан документ за обхвата и правилата на ангажимента, посочени лица за контакт, матрицата на акаунтите и клиентските организации и приоритизиран списък със сценарии за атака.

2 · Изброяване на повърхността и административния слой

Картографираме действителната повърхност, а не документираната: крайните точки на AI услугата наред със съществуващите API на приложението, конфигурацията на промптите и шаблоните, операциите по жизнения цикъл на модела, задействането на зареждане, записите с данни за достъп и връзки, пространството на идентификаторите на хранилището и описа на инструментите на агента. Административната повърхност се изброява първо с административен акаунт, така че да бъде видяна да работи както е предвидено, преди да бъде опитано каквото и да е отдолу.

Произвежда: опис операция по операция на административната повърхност на AI, обектите с данни, които функцията притежава, и договора за инструментите на агента.

3 · Ръчно използване на уязвимости

Тук ангажиментът се печели или губи и това е изцяло човешка работа. Всяка променяща операция се повтаря с най-обикновения токен, който клиентската организация би издал, и се записва статусът, който съвърхът действително връща, а не това, което интерфейсът подсказва. Обекти се създават като двама потребители и пространството на идентификаторите се обхожда в двете посоки. Агентът бива помолен да направи това, което искащият потребител не може. Инжектирането на подсказки се подрежда според това, което достига: извикване на инструмент, данни за достъп, данни на друга клиентска организация.

Произвежда: проверени констатации с доказателство, път за възпроизвеждане и демонстрирано въздействие.

4 · Отчитане

Всяка констатация се описва с доказателството си, стъпките за възпроизвеждането ѝ, какво означава за бизнеса и конкретна поправка. Там, където няколко констатации споделят една първопричина, го посочваме и предлагаме единната промяна, защото резултат от административния слой обикновено е единна проверка на правата, а не списък от кръпки. Втори старши консултант възпроизвежда всяка констатация преди издаването.

Произвежда: този документ, машинно четим експорт на констатациите и разговор по резултатите.

5 · Повторен тест

Щом отстраняването е завършено, тестваме повторно всяка констатация и преиздаваме отчета с обновени статуси и подписано писмо за приключване — включено по стандарт, защото непроверена поправка е план, а не резултат.

Произвежда: обновен отчет със статус за всяка констатация и писмо за приключване.

МЕТОДОЛОГИЯ · ПРОДЪЛЖЕНИЕ

Шестте области, инструментите и безопасността

Какво покрива един тест за проникване в AI

Шест области. Първите две произвеждат по-голямата част от сериозните констатации — затова се тестват първи, а не последни.

Област	Какво се опитва	Констатации тук
Административният слой на AI	Всяка административна операция, която AI компонентът излага — конфигурация на подсказки и шаблони, точки за разширение, записи с данни за достъп и връзки, създаване на модели, обучение и изтриване, внедряване, задействане на зареждане — опитана с акаунта с най-малко права, който може да се удостовери. Четенето и записът са тествани отделно.	F-01, F-03, F-05
Оторизация и изолация на AI данните между клиентски организации	Съдържание, създадено като двама потребители в една клиентска организация и като потребители в две клиентски организации, с разменени идентификатори в двете посоки, като извличането, метаданните, експортирането и изтриването са тествани отделно. Производните артефакти — векторни представяния, резюмета — са третирани като самостоятелни обекти.	F-02
Границата на оторизация на агента и инструментите	Чия власт носи всяко извикване на инструмент: тази на крайния потребител, оторизирана на нивото на инструмента, или една служебна идентичност, достатъчно широка да обслужва всички. Опитано през агента това, което на искащия потребител се отказва директно.	F-04
Инжектиране на промпти, пряко и косвено	Инструкции в собствения ход на потребителя и инструкции, поставени в съдържание, което системата зарежда. Подредени по последствие, а не по полезен товар: какво е достигнато, не че е проработило.	F-04
Пътят на данните и съхранението	Къде отиват подсказките, разговорите, качените файлове и изходите, кои компоненти ги виждат, какво се записва в журнал и с каква подробност, колко дълго всяко хранилище го пази и дали хранилището на артефактите е достижимо без сесия.	F-09 (положителна)
Обикновена уеб и API сигурност на AI крайните точки	Удостоверяване на всеки маршрут, включително поточните, оторизация на всеки от тях, ограничения на честотата и разходите, ограничения на размера на входните данни, обработка на грешки, политика за различен произход и тайни в пакета от страна на клиента.	F-06, F-07, F-08

Как се използват инструментите

Използваме автоматизирани инструменти интензивно и им се доверяваме за точно едно нещо: покритие. Скенерът може да изброи голяма повърхност по-бързо и по-последователно от човек. Това, което не може, е да разбере за какво служи функцията — кои записи един акаунт има право да вижда, дали на един агент е трябвало да бъде позволено да направи определено извикване, или кой от два отговора с правилен вид представлява прекосена граница. При AI функционалностите тази разлика е по-голяма от обичайното, защото публикуван списък с категории риск не е тест, а генериран корпус за jailbreak не ви казва към какво е свързан моделът.

Какво добавя автоматизацията

- Откриване на крайни точки, маршрути и параметри по повърхността на AI и приложението.
- Повтаряне на записан набор операции през няколко токена и клиентски организации.
- Разпознаване на версии и конфигурации в голям обем.
- Широчина през корпуси за инжектиране, за триаж, а не за констатации.
- Покритие за регресии при повторния тест.

Какво добавя само човекът

- Решение кои административни операции съществуват и имат значение.
- Тестване на оторизацията през акаунти, роли и клиентски организации.
- Преценка какво едно инжектиране действително е достигнало и колко струва това.
- Забелязване, че една липсваща проверка обяснява няколко констатации.
- Решение какво е безопасно да бъде изпълнено срещу система, която изразходва ресурси.

Безопасност и правила на ангажмента

- 1 Разрешението е първо. Никакво тестване не започва, докато не е налице подписано разрешение, назоваващо системите в обхвата, и екипът го носи през цялото време.
- 2 Наличността и разходите са защитени. Отказът на услуга и тестването под натиск са изключени по подразбиране. Операции, които изразходват изчислителен ресурс или бюджет — зареждане на данни, обучение, масово генериране — са ограничени, договорени писмено предварително и се изпълняват срещу тестова среда навсякъде, където моделът на оторизация е същият.
- 3 Минимално необходимо доказателство. Използването на уязвимостта спира в момента, в който въздействието е демонстрирано. Предпочитаме да демонстрираме пропуск в оторизацията срещу запис, който сме създали сами; не извличаме данни масово, а конфигурацията, която променяме, се възстановява незабавно.
- 4 Нищо не се поставя там, където други ще го прочетат. Съдържанието, използвано за тестване на косвено инжектиране, отива в корпус, принадлежащ на тестерите, е ясно обозначено и се премахва след това. Никаква инструкция не се оставя там, където сесия на действителен потребител би могла да я достигне.
- 5 Не се публикуват работещи експлойти. Възпроизвеждането е написано на нивото, което използва един бюлетин на доставчик — операцията, пътят и отговорът — което е достатъчно за вашите инженери и недостатъчно за всеки друг.
- 6 Жив контакт през цялото време. Посочените лица за контакт от двете страни са достъпни през целия период, а констатация с критичност High се ескалира в същия ден, в който бъде потвърдена, със съвет за междинно ограничаване, вместо да бъде задържана за отчета.
- 7 Чисто излизане. Тестовите акаунти, качените документи, създадените модели и всеки друг артефакт се премахват и изброяват, за да може клиентът да провери, че средата е както е била.

Защо скучният акаунт има значение. Най-ценните данни за достъп в един AI ангажимент са тези, които не твърдят абсолютно нищо. Административният акаунт ви показва административната повърхност, работеща както е предвидено; обикновеният акаунт ви казва дали сървърът всъщност го е грижа. По-голямата част от сериозните констатации в този отчет произлизат от извикване на административна операция с токен, който не претендира за никакви права, и получаване на отговор.

Инструментът може да ви каже, че една крайна точка съществува. Само човек може да ви каже, че обикновен потребител никога не е трябвало да я извиква.

РАЗДЕЛ 3 · НАКРАТКО

РАЗДЕЛ 4

Оценки на критичността

Констатациите се оценяват по бизнес риск — какво струва проблемът на тази организация, ако бъде използван — като CVSS v3.1 информира оценката, но не я диктува.

Оценка	Какво означава	Очакван отговор
CRITICAL	Незабавно и тежко бизнес въздействие; лесна експлоатация. Обикновено без удостоверяване и надеждно, водещо директно до загуба на данни, финансова загуба или контрол над средата.	Поправете веднага. Ескалирайте същия ден; приложете междинно ограничаване, ако пълната поправка отнема време.
HIGH	Значително въздействие и реалистична възможност за експлоатация, макар да може да изисква позиция, акаунт или предварително условие. Често решаващото звено във верига.	Поправете в текущия цикъл. Не допускайте да влезе в общия списък със задачи.
MEDIUM	Умерено въздействие или сериозно въздействие, което зависи от условия извън пълния контрол на нападателя.	Планирайте отстраняване с договорен срок в следващия цикъл или два.
LOW	Ограничено пряко въздействие. Укрепване на защитата и защита в дълбочина; полезно за нападател главно в комбинация с нещо друго.	Планирайте го наред със свързаната работа.
INFORMATIONAL	Няма пряко въздействие върху сигурността. Наблюдение, отклонение от добрата практика или — както тук — контрол, потвърден като работещ.	Отбележете го. Действайте по ваша преценка.

Как се използва CVSS — и къде спира

Всяка констатация тук носи базова оценка и вектор по CVSS v3.1. CVSS е общ език: той позволява на одитор, доставчик и инженер да сравнят две констатации, без да спорят за прилагателни. Това, което умишлено не знае, е вашият бизнес — дали засегнатата система съдържа регулирани лични данни или тестово копие на миналогодишния каталог, или дали консуматорът на записваем конфигурационен запис вече е свързан. Затова използваме базовата оценка като входни данни и оценяваме по бизнес риск.

Там, където двете се разминават, констатацията посочва оценката, резултата и причината. F-03 е разработеният пример в този отчет: илюстративната ѝ базова оценка е 7.1 и е оценена като Medium, защото консумиращата сървърна част не беше свързана в тестваната среда. Дефектът в оторизацията е пълен и действащ; липсва само консуматорът. Оценена само по резултата, тя би изпреварила разкриване на данни между клиентски организации, което би било погрешно — а отхвърлена заради липсващия консуматор, би било още по-лошо. F-08 показва обратното: позволяваща политика за различен произход изисква влязла жертва и съдържание, контролирано от нападателя, затова е Low сама по себе си, а раздел 7 я подрежда наред с F-01 и F-03, защото пред неохраняван административен запис тя премахва последното предварително условие, което тези констатации са имали.

Оценките в този примерен документ са илюстративни. Те са представителни примери, подбрани да съответстват на всеки вид констатация, а не оценки, присвоени на действителна система.

CVSS ви казва колко лоша е уязвимостта. Само вашият контекст ви казва колко лош ще бъде вторникът.

РАЗДЕЛ 4 · НАКРАТКО

РАЗДЕЛ 5

Обобщение на констатациите

Всичките девет вписвания, в реда на отстраняване. Раздел 6 излага доказателството, пътя за възпроизвеждане, въздействието и препоръката за всяко от тях.

ID	Констатация	Компонент	Критичност	CVSS	Статус
F-01	AI pipeline storage credentials readable and writable by a non-administrator	AI administration service — storage destination record	HIGH	8.1	Отворена
F-02	Cross-tenant access to conversations, documents and embeddings	AI datastore — threads, documents, embeddings, generated summaries	HIGH	7.7	Отворена
F-03	Generative prompt configuration writable by an ordinary tenant user	AI administration service — prompt extension points	MEDIUM	7.1	Отворена
F-04	Agent tool boundary steerable by instructions planted in an ingested document	Agent orchestration layer and its tool authorisation	MEDIUM	6.5	Отворена
F-05	Model lifecycle and data ingestion operable without an administrator role	AI administration service — model lifecycle and ingestion triggers	MEDIUM	6.3	Отворена
F-06	Unauthenticated AI endpoint discloses internal architecture in its error body	AI generation endpoint — error handling	LOW	5.3	Отворена

ID	Констатация	Компонент	Критичност	CVSS	Статус
F-07	No rate limiting or cost control on the generation endpoint	AI generation endpoint — throttling and quota	LOW	4.3	Отворена
F-08	Permissive CORS on the AI API: arbitrary origin reflected with credentials allowed	AI API gateway — cross-origin resource sharing policy	LOW	4.3	Отворена
F-09	Positive observation: the administrator gate exists, one service away	Platform identity service, compared with the AI services	INFORMATIONAL	n/a	Observation

Статусът е към датата на този отчет; след повторния тест всеки ред се преиздава като затворен, частично отстранен или отворен. Оценките са илюстративни. Констатациите се предават и като експорт в CSV и JSON.

Как се чете таблицата

- 1 Три от осемте констатации са една липсваща проверка. F-01, F-03 и F-05 са един и същ дефект, достигнат през три операции: административните услуги на AI не налагат проверка за администратор по пътя на запис. Третирайте ги като една промяна с три потвърждения, а не като три задачи — F-09 показва услуга на същата платформа, която вече налага тази проверка.
- 2 Констатацията за инжектиране се оценява според това, което е достигнала. F-04 не е Medium, защото инжектирането е успяло; тя е Medium, защото инструментът, който е бил насочен, връща бизнес записи през граница, през която на искащия потребител се отказва директно. Ако не беше достигнала нищо, нямаше да бъде в тази таблица.
- 3 Трите Low не са запълване и една от тях се натрупва. F-08 е Low сама по себе си и съществено по-лоша в комбинация с F-01 и F-03 — затова раздел 7 я подрежда наред с тях, а не във вълната по закаляване.

РАЗДЕЛ 6

Подробни констатации

По един раздел за всяка констатация: критичност и илюстративен CVSS, засегнат компонент, какво беше открито, доказателството, път за възпроизвеждане, въздействието и поправка.

За доказателствата по-долу. В действителен отчет всеки запис е заличена снимка на екрана от собствената сесия на инженера, с пълните двойки заявка и отговор в приложението. Всеки запис тук е синтетична реконструкция — операциите са представителни; услугите, клиентските организации, идентификаторите, ключовете и изходът са измислени. В този документ никъде не присъства работещ полезен товар или експлойт: възпроизвеждането е написано на нивото, което използва един бюлетин на доставчик.

HIGH

F-01 AI pipeline storage credentials readable and writable by a non-administrator

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
High	8.1 AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N	AI administration service — storage destination record	Отворена

Class: Missing authorisation (CWE-862) · insufficiently protected credentials (CWE-522) · Reference: OWASP LLM Top 10 — supply chain and sensitive information disclosure · OWASP API Top 10 API5:2023 · MITRE ATLAS — ML supply chain compromise · Composite element: embedded generative-AI feature in an enterprise application

Описание

The AI pipeline reaches object storage for datasets, embeddings and generated artefacts through a destination record held by the AI administration service. That record contains the storage endpoint, the container and the credential itself, and writing it is how an administrator rotates the credential. It is also how an attacker redirects the pipeline.

Requested with an ordinary business account — a token whose claims name a non-administrative role and carry an explicit non-administrator flag — the record returned in full, credential included, and a replacement was accepted. The administration interface hides this screen for that role, and hides it correctly; the decision is taken in the browser and the service behind it does not repeat the check. The consequence runs in both directions: the existing credential is disclosed to anyone who can authenticate, and the destination the pipeline writes to becomes the attacker's choice, so training and inference data lands wherever they say.

Доказателство**AI pipeline storage credentials readable and writable by a non-administrator**

SYNTHETIC EXAMPLE

An ordinary business account reads the pipeline's object-store credential and replaces the destination record.

```
analyst@wkstn ~$ bash

analyst@wkstn:~/ai$ cat bob-token-claims.json
Synthetic transcript — invented tenant, service, destination and key material
{"sub":"bob","tenant":"t-larkmoor","roles":["business.user"],"is_admin":false}

analyst@wkstn:~/ai$ curl -s -H "Authorization: Bearer $B0B" \
  https://ai.example.test/ai-svc/config/storage-destination
HTTP/1.1 200 OK (the UI hides this screen for this role)
{"name":"pipeline-artefacts","endpoint":"objstore.example.test",
 "container":"ai-artefacts", "sas":"sv=2099-01&sig=KxAb...<truncated - fabricated value>"}

analyst@wkstn:~/ai$ curl -s -o /dev/null -w '%{http_code}\n' -X PUT \
  -H "Authorization: Bearer $B0B" -H 'Content-Type: application/json' \
  --data @destination-we-control.json \
  https://ai.example.test/ai-svc/config/storage-destination
204 -- no administrator claim asserted, no role check on write
performed against a destination record we created, with written permission

▶ An is_admin:false token reads the pipeline's storage credential and redirects its output.
```

Illustrative example — synthetic data, not client evidence.

Evidence for F-01 — a token asserting a non-administrative role reads the pipeline's storage destination record, including its credential, then replaces the record. Illustrative reconstruction using synthetic data on reserved example.test names, the fictional tenant t-larkmoor and the placeholder account bob; not client evidence.

Стъпки за възпроизвеждане

- 1 Authenticate as the most ordinary account the tenant will issue and decode the token's claims, confirming it names a business role and asserts no administrative privilege.

- 2 Request the AI service's storage destination configuration directly, bypassing the administration interface entirely. In the sample transcript the response is **200 OK** and carries the endpoint, the container and the credential.
- 3 Replay the same request as a **PUT** carrying a destination record under your own control. The sample shows **204** — accepted, with no administrative claim asserted at any point.
- 4 Read the record back to confirm the replacement persisted. Testing stopped there: the destination was restored immediately, and no pipeline run was allowed to write to a tester-controlled store.

Въздействие

Two distinct losses. First, a credential for the store that holds every dataset, embedding and generated artefact the feature produces is handed to anyone who can log in, which for a tenant with ordinary staff accounts is a large population. Second, and worse, the attacker chooses the destination: subsequent training and inference data is written where they direct it, continuously, as part of normal operation and with no anomaly for anyone to notice. There is no application compromise to detect, because the application is working exactly as configured — it is the configuration that was taken. Any personal or commercially sensitive data that reaches the corpus follows the destination record.

Препоръка

- 1 Treat the credential in the destination record as compromised and rotate it now, before any code change. Check the storage account's own access logs for reads from outside the expected pipeline identity.
- 2 Require a server-side administrator check on this operation specifically, on the write path as well as the read path. Read access to a configuration area is not consent to write it.
- 3 Stop returning the credential at all. The configuration API should expose the destination's identity and status and never its secret material; rotation should write a reference to a managed secrets store rather than a value.
- 4 Prefer a workload identity over a long-lived shared credential, so that the pipeline's access to storage is bound to the pipeline rather than to a string that can be copied out of a configuration record.

HIGH

F-02 Cross-tenant access to conversations, documents and embeddings

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
High	7.7 AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N	AI datastore — threads, documents, embeddings, generated summaries	Отворена

Class: Authorisation bypass through user-controlled key (CWE-639) · incorrect authorisation (CWE-863) · Reference: OWASP LLM Top 10 — sensitive information disclosure · OWASP API Top 10 API1:2023 (BOLA) · OWASP WSTG-ATHZ · Composite element: multi-tenant AI feature with its own datastore

Описание

AI features generate and store a great deal: conversation threads, uploaded and ingested documents, embeddings derived from them, and generated summaries and artefacts. These usually live in a new service with its own datastore and its own identifier space, wired into the platform session for authentication but not into the platform's existing object-authorisation model. That is what was found here.

Content was created by an account in one tenant and its identifiers were then requested by an account in another, with no relationship between the two. Thread messages, the source document, the derived embeddings and the generated summary all returned successfully. The delete verb on the same object was correctly refused,

which makes the finding unambiguous: an ownership check exists in this service and was not applied to the read, export or metadata paths. The generated summary deserves separate mention: it can be more sensitive than any record behind it, because the analysis is the value.

Доказателство

One tenant's user reads another tenant's conversations, documents and embeddingsSYNTHETIC EXAMPLE

The AI datastore has its own identifier space and was never wired into the platform's object-authorisation model.

```
analyst@wkstn - bash

Synthetic transcript – invented tenants, identifiers and responses
carol (tenant t-pentreath) created the thread and uploaded the document
POST /ai-svc/threads as carol, t-pentreath
-> 201 {"thread_id":"th_9f41c0a7"}
every request below carries bob's token: tenant t-larkmoor, business.user
GET /ai-svc/threads/th_9f41c0a7/messages
-> 200 another tenant's conversation, in full
[{"role":"user","content":"summarise the attached pricing note"}, ...]
GET /ai-svc/documents/doc_41b8e2/content
-> 200 the source document itself, 148 kB
GET /ai-svc/embeddings?document_id=doc_41b8e2
-> 200 {"vectors": 1024, "model": "..."}
GET /ai-svc/threads/th_9f41c0a7/summary
-> 200 the generated summary – the analysis, not the record
DELETE /ai-svc/threads/th_9f41c0a7
-> 403 the ownership check exists – on one verb only

> Retrieval, export and metadata cross the tenant boundary; delete does not. One verb was checked.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-02 — content created in one tenant is retrieved by an account in another across four different paths, while the delete verb on the same object is refused. Illustrative reconstruction using synthetic data, the fictional tenants t-larkmoor and t-pentreath and the placeholder accounts bob / carol; not client evidence.

Стъпки за възпроизвеждане

- 1 Using an account in the first tenant, create a conversation thread and upload a document to it. Record the identifiers the service returns.
- 2 Authenticate as an unrelated account in the second tenant and request the thread's messages by identifier. The sample shows **200** and the conversation in full.
- 3 Repeat for the document content, the embeddings derived from it and the generated summary, testing each path separately rather than assuming they share a check.
- 4 Attempt the delete verb on the same object to establish whether the service has any ownership check at all; the sample shows **403**, which proves the check exists and was not applied consistently. No data belonging to anyone other than the tester's own accounts was retrieved.

Въздействие

One tenant can read another's conversations, source documents, derived embeddings and generated analysis. For a multi-tenant product this is the failure customers ask about by name in security questionnaires, and the one that converts a technical defect into a contractual and regulatory event: a confidentiality breach affecting identified third parties, discoverable by anyone with an ordinary login and a valid identifier. Because the identifiers appear in URLs, exports and support tickets, the precondition is weaker than it looks.

Препоръка

- 1 Authorise every object access against the requesting principal's tenant and entitlement, inside the data-access layer rather than in each handler, so a new endpoint inherits the check instead of having to remember it.

- 2 Apply the check per verb and per derived object. Retrieval, metadata, export, regeneration and deletion are separate paths, and embeddings and summaries are separate objects that inherit the sensitivity of their source.
- 3 Make identifiers unguessable as defence in depth, but do not treat that as the control — the finding is the missing check, not the shape of the identifier.
- 4 Add an automated test that creates an object as one tenant and asserts a refusal on every verb as another, and run it in the pipeline so the next endpoint cannot regress it quietly.

MEDIUM

F-03 Generative prompt configuration writable by an ordinary tenant user

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
Medium	7.1 AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:H/A:N	AI administration service — prompt extension points	Отворена

Class: Missing authorisation (CWE-862) · improper control of dynamically-managed code resources (CWE-913) · Reference: OWASP LLM Top 10 — prompt injection and improper output handling · OWASP API Top 10 API5:2023 · MITRE ATLAS — poison training or prompt data · Composite element: embedded generative-AI feature in an enterprise application

Описание

Each generative scenario in the feature executes a base prompt plus whatever a tenant has configured at an extension point — tone, terminology, policy text, output shape. Those extension points are ordinary configuration records held by the AI administration service, and they are the live instructions the model receives on every session of that scenario. The administration interface restricts them to an administrative role. The service does not.

An account with no administrative claim listed the extension points bound to live scenarios, replaced the content of one with a benign marker string and read it back as the active configuration. In this environment the consuming backend was not fully wired, so nothing written was ever executed by a model. That is why the rating is Medium while the illustrative CVSS base score is higher: the authorisation defect is live and complete, and only the consumer is absent. A test looking for prompt injection alone would have recorded a dead end and moved on — the write path is the finding, and it will be fully exploitable the day the consumer ships.

Доказателство

Generative prompt configuration writable by an ordinary tenant user

The live instruction store behind a generative scenario accepts a write from a token that asserts no privilege.

analyst@wkstn — bash

```
Synthetic transcript — invented service, scenario and configuration record
bob's token throughout: tenant t-larkmoor, roles [business.user], is_admin false

GET /ai-svc/config/prompt-extensions
-> 200 lists every extension point bound to a live scenario
[{"id":"pe-004","scenario":"reply-draft","state":"active"}, ...]

PUT /ai-svc/config/prompt-extensions/pe-004
-> 204 instruction content replaced; no administrator check

GET /ai-svc/config/prompt-extensions/pe-004
-> 200 our text read back as the live extension
{"appended":"<benign marker string written by the tester>"}

the consuming backend was not wired in this environment, so nothing
we wrote was ever executed by a model — the write path is the finding

> Writing the prompt store is writing the instructions every session of that scenario gets.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-03 — the extension points bound to live generative scenarios are listed, written and read back by a token asserting no administrative privilege. Illustrative reconstruction using synthetic data on reserved example.test names and the placeholder account bob; not client evidence.

Стъпки за възпроизвеждане

- 1 As an ordinary business account, list the AI service's prompt extension points and note which are bound to scenarios marked active.
- 2 Issue a **PUT** against one of them carrying a clearly marked, benign test string rather than an instruction. The sample shows **204**.
- 3 Read the record back and confirm the written content is returned as the live extension for that scenario.
- 4 Establish whether a model consumes it, by exercising the scenario in the application and comparing output. In the sample the consumer was not wired, which is recorded as a fact about the environment and not as a mitigation. The original content was restored immediately.

Въздействие

Write access to a prompt store is write access to the system instructions every user of that scenario receives. An attacker who holds it does not need prompt injection, because they can edit the instructions directly: steer output, suppress a refusal, insert a direction to disclose retrieved content, or change what the feature tells staff to do. It is also silent — the output still looks like the product, and the change lives in a configuration record that is not versioned, diffed or alerted on. Where the feature's output informs a business decision, the integrity of that decision is in the hands of anyone with a login.

Препоръка

- 1 Enforce the administrator check server-side on the write path for every prompt and template record, independently of the read path and of service entry.
- 2 Version prompt and extension content. A change to live instruction content should produce a diff, a named author, an alert and a rollback path, in the same way a code change does — because that is what it is.

- 3 Separate the scope that permits everyday generative use from the scope that permits configuring it. Where the platform ships a single service-wide scope, raise it with the vendor and meanwhile restrict the role carrying it to a named administrative group rather than a default business role.
- 4 Review the current content of every extension point against what it is supposed to say, and keep that comparison as a recurring check rather than a one-off.

MEDIUM

F-04 Agent tool boundary steerable by instructions planted in an ingested document

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
Medium	6.5 AV:N/AC:H/PR:L/UI:R/S:C/C:H/I:L/A:N	Agent orchestration layer and its tool authorisation	Отворена

Class: Improper neutralisation of input used for LLM prompting (CWE-1427) · confused deputy / excessive agency (CWE-441) · Reference: OWASP LLM Top 10 — prompt injection and excessive agency · OWASP Agentic Security Initiative · MITRE ATLAS — LLM prompt injection · Composite element: AI assistant with tool access inside a business application

Описание

The assistant can act as well as answer: it calls internal services to fetch records, run a summary pipeline and write artefacts. The question that matters is whose authority it acts under. Here each tool call was authorised against the orchestrator's own service identity, which is broad enough to serve every user, rather than against the identity of the person who asked.

A document was uploaded to the tester's own corpus containing an ordinary-looking instruction line in its body. A later, entirely benign question in a fresh thread caused the agent to read that document and act on the instruction, issuing a tool call scoped to a tenant the requesting account is not entitled to. The same operation, attempted directly with the same account's token, was refused. The injection is not the finding; what it reached is. The rating reflects exactly that: the tool in question returns business records across tenant boundaries, so a steerable boundary around it is serious, while the attack requires content to be ingested and a user to ask a question that touches it.

This report does not publish a working payload. The reproduction below describes the mechanism at the level a vendor advisory uses, which is sufficient for your engineers to confirm the behaviour and insufficient to hand anyone a tool.

Доказателство

Agent tool boundary steerable by instructions planted in an ingested document

SYNTHETIC EXAMPLE

The tool layer authorises the orchestrator's identity rather than the requesting user's.

```

analyst@wkstn ~$ sed -n '41,43p' tester-note.md
Synthetic transcript — invented agent, tool names and responses
analyst@wkstn:~/ai$ sed -n '41,43p' tester-note.md
paraphrased, not a published payload: an ordinary-looking instruction
line inside a document the tester uploaded to the tester's own corpus
[instruction directing the agent to call fetch_record for
entities in tenant t-pentreath and return the result]

a benign question is then asked in a new thread, as bob
POST /ai-svc/threads/th_2c71/messages
-> 200 the agent acts on the document's instruction
{"tool_calls": [{"name": "fetch_record",
"args": {"entity": "account", "tenant": "t-pentreath"}}]}

the same operation attempted directly, with the same user's token
GET /records?entity=account&tenant=t-pentreath
-> 403 refused: bob is not entitled to that tenant

Through the agent, the user obtained a result the platform refuses them directly.

```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-04 — an instruction carried in an ingested document causes the agent to issue a tool call scoped to another tenant, while the same call made directly with the requesting user's token is refused. The planted text is paraphrased, not reproduced. Illustrative reconstruction using synthetic data, invented tool names and the fictional tenants t-larkmoor / t-pentreath; not client evidence.

Стъпки за възпроизвеждане

- 1 Enumerate the tools the agent can invoke, their parameters and the identity each runs as. Where no such inventory exists, asking the assistant about its own capabilities in plain language will usually produce one.
- 2 Place a document containing a directive sentence into a corpus the agent ingests, using an account and a corpus belonging to the tester.
- 3 In a new thread, ask an ordinary question that causes the document to be retrieved, and record the tool calls the orchestration layer emits in response.
- 4 Attempt the same operation directly, with the same user's token and no agent in the path. A refusal there and a success through the agent together establish that the tool layer is authorising the orchestrator rather than the caller.

Въздействие

A user obtains, through the assistant, a result the platform refuses them directly — and the instruction that caused it can be placed by anyone able to get content into the corpus, which in a feature that ingests tickets, documents or shared files may include people with no account at all. The agent becomes a universal proxy with the broadest identity in the system, and the request looks legitimate in every log because the orchestrator was entitled to make it. Any entitlement model enforced in the application tier is bypassed rather than broken, and the tools that write rather than read turn the same path into an integrity problem.

Препоръка

- 1 Propagate the end user's identity into every tool call and authorise at the tool, against that identity. An agent should never hold an identity broader than the narrowest user it serves.

- 2 Treat all retrieved content as untrusted data rather than as instructions: separate the instruction channel from the content channel in the prompt structure, and label retrieved material explicitly as material to be reasoned about.
- 3 Constrain the tool surface. Give each scenario the smallest tool set it needs, require confirmation for state-changing tools, and apply the caller's own tenant and entitlement filters inside the tool rather than in the arguments the model supplies.
- 4 Log each tool call with the requesting principal, the arguments, the authorisation decision and the retrieval that preceded it, so the chain from ingested content to action is reconstructable after the fact.

MEDIUM

F-05 Model lifecycle and data ingestion operable without an administrator role

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
Medium	6.3 AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:L	AI administration service — model lifecycle and ingestion triggers	Отворена

Class: Missing authorisation (CWE-862) · execution with unnecessary privileges (CWE-250) · Reference: OWASP LLM Top 10 — supply chain and unbounded consumption · MITRE ATLAS — poison training data · OWASP API Top 10 API5:2023 · Composite element: embedded generative-AI feature in an enterprise application

Описание

Creating a model definition, attaching a dataset, starting a training run, promoting it and deleting it are plain operations over HTTP, and so is triggering the job that pulls business data into the corpus. All of them were available to an account asserting no administrative privilege, gated only by the coarse scope that also permits everyday generative use.

Ingestion is the one to look at twice. It decides what the feature reads, which makes it a data-poisoning path, and it moves volume, which makes it a convenient engine for relocating tenant data — particularly in combination with F-01, where the destination that data lands in is also writable. Model deletion is the quiet member of the set: it is an availability and integrity action that leaves the feature degraded or reverted without anything resembling an attack in the logs.

Доказателство

Full model lifecycle and data ingestion operable without an administrator role

SYNTHETIC EXAMPLE

Create, train, ingest and delete, all from a token that asserts no administrative privilege.

```
analyst@wkstn ~$ bash

Synthetic transcript — invented service, model and run identifiers
bob's token throughout: tenant t-larkmoor, roles [business.user], is_admin false

POST /ai-svc/models
  -> 201 {"model_id":"m-0f12"}
POST /ai-svc/models/m-0f12/train
  -> 202 {"run_id":"tr-7731","status":"queued"}
POST /ai-svc/ingestion/runs
  -> 202 {"run_id":"ig-2048","scope":"tenant"}
GET /ai-svc/ingestion/runs/ig-2048
  -> 200 {"records":0,"state":"cancelled-by-tester"}
DELETE /ai-svc/models/m-0f12
  -> 204 the model we created, removed by the same token

every operation above ran against records the tester created, with written
permission; the ingestion run was cancelled before it read any data

> Lifecycle and ingestion are administrative, gated only by the everyday-use scope.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-05 — model creation, a training run, an ingestion run and a deletion, all accepted from a token asserting no administrative privilege. Every operation shown ran against records created by the tester and the ingestion run was cancelled before reading data. Illustrative reconstruction using synthetic data and invented identifiers; not client evidence.

Стъпки за възпроизвеждане

- 1 As an ordinary business account, create a model definition against the AI service and record the identifier returned.
- 2 Start a training run against that definition and confirm the service accepts it (202 in the sample), then trigger an ingestion run and read its status.
- 3 Delete the model definition created in step one, confirming the full lifecycle is available to the same token.
- 4 Repeat each operation with an administrative account and compare the responses, to establish that the service is not distinguishing between the two. Scope rules required written permission for the resource-consuming operations; the ingestion run was cancelled before it read any data and the model definition was removed afterwards.

Въздействие

An ordinary user can choose what the AI feature learns from and when, start work that consumes compute and budget, and delete the models the feature depends on. Poisoned or redirected ingestion is the most durable of these, because it affects every answer the feature gives afterwards and is almost impossible to attribute later without ingestion-level logging. Combined with F-01 the pair is materially worse than either alone: one finding chooses what is read, the other chooses where it is written.

Препоръка

- 1 Gate model lifecycle and ingestion behind a narrower scope than generative usage, and enforce that scope server-side on each operation rather than once at service entry.
- 2 Require approval or rate limiting for ingestion runs rather than allowing any scope holder to start one on demand, and record what each run read and where it wrote.

- 3 Make deletion recoverable: soft-delete model definitions with a retention window and an alert, so a destructive action is reversible and visible.
- 4 Audit the records that already exist. Any model, dataset or ingestion run created outside the expected administrative group should be reconciled before the gate is applied, not after.

LOW

F-06 Unauthenticated AI endpoint discloses internal architecture in its error body

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
Low	5.3 AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	AI generation endpoint — error handling	Отворена

Class: Generation of error message containing sensitive information (CWE-209) · exposure of sensitive information (CWE-200) ·
Reference: OWASP LLM Top 10 — sensitive information disclosure · OWASP API Top 10 API8:2023 · OWASP WSTG-ERRH ·
Composite element: AI endpoints published alongside an existing application

Описание

A malformed request sent to the generation endpoint with no credential at all returned a verbose error. The body named the orchestration host and port behind the gateway, the vector store and its port, the inventory of tools the agent can invoke, and a source file and line number from the orchestration service.

Individually these are information leaks of the dulllest kind. Collected in one anonymous response they are a map: the tool inventory is the live contract rather than the documented one, and it tells an attacker exactly which of the other findings in this report are worth pursuing and in what order. The adjacent streaming route did require a session, which indicates the authentication requirement was applied route by route rather than at the service.

Доказателство**Unauthenticated AI endpoint returns internal architecture in its error body**

SYNTHETIC EXAMPLE

A malformed request with no session discloses internal hosts, ports, the tool inventory and a source path.

```
analyst@wkstn ~$ curl -s -i -X POST \
  Synthetic transcript - invented internal hostnames, ports and stack frame
  -H 'Content-Type: application/json' --data '{"prompt":}' \
  https://ai.example.test/ai-svc/v1/generate # no Authorization header

HTTP/1.1 500 Internal Server Error
{"error":"orchestrator_failure",
 "detail":{"upstream http://ai-orch-02.int.example.test:8711/v1/plan",
 "vector_store":"vec-01.int.example.test:6333",
 "tools":["fetch_record","run_summary_pipeline","write_artefact"],
 "trace":"at Orchestrator.plan (/srv/ai-orch/plan.js:214)"}

analyst@wkstn ~$ curl -s -o /dev/null -w '%{http_code}\n' \
  https://ai.example.test/ai-svc/v1/generate/stream # no credential
401 -- the streaming route does require a session; this one does not

> An anonymous request is handed internal hostnames, ports, tool names and a source line.
```

Illustrative example — synthetic data, not client evidence.

Evidence for F-06 — an anonymous malformed request returns internal hostnames and ports, the agent's tool inventory and a source path, while the adjacent streaming route correctly requires a session. Illustrative reconstruction using synthetic data on reserved example.test names and invented internal host names; not client evidence.

Стъпки за възпроизвеждане

- 1 Send a syntactically malformed request body to the generation endpoint with no authorisation header.
- 2 Record the response status and body. The sample returns **500** with the upstream URL, the vector store address, the tool list and a stack frame.
- 3 Enumerate the sibling routes on the same service and compare which require a session, to establish whether authentication is applied at the service or per route.
- 4 Cross-reference the disclosed internal names against the external estate to confirm they are not otherwise published. No attempt was made to reach the internal hosts directly.

Въздействие

Reconnaissance that would otherwise take an attacker time, or an account, is handed over anonymously. The tool inventory in particular narrows the search for the agent boundary described in F-04, and the internal host and port give an attacker who later gains any foothold a specific destination rather than a subnet. The rating is Low because nothing here is directly exploitable on its own; the value is entirely in what it accelerates.

Препоръка

- 1 Return a generic error body with a correlation identifier and keep the detail in the server-side log. The client needs to know that something failed and how to reference it, and nothing more.
- 2 Require authentication at the service rather than per route, so a new endpoint is protected by default and an omission is a deliberate exception rather than an oversight.
- 3 Remove stack frames and upstream URLs from production responses, and confirm the framework's debug mode is disabled in every environment that is reachable from outside.
- 4 Treat the tool inventory as sensitive configuration. It should never be reachable without a session, and never from an error path.

LOW

F-07 No rate limiting or cost control on the generation endpoint

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
Low	4.3 AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:L	AI generation endpoint — throttling and quota	Отворена

Class: Uncontrolled resource consumption (CWE-400) · allocation without limits (CWE-770) · Reference: OWASP LLM Top 10 — unbounded consumption · OWASP API Top 10 API4:2023 · Composite element: AI endpoints published alongside an existing application

Описание

A bounded burst of generation requests from a single ordinary account, agreed in writing beforehand and run against a non-production tenant, met no throttle. Every request was accepted, none was refused, and the responses carried no rate-limit headers, no retry guidance and no quota of any kind. Response times lengthened materially across the burst.

Denial of service was explicitly out of scope and no outage was induced or attempted; the finding is the absence of a control rather than a demonstrated outage, and it is rated accordingly. It matters more for AI endpoints than for ordinary ones for a reason that is commercial rather than technical: each request carries a direct unit cost, so an unthrottled endpoint is a billing exposure as well as an availability one, and the bill arrives whether or not

anyone noticed.

Доказателство

No rate limiting or cost control on the generation endpointSYNTHETIC EXAMPLE

A bounded, authorised burst from one ordinary account met no throttle, no quota and no 429 at any point.

```
analyst@wkstn ~$ for i in $(seq 1 120); do \
  curl -s -o /dev/null -w '%{http_code} %{time_total}\n' \
  -H "Authorization: Bearer $B0B" -d @prompt.json \
  https://ai.example.test/ai-svc/v1/generate & done | sort | uniq -c

count status
-----
120 200 -- 120 of 120 accepted
  0 429 -- none throttled

analyst@wkstn ~$ curl -sI ... /ai-svc/v1/generate | grep -iE 'ratelimit|retry'
(no output) -- no RateLimit-* headers, no Retry-After, no quota

time_total rose from 0.9 s to 11.4 s across the burst; no outage was
induced and none was attempted — the finding is the absent control

► One ordinary account can consume the tenant's generation capacity, and its budget.
```

Illustrative example — synthetic data, not client evidence.

Evidence for F-07 — a bounded, authorised burst from one ordinary account is accepted in full, with no refusals and no rate-limit or quota headers in any response. Illustrative reconstruction using synthetic counts and timings against a non-production tenant; not client evidence.

Стъпки за възпроизвеждане

- 1 Agree the burst size, the target environment and the window in writing before testing, since this operation consumes resources.
- 2 Issue a bounded set of concurrent generation requests from a single ordinary account and tabulate the response statuses.
- 3 Inspect the responses for rate-limit, quota or retry headers, and the account's own usage view for any consumption counter.
- 4 Record response-time drift across the burst as an observation. Stop at the agreed ceiling: the objective is to establish that no limit exists, not to find where the service fails.

Въздействие

One ordinary account — or one credential leaked from a client bundle — can consume the tenant's generation capacity and its budget, degrading the feature for every other user and producing a cost that is only visible on the invoice. Without per-account counters there is also no way to attribute the consumption afterwards, so the first clear signal is a finance question rather than a security alert.

Препоръка

- 1 Apply a per-account and per-tenant rate limit on the generation endpoints, and return the standard rate-limit and retry headers so clients can behave correctly.
- 2 Set a token or spend budget per tenant with an alert threshold below the hard limit, so unusual consumption is noticed before it is paid for.
- 3 Bound input size and generation length at the gateway, and reject rather than truncate oversized requests.

- 4 Expose per-account consumption in the tenant's own administration view, so the organisation can attribute usage without raising a support case.

LOW

F-08 Permissive CORS on the AI API: arbitrary origin reflected with credentials allowed

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
Low	4.3 AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N	AI API gateway — cross-origin resource sharing policy	Отворена

Class: Permissive cross-domain policy with untrusted domains (CWE-942) · origin validation error (CWE-346) · Reference: OWASP API Top 10 API8:2023 · OWASP WSTG-CLNT · OWASP LLM Top 10 — sensitive information disclosure · Composite element: AI endpoints published alongside an existing application

Описание

The AI API echoes whatever origin a request presents, and returns **Access-Control-Allow-Credentials: true** alongside it. The null origin is accepted too, so a sandboxed frame qualifies, and no **Vary** header accompanies the reflection, which makes it cacheable. The policy is not an allowlist with a permissive entry; there is no allowlist.

In effect the browser's own boundary is switched off for these endpoints. Any page a signed-in user visits can issue credentialed requests to the AI API as that user and read the responses. The rating is Low in isolation, because the attacker still needs the victim to be authenticated and to visit content they control, and because the routes involved are the user's own. It is listed last and rated last for that reason — and the remediation plan in section 7 notes that it should be read together with F-01 and F-03, since a permissive policy in front of an endpoint with no server-side role check turns an authorisation defect into a drive-by one.

Доказателство

Permissive CORS on the AI API: arbitrary origin reflected, credentials allowed
Any page the user visits can drive the AI endpoints as that signed-in user and read the responses.

SYNTHETIC EXAMPLE

```
analyst@wkstn ~$ curl -s -i -X OPTIONS \
  Synthetic transcript — invented origin and response headers
  -H 'Origin: https://unrelated.example.test' \
  -H 'Access-Control-Request-Method: POST' \
  https://ai.example.test/ai-svc/v1/generate

HTTP/1.1 204 No Content
Access-Control-Allow-Origin: https://unrelated.example.test <- reflected
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: Authorization, Content-Type
Vary: (absent) <- the reflection is cacheable

analyst@wkstn ~$ curl -s -i -X OPTIONS -H 'Origin: null' ... | head -2
HTTP/1.1 204 No Content
Access-Control-Allow-Origin: null <- a sandboxed frame is also allowed

> Origin is echoed rather than checked, and credentials are permitted with it.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-08 — a pre-flight request from an unrelated origin is answered with that origin reflected, credentials allowed, no **Vary** header, and the null origin also accepted. Illustrative reconstruction using synthetic data on reserved example.test names; not client evidence.

Стъпки за възпроизвеждане

- 1 Issue a pre-flight **OPTIONS** request to an AI endpoint carrying an unrelated **Origin** header.
- 2 Inspect the response for **Access-Control-Allow-Origin**, **Access-Control-Allow-Credentials** and **Vary**. The sample shows the origin reflected, credentials allowed and no **Vary**.
- 3 Repeat with the null origin to establish whether sandboxed contexts are accepted.
- 4 Confirm which routes the policy covers by repeating against several paths on the same service; a policy applied at the gateway usually covers all of them.

Въздействие

A page outside the organisation's control can act as the signed-in user against the AI API and read what comes back — conversations, documents, generated output — without ever handling the user's credential. Where that same API also accepts administrative writes without a server-side role check, as described in F-01 and F-03, the combination removes the last precondition those findings had: the attacker no longer needs an account of their own, only a link a member of staff will open.

Препоръка

- 1 Replace reflection with an explicit allowlist of origins the product genuinely serves, and refuse everything else. Never reflect, and never accept the null origin.
- 2 Keep **Access-Control-Allow-Credentials: true** only where a credentialed cross-origin call is actually required, and remove it elsewhere.
- 3 Send **Vary: Origin** on every response that varies by origin, so an intermediate cache cannot serve one caller's policy to another.
- 4 Apply the policy once at the gateway rather than per service, and add a check to the pipeline that fails the build if a response reflects an arbitrary origin.

INFORMATIONAL

F-09 Positive observation: the administrator gate exists, one service away

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
Informational	n/a не е уязвимост; не е присвоена оценка	Platform identity service, compared with the AI services	Observation

Class: Positive control observation — not a vulnerability · Reference: Recorded as a control comparison supporting F-01, F-03 and F-05 · Composite element: control comparison on the same platform and tenant

Описание

Not every observation in a report is a defect, and this one is the most useful page in the document for the team that has to fix the rest. With the same token, in the same tenant, on the same platform, the AI administration services accepted privileged writes while the platform's own identity service refused the equivalent operations outright, with a clean denial naming the required role.

That comparison settles the question that usually delays remediation. The identity context was present and correct; a service standing next to the AI services was already reading it on write. So the gate is not a platform limitation or an architectural impossibility — it is a check that exists and was never applied to this surface, which makes the conversation with a platform vendor considerably shorter. One further positive, recorded in the same pass: the shared observability store held latency, token counts and tenant identifiers for AI requests but not

prompt or completion bodies, where the opposite is common.

Доказателство

SYNTHETIC EXAMPLE

Positive observation: the administrator gate exists, one service away

The same token, the same tenant and the same platform — refused by the identity service, accepted by the AI services.

```
analyst@wkstn ~ bash

Synthetic transcript — invented services and responses; the comparison
is the point, not the individual status codes
{"sub": "bob", "tenant": "t-larkmoor", "roles": ["business.user"], "is_admin": false}

AI administration surface — same token:
PUT    /ai-svc/config/storage-destination
-> 204 accepted
PUT    /ai-svc/config/prompt-extensions/pe-004
-> 204 accepted
POST   /ai-svc/ingestion/runs
-> 202 accepted

Platform identity service — same token, same tenant:
GET    /identity-svc/users
-> 403 {"error": "administrator role required"}
PUT    /identity-svc/users/u-0012/roles
-> 403 same denial on the write path

> The identity context was present. One service evaluated it on write; the AI services did not.
```

Illustrative example — synthetic data, not client evidence.

Evidence for F-09 — the same token is accepted on three AI administration operations and refused on the identity service's equivalent operations. Illustrative reconstruction using synthetic data and invented service names; not client evidence.

Как беше установено това

- 1 Take a single ordinary business token and replay the administrative operations of the AI services, recording the status returned for each.
- 2 Replay the equivalent administrative operations of a neighbouring service on the same platform, with the same token in the same session.
- 3 Compare the two sets. A divergence establishes that the identity context is available and that enforcement is a per-service decision. Inspect the shared trace store in the same pass, to confirm which fields of an AI request are retained.

Въздействие

None directly — this observation records working controls. Its value is evidential: it removes the feasibility argument from the remediation discussion for F-01, F-03 and F-05, and gives your engineers, or your vendor, a correctly behaving service on the same stack to model the fix on.

Какво да направите с това наблюдение

- 1 Use the identity service's behaviour as the reference when specifying the gate for the AI administration surface, and compare the two endpoint by endpoint after remediation.
- 2 Keep prompt and completion bodies out of the shared observability store, record both observations in the control inventory, and state the intention somewhere durable so a future tracing change does not quietly reverse it.

Напомняне за този раздел. F-01 до F-09 са взети от формата на няколко анонимизирани ангажимента и представени, сякаш идват от един. Всеки запис, име на услуга, клиентска организация, идентификатор, акаунт и оценка по-горе са синтетични.

РАЗДЕЛ 7

План за отстраняване

Същите препоръки, групирани на вълни и подредени. Последователността има по-голямо значение от списъка: единна проверка на правата затваря три констатации, а изпълнението му преди работата по закаляване прави тази работа по-малка.

Последователността и защо е именно тази

Ограничаването е първо, защото данни за достъп, които са били четими от всеки акаунт в клиентската организация, трябва да се третират като вече известни, а подмяната им не изисква издание. Налагането е второ, защото е една промяна, която затваря три констатации и премахва усилвателя зад останалите. Изоляцията следва, защото е друг екип и друга кодова база и може да върви паралелно. Границата на агента е следваща, защото е работа по проектиране, а не конфигурация. Закаляването на крайните точки е последно, с едно изключение, отбелязано в таблицата: F-08 е подредена рано, въпреки оценката си Low, защото позволяваща политика за различен произход пред неохраняван административен запис е съществено различен проблем от позволяваща политика сама по себе си.

Вълна	Какво се променя	Затваря	Отговорник	Последователност
1 — Ограничаване	Подменете данните за достъп до хранилището на процеса и тези за връзката с бизнес системата. Прегледайте журналите за достъп до хранилището и текущото съдържание на всеки запис за местоназначение, разширение на подсказка и модел за неочаквани промени.	Ограничава F-01	Отговорник за платформата / AI услугата	Незабавно, преди каквато и да е промяна в кода
2 — Налагане	Една проверка за оторизация само за администратори върху цялата административна повърхност на AI, налагана от сървъра поединично при всяка променяща операция, по модела на услугата за идентичност в F-09. Спрете да връщате материал с данни за достъп от конфигурационните крайни точки.	F-01, F-03, F-05	Отговорник за AI услугата, заедно с доставчика на платформата, където услугата е предоставена от доставчик	Текущият цикъл; започва, щом вълна 1 е завършена
3 — Изолиране	Собственост на ниво обект и оторизация по клиентска организация в слоя за достъп до данните на AI хранилището, приложени към всяка операция и към производните обекти — векторни представяния, резюмета, експорти.	F-02	Екип по услугата за AI данни	Текущият цикъл; върви паралелно с вълна 2

Вълна	Какво се променя	Затваря	Отговорник	Последователност
4 — Ограничете агента	Пренасяйте идентичността на крайния потребител във всяко извикване на инструмент и оторизирайте на нивото на инструмента. Разделете канала с инструкции от извлеченото съдържание. Сведете набора инструменти на всеки сценарий до необходимото.	F-04	Екип по AI платформата / оркестрацията	Следващият цикъл; промяна в проектирането, не конфигурация
5 — Закалете крайните точки	Общи тела на грешките с идентификатор за съотнасяне; удостоверяване, изисквано на нивото на услугата, а не за всеки маршрут; ограничения на честотата за всеки акаунт и клиентска организация и бюджет за разход; изричен списък с разрешени произходи с Vary.	F-06, F-07, F-08	Екип по API шлюза и приложението	F-08 наред с вълна 2; F-06 и F-07 следващия цикъл
6 — Поддържайте го затворено	Въведете версии и известия за промените в подсказките и конфигурацията. Записвайте в журнал субекта, твърденията, операцията и решението за всяка оторизация в AI услугата. Добавете регресионни тестове между клиентски организации и за различен произход в процеса.	Предотвратява повторение	Отговорник за AI услугата със сигурностно инженерство	Непрекъснато, с начало в текущия цикъл

Обобщение по констатация

ID	Какво да направите	Вълна	Усилие
F-01	Подменете данните за достъп; охранете записа; спрете да връщате материал с тайни	1 и 2	Часове, после дни
F-02	Оторизирайте достъпа до обектите за всяка операция в слоя за достъп до данните	3	Дни
F-03	Охранете записа; въведете версии и известия за конфигурацията на промптите	2 и 6	Дни
F-04	Пренасяйте идентичността на потребителя във всяко извикване на инструмент и оторизирайте там	4	Промяна в проектирането
F-05	По-тесен обхват за жизнения цикъл и зареждането; одобрение за изпълненията по зареждане	2	Дни
F-06	Общо тяло на грешката; удостоверяване на нивото на услугата, не за всеки маршрут	5	Часове
F-07	Ограничения на честотата за всеки акаунт и клиентска организация и бюджет за разход	5	Дни
F-08	Изричен списък с разрешени произходи, без отразяване, без null, с Vary	2 (рано)	Часове
F-09	Използвайте като референтна реализация за вълна 2	2	Няма — наблюдение

Усилието е нашата оценка за самата работа, не за контрола на промените около нея, и в този примерен документ е илюстративно като всичко останало. Отговорниците са ролите, които бихме очаквали да държат всяка промяна; вашата организация може да ги съотнесе различно.

Ако AI функционалността идва от доставчик на платформа. Вълни 2 и 3 може да са промяна на доставчика, а не ваша. Раздел 6 записва операцията и върнатия отговор за всеки опит, така че съответните констатации могат да бъдат предадени както са, а F-09 ви дава услуга на същата платформа, която вече налага проверката, която искате. Междувременно частта, която е ваша, е съотнасянето на ролите: ограничете обхвата, който покрива AI услугата, до посочена административна група, вместо до роля по подразбиране, и третирайте включването на AI функционалност като промяна, която дава права, с прегледа, който това предполага.

РАЗДЕЛ 8

Повторен тест и приключване

Отстраняването е завършено само когато някой е атакувал поправката. Повторният тест е включен по стандарт — той не е добавка и не се остойностява отделно.

Какво покрива повторният тест

Когато екипът ви потвърди, че отстраняването е завършено, се връщаме и тестваме отново всяка констатация. Повторният тест не е преглед на документация: всяка констатация се атакува отново по пътя за възпроизвеждане в раздел 6, така че поправката се потвърждава чрез това, че атаката вече не работи, а не чрез описание.

- 1 Повторно се тества всяка констатация, не извадка. Всичките девет реда в раздел 5, включително Low и положителното наблюдение — контрол, който е работел, си заслужава да бъде потвърден, че продължава да работи след промяната около него.
- 2 Тества се поправката, както и обходният път около нея. Проверяваме, че пътят за възпроизвеждане вече се проваля, и търсим очевидното заобикаляне: проверка, приложена към една операция, но не към съседната ѝ, или данни за достъп, подменени на едно място, но не на друго.
- 3 Вълните по налагане се проверяват операция по операция. За вълна 2 повтаряме пълния набор административни операции с обикновен токен и записваме всеки статус, така че приключването се основава на опис, а не на извадка.
- 4 Регресия в засегнатия компонент. Там, където една поправка променя удостоверяването, оторизацията, обхвата на инструментите или обработката на изхода, повторно задействае обкръжаващата функционалност — включително че функцията продължава да работи за правилните роли.
- 5 Ясен резултат за всяка констатация. Всяка констатация се преиздава като затворена, частично отстранена или отворена, с бележка за всеки статус, различен от затворен. Нов проблем, въведен от работата по отстраняването, също се докладва, макар повторният тест да не е втора пълна оценка.

Какво заявява писмото за приключване

След повторния тест издаваме подписано писмо за приключване наред с преиздадения отчет. То е написано, за да бъде споделяно: с клиенти, които извършват проверка на доставчици, с онзи, който води работата ви по управление на AI, и с борд, който иска една страница.

Писмото отразява

- ▢ Какво е било тествано, от кого, в кои дати и на кои препратки се опира.
- ▢ Колко констатации са били повдигнати и с каква критичност.
- ▢ Колко са потвърдени като затворени и датата на повторния тест.
- ▢ Всяка констатация, която остава отворена, с нейната критичност и заявения план на клиента.
- ▢ Името и подписът на консултанта, който го е проверил.

Писмото умишлено не отразява

- ▢ Стъпки за възпроизвеждане, полезни товари, доказателства или крайни точки.
- ▢ Имена на услуги, описи на инструменти или архитектурни подробности.
- ▢ Всичко, което би помогнало на нападател, който се е сдобил с писмото.
- ▢ Каквото и да е твърдение, че средата вече е без уязвимости.
- ▢ Каквото и да е сертификация или изявление за съответствие — ние не сме одитор, оценител или сертифициращ орган.

Защо включваме повторния тест, вместо да го продаваме. Поправка, която никога не е била проверена, е план, не резултат — особено там, където правилността е невидима от интерфейса.

ПРИЛОЖЕНИЕ А

Достъп, акаунти и инструменти

AI ангажиментите са необичайно чувствителни към подготовката, защото по-голямата част от стойността е в тестването на оторизацията, а тестването на оторизацията е невъзможно без акаунти.

Какво поискахме и защо

Какво	Защо има значение
По два акаунта за всяка роля, в един и същ клиентска организация	Двама потребители на едно и също ниво на права е това, което прави възможно тестването на хоризонталния контрол на достъпа — един достига разговорите и документите на другия. Един акаунт за всяка роля тества само вертикалните граници, което е по-лесната половина.
Един административен акаунт	Противоинтуитивно, но необходимо: тестерът трябва да види административната повърхност да работи както е предвидено, за да може да я изброи, и само тогава може да опита всяка операция отдолу. Без това изброяването е догадка.
Втори клиентска организация	Изоляцията между клиентски организации не може да бъде тествана отвътре на една клиентска организация. Осигуряването често е бавно, затова това се заявява, преди ангажиментът да бъде запазен, а не на първия ден.
Среда с данни	Празна база знания и липса на история на разговорите означава, че няма нищо, което да бъде достигнато през граница. Ясно обозначено синтетично съдържание в корпуса струва повече от достъп до продукционната среда и избягва цяла категория трудни разговори.
Списъкът с инструменти на агента, с параметрите	Всеки инструмент, който моделът може да извика, какво прави, какво приема и под коя идентичност работи. Там, където такъв документ не съществува, самото му написване е ценно — а липсата му обикновено е констатация в изчакване.
Писмено изявление какво функцията има право да прави	Кои роли могат да я използват, кои области от данни може да докосва, кои действия може да извършва от името на потребител, какво трябва да откаже. Без това тестерът може да отчете какво прави системата, но не и къде тя се разминава с това, което е било предвидено — а именно тази разлика е една констатация за оторизация.
Бележки по архитектурата и достъпен инженер	Къде се намира слой за оркестрация, кои хранилища съдържат подсказки, векторни представяния и артефакти, кои данни за достъп държи процесът и кои части са управлявани от доставчик. Плюс писмено решение дали зареждането на данни и обученията могат да бъдат задействани.

Използвани инструменти

Етап	Обичайни инструменти	За какво беше използвано
Изброяване на повърхността	Прихващащ прокси, откриване на маршрути и съдържание, изброяване през certificate transparency и DNS	Установяване на действителната повърхност: крайните точки на AI услугата наред със съществуващите на приложението и всичко публикувано без намерение.
Тестване на административния слой	Прихващащ прокси с ръчен повторител и скриптирано повтаряне на записан набор операции през няколко токена	Повтаряне на всяка променяща операция с най-обикновения токен, който клиентската организация би издал, и записване на това, което сървърът е наложил, а не което интерфейсет е подсказвал.
Оторизация и изолация	Инструменти за повтаряне на заявки, задвижвани от матрица на акаунтите и клиентските организации, със сравнение на отговорите	Обхождане на пространството на идентификаторите в двете посоки през акаунти и клиентски организации, като извличането, метаданните, експортирането и изтриването са задействани отделно.
Тестване на агента и инжектирането	Собственият интерфейс и API на продукта, контролиран корпус от съдържание, притежавано от тестерите, и проследяване на извикванията на инструменти	Установяване на договора за инструментите, след което определяне какво инструкция в заредено съдържание действително би могла да достигне през него.
Проверка и отчитане	Скриптирано повторение на заявки, събиране на доказателства и вътрешен процес за отчитане	Независимо възпроизвеждане на всяка констатация, събиране на доказателства и генериране на този документ и машинно четимия експорт.

Инструментите не произвеждат констатации. Всичко в раздел 6 беше проверено на ръка и, където беше безопасно, задействано, за да се демонстрира действително въздействие. Изходът на един инструмент е кандидат за проучване, никога ред в отчета — а при AI функционалностите това има по-голямо значение от обичайното, защото генериран корпус от опити за инжектиране ще произведе огромно количество изход и почти никакви констатации. Кандидати, които се оказаха да не достигат нищо в контекста, бяха отхвърлени, вместо да бъдат отчетени.

ПРИЛОЖЕНИЕ Б

Извън обхвата и какво този тест не е

Какво беше изключено, защо и какво означава това за увереността, която този отчет дава. Всичко изброено тук не беше тествано и нищо в този отчет не бива да се чете като твърдение за него.

Изключено	Причина	Какво означава това за вас
Хостваната услуга за модела на трета страна	Не е на клиента да го разрешава. Тестването на система, която не притежавате, изисква писменото разрешение на нейния собственик, а никой доставчик не го дава на тестер на клиент.	В обхвата бяха вашето приложение, слойът за оркестрация, подсказките, пътят на данните и данните за достъп — включително как вашата страна обработва това, което моделът връща. Всеки, който предлага да тества самата услуга за модела, не може законно да го изпълни.
Отказ на услуга и тестване под натиск	Изключено по договореност. Умишленото изтощаване на производствена услуга е неразлично от прекъсване, а при AI крайна точка с отчитане е и скъпо.	F-07 отразява липсата на ограничаване на честотата като наблюдение от ограничен, разрешен пик. Това не е измерване на това как услугата се държи при обемна атака.
Red teaming на безопасността на модела	Различно упражнение с различни входни данни: то изисква писмена политика за съдържанието, спрямо която да се измерва, голям обем опити и проверяващи с подходяща подготовка.	Този отчет не казва нищо за това дали асистентът може да бъде подтикнат да произведе вредно, предубедено или извън политиката съдържание. Отчитаме това, на което се натъкнем; това не е покритие.
Оценка на модела	Точността, честотата на халюцинации, релевантността на извличането и регресията между версиите на модела се измерват с бенчмарк и маркиран набор от данни.	Нищо тук не казва дали функцията е добра. Това се измерва непрекъснато от екипа, който я притежава, а не в тест по сигурността в един момент от времето.
Управление, увереност и сертификация	Ние сме фирма за офанзивно тестване по сигурността, не одитор, оценител или сертифициращ орган.	Този отчет произвежда техническо доказателство, което онзи, който води работата ви по управление или увереност, може да използва. Той не ви прави съответстващи на каквото и да е и не сертифицира нищо. Вижте Приложение В.
Социално инженерство срещу служители на клиента	Изключено по договореност. Фишингът и претекстването се поръчват отделно, за да могат съгласието и работата с данните на служителите да бъдат уговорени самостоятелно.	Човешкият слой не беше тестван. До няколко от констатациите тук нападател, който първо е фишал обикновен акаунт, би стигнал по-бързо — а това, което е показателно, са всички права, от които повечето от тях имат нужда.
Всичко, което не е названо в подписания обхват	Тестът за проникване е разрешен срещу конкретен списък.	Системи, открити по време на тестването, но извън подписания обхват, бяха докладвани като наблюдения и не бяха изследвани.

Извън обхвата означава нетествано, не сигурно. Това е най-често погрешно разчитаният ред във всеки отчет от тест за проникване. Ако една система не присъства в раздел 2, този документ не твърди абсолютно нищо за нея — нито добро, нито лошо. Там, където едно изключение оставя риск, който ви интересува, разговорът по резултатите е правилното място той да бъде включен в следващата работа.

ПРИЛОЖЕНИЕ В

Препратки и какво представляват

Този ангажимент се опря на документите по-долу. Всеки е описан като това, което действително е. Нито един от тях не е стандарт, по който този отчет сертифицира каквото и да е.

Препратка	Какво представлява	Как беше използвано тук
OWASP Top 10 for LLM Applications	Поддържан от общността списък на категориите риск, най-често наблюдавани в приложения, изградени върху големи езикови модели. Документ за осведоменост и покритие, не рамка за контроли и не подлежи на сертификация.	Като контролен списък за покритие при определянето на обхвата и отново при прегледа, за да се потвърди, че никоя категория не е останала неразгледана. Няколко констатации тук се съотнасят към повече от една категория, а някои категории не произведоха нищо.
OWASP Agentic Security Initiative	Проект на OWASP, документиращ заплахите и референтните архитектури, специфични за агентните системи — използване на инструменти, автономност, памет и делегиране. Публична препратка, не стандарт.	За да бъде поставено в рамка тестването на оторизацията на агента и инструментите зад F-04, и по-специално моделът на „обърканото доверено лице“, при който инструментът оторизира оркестратора, а не извикващия.
MITRE ATLAS	База знания за тактиките и техниките на противника, наблюдавани срещу системи с AI, структурирана по модела на ATT&CK. Таксономия, не сертификация.	За да бъдат назовани техниките последователно в редовете с класа на констатацията, така че собствената ви работа по откриване и разузнаване на заплахите да може да се съотнесе към същия речник.
NIST AI Risk Management Framework	Доброволна рамка, публикувана от NIST, за управление на риска през целия жизнен цикъл на една AI система, организирана около функциите управление, картографиране, измерване и справяне. Тя не е стандарт, подлежащ на сертификация, и NIST не издава сертификация по нея.	Като контекст. Там, където собствената програма на организацията за AI риск изисква независимо техническо тестване на система, отчет от този вид е входни данни за тази дейност. Не правим оценка на вашата програма и не твърдим нищо за съответствието ви с нея.
OWASP API Security Top 10 и OWASP Web Security Testing Guide	Утвърдени публични препратки за обикновения уеб и API слой. Методология за тестване и материал за осведоменост.	AI крайните точки са уеб крайни точки, а констатациите по закаляване F-06 до F-08 идват директно от този слой. AI функционалностите често се изграждат бързо и встрани от основното приложение, затова пропускат защитите на нивото на рамката, които останалата част от платформата получава безплатно.

Препратка	Какво представлява	Как беше използвано тук
ISO/IEC 42001 и Регламентът на ЕС за изкуствения интелект	Съответно стандарт за система за управление и правен инструмент. И двете са режими за съответствие, прилагани от акредитирани сертифициращи органи и регулатори, а не от фирми за тестване.	Споменато само защото клиентите питат къде се вписва тестването. Вписва се като доказателство, предадено на онзи, който води работата ви по съответствие. Ние не сме одитор, оценител или сертифициращ орган; не издаваме сертификат или атестация и нищо в този отчет не бива да бъде представяно като такова.

Какво е този отчет, в едно изречение. Той е записът на офанзивен тест по сигурността на определен набор системи през определен период, написан така, че констатациите да могат да бъдат възпроизведени, поправени и проверени. Той е техническо доказателство. Не е одит, оценка, атестация, сертификация или изявление за съответствие с каквато и да е рамка, а PXLSecurity LTD. не притежава акредитация да издава такова.

ПРИЛОЖЕНИЕ Г

Как да четете този отчет

Различните читатели имат нужда от различни части на този документ. На никого не му трябва целият, а четенето му от начало до край обикновено е най-бавният начин да се действа по него.

- Ако сте ръководител или член на борда

Прочетете раздел 1 и спрете. Той заявява какво беше тествано, колко сериозен е резултатът и трите неща, които да поправите първо, без жаргон. Раздел 4 обяснява думите за критичност.
- Ако отговаряте за програмата по отстраняване

Прочетете раздели 1, 5 и 7. Раздел 7 е планът: вълни, отговорници и последователност. Устоявайте на изкушението да разделите вълна 2 на три задачи — тя е единна проверка на правата с три потвърждения.
- Ако сте инженер, който поправя констатация

Отидете направо на вашата констатация в раздел 6. Стъпките за възпроизвеждане ви позволяват да видите проблема, преди да промените каквото и да е, и да потвърдите поправката, като ги изпълните отново и видите, че се провалят. Препоръката е подредена: първата точка е какво да направите днес, останалото е това, което прави поправката трайна.
- Ако AI функционалността идва от доставчик на платформа

Прочетете раздел 6 за операциите и отговорите и акцентирания текст в края на раздел 7. Съответните констатации могат да бъдат предадени на доставчика както са, с F-09 като вътрешна референтна реализация, към която да се посочи. Съотнасянето на ролите в собствената ви клиентска организация е частта, която е ваша да промените днес.

- Ако сте одитор или водите програмата за управление на AI
Прочетете раздели 2, 3 и 4 за това какво беше тествано и как, Приложение В за това какво са и какво не са препратките, после раздел 5 за резултата. След отстраняването писмото за приключване в раздел 8 е документът, предназначен за вас.
- Ако решавате дали да поръчате тест
Този документ е предаваният продукт. Това, което бихте получили, е тази структура, запълнена с вашите системи: доказателства, път за възпроизвеждане, който разработчиците ви могат да следват, поправка, която пасва на вашата технология, подреден план и повторен тест.

Устройството на една констатация

Всяка констатация в раздел 6 съдържа

- ▢ Идентификатор и заглавие — устойчиви, за да може да бъде проследена до приключване.
- ▢ Критичност и CVSS v3.1 — оценката, с резултата зад нея.
- ▢ Засегнат компонент — какво да отидете и погледнете.
- ▢ Клас на констатацията — препратка към CWE или техника.
- ▢ Описание — какво беше открито и защо има значение.

- ▢ Доказателство — записът, който я установява.
- ▢ Възпроизвеждане — стъпки на нивото на бюлетин, без работещ полезен товар.
- ▢ Въздействие — какво струва на бизнеса, ако бъде използвана.
- ▢ Препоръка — конкретна поправка, в реда, в който бихме я приложили.
- ▢ Статус — отворена при издаването; обновен при повторния тест.

Полезният товар не е констатацията. Това, което е достигнал, е.

ПРИЛОЖЕНИЕ Г · НАКРАТКО

Това е крайният доклад. Сега го поръчайте написан за вашата AI функционалност.

Всичко в този примерен документ — доказателствата, пътищата за възпроизвеждане, поправките, подреденият план, повторният тест — е това, което произвежда един ангажимент на PXLSec. Единственото, което липсва, е вашата среда. Кажете ни какво искате да бъде тествано и старши консултант, а не търговец, ще се върне с въпроси, предложен подход и оферта.

- Тестване за проникване в AI и LLM — първо административният слой, после изолацията, агентите и инжектирането.
- Тестване за проникване — уеб, API, мобилни, облак, вътрешна и външна инфраструктура.
- Оценка на Active Directory и вътрешната мрежа — там, където в действителност започват повечето вътрешни пробиви.
- Red teaming и симулация на противник, отчетени като разказ за това как средата наистина пада.
- Повторните тестове са включени по стандарт, защото поправка, която никога не е била проверена, е план, не резултат.

pxlsec.com/bg/contact · hello@pxlsec.com

PXLSecurity LTD. · бул. „Васил Левски“ 12, 1142 София, България · VAT BG203239331 · независими от 2014 г.

Този документ е обобщен примерен документ. Той беше сглобен от формата на няколко анонимизирани ангажимента на PXLSec и преписан със синтетични, илюстративни данни. Той не описва нито един действителен клиент. Всички имена използват резервирания домейн example.test, TEST-NET адреси в 192.0.2.0/24 и измислените клиентски организации t-larkmoor и t-pentreath; всички акаунти, идентификатори, ключове, записи и оценки по CVSS са измислени за илюстрация; не е назован доставчик, платформа или продукт. В него никъде не присъства име на клиент, име на хост, адрес, данни за достъп, човек или дата на ангажимент. PXLSecurity LTD. е фирма за офанзивно тестване по сигурността и не е одитор, оценител или сертифициращ орган; нищо в този документ не сертифицира и не атестира съответствие с каквато и да е рамка.