ورقة بحثية أمنية

من السياق إلى الاختراق:

تحصين وكلاء الذكاء الاصطناعي

عمر آل سميح

المملكة العربية السعودية

من السياق إلى الاختراق: تحصين وكلاء الذكاء الاصطناعي

الملخص

يُعرف بروتوكول سياق النموذج – Model Context Protocol (MCP) – بوصفه معياراً مفتوحاً يُمكّن نماذج الذكاء الاصطناعي من الاتصال بالأنظمة الخارجية عبر طبقة تكامل موحدة تكشف البيانات وواجهات برمجة التطبيقات وأدوات التنفيذ لوكلاء الذكاء الاصطناعي. وقد أصبح هذا البروتوكول طبقة تكامل محورية بين نماذج اللغة والأنظمة الخارجية؛ كالمرجعيات، وواجهات برمجة التطبيقات، وقواعد البيانات، والمتصفحات، وواجهات الطرفية أو سطر الأوامر، وتطبيقات الأعمال، والخدمات السحابية. ويُحتم هذا التحول فرض رقابة صارمة، لأن أنظمة الذكاء الاصطناعي لم تعد تقتصر على توليد الإجابات، بل باتت تملك القدرة على استرداد سياقات خاصة، واختيار أدوات، وتنفيذ عمليات تُعدّل أنظمة حقيقية. وبينما يعمل البروتوكول على تحسين قابلية التشغيل البيني، فإنه يُوسع السطح الهجومي حول الذكاء الاصطناعي بشكل جذري. الخطر الأساسي لا ينحصر في الإجابات الخاطئة، بل في قدرة النموذج المتصل بالأدوات على تسريب البيانات، أو التورط في تنفيذ إجراءات تدميرية، أو الوثوق بسياقات مسمّمة، أو تجاوز الصلاحيات الممنوحة. تُشرّح هذه الورقة هذا البروتوكول من منظور أمني بحت، وتُقر "سلسلة التعرض من السياق إلى الفعل"، كإطار عمل إلزامي لتتبع انتقال الخطر من السياق الخارجي إلى الفعل الفعلي. وتُقر الورقة حقيقة قاطعة: التبني الآمن لبروتوكول سياق النموذج يفرض التعامل مع النماذج كمفسرات غير موثوقة داخل بنى معمارية مقيدة بصلاحيات صارمة، ويُحظر تماماً اعتبارها كيانات أمنية قادرة على اتخاذ قرارات التفويض.

1. المقدمة

لقد انتهى عصر الذكاء الاصطناعي المعزول في صناديق المحادثة المغلقة؛ لتقتحم هذه الأنظمة اليوم صلب البنية التحتية وتحتل موقع الصدارة كواجهات

تنفيذية تمتلك سلطة الفعل المباشر. في الماضي القريب، كان الخطر الأمني غير مباشر ومحصوراً في قدرة النموذج على توليد إجابة مضللة أو اقتراح شيفرة ضعيفة، دون أن يمتلك أي صلاحية أو قدرة فعلية على المساس بالأنظمة التشغيلية المحيطة به.

لكن هذه الحدود سقطت تماماً؛ فقد مُنح وكلاء الذكاء الاصطناعي اليوم مفاتيح العبور لعمق الأنظمة، وباتوا يمتلكون السلطة الفعلية لفحص المستودعات البرمجية، واستخراج الملفات، وتمرير الأوامر عبر واجهات الطرفية أو سطر الأوامر، واستدعاء واجهات برمجة التطبيقات الداخلية. لم يعد النموذج اللغوي مجرد نظام يقرأ النصوص ليُفسرها البشر، بل تحوّل إلى طبقة استدلال تنفيذية تملك القدرة والسطوة على اتخاذ إجراءات حاسمة وتغيير حالة الأنظمة في أجزاء من الثانية.

وتبرز أهمية بروتوكول سياق النموذج في توحيد طبقات هذا التكامل، مانحاً الوكلاء واجهة موحدة للوصول إلى البيانات واتخاذ الإجراءات. لكن من منظور أمني، تفتح هذه الإمكانية سطحاً هجومياً حرجاً؛ إذ لا يقتصر دور البروتوكول على توسيع نطاق قدرات الذكاء الاصطناعي، بل يوسع مساحة التهديد لتشمل كل نقطة اتصال يمتلكها الوكيل. فالنموذج القادر على قراءة الملفات سيُستغل لتسريبها، والنموذج القادر على الاستهلاك الخارجي للسياق سيتعامل مع الأوامر الخبيثة كتعليمات تشغيلية إلزامية.

لذا، يُعد الاكتفاء بالنماذج الأمنية المصممة لواجهات برمجة التطبيقات التقليدية قصوراً أمنياً خطيراً. وتؤكد هذه الورقة كحقيقة قاطعة أن البروتوكول يخلق سطحاً أمنياً معقداً بين المستخدم، والنموذج، ومصدر البيانات، والفعل. وتتمثل الاستجابة الأمنية الإلزامية في تصميم الأنظمة القائمة عليه كأنظمة تكامل عالية المخاطر مبنية على مبدأ انعدام الثقة الافتراضية (Zero Trust)، حيث يُحظر اعتبار الموجهات سياسات أمنية، ويُمنع اعتبار المواءمة تحكماً في الوصول، ولا تُقبل نوافذ الموافقة كبديل عن البنى المعمارية المحصّنة.

2. الخلفية

يُعد بروتوكول سياق النموذج معياراً لربط تطبيقات الذكاء الاصطناعي بالإمكانات التشغيلية. في التصميم النمطي، يكشف خادم بروتوكول سياق النموذج عن الأدوات والموارد، بينما يربط عميل بروتوكول سياق النموذج تلك الإمكانات بتطبيق الذكاء الاصطناعي لتمكينه من استدعائها وطلبها.

الخطر التشغيلي الصريح يكمن في أن البروتوكول يُحوّل السياق إلى أمر تنفيذي. لم يعد الملف مجرد نص للتلخيص، بل موجهاً يُحدد الأداة التي سيختارها النموذج. ولم تعد مشكلة في مستودع برمجي مجرد طلب مرجعي، بل مدخلاً توجيهياً لوكيل يمتلك صلاحيات التعديل.

تُؤطر هذه الورقة الأزمة عبر "سلسلة التعرض من السياق إلى الفعل"، لتفصيل مسار تحول الخطر إلى كارثة تشغيلية وفق ست مراحل إلزامية التتبع: 1. **دخول السياق الخارجي:** اختراق النظام عبر مستند، بريد إلكتروني، مشكلة برمجية، إدخال في السجلات، أو مخرجات أداة. 2. **تفسير النموذج للسياق:** إخفاق النموذج في التمييز بين الأوامر الموثوقة والبيانات الخبيثة. 3. **اختيار الأداة:** تورط النموذج في اختيار أداة حيوية استناداً لسياق مضلل. 4. **إنفاذ الصلاحيات:** استدعاء الأداة مُتسلحةً بصلاحيات النظام أو المستخدم. 5. **التغيير في العالم الحقيقي:** تعديل، قراءة، أو حفظ في مستودع برمجي بناءً على الاستدعاء. 6. **إعادة البناء الجنائي:** محاولة طبقات المراقبة استيعاب ما حدث.

وتفرض هذه السلسلة نهجاً دفاعياً مركباً؛ فالحوادث هنا لا تعترف بالأخطاء المعزولة، بل تقع عندما تتحد السياقات المعادية، وتتجاوز الصلاحيات الواسعة حدودها في ظل منظومة تدقيق هشة.

3. نموذج التهديد

يُلزم بناء أي نظام يعتمد على البروتوكول بوضع نموذج تهديد يفترض وجود محرك احتمالي عُرضة للتلاعب المباشر داخل مسار التحكم.

الأصول: تشمل الشيفرة المصدرية، مفاتيح واجهات برمجة التطبيقات، الرموز، ملفات الاعتماد، وسجلات التدقيق (Audit Logs). وتعتبر أسرار البيئات التشغيلية والبيانات المالية للعملاء أصولاً يُحظر المساس بها.

الجهات الفاعلة: يُشترط افتراض سوء النية؛ بدءاً من المهاجمين الخارجيين المتحكمين في المستندات والروابط، مروراً بالمطلعين الخبثاء، وصولاً لمطوري خوادم البروتوكول التابعين لجهات خارجية.

حدود الثقة: تُفرض حدود أمنية صارمة بين المحتوى الخارجي والأوامر الموثوقة. وتُعتبر حدود الأوامر/البيانات داخل سياق النموذج هي النقطة الأضعف؛ إذ تعجز النماذج الحالية عن ضمان هذا العزل. وقد صنف المركز الوطني للأمن

السيبراني في المملكة المتحدة حقن الأوامر داخل السياق كتهديد حرج لا ينبغي الاستهانة به.

أسطح الهجوم: تشمل أوامر واجهات الطرفية أو سطر الأوامر، استعلامات قواعد البيانات، ومنافذ واجهات برمجة التطبيقات. والمخاطر تتسع لتشمل مشكلة الوسيط المخوّل المخدوع (Confused Deputy)، واختطاف الجلسات.

4. المخاطر الأمنية

4.1 تجاوز صلاحيات الأدوات

يُعد تجاوز صلاحيات الأدوات خرقاً معمارياً فادحاً يحدث بمجرد منح وكيل الذكاء الاصطناعي وصولاً يتجاوز الحد الأدنى القطعي لمهمته. كأن يُمنح وكيل يحتاج لقراءة ملف معين صلاحية الوصول لكامل مساحة العمل، أو يكتسب صلاحية الكتابة وتنفيذ أوامر واجهات الطرفية أو سطر الأوامر بدلاً من القراءة فقط.

يُحتم التصميم الأمني أن تكون الصلاحيات مقيدة جراحياً؛ فالنموذج ليس جهة أمنية قادرة على تحجيم سلطاتها ذاتياً، بل كيان قد يسيء استخدام الصلاحيات بسبب خطأ استدلالي بسيط أو استجابة لأمر خبيث.

القاعدة الأمنية الصارمة: يُحظر منح النموذج أي صلاحيات بناءً على احتمالية الحاجة المستقبلية، ويُشترط تقييده وفق الضرورة الماسة للحظة التشغيل (مبدأ أقل صلاحية).

4.2 حقن الأوامر عبر المصادر المتصلة

تتحول ثغرة حقن الأوامر داخل السياق إلى تهديد تنفيذي مباشر ومدمر عندما يمتلك النموذج أدوات تشغيلية. فالتعليمات الخبيثة المزروعة داخل تذكرة دعم أو مستند، والتي كانت تُعتبر ساكنة في الأنظمة التقليدية، تصبح قادرة على توجيه الوكيل لاستدعاء واجهات برمجة تطبيقات داخلية لتنفيذ هجومها.

ويُعظم بروتوكول سياق النموذج من هذا التهديد؛ لأن سحب البيانات الخارجية هو جوهر عمله اليومي. يُمنع الاعتماد الحصري على الموجهات النظامية للحماية. الموجه الذي يطلب من النظام "تجاهل الأوامر الخبيثة" هو إجراء شكلي يفتقر للفاعلية. يُلزم النظام بفرض تصنيفات صارمة لمصدر البيانات وتطبيق سياسات حازمة لا تعتمد على وعي النموذج.

4.3 تسميم السياق

يُعد تسميم السياق التلاعب المتعمد بالمعلومات التي يعتمد عليها النموذج لبناء قراراته. وهو تهديد أشد خطراً وأكثر تخفياً من حقن الأوامر المباشر؛ حيث لا يوجه المهاجم أمراً واضحاً، بل يعبث بالحقائق المرجعية؛ كأن يُتلاعب بإدخال في السجلات لجعل الوكيل يشخِّص خطأً وهمياً في مكون سليم ويقوم بإيقافه.

يُصنف هذا التهديد كخطر حرج لأن الوكلاء مصممون للوثوق بالسياق المسترد واعتباره دليلاً مبرراً. وفي البنية المعمارية الآمنة، يُعامل السياق الخارجي كمدخلات غير موثوقة افتراضياً، ويُحظر بناء قرارات التفويض الحساسة بناءً عليها دون توثيق مرجعي مستقل.

4.4 مشكلة الوسيط المخوّل المخدوع

تتجسد هذه الثغرة عندما ينجح المهاجم في دفع وسيط يمتلك الصلاحيات (الوكيل) لتنفيذ مهام تدميرية نيابة عنه. فعندما يثق الوكيل بنص خبيث داخل تذكرة دعم ويستخدم صلاحياته الداخلية العالية لإجراء تعديلات غير مصرح بها نيابة عن المستخدم، فإنه يتحول إلى أداة اختراق طيعة.

وللقضاء على هذا التهديد، تُلزم الأنظمة بإرساء ضوابط متعددة: يُحظر على النموذج اتخاذ قرارات تفويضية نهائية، وتُلزم بوابة الأدوات بفحص الإجراءات عالية المخاطر، ويُحتم على واجهات برمجة التطبيقات إنفاذ قواعد الأعمال بصورة مستقلة ومُعزولة تماماً عن الوكيل.

4.5 تسريب البيانات

يُعد تسريب البيانات اختراقاً كارثياً يحدث عند تمكن النموذج من تمرير محتوى حساس ومحمي من الأنظمة الداخلية إلى بيئات عامة؛ كإرفاق أسرار البيئة في طلب سحب عام، أو استخراج سجلات مالية عبر أداة اتصال خارجية. النمط الأشد فتكاً هو انتقال البيانات العكسي عبر الحدود الأمنية.

تُلزم الأنظمة القائمة على بروتوكول سياق النموذج بفرض رقابة حازمة على حركة البيانات. يُمنع الاكتفاء بتأمين كل أداة بشكل منفصل؛ بل تُلزم بوابات المراقبة باعتراض مسار الوجهات لمنع إرسال بيانات إلى خدمات خارجية مهما برر النموذج ذلك.

4.6 الثقة في مخرجات الأدوات

القاعدة الذهبية: يُشترط التعامل مع مخرجات الأداة كمدخلات مشبوهة ما لم يثبت صدورها من مصدر مرجعي مصدّق. اختراق أداة واحدة لتقديم مخرجات مسممة يكفي لاختطاف حلقة الاستدلال الخاصة بالنموذج بأكملها.

يُحظر، تحت أي ظرف أو مبرر تشغيلي، السماح لمخرجات الأداة بتعديل السياسات الأمنية أو الالتفاف على الضوابط المعتمدة الأساسية.

4.7 سلاسل الهجوم عبر الأدوات

الحوادث الحرجة لن تقتصر على استدعاءات فردية، بل ستُنفذ كسلاسل هجومية مركبة. قد تبدأ بقراءة بريد إلكتروني بريء ظاهرياً، ثم جلب ملف ضار، ليليه استدعاء أداة واجهة الطرفية أو سطر الأوامر لتحويل خطأ استدلالي إلى تنفيذ فعلي يُدمر النظام.

يُلزم التصميم الدفاعي بتقييم مسار العمل برمته ككتلة واحدة مترابطة؛ قراءة ملف أمر طبيعي، لكن قراءته وتسريبه فوراً إلى عنوان شبكي خارجي يُعد خرقاً يستوجب الإيقاف الفوري.

4.8 فجوات قابلية التدقيق والأدلة الجنائية

تفرض الحوادث القائمة على البروتوكول تحديات تحقيق استثنائية. فالسجل التقني الذي يوثق استدعاءً لخدمة معينة لا يوضح ما إذا كان الدافع هو أمر شرعي أو استجابة لتوجيه خبيث مزروع في مستند مقروء.

تتطلب الأدلة الجنائية الصارمة توثيق السلسلة كاملة. يُحتم التصميم الأمني تخصيص سجلات غير قابلة للتغيير ترصد بدقة نية المستخدم والمصادر والأدوات المستخدمة، مع فرض التشفير والتقييد الصارم لمنع تحول هذه السجلات إلى هدف استراتيجي للمهاجمين.

4.9 إرهاق الموافقة البشرية

تُعد مطالبات الموافقة المتكررة للعمليات الروتينية ثغرة هندسة اجتماعية؛ حيث يضطر المستخدمون للنقر الأعمى لتسيير العمل.

يُحظر طلب الموافقة البشرية إلا في الإجراءات عالية المخاطر (كالتعديلات المالية أو تغييرات قواعد البيانات). ويُشترط أن تُصاغ المطالبات كتحذيرات أمنية مفصلة لا تقبل التأويل البصري المضلل.

4.10 مخاطر سلسلة التوريد وخوادم الجهات الخارجية

تُمثل خوادم البروتوكول التابعة لجهات خارجية تهديداً صريحاً لسلسلة التوريد. هي ليست مجرد إضافات أو ملفات إعداد؛ بل هي حزم برمجية ذات امتيازات حية داخل قلب مسار الذكاء الاصطناعي.

يُحظر قطعيّاً تثبيت أي خادم في البيئات التشغيلية الحساسة دون إخضاعها لعزل تشغيلي (Sandboxing) متقدم، وتثبيت جبري للإصدارات، وفرض رقابة شبكية صارمة تمنع الخادم من التجسس الصامت على سياق العمليات.

5. سيناريوهات هجوم واقعية

5.1 مستند خبيث يتسبب في تسريب ملفات داخلية

الوصول الأولي: يفتح الوكيل مستنداً يبدو عادياً ولكنه يُبطن أوامر مخفية تحثه على قراءة البنية المعمارية الحساسة وإدراجها في الرد. **نقطة الضعف:** امتلاك الوكيل لصلاحيات قراءة غير مقيدة عبر كامل النظام التشغيلي. **التدابير الوقائية الإلزامية:** يُحظر منح أداة نظام الملفات وصولاً مفتوحاً. يُشترط تقليص الصلاحيات لتنحصر حصراً في دليل العمل المؤقت، وتُفرض سياسات حجب قطعية تمنع قراءة وتمرير أنماط الملفات الحساسة.

5.2 مشكلة مسممة تؤثر على وكيل البرمجة

الوصول الأولي: مستخدم يطرح مشكلة برمجية تحتوي توجيهاً خبيثاً يُجبر الوكيل على إضعاف بروتوكول المصادقة في شفرة المستودع. **نقطة الضعف:** ثقة الوكيل التامة بالنص المطروح واعتباره دليلاً هندسياً مشروعاً لإجراء عمليات كتابة. **التدابير الوقائية الإلزامية:** يُمنع النموذج آلياً من تنفيذ أي تعديلات على الملفات الأمنية وملفات البيئة. يُفرض التحليل البرمجي الثابت وفحوصات السلامة الإلزامية كشرط مسبق قبل إنشاء أو قبول أي طلب سحب.

5.3 تذكرة دعم مزيفة تتسبب في تنفيذ واجهة برمجة تطبيقات

الوصول الأولي: تذكرة دعم تمت صياغتها لتأمر وكيل الدعم الفني بإجراء تعديلات جذرية في امتيازات أحد الحسابات. **نقطة الضعف:** خضوع واجهات برمجة التطبيقات لأوامر الوكيل دون أي فحص وتثبت استقلالي لشرعية الطلب. **التدابير الوقائية الإلزامية:** يُحظر تماماً اعتبار نصوص التذاكر دليلاً

قانونياً للتفويض. يُلزم بفرض التحقق المزدوج المستقل داخل واجهات برمجة التطبيقات ذاتها لرفض أي طلب لا يستوفي الشروط بمعزل عن رغبة الوكيل.

5.4 خادم جهة خارجية يجمع سياقاً حساساً في صمت

الوصول الأولي: خادم أُضيف لتسريع مهام البحث، لكنه يقوم بجمع واستخراج المقتطفات البرمجية وأسرار التطوير بصمت لأسابيع. **نقطة الضعف:** غياب العزل التشغيلي والافتقار للرقابة على البيانات الصادرة. **التدابير الوقائية الإلزامية:** يُحظر استخدام الخوادم الخارجية دون تفعيل بوابات الشبكة لتقييد خروج البيانات (Egress Filtering)، ويُلزم بتشغيل أي خادم غير داخلي داخل بيئة معزولة بإحكام.

6. الضوابط الأمنية

تُفرض الضوابط الأمنية على كامل مسار "سلسلة التعرض". ويُحظر الاعتماد على طبقة دفاعية واحدة أو افتراض سلامة النوايا.

في مرحلة **السياق الخارجي**، يُشترط تصنيف مصدر المعلومات كغير موثوق افتراضياً لكل ما يرد من خارج حدود المؤسسة. تساهم تصفية حقن الأوامر في تقليل الهجمات الواضحة، لكن لا يمكن الاعتماد عليها كضابط رئيسي وحيد.

في مرحلة **تفسير النموذج**، يُلزم بوضع حدود صارمة وواضحة للسياق، مع تجريد النموذج من أي حق في إبرام قرارات التفويض.

في مرحلة **اختيار الأداة**، تُمثل قوائم سماح الأدوات جدار الحماية الإلزامي الأول. ويُمنع نهائياً استخدام أدوات ذات صلاحيات مفتوحة؛ فأداة واجهة الطرفية أو سطر الأوامر التي تُجيز "تشغيل أي شيء" تمثل خطيئة معمارية لا تُغتفر.

في مرحلة **تنفيذ الصلاحيات**، يُطبق مبدأ أقل صلاحية كدستور أمني غير قابل للتجاوز، مع فرض تخصيص النطاقات بدقة وتطبيق مبدأ القراءة فقط كوضع افتراضي دائم.

في مرحلة **التأثير على العالم الحقيقي**، تُلزم بوابات الأدوات بإعاقة تسريب البيانات، وفرض قيود المعدلات الصارمة لعرقلة محاولات الاستخراج الجماعي والشامل.

في مرحلة **إعادة بناء التدقيق**، تُشترط السجلات المحصنة وغير القابلة للتغيير التي تربط بصمات البيانات بالنية والأداة المستخدمة.

وتشمل قائمة الضوابط الحتمية: الصلاحيات المقيدة جراحياً، العزل التشغيلي المتقدم، عزل الأسرار عن متناول النموذج، إخضاع المخرجات للمطابقة الحتمية، وحرمان الخوادم غير المُدارة داخلياً من أي امتيازات تشغيلية حرجة.

7. مقترح لبنية معمارية آمنة

تتأسس البنية المعمارية الحصينة لبروتوكول سياق النموذج على فرض ضوابط حتمية تُقيد المحرك الاحتمالي وتؤطره. ويُحظر منح النموذج سلطة اتخاذ قرارات الصلاحيات.

طبقة نية المستخدم: تُلزم بتأطير الطلبات بدقة. وتُرفض الطلبات الغامضة آلياً؛ لا مجال لقبول أمر فضفاض.

طبقة استدلال النموذج: تقتصر صلاحيتها على ترشيح الإجراءات وفق سياقاتها المقيدة والموسومة بعلامات التحذير، دون السماح لها بالتنفيذ المباشر قيد أُنملة.

طبقة إنفاذ السياسات: تُمثل خط الدفاع الصلب. تُنفذ تقييماً صارماً للمخاطر وتصدر قرارات الحظر أو السماح استناداً لمعايير مُبرمجة سلفاً لا دخل للنموذج بها.

طبقة بوابة الأدوات: تُمثل الحارس الإلزامي؛ يُمنع أي استدعاء يتجاوز هذه الطبقة، وتتولى إنفاذ قوائم السماح وتقييد النطاقات بقوة حازمة لا تقبل المساومة.

طبقة الوصول إلى البيانات: يُحظر وصول قواعد البيانات المباشر المفتوح (Raw Access) تحت أي ذريعة تشغيلية. وتلتزم واجهات برمجة التطبيقات بمسؤوليتها الذاتية عن فحص التفويض وقواعد الأعمال.

طبقة التسجيل والتدقيق: توثق مسارات الحركة بالكامل في أقبية بيانات آمنة وغير قابلة للمسح.

طبقة الموافقة البشرية: صمام الأمان النهائي الذي لا يُستدعى إلا حين يواجه النظام إيعازاً بتغييرات جذرية أو عمليات نقل بيانات فائقة الحساسية.

هذا المعمار ليس ترفاً، بل ضرورة تشغيلية؛ يُحظر كلياً ربط الأدوات بالنموذج بشكل مباشر تحت ذريعة الموجهات المحسنة.

8. المناقشة

لا تهاون ولا مساومة في المقايضة بين الأمان والمرونة. الصلاحيات غير المرئية والخفية التي تمنح الوكيل وصولاً مفتوحاً لإرضاء تجربة المستخدم هي إخفاق جذري في بنية النظام.

سيبقى بروتوكول سياق النموذج تقنية أساسية للمستقبل، لكن يتعين على المؤسسات التخلي الفوري عن نهج "الوصول الشامل" وتبني العقيدة الصارمة المتمثلة في "الوصول المقيّد والمبرر للعملية الحالية فقط".

تُبنى العقيدة الأمنية للبروتوكول على يقينيات وثوابت غير قابلة للنقاش: السياق مُعادٍ افتراضياً، النموذج مخترق استدلالياً بشكل وارد ومستمر، خوادم الأدوات هي مسارات تسريب محتملة، والسجلات المعصومة من التلاعب هي ضرورة تشغيلية وليست خياراً تصميماً.

9. الخاتمة

يُعيد بروتوكول سياق النموذج رسم السطح الهجومي، ناقلاً الخطر من التوليد النصي العابر إلى التنفيذ الفعلي المؤثر. لم يعد السؤال "هل سيخطئ النموذج في إجابته؟" بل "ما هو حجم الدمار الذي سيخلفه النموذج الممتلك للأدوات إن تعرض للاختراق أو التلاعب الدلالي؟"

الصلاحيات غير الخاضعة للرقابة هي مصدر الخطر الحقيقي، والمسؤولية عن الكوارث التشغيلية تقع بالكامل على عاتق البنية المعمارية وليس على النموذج.

إن تأمين وكلاء الذكاء الاصطناعي لا يتحقق إطلاقاً بتقديم مناشدات للنماذج لتوخي الحذر، بل عبر فرض بنى معمارية حديدية تُجرد النماذج من القوة المطلقة، وتتعامل مع كل مستند وتذكرة ومشكلة كقنبلة موقوتة، وتقطع مسار التنفيذ الخبيث بقوة السياسات المبرمجة قبل أن يتحول إلى استدعاء واقعي يدمر الأنظمة.

المراجع

[1] Anthropic, “Introducing the Model Context Protocol,” Nov. 25, 2024. https://www.anthropic.com/news/model-context-protocol

[2] Model Context Protocol, “Tools,” MCP Specification, Protocol Revision 2025-06-18. https://modelcontextprotocol.io/specification/2025-06-18/server/tools

[10] NIST, “Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations,” NIST AI 100-2e2025, Mar. 2025. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2025.pdf

[11] MITRE, “MITRE ATLAS.” https://atlas.mitre.org/

[12] K. Greshake et al., “Not What You’ve Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection,” arXiv:2302.12173, 2023. https://arxiv.org/abs/2302.12173

[13] N. Maloyan and D. Namiot, “Breaking the Protocol: Security Analysis of the Model Context Protocol Specification and Prompt Injection Vulnerabilities in Tool-Integrated LLM Agents,” arXiv:2601.17549, 2026. https://arxiv.org/abs/2601.17549

[14] Cloud Security Alliance AI Safety Initiative, “MCP by Design: RCE Across the AI Agent Ecosystem,” Apr. 20, 2026. https://labs.cloudsecurityalliance.org/research/csa-research-note-mcp-by-design-rce-ox-security-20260420-csa/

[15] UK National Cyber Security Centre, “Prompt Injection Is Not SQL Injection: It May Be Worse,” Dec. 2025. https://www.ncsc.gov.uk/blog-post/prompt-injection-is-not-sql-injection

[3] Model Context Protocol, “Resources,” MCP Specification, Protocol Revision 2025-06-18. https://modelcontextprotocol.io/specification/2025-06-18/server/resources

[4] Model Context Protocol, “Authorization,” MCP Specification Draft. https://modelcontextprotocol.io/specification/draft/basic/authorization

[5] Model Context Protocol, “Security Best Practices.” https://modelcontextprotocol.io/docs/tutorials/security/security_best_practices

[6] Anthropic, “Claude Code Security.” https://code.claude.com/docs/en/security

[7] OWASP Foundation, “OWASP Top 10 for Large Language Model Applications.” https://owasp.org/www-project-top-10-for-large-language-model-applications/

[8] E. Tabassi, “Artificial Intelligence Risk Management Framework (AI RMF 1.0),” NIST AI 100-1, Jan. 2023. https://doi.org/10.6028/NIST.AI.100-1

[9] NIST, “Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile,” NIST AI 600-1, Jul. 2024. https://doi.org/10.6028/NIST.AI.600-1