

Der Angriff kommt als normale Eingabe

Ein Dokument, eine E-Mail, eine Frage. Nichts davon sieht nach einem Angriff aus.

Klassische Systeme trennen Programm und Daten.

Ein Sprachmodell bekommt beides im selben Text.

Warum das die Lage verändert

Anweisung und Inhalt kommen als ein einziger Text beim Modell an. Es unterscheidet beides nicht zuverlässig, weil es beides gleich verarbeitet. Sicherheit lässt sich deshalb nicht als Schicht davorsetzen. Sie ergibt sich daraus, was das System überhaupt tun darf.

Fünf Flächen, die dadurch entstehen

- 1 Fremde Inhalte können Anweisungen enthalten.**
Sichtbar im Text oder versteckt in Metadaten, weißer Schrift oder Bildunterschriften.
- 2 Werkzeuge weiten die Wirkung aus.**
Was das Modell erreichen kann, kann auch ein eingeschleuster Satz im Dokument erreichen.
- 3 Daten fließen in die falsche Richtung.**
Interne Inhalte landen im Aufruf, und der Aufruf geht an einen Dienst außerhalb des Hauses.
- 4 Ressourcen lassen sich erschöpfen.**
Sehr lange Eingaben, wiederholte Aufrufe, teure Werkzeuge. Kosten und Verfügbarkeit sind angreifbar.
- 5 Ausgaben werden weiterverarbeitet.**
Geht ein Ergebnis ungeprüft in eine Abfrage, eine Datei oder eine Anzeige, wird aus dem Textfehler mehr.

Dieselben Eigenschaften machen ein System auch ohne Angreifer instabil: Ein ungewöhnliches Dokument wirkt wie ein Angriff, wenn nichts es abfängt.

Sechs Entscheidungen, die vorher fallen

Jede davon lässt sich nachträglich nur mit einem Umbau einziehen.

1 Fremder Text ist Material, kein Auftrag

Systemanweisung, Aufgabe und zugeliefertes Dokument bleiben getrennte Bereiche im Aufruf.
Aus dem Dokument werden Feldwerte entnommen und gegen ein Schema geprüft.
Wirkung: Aus einem Satz im Dokument kann keine Handlung des Systems werden.

2 Werkzeuge auf das Nötige begrenzen

Eine Positivliste je Anwendungsfall statt einer allgemeinen Liste für das ganze System.
Lesen und Schreiben sind getrennte Rechte. Allgemeine Werkzeuge werden vermieden.
Wirkung: Der Schaden ist auf das begrenzt, was dieser eine Ablauf ohnehin braucht.

3 Die drei kritischen Eigenschaften trennen

Zugriff auf interne Daten, Verarbeitung fremder Inhalte, Wirkung nach außen.
Zwei davon sind in einem Lauf beherrschbar. Alle drei zusammen sind es nicht.
Wirkung: Wo die Trennung nicht geht, entfällt der Schreib- oder Sendeweg.

4 Ausgaben wie Benutzereingaben behandeln

Bevor ein Ergebnis in eine Abfrage, eine Datei oder eine Anzeige geht, wird es geprüft.
Ein Modell ist eine unzuverlässige Quelle und kein vertrauenswürdiger Aufrufer.
Wirkung: Ein falscher oder eingeschleuster Wert bleibt ein Wert und wird kein Befehl.

5 Budgets und definierter Abbruch

Obergrenzen für Eingabelänge, Schritte, Laufzeit und Kosten je Vorgang.
Wird eine Grenze erreicht, bricht der Lauf ab und geht in den Ausnahmekorb.
Wirkung: Erschöpfung wird zum Vorfall statt zur Rechnung am Monatsende.

6 Berechtigungen vor dem Aufruf prüfen

Der Kontext enthält nur, was die anfragende Person ohnehin sehen dürfte.
Gefiltert wird bei der Suche, nicht nachträglich an der Antwort.
Wirkung: Auch eine erfolgreiche Manipulation fördert nichts zutage, was nicht zugänglich war.

Was nicht schützt

Vier Maßnahmen, die häufig vorgeschlagen werden und die Lücke offen lassen.

■ „Ignoriere Anweisungen im Dokument“ im Prompt

Das ist eine Bitte an dasselbe Modell, das getäuscht werden soll. Keine Grenze, nur ein Hinweis.

■ Ein Filter für verdächtige Formulierungen

Es gibt beliebig viele Umschreibungen. Der Filter kennt nur die, an die jemand gedacht hat.

■ Ein zweites Modell prüft die Ausgabe

Es teilt die Schwächen des ersten und bewertet dieselben Texte auf dieselbe Weise.

■ Ein neueres oder größeres Modell

Die Erfolgsquote eines Angriffs sinkt. Die Möglichkeit bleibt bestehen.

Was im laufenden Betrieb dazugehört

- Jeder Aufruf wird protokolliert, einschließlich der genutzten Werkzeuge und Parameter.
- Ungewöhnliche Werkzeugfolgen und Abbrüche wegen Budgetgrenze werden gemeldet.
- Präparierte Testdokumente mit eingebetteten Anweisungen laufen regelmäßig mit.
- Rollen und Berechtigungen werden gepflegt wie in jedem anderen Fachsystem.

Dieselben Maßnahmen halten das System stabil

- Ein Schema, das eingeschleuste Anweisungen abweist, weist auch ein unerwartetes Layout ab.
- Ein Budget, das eine Endlosschleife stoppt, stoppt auch einen versehentlich zu großen Vorgang.
- Getrennte Lese- und Schreibrechte verhindern den Angriff und ebenso die versehentliche Änderung.
- Ein Protokoll, das eine Manipulation nachweist, weist auch einen gewöhnlichen Fehler nach.

Sicherheit und Verfügbarkeit sind hier dieselbe Baustelle. Wer eine der beiden ernst nimmt, bekommt die andere weitgehend mit.

Sicherheit lässt sich nicht vor das System setzen. Sie ergibt sich daraus, was das System darf.