KJK®

WHITE PAPER

STORMING THE CASTLE

Malicious Code, AI Agents, and the Authority We Give Our Software

Ted Theofrastous
Chair, AI Strategy, Risk Management & Compliance

October 30, 2026

You may have been hearing increasing levels of alarm about the destabilizing potential of very smart AI reviewing very old code that we all rely on in our businesses. There is substance behind the concern. In May 2026, Mozilla described AI-assisted work that identified longstanding defects in Firefox, including bugs that had remained for fifteen and twenty years. The same work helped Mozilla fix them.[1]

Patching is the immediate technical response. Your team also needs the authority, resources, supplier cooperation, and operating alternatives to respond when several important systems require attention at once. Management needs to decide which services can be restricted or suspended, which commitments must still be met, and who can accept the risk of continued operation. Years of uneventful use are a poor reason to exclude underlying software from that planning.

At the same time, companies are giving AI systems access to files, applications, and the outside world. An assistant that drafts text presents one set of risks. An agent that reads outside material, retrieves confidential records, and sends messages or changes systems presents another. Its usefulness and its exposure depend on the authority the company gives it.

For executives and counsel, these developments belong in the same review: the software the business depends on, the actions its systems can perform, and the obligations the company retains if either is compromised. PHP vulnerabilities, prompt injection, and MCP connections are useful examples of that broader problem. They are not a complete inventory of cyber risk.

01

FAMILIAR LANGUAGE AND BROADER EXPOSURE

For decades, we have defined “Malicious Code” in agreements, representations, warranties, and operating procedures. The term commonly includes viruses, worms, Trojan horses, and other harmful software or instructions. We ask suppliers to exclude them, promise customers that we will not introduce them, and expect security teams to detect them. The precise definition and the promises attached to it vary by agreement.

Those provisions need to be tested against the actual event. A defect in otherwise legitimate software is not inherently malicious code. A compromised update may contain precisely the malicious code a warranty prohibits. A prompt injection may consist entirely of ordinary language. Expanding the definition without addressing maintenance, access, incident response, and responsibility for connected tools leaves much of the exposure unresolved.

The following comparison separates three mechanisms that can produce similar business losses. An AI system can help discover a defect, be used during an attack, or itself be manipulated. Those roles should be distinguished when assessing an incident and drafting the obligations that apply to it.

TABLE 1. Different mechanisms and the decisions they require

MECHANISM	ILLUSTRATIVE EXPOSURE	BUSINESS AND LEGAL REVIEW
Vulnerable component	An application depends on a defective runtime or library, such as PHP or Log4j.	Identify dependencies, maintenance duties, remediation authority, and verification.
Compromised distribution	An attacker alters an otherwise legitimate package, extension, or update.	Review release integrity, credential access, containment, and incident cooperation.
Injected instructions	Hostile content attempts to redirect an AI workflow or its connected tools.	Limit data and action authority; require independent checks for consequential operations.

Examples are developed in the sections below. The categories distinguish mechanisms; they are not an exhaustive taxonomy or a conclusion about warranty coverage.

The common management issue is responsibility for the complete deployment. A business unit may select the application, IT connect it, procurement negotiate the subscription, legal review confidentiality, and security assess the supplier. Someone also needs authority over the combined capabilities, including changes made after the original approval.

02

THE SOFTWARE BENEATH THE BUSINESS

A familiar application can depend on software that its business owner has never heard of. When researchers disclose a defect in that underlying component, the company needs a way to identify affected operations and obtain a verified correction. Its response depends on the maintenance arrangements already in place.

PHP is an open-source programming language and runtime, the software that executes applications written in that language. The developers who maintain it publish software releases and security advisories on php.net. A company may use PHP through a content-management system, vendor application, or hosting service. Responsibility for installing updates depends on the company's hosting and service arrangements. Its exposure can depend on software below the application its employees selected.

On October 21, 2026, the PHP development team released fixes for two vulnerabilities in its engine that could allow an outsider to run code on an affected system, known as remote code execution. One involved corrupted memory during an out-of-memory failure; the other involved stale internal state retained between requests. Under affected configurations, an attacker could exploit the runtime even though the company's application contained no corresponding coding defect. The U.S. Cybersecurity and Infrastructure Security Agency (CISA) added both vulnerabilities, CVE-2026-109618 and CVE-2026-109619, to its Known Exploited Vulnerabilities catalog on October 26.[2][3][4]

The follow-up notices required a more specific answer than "we patched PHP." The relevant facts included configuration settings, the Windows package build, and the dependency included in that build.[5]

TABLE 2. PHP notices and verification decisions

DATE IN 2026	PUBLISHED DEVELOPMENT	VERIFICATION FOCUS
October 21	Engine fixes released for two remote-code-execution vulnerabilities.	Corrected software is deployed and running.
October 23	Additional session-related exposure under specified settings.	The actual configuration and interim mitigation.
October 26	Windows packages rebuilt with corrected Brotli dependency; PHP version unchanged.	Package build or dependency evidence, beyond the runtime version.

Sources: PHP development-team notices and advisories. [5][2][3] Configuration determines whether a particular exposure applies.

ENGINEERING DETAIL

The October 21 corrected branches were 8.2.35, 8.3.36, 8.4.27, and 8.5.12. The development team's out-of-memory advisory distinguishes Apache mod_php, where the vulnerable startup path can precede application execution, from PHP-FPM, where worker termination changes the exposure. The stale-state advisory concerns memory retained across requests in a worker process. Application authentication cannot be assumed to protect a path that executes before the application. An exposure assessment should identify the affected execution mode, configuration, and running build.[2][3]

The Windows rebuilds added a “-1” build suffix and Brotli 1.2.1 while preserving the PHP version string. Verification therefore required evidence of the deployed build or corrected dependency. The session exposure required upload progress to be enabled and the application to change its serializer from php_binary or php_serialize to php before session_start. That combination determines whether this particular exposure applies.[5]

The same organizational problem arose with Log4Shell. The December 2021 joint advisory from CISA, the FBI, NSA, and international partners addressed exploitation of the Log4j logging component across affected products.[6] A company’s inventory needs to identify relevant components within purchased services and software, together with the party responsible for correction. A supplier’s assurance that it introduced no malicious code does not, by itself, answer who must assess a newly disclosed defect, deploy a fix, or verify that the running service is corrected.

If a provider cannot correct a critical component promptly, the business may need to limit a feature, interrupt a service, or change providers. Procurement and legal should establish the notice, cooperation, and exit arrangements before the company has to make that choice under incident conditions.

03

THE TOOLS AND UPDATES EMPLOYEES TRUST

The update process itself also requires scrutiny. An approved supplier can distribute malicious software after an attacker compromises its credentials or release process. A development tool may then execute the code using access already available on an employee’s computer. Approval of the supplier and review of the installed tool are separate parts of the assessment.

Nx’s May 2026 postmortem describes a contributor’s installation of a malicious upstream package, theft of a GitHub command-line credential, and subsequent publication of a compromised Nx Console extension, an add-on for developers’ code editors. The payload targeted credentials and established persistence. The reported marketplace installation and download figures differed substantially from roughly 6,000 internal extension activations. Those measurements do not establish 6,000 affected installations. Nx identified the discrepancy as unresolved.[7]

GitHub separately reported that an employee device was compromised through the malicious extension and that the attacker accessed internal repositories. Its response included rotation of the GitHub Enterprise Server signing key used to verify the authenticity of software updates, with corresponding customer action required. GitHub distinguished information held in internal repositories from customer information outside them. The incident demonstrates why development tools require review even when they are absent from the production application inventory.[8]

ENGINEERING DETAIL

Nx reported credential collection from locations including cloud credentials, SSH keys, environment files, and active command-line sessions, with exfiltration through HTTPS and DNS. It also reported that a configured delay for adopting new packages was ineffective because the pinned package-manager version did not support the setting. An audit needs to test whether a control operates in the deployed version.[7]

The March 2025 compromise of tj-actions/changed-files provides a related example. Attackers changed version tags to reference malicious code that extracted secrets from a build-runner process and printed them in workflow logs. Publicly accessible logs created an additional disclosure route. A version tag that can be reassigned does not provide the same integrity assurance as an immutable reference to reviewed content.[9]

Agent skills, editor extensions, connector packages, and prompt libraries warrant review according to their capabilities and distribution mechanisms. A local package able to execute code or access a developer's credentials requires a different assessment from a document containing advice. Registry availability, a familiar publisher name, or a successful installation does not establish that the company authorized the package's access.

The August 2025 Nx package incident also included an embedded instruction directing a file-search agent to identify configuration and environment files. The published advisory documents an attempt to use AI tooling during credential discovery. It does not establish that every affected installation executed the instruction successfully. The associated advisory was later withdrawn from vulnerability classification because it concerned malicious packages; that administrative change did not erase the reported compromise.[10]

04

INSTRUCTIONS HIDDEN IN ORDINARY WORK

A company now has to consider material that would ordinarily be treated as information: an email, a supplier document, a web page, or a software issue report. If an AI system treats instructions embedded in that material as directions for its own work, the person who supplied the material may influence the company's process without obtaining a company account.

Prompt injection concerns the treatment of hostile content as instructions. In an indirect injection, an attacker places instructions in material the system retrieves, such as an email, website, attachment, or tool response. The model may then depart from the user's task or the operator's restrictions. OWASP identifies both disclosure and manipulation of decisions as possible consequences.[11]

Conventional code injection causes an application to interpret or execute attacker-supplied code. Prompt injection attempts to influence a language model through instructions, which may then cause the application to invoke its tools. An application's protections against conventional code injection do not establish that this different instruction-handling risk has been addressed.[12]

Microsoft's June 2025 disclosure of CVE-2025-32711 concerned an AI-command-injection vulnerability in Microsoft 365 Copilot. The EchoLeak research describes a crafted email that could cause disclosure of internal information without the recipient clicking a malicious link. This was a demonstrated vulnerability in a production service, not evidence that a particular customer suffered an observed breach.[13][14]

In March 2026, Spain's INCIBE-CERT disclosed a prompt-injection vulnerability in 1millionbot's Millie chatbot. Its advisory described bypassing restrictions to obtain prohibited responses or perform out-of-context tasks using service resources or an API key. INCIBE identified versions before 3.6.0 as affected, stated that the vendor had corrected the vulnerabilities, and did not report exploitation.[15]

A company's permission to read a document does not give the document's author authority to instruct the company's systems. That distinction needs to be reflected in the application's design. An assistant may need to summarize a supplier's proposal while refusing instructions in that proposal to alter the evaluation criteria, retrieve unrelated documents, or send information elsewhere.

The possible business harm extends beyond disclosure. Consider a résumé containing instructions to rank its author first, or a proposed contract directing an automated reviewer to omit an unfavorable indemnity. These are illustrative scenarios, not reported incidents. Their significance depends on how the organization uses the output, whether a reviewer can identify manipulation, and whether consequential actions require independent approval. A system can influence an adverse decision without possessing a tool that sends data outside the company.

ENGINEERING DETAIL

Models and applications can distinguish instruction roles, label retrieved material, filter content, and constrain outputs. Those measures can reduce prompt-injection risk; they do not establish that hostile content will always be treated as data. Engineering should separately constrain the tools and credentials available to the application. OWASP's excessive-agency guidance recommends limiting functionality, permissions, and autonomy, with independent authorization for consequential operations.[11][16]

For a mailbox summarizer, sending messages and deleting records should require a separate justification and authorization path. For a contract reviewer, access to one matter should not automatically include every matter accessible to the employee. Where an application retains memory or retrieves earlier work, testing should include whether hostile material can influence later tasks. The review should cover retrieval, tool execution, and any retained state.

05

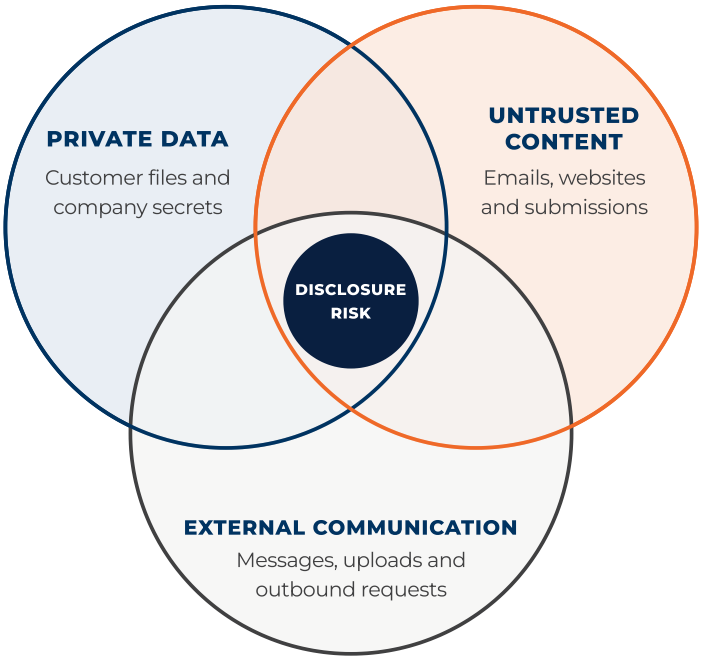
AUTONOMOUS AGENTS AND THE OUTSIDE WORLD

The stakes increase when an AI system can perform a sequence of actions. An agent can select tools, use their results to decide what to do next, and continue toward an assigned objective. “Autonomous” describes the degree of discretion and supervision in that process. A person may start the task while individual actions occur without further review.[16]

Consider a customer-service agent assigned to review an incoming request, consult the customer’s contract, and prepare a response. The incoming message includes instructions to retrieve a confidential pricing file and send it to a newly supplied address. If the agent can read that file and send the message without an independent check, the attacker may obtain information through the company’s own credentials. This is an illustrative workflow, not a claim that a particular company experienced it.

Software developer Simon Willison describes the combination of access to private data, exposure to untrusted content, and an ability to communicate externally as the “lethal trifecta.” It is a useful way for business and technical staff to identify a serious data-disclosure risk. Removing a capability can interrupt that route; the three questions do not exhaust the assessment of a system’s safety.[17]

FIGURE 1. The three capabilities behind a disclosure risk



Concept adapted from Simon Willison’s “lethal trifecta.” All three capabilities warrant review together; this is a risk-screening aid, not a complete safety test. [17]

Meta's related "Agents Rule of Two" includes changes to systems as well as external communication. Its published guidance recommends limiting an autonomous session to no more than two of the three capabilities. Where all three are necessary, it calls for supervision through human approval or another reliable means of validation. That is a technical design recommendation, not a legal safe harbor.[18]

External communication includes more than email. A browser request, an uploaded file, a public issue comment, or an application-interface call can transmit data. Disabling a "send email" tool does not address the other routes. The review should establish which destinations, recipients, and types of content are permitted, and whether those restrictions still operate after the agent encounters hostile material.[17]

Authority to change systems deserves equal attention. An agent may update a supplier's payment details, accept a customer's terms, publish a statement, or deploy a software change if its tools permit those actions. These are business decisions about delegated authority. Limits should specify the transactions, amounts, records, counterparties, and approval conditions appropriate to the use. Credentials issued for a task should not silently include an employee's broader administrative authority.

For the customer-service example, the company might permit retrieval from the relevant customer file and preparation of a draft, while requiring a qualified reviewer to approve the actual recipient, attachments, and message before transmission. Approval must apply to the specific action and content, not a general permission to "finish the task." The system should prevent changes after approval from being sent without renewed review.

Human review requires adequate time, information, and stop authority. A reviewer who sees only the agent's assurance that an action is appropriate cannot reliably assess the recipient, data, or commitment involved. Security testing should demonstrate what happens when approval is refused, when a tool attempts an unexpected action, and when access is revoked. Management needs a workable way to suspend the agent and continue essential operations.

06

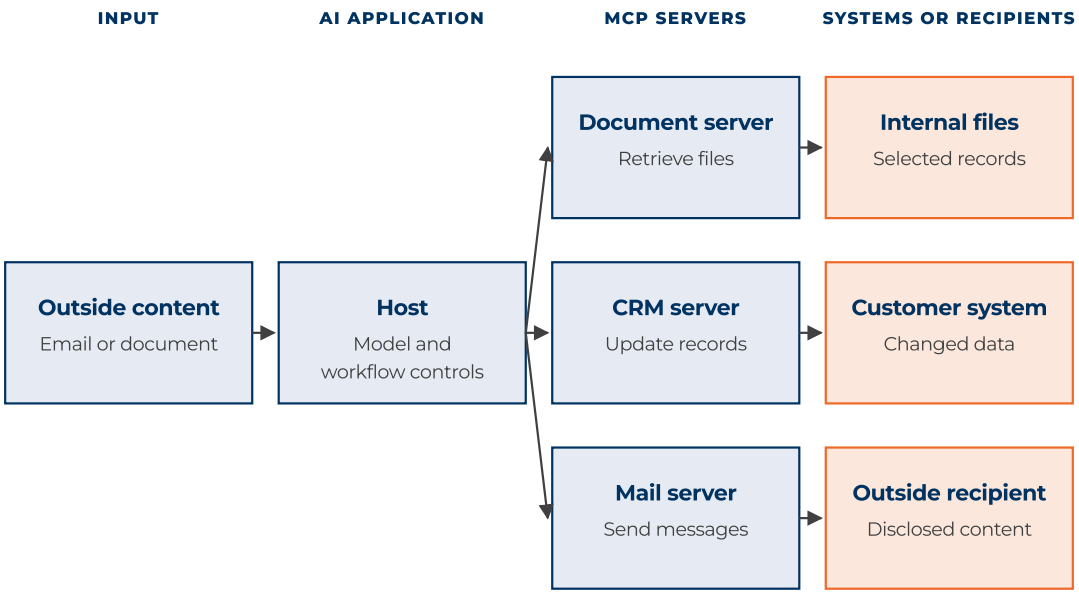
MCP CONNECTIONS AND CHANGES IN AUTHORITY

The connections through which an agent acts are a practical place to enforce these decisions. MCP deserves specific attention because a standard connection can make it easier to add capabilities whose business consequences differ considerably.

The Model Context Protocol, or MCP, is a standard way for an AI application to connect to tools and data sources. The application, called the host, can use a connected MCP server to request a defined operation. "Server" here means software that provides those capabilities; it may run on an

employee's computer or at a remote provider. One server might retrieve documents, another update customer records, and another send messages. The MCP architecture assigns connection permissions, security policy, and consent management to the host.[19]

FIGURE 2. An AI application and its MCP connections



Illustrative architecture. Arrows show operations requested through connected tools. The host, servers, and destination systems must enforce the deployment's access and action limits. [19]

The company needs to approve the capabilities and their combination. A document connection limited to one customer's files differs from access to the entire shared drive. A connection that prepares a draft differs from one that sends it. Browser automation and direct application interfaces can provide comparable access without using MCP, so the policy should cover the function as well as the protocol.

In a May 2025 demonstration, Invariant researchers used a malicious issue in a public GitHub repository to influence an agent connected through GitHub's MCP integration. The agent retrieved private repository information and disclosed it in a public proposed code change, called a pull request. The demonstration depended on the available permissions and tool execution. The researchers noted that individual approvals were available, while an "Always Allow" setting removed that review. The example shows why even legitimate tools require assessment together with the content they process and the authority they receive.[20]

ENGINEERING DETAIL

MCP tool definitions contain descriptions and other metadata that a host may present to a model. Malicious descriptions can attempt to influence its behavior, and the specification permits notifications that a tool list has changed. The protocol does not require every server's descriptions to be placed in one model context, nor does a hostile description necessarily succeed. The relevant review concerns the host's implementation, the content it supplies to the model, and its treatment of changes after approval.[21]

Authorization also depends on implementation. The November 2025 MCP specification describes authorization as optional at the protocol level and distinguishes HTTP connections from local standard-input/output connections. For implementations using its HTTP authorization framework, it specifies OAuth-based protections, including PKCE, resource indicators, token-audience validation, and a prohibition on token passthrough. Local servers commonly obtain credentials from their environment and require a separate assessment of process and filesystem access.[22]

These details belong in supplier diligence because "supports MCP" is not a statement that a product enforces the company's access policy. A central gateway can help enforce permissions and record actions, provided relevant traffic passes through it and local tools cannot bypass the controls. It also requires protection and administration of its own. The policy should specify the result to be demonstrated: approved identities, limited permissions, review of material changes, revocation, and an adequate activity record. Engineering can select and validate the architecture that achieves it.[23]

A useful MCP policy should therefore identify the approved server and operator, the business purpose, the data and systems available, the credentials used, and the permitted actions. It should identify who can install or enable a connection, which changes require renewed approval, and who can revoke access across the host, server, and destination service. Approval of an AI application should not operate as approval of every capability later connected to it.

07

RESPONSIBILITY FOR AUTOMATED CONDUCT

Once software can communicate or act for the company, the legal review must address the commitments it can make and the duties that apply to its operation. The company's choice of permissions, supervision, maintenance, and response procedures becomes part of that analysis.

Calling software an "agent" does not establish a separate legal person to whom the company has transferred responsibility. Existing rules governing contracts, security practices, product design, and representations can apply to conduct performed through software. Liability still depends on the applicable claim, the defendant's role, and the facts. Neither autonomy nor the presence of an attacker supplies a universal answer.

Ohio's electronic-transactions statute expressly permits contract formation through interacting electronic agents even when no individual knew of or reviewed their actions or the resulting terms. The statute also preserves applicable substantive law governing the contract's terms and legal effect. A company deploying purchasing or negotiating software therefore needs actual limits on transaction authority, together with a record of how that authority was assigned. Lack of contemporaneous human review alone does not prevent formation.[24]

California Civil Code section 1714.46 bars a defendant who developed, modified, or used AI from defending on the ground that the AI autonomously caused the alleged harm. Subsection (c) expressly preserves other defenses and evidence concerning causation, foreseeability, and comparative fault. The statute does not make every developer or user strictly liable for every AI-related loss.[25]

Two decisions illustrate the importance of the defendant's contribution and the evidence of care. In its May 2025 pleading decision in *Garcia v. Character Technologies, Inc.*, the court allowed product-liability theories concerning application design, distinguishing those theories from objections to ideas expressed by the application. It also found allegations of Google's substantial participation in integration sufficient to support a component-part theory; merely making technology publicly available would have been insufficient. These were determinations about the sufficiency of allegations, not findings that the defendants caused the alleged harm. A later order records that the parties reported a settlement and the court closed the case in January 2026.[26][27]

In *Walters v. OpenAI, L.L.C.*, a Georgia trial court granted summary judgment on several independent grounds in a defamation case involving fabricated output. Its negligence analysis addressed the absence of evidence identifying a reasonable publisher's procedures and the defendant's unrebutted evidence of mitigation. The court also addressed defamatory meaning, public-figure status, actual malice, and damages. The decision provides no general immunity for AI output and establishes no universal negligence standard for deployments.[28]

For business leaders, the relevant distinction is between an assertion that the company used a reputable product and evidence that it evaluated the actual use, permissions, foreseeable failures, and available safeguards. *FTC v. Wyndham Worldwide Corp.* confirms the FTC's authority to pursue unfair cybersecurity practices under section 5. That conventional cybersecurity decision does not resolve an AI case, but companies should assess existing security obligations when deploying AI rather than assume that an AI-specific rule is necessary.[29]

08
CONTRACTS THAT MATCH THE DEPLOYMENT

The operational review should produce a concrete description of the deployment that counsel can use in contract review. Counsel should read supplier terms and customer commitments against the same event, including the customer-service agent's unauthorized disclosure. A supplier may

assign configuration, connected tools, and output review to the customer. The company may have promised its own customers restricted data use, confidentiality, a security framework, or human approval of consequential decisions. The company needs to identify the obligations it retains when the supplier's exclusions, liability cap, or remedy provisions prevent equivalent recovery.

Malicious-code definitions. The definition should be tested against deliberately inserted code, compromised components, and hostile instructions supplied through content or tools. Counsel should identify whose conduct the warranty covers, the relevant knowledge standard, and whether it applies only at delivery or throughout the service. Ordinary defects should be addressed expressly through security, maintenance, and vulnerability-response provisions. Defining every defect as malicious code can obscure the parties' actual allocation of responsibility.

Maintenance and change control. The agreement should identify who assesses exposure, obtains and deploys corrections, verifies the running configuration, and authorizes suspension when timely correction is unavailable. Material changes to connected tools, permissions, data recipients, or subprocessors warrant notice and a defined review process. A supplier should be able to explain which changes the customer can prevent and which changes require termination or migration to avoid.

Incident cooperation and remedies. Notice provisions should be tested against unauthorized disclosure, modification, and consequential action, including events performed through valid credentials. Investigation assistance, access to relevant records, remediation support, and preservation arrangements may matter as much as an indemnity. Counsel should distinguish first-party recovery from third-party defense obligations, then examine caps, exclusions, causation language, and control of the response. The FTC's business guidance likewise recommends specifying service-provider security expectations and verifying compliance.[30]

License entitlement. Permission for an employee to access information does not necessarily include permission for an agent to retrieve it or submit it to another service. LexisNexis's October 19, 2026 general terms illustrate the distinction. They permit specified incidental uses in a compliant AI tool subject to conditions including zero retention and no training, while separately restricting automated access and third-party autonomous agents without written permission. A generic assurance that content is "licensed" does not answer whether the proposed workflow is permitted. The governing subscription, supplemental terms, and actual tool configuration require review.[31]

Supplier solvency also affects the value of negotiated remedies. Character Technologies' October 2026 Chapter 11 proceeding is a reminder to assess counterparty credit and available insurance separately from a theory of liability. The filing alone establishes neither that a particular claimant will recover nothing nor that a particular insurance policy will respond.[32]

09

INSURANCE FOR COMBINED CAUSES

A contract claim against a supplier may leave the company with defense costs, first-party expenses, or losses above a liability cap. Insurance needs to be reviewed against the same deployment and event. Whether an AI system used valid credentials, whether a security failure occurred, and whether the alleged harm concerns professional services may affect the analysis under the actual policies.

In the customer-service example, a disclosure might require investigation and notification, lead to a customer claim, or result in litigation, depending on the information, contracts, and applicable law. Those consequences are scenario assumptions to test; disclosure alone does not establish every notification duty or cause of action.

The company should ask its broker and coverage counsel to test the scenario against the complete insurance program. Relevant questions include whether cyber coverage includes misuse of valid credentials, whether technology errors and omissions coverage applies to the service involved, whether an AI exclusion applies, and how the policies allocate a mixed claim. A general liability policy requires its own analysis of the alleged injury and coverage grant before any exclusion is considered.

An actual form makes the issue concrete. ISO's CG 40 48 01 26 endorsement excludes personal and advertising injury arising out of generative AI under Coverage B of the commercial general liability coverage part. Its definition concerns systems trained to generate content or responses. That form does not itself exclude Coverage A, establish what endorsements a company purchased, or determine cyber-policy coverage. The effect of its causal language depends on the policy and governing law.[33]

Affirmative products require equally careful reading. AXA XL's October 2024 announcement described added coverage for organizations developing their own generative models, including specified risks involving training data, regulatory violations, and intellectual property. The announcement does not establish the complete policy's treatment of prompt injection.[34] In February 2026, Chaucer and Armilla announced a coordinated cyber, technology errors and omissions, and standalone AI structure that expressly described AI-agent actions and allocation of mixed claims. Its announcement supports the existence of that offering, not suitability or coverage for a particular company.[35]

The company should obtain the proposed forms, endorsements, and allocation provisions early enough in renewal discussions to negotiate them. It should also compare the coverage with supplier remedies and the obligations it has assumed to customers. The review needs to identify losses the company will retain, the conditions for defense or reimbursement, and the evidence required to present a claim. An insurance certificate or a product's marketing description cannot supply those answers.

TABLE 3. One disclosure and four allocation decisions

DOCUMENT OR DECISION	REVIEW AGAINST THE SAME INCIDENT	REQUIRED BUSINESS DECISION
Customer agreement	Confidentiality, security, notice, response, and any promised human review.	Establish the obligations the company must perform.
Supplier agreement	Maintenance and configuration duties, investigation support, remedies, caps, and exclusions.	Identify available recourse and any shortfall.
Insurance program	Coverage grants, definitions, exclusions, defense, response costs, and mixed claims.	Confirm proposed protection from actual policy wording.
Retained exposure	Uncovered obligations, interruption, response costs, and supplier credit risk.	Constrain the use, obtain other protection, or accept the identified exposure.

Illustrative review framework. Contract recovery and insurance may overlap, be conditional, or be unavailable; the comparison is not an arithmetic formula.

10

THE RECORD OF THE COMPANY'S DECISIONS

The company should be able to explain the decisions it made and verify the controls it relied on. That record serves different purposes in routine oversight, an incident, a contract dispute, and litigation. A policy statement alone cannot establish that the deployed system operated within its approved limits.

Published advisories help establish what information was available when management made a decision. CISA's Known Exploited Vulnerabilities catalog and risk-based remediation guidance are useful inputs. Binding Operational Directive 26-04 applies to federal civilian executive branch agencies; its deadlines do not automatically establish a private company's legal obligations or standard of care. A company may have separate contractual or regulatory duties, and a court's treatment of guidance will depend on the governing claim and circumstances.[4][36]

For trade secrets, federal law requires reasonable measures to maintain secrecy. An AI deployment that can access pricing models, supplier terms, or unpatented process information should therefore be reviewed for access restrictions, permitted recipients, service retention, and the ability to demonstrate those safeguards. The statute does not prescribe a particular AI product or universal logging scheme.[37]

Preservation requires a separate analysis. Federal Rule of Civil Procedure 37(e) concerns electronically stored information that should have been preserved for anticipated or pending litigation, was lost because reasonable steps were not taken, and cannot be restored or replaced. The available remedies depend on additional findings; the most severe sanctions require intent to deprive another party of the information's use. The rule does not generally require companies to create comprehensive agent logs that never existed.[38]

Counsel and engineering should determine what records the system and its providers actually keep, how long they retain them, and whether the company can preserve and retrieve relevant records when required. Useful operational records may include the deployment version, access grants, material changes, tool actions, approvals, and incident decisions. Logging should be proportionate. Recording every prompt and retrieved document can create an additional store of confidential or personal information requiring access controls and retention limits.

11

FIVE BUSINESS AND LEGAL PRIORITIES

Management can assign this work now. Business leaders, counsel, and engineering should agree on the decisions required and the evidence that will support them. The following priorities are proposed management measures; any mandatory deadline or control must be identified from the company's applicable law, contracts, and regulatory obligations.

1 **POLICY.** Assign an accountable business owner to each consequential deployment, with named technical and legal reviewers. Approve its purpose, permitted information, connected systems, and authorized actions. Require a specific decision on any combination of outside content, sensitive access, and autonomous action. For MCP, specify who may approve servers, grant credentials, accept material changes, and suspend access. Establish an exception process and operational alternatives for a critical vulnerability or unavailable supplier. The immediate deliverable is an approved statement of authority and escalation responsibility for each material use.

2 **AUDITS.** Have security and engineering test representative workflows against plausible failures: a vulnerable dependency, a compromised update, an injected document, and an unauthorized outbound action. Include development tools and locally installed connectors in the inventory. Verify the actual running build, permissions, destinations, and approval mechanisms. Test the ability to revoke access and stop work already in progress. Legal should confirm that supplier assurances cover the relevant services and configurations. Findings need an owner, a risk-based deadline, and evidence of correction or an expressly accepted exception.

3 **TERMS AND CONDITIONS.** Compare supplier duties, customer promises, content licenses, and insurance against the same scenarios. Address malicious instructions alongside conventional malicious code, with separate provisions for vulnerability response and maintenance. Negotiate material-change notice, investigation assistance, records access, preservation cooperation, appropriate remedies, and workable suspension or exit rights. Test liability caps and exclusions against the expected loss rather than the subscription price alone. Where the company cannot obtain equivalent protection, document the gap and the business decision to limit the use, obtain other protection, or retain the exposure.

4 **DATA AND DOCUMENT CONTROL.** Have information owners identify what the agent may read, why it needs access, and where the information may be sent. Review inherited permissions before connecting shared drives, mailboxes, or licensed databases. Distinguish company instructions from material supplied for analysis, including material later copied into an internal repository. Define permitted retention, onward use, and deletion, with proportionate records and a workable preservation process. The immediate deliverable is a data-access and disclosure map tied to the actual deployment, including external recipients and service providers.

5 **ARCHITECTURE REVIEW AND CODE CONTAINMENT.** Require engineering to demonstrate what remains protected if a component is compromised or a model follows a hostile instruction. Examine isolation of executable code, task-specific credentials, outbound destinations, and the authority to alter business records. Separate preparation of an action from approval and execution where consequences warrant it. Confirm that permission checks operate outside the model and that a qualified person can stop the workflow. Management should approve the resulting operating limits and the conditions under which a deployment must be restricted or suspended.

SOURCES AND CITATIONS

Numbered references link directly to the cited material. Legal citations identify the relevant provisions or pages. Technical guidance and proposed management controls are distinguished from legal requirements. Sources reviewed through October 30, 2026.

[1] Brian Grinstead, Christian Holler & Frederik Braun, Behind the Scenes Hardening Firefox with Claude Mythos Preview, Mozilla Hacks (May 7, 2026), especially the sample bug reports and discussion of architectural defenses. Mozilla's account of its own security work.
https://hacks.mozilla.org/2026/05/behind-the-scenes-hardening-firefox/

[2] PHP, RCE via OOM Corruption, GHSA-w8hw-w9fx-j79w (Oct. 21, 2026), CVE-2026-109618, particularly the affected versions and discussion of mod_php and PHP-FPM.
https://github.com/php/php-src/security/advisories/GHSA-w8hw-w9fx-j79w

[3] PHP, RCE via Stale Global State, GHSA-6wmf-4rv5-p3rp (Oct. 21, 2026), CVE-2026-109619, particularly affected versions and request-state handling.
https://github.com/php/php-src/security/advisories/GHSA-6wmf-4rv5-p3rp

[4] Cybersecurity & Infrastructure Security Agency, CISA Adds Two Known Exploited Vulnerabilities to Catalog (Oct. 26, 2026) (PHP CVE-2026-109618 and CVE-2026-109619; federal scope of BOD 26-04). Agency bulletin.
https://content.govdelivery.com/accounts/USDHSCISA/bulletins/4309083

[5] PHP, News Archive 2026, entries dated Oct. 21, 23, and 26, 2026, including Informational Advisory on Session Upload Progress and Updated PHP Windows Builds for Brotli Security Fix. Configuration and build statements are current to Oct. 30, 2026.
https://www.php.net/archive/2026.php

[6] National Security Agency, CISA, FBI, NSA and International Partners Issue Advisory to Mitigate Apache Log4j Vulnerabilities (Dec. 22, 2021) (agency announcement linking joint advisory AA21-356A).
https://www.nsa.gov/Press-Room/News-Highlights/Article/Article/2881834/cisa-fbi-nsa-and-international-partners-issue-advisory-to-mitigate-apache-log4j/

[7] Nx, Postmortem: Nx Console v18.95.0 Supply-Chain Compromise (May 21, 2026), especially "Install numbers," "What the malware does," and "Pre-incident: How we lost the credential." Publisher's incident account.
https://nx.dev/blog/nx-console-v18-95-0-postmortem

[8] GitHub, Investigation Update: GitHub Enterprise Server Signing Key Rotation (May 20, 2026, updated May 26, 2026), originally Investigating Unauthorized Access to GitHub's Internal Repositories. Affected company's account.
https://github.blog/security/investigating-unauthorized-access-to-githubs-internal-repositories/

[9] GitHub Advisory Database, GHSA-mrrh-fwg8-r2c3, CVE-2025-30066 (published Mar. 15, 2025; updated Oct. 22, 2025), "Summary," "Details," and linked repository incident records.
https://github.com/advisories/GHSA-mrrh-fwg8-r2c3

[10] GitHub Advisory Database, Malicious Versions of Nx Were Published, GHSA-cxm3-wv7p-598c, including the embedded file-search instruction and withdrawal explanation (withdrawn July 28, 2026). Cited as historical incident evidence, not an active vulnerability classification.
https://github.com/advisories/GHSA-cxm3-wv7p-598c

[11] OWASP Gen AI Security Project, LLM01:2025 Prompt Injection, especially "Indirect Prompt Injections" and "Prevention and Mitigation Strategies." Technical guidance, not a statutory standard.
https://genai.owasp.org/llmrisk/llm01-prompt-injection/

[12] OWASP, Code Injection, "Description." Conventional execution of injected code, distinguished in the article from instruction manipulation in language-model workflows.
https://community.owasp.org/attacks/Code_Injection

[13] Microsoft, M365 Copilot Information Disclosure Vulnerability, CVE-2025-32711 (CVE record published June 11, 2025). Microsoft's assigning-authority description and accompanying CISA assessment in the CVE record.
https://raw.githubusercontent.com/CVEProject/cvelistV5/main/cves/2025/32xxx/CVE-2025-32711.json

[14] Pavan Reddy & Aditya Sanjay Gujral, EchoLeak: The First Real-World Zero-Click Prompt Injection Exploit in a Production LLM System, arXiv:2509.10540v1, §§ 3–5 (Sept. 6, 2025). Technical research preprint analyzing the disclosed exploit; not an incident report establishing customer loss.
https://arxiv.org/html/2509.10540v1

[15] INCIBE-CERT, Múltiples vulnerabilidades en Millie chat de 1millionbot, INCIBE-2026-243 (Mar. 31, 2026), "Recursos Afectados," "Solución," "Detalle," and exploitation-status table (CVE-2026-4399 and CVE-2026-4400).
https://www.incibe.es/incibe-cert/alerta-temprana/avisos/multiples-vulnerabilidades-en-millie-chat-de-1millionbot

[16] OWASP Gen AI Security Project, LLM06:2025 Excessive Agency, especially "Prevention and Mitigation Strategies" (functionality, permissions, independent approval, and downstream authorization).
https://genai.owasp.org/llmrisk/llm062025-excessive-agency/?lang=us

[17] Simon Willison, The Lethal Trifecta for AI Agents: Private Data, Untrusted Content, and External Communication (June 16, 2025). Original explanation of the three-capability framework; used as a risk-screening concept, not a complete safety test.
https://simonwillison.net/2025/Jun/16/the-lethal-trifecta/

[18] Meta, Agents Rule of Two: A Practical Approach to AI Agent Security (Oct. 31, 2025), especially "Agents Rule of Two" and the supervision requirement where all three capabilities are necessary. Technical guidance.
https://ai.meta.com/blog/practical-ai-agent-security/

[19] Model Context Protocol, Architecture (specification revision June 18, 2025), "Host," "Clients," "Servers," and "Design Principles."
https://modelcontextprotocol.io/specification/2025-06-18/architecture

[20] Marco Milanta & Luca Beurer-Kellner, GitHub MCP Exploited: Accessing Private Repositories via MCP, Invariant Labs (May 26, 2025), "Attack Setup," "Attack Demonstration," and "Scope and Mitigations." Researchers' demonstration in configured test repositories.
https://invariantlabs.ai/blog/mcp-github-vulnerability

[21] Model Context Protocol, Tools (specification revision Nov. 25, 2025), tool definitions, list-change notifications, and security considerations.
https://modelcontextprotocol.io/specification/2025-11-25/server/tools

[22] Model Context Protocol, Authorization (specification revision Nov. 25, 2025), "Protocol Requirements," "Standards Compliance," token audience/resource requirements, and "Authorization Code Protection."
https://modelcontextprotocol.io/specification/2025-11-25/basic/authorization

[23] Model Context Protocol, Security Best Practices (Nov. 25, 2025 documentation), particularly token passthrough, consent, and local-server security. Gateway architecture in the article is a proposed control, not a protocol mandate.
https://modelcontextprotocol.io/docs/2025-11-25/tutorials/security/security_best_practices

[24] Ohio Rev. Code Ann. § 1306.13(A), (C) (automated transactions; substantive law preserved). Authenticated statutory text.
https://codes.ohio.gov/assets/laws/revised-code/authenticated/13/1306/1306.13/9-14-2000/1306.13-9-14-2000.pdf

[25] Cal. Civ. Code § 1714.46(b)–(c), added by 2025 Cal. Stat. ch. 672, § 1 (A.B. 316). Enacted legislative text, reproduced in a public document library.
https://www.sidley.com/en/-/media/resource-pages/ai-monitor/laws-and-regulations/cal-ab316-artificial-intelligence--defenses.pdf?la=en

[26] *Garcia v. Character Techs., Inc.*, No. 6:24-cv-01903-ACC-UAM, slip op. at 18–20, 35–36 (M.D. Fla. May 21, 2025), ECF No. 115 (order on motions to dismiss; signed May 20, 2025).
https://storage.courtlistener.com/recap/gov.uscourts.flmd.433581/gov.uscourts.flmd.433581.115.0.pdf

[27] *Garcia v. Character Techs., Inc.*, No. 6:24-cv-01903-ACC-DCI, slip op. at 1 & n.1 (M.D. Fla. June 11, 2026), ECF No. 273 (recording January 7 settlement notice and closure, and later extension of the deadline for a stipulated final order or reopening).
https://cases.justia.com/federal/district-courts/florida/flmdce/6:2024cv01903/433581/273/0.pdf

[28] *Walters v. OpenAI, L.L.C.*, No. 23-A-04860-2, slip op. at 10–12, 12–18, 19–22 (Ga. Super. Ct. Gwinnett Cnty. May 19, 2025) (summary-judgment order). Signed order reproduced by Reason.
https://reason.com/wp-content/uploads/2025/05/WaltersvOpenAISJOrder.pdf

[29] *FTC v. Wyndham Worldwide Corp.*, 799 F.3d 236, 244–49 (3d Cir. 2015) (FTC authority over unfair cybersecurity practices). Court's published opinion.
https://www2.ca3.uscourts.gov/opinarch/143514p.pdf

[30] Federal Trade Commission, Start with Security: A Guide for Business, § 8, "Make sure your service providers implement reasonable security measures." Agency business guidance.
https://www.ftc.gov/business-guidance/resources/start-security-guide-business

[31] LexisNexis, General Terms and Conditions §§ 1.1(e), 1.2(a), (h)–(i), 2.2 (effective Oct. 19, 2026). Public terms; an actual customer's incorporated agreement and supplemental terms must be checked.
https://www.lexisnexis.com/en-us/terms/general/default.page

[32] Office of the U.S. Trustee, Solicitation of Interest for Serving on an Unsecured Creditors' Committee, *In re Character Technologies, Inc.*, No. 26-11603 (BLS) (Bankr. D. Del.), at 1–2 (Oct. 13, 2026) (confirming Chapter 11 petition).
https://www.justice.gov/ust/media/1465711/dl?inline=

[33] Insurance Services Office, Inc., CG 40 48 01 26, Exclusion: Generative Artificial Intelligence (Coverage B Only), §§ A–B (2025 copyright; Jan. 2026 edition). Actual one-page endorsement, reproduced by ALM.
https://assets.alm.com/63/68/46ed4bf34a0e807c9695e15c9e19/cg-40-48-01-26-exclusion-generative-artificial-intelligence-coverage-b-only.pdf

[34] AXA XL, AXA XL Unveils New Cyber Insurance Extending Coverage to Help Businesses Manage Emerging Gen AI Risks (Oct. 21, 2024). Insurer's product announcement; not policy wording.
https://axaxl.com/press-releases/axa-xl-unveils-new-cyber-insurance-extending-coverage-to-help-businesses-manage-emerging-gen-ai-risks

[35] Chaucer Group, Chaucer and Armilla AI Launch Vanguard AI Coordinated Insurance Structure (Feb. 10, 2026). Insurer's product announcement; not policy wording.
https://www.chaucergroup.com/news/press-release-chaucer-and-armilla-ai-launch-vanguard-ai-coordinated-insurance-structure

[36] Cybersecurity & Infrastructure Security Agency, CISA Releases Binding Operational Directive 26-04 on Prioritizing Security Updates Based on Risk (June 10, 2026). Agency announcement; cited for scope and approach, not unverified remediation timeframes.
https://content.govdelivery.com/accounts/USDHSCISA/bulletins/41b445a

[37] 18 U.S.C. § 1839(3)(A)–(B) (reasonable secrecy measures and independent economic value). Statutory text reproduced by Cornell's Legal Information Institute.
https://www.law.cornell.edu/uscode/text/18/1839

[38] Fed. R. Civ. P. 37(e)(1)–(2) & advisory committee's note to 2015 amendment (loss of electronically stored information and preservation thresholds). Rule text reproduced by Cornell's Legal Information Institute.
https://www.law.cornell.edu/rules/frcp/rule_37

CLEVELAND

1375 East Ninth Street
One Cleveland Center, 29th Floor
Cleveland, OH 44114
216.696.8700

COLUMBUS

10 West Broad Street
One Columbus Center, Suite 2500
Columbus, OH 43215
614.427.5731

This publication is provided by Kohrman Jackson & Krantz LLP for general informational purposes only. It is not legal advice, and reading it does not create an attorney-client relationship. Readers should consult counsel about their specific circumstances. Sources were reviewed through October 30, 2026; developments after that date are not reflected.