Managing the cyber risk of agentic AI

Use safeguards, sandboxing and active oversight to realise the benefits of autonomous systems while limiting the unintended activity.

Agentic AI systems have shown potential to transform how organisations work, at times delivering unparalleled productivity gains. They can automate complex workflows, reduce routine effort and enable people to focus on higher-value tasks.

As organisations deploy increasingly autonomous uses of AI at pace, it is important to consider how these systems behave when they do not function as envisaged or expected - and plan accordingly.

Recently, there have been several incidents involving AI models and agentic AI systems carrying out unsanctioned or unintended activity. These events highlight why organisations need to carefully consider how these technologies are deployed, constrained, observed and responded to.

- In this blog
  - Advice and effective practice will continue to evolve
  - Who this blog is for
  - Assess how much autonomy is needed
  - Understand your model's built-in safeguards and protections
  - Plan for additional safeguards
- Key considerations
  - Identify what could go wrong
  - Prompt carefully
  - Set the right level of oversight
  - Control the AI agent's environment with a robust sandbox
  - Observability: log, audit and monitor agentic AI activity as part of security operations
  - Make your AI activity easy to attribute
  - Emergency shutdown: maintain the ability to 'pull the plug'
- Further reading

Advice and effective practice will continue to evolve

AI cyber security continues to be a rapidly evolving field, and effective practice will continue to evolve as the technology matures and evidence bases are built. We will work with the wider cyber security community to continue to research, develop and evaluate approaches that help organisations deploy increasingly autonomous systems securely and confidently.

The NCSC has been researching and experimenting in this area for some time and is working with partners to develop formal guidance which will build upon, and ultimately supersede, this blog.

Until formal guidance is published, the NCSC is sharing interim practical advice, based on NCSC research to date, to help organisations make informed decisions about how to deploy agentic AI systems securely within their environments.

Who this blog is for

This blog is aimed at system designers and operators who are:

- building environments where AI agents are intended to operate with significant degrees of autonomy
- concerned about the implications of an AI agent performing unintended actions because of the instructions it receives, the tools available to it, or the systems it can access

Assess how much autonomy is needed

The recommendations in this blog should be applied proportionately based on the level of autonomy each agentic AI system is intended to have.

Not every deployment requires the same degree of autonomy. Some agents may simply assist a human user by making suggestions or carrying out tightly constrained tasks in a low-risk manner. Others may be trusted to take actions, access production systems and make decisions with little or no human intervention in potentially high-risk scenarios.

The greater an agent's autonomy, the greater the potential impact if it:

- malfunctions unexpectedly
- accesses information it should not
- takes actions outside its intended scope

And therefore the greater the need for controls.

Being clear on the level of autonomy, and the level of risk you are willing to tolerate, will inform the design of controls that are proportionate to both the organisational benefits you aim to achieve and the consequences of a failure.

Understand your model's built-in safeguards and protections

Once you have identified how much autonomy an agent will have and the level of risk you are prepared to accept, it is important to understand the safeguards that already exist within the AI model, inference service and harnesses you are using.

Many AI models include mechanisms designed to detect or prevent certain types of unwanted behaviour. These controls can provide a baseline level of protection. However, they should not be treated as holistic. Safety controls built into models may:

- be bypassed
- not provide adequate protection in higher-risk environments
- not be sufficient to manage risk appropriately on their own

Understanding the safeguards that are available, their efficacy and their limitations, will enable you to determine what additional controls are needed.

Plan for additional safeguards

For applications where the consequences of failure are above tolerance, organisations should implement additional safeguards rather than relying solely on model-level or harness-level protections.

Examples of further safeguards can include classifiers, deterministic provers and wider controls.

Irrespective of the inbuilt controls, all deployments should be subject to robust observability, operational monitoring and response procedures.

Key considerations

The following considerations will help you decide:

- whether an autonomous AI agent is appropriate given your risk tolerance
- what safeguards may be needed to manage the associated risks

1

Identify what could go wrong

Document what is within and outside the scope of your intended activity, highlighting any ‘red lines’ that the agentic AI system must stay within.

Before deployment, carry out threat modelling to identify the failure scenarios during the agent’s activities. Consider:

- the prompt given to the agent
- the networks and services the agent may be able to access and communicate with through the configured sandbox – either directly or indirectly

Remember that an AI agent is not human. It does not have common sense or human traits, and may interpret instructions and goals in literal or unexpected ways.

Use the results from your threat modelling to assess whether you accept the risks identified and whether any additional controls should be placed in and around the agentic AI system.

2

Prompt carefully

The instructions and context given to an agentic AI system and its underlying model will influence the activities it carries out.

AI agents can pursue goals in ways that are unintended or undesirable if those goals, or how you expect the agent to achieve them, are not clearly defined. The way a task is described can have a significant impact on the agent's behaviour.

Before running an agent, think carefully about:

- what it is to achieve
- what actions it should be allowed to take
- when it should stop and seek human approval – and ensure those approvals will be both guaranteed and gated

Where possible, make your expectations explicit. Instruct the agent in both what it should do and what it should not do.

For longer-running tasks, review how the agent handles context over time. Some agents may compress their context windows to use tokens more efficiently. Where appropriate, repeat critical constraints so they remain throughout the task.

However, do not rely on prompting alone. You should combine prompts with technical and operational controls to provide defence in depth.

3

Set the right level of oversight

Consider how much autonomy you want the agent to have. There are several approaches:

- Human-in-the-loop: humans approve actions before they happen.
- Human-on-the-loop: humans monitor actions and can intervene if needed.
- Human-out-of-the-loop: AI acts autonomously without human review.

The model you choose should inform your risk tolerance and the potential consequences if the agent behaves unexpectedly.

Where unintended activity would have significant consequences, we recommend:

- making named individuals or group responsible for agentic AI activities
- maintaining human oversight of the AI agent's actions
- instructing agents to identify blockers or behaviours directly
- using real-time monitoring alerts to identify unexpected or concerning behaviour
- monitoring what the agent is doing so it can be stopped if necessary

Using additional 'judge' AI models and agents can help manage an agent's activities. However, these agents and the approach should be independently evaluated for performance noting that they may also have limitations and second order effects.

For higher risk scenarios, we recommend having human oversight alongside technically enforced controls. When events of consequence are flagged, humans should be able to investigate quickly and take action to limit any unintended impact.

4

Control the AI agent's environment with a robust sandbox

Always run AI agents within a sandboxed environment that controls and manages what resources can and cannot be communicated with, both locally and over a network.

a. Decide what your agent can connect to

For high-risk activities, the most robust approach is to run your AI agent within an isolated and disconnected environment, with access to pre-downloaded tools and information.

However, many use cases require the AI agent to communicate with external environments, systems and services including for access to the model. When this is necessary, apply sandbox isolations.

b. Define the boundaries of the sandbox

When designing a sandbox environment, think beyond the system where the:

- AI agent scaffolding runs
- AI agents commands run
- inference infrastructure is - whether local, remote or as a service

Where possible, these elements should be segregated from one another.

Instead, consider everything the agent can access or influence, whether directly or indirectly. This includes:

- Execution: between scaffolding, agent command execution sandbox and inference infrastructure
- Network: what it can remotely access
- Compute: what it can locally access
- Credentials: what identity and roles it can impersonate or otherwise have access to, intentionally or not
- Data: what data it has access to

Some autonomous AI tools, such as OpenClaw and Hermes Agent, are designed to interact with large numbers of systems and services. The wider the access, the greater the potential impact if the AI agent malfunctions unexpectedly or is compromised. To help limit the agent's potential impact - sometime described as its 'blast radius', you should try to restrict access to just the resources needed for the task being performed whilst monitoring for attempts at wider access.

c. Restrict network access

Where possible, deny all inbound and outbound network traffic to the AI agent's environment by default. Then, only allow connections that are required, using allowlists.

When selecting network sandboxing technologies, consider the security properties described in our cross domain approach and architecture guidance.

Some use cases require internet access and allowlists may not always provide the level of control or granularity required. In these cases, consider using protocol- or service-aware proxies that allow connections only by exception and require manual approval before permitting communications outside of approved scopes.

The following maturity model will help you assess your approach:

- **Level 1 (lowest)**: unrestricted network access
- **Level 2**: access restricted to an allowlist of approved domains
- **Level 3**: access restricted to just the API of the model
- **Level 4 (highest)**: no external network access, with the model hosted locally inside the network sandbox

d. Strengthen compute and host isolation

Many AI agent solutions include built-in sandboxing technologies. For some use cases, these may provide a sufficient level of protection. However, you should evaluate whether they are appropriate for your use case.

For higher risk activities, consider enforcing stronger isolation using additional controls and solutions.

When prompted or trained on the underlying model to do so, AI agents may be able to discover and exploit configuration issues, or potentially even vulnerabilities, in certain technical controls. If this event occurs, it could lead to a sandbox escape.

To reduce this risk, use multiple layers of isolation and control whilst regularly validating configurations, potentially via agentic AI approaches, to identify weaknesses or misconfiguration. This should include explicit prompts to not:

- connect to domains outside an allowlist
- attempt to escape the sandbox

Not all sandboxing technologies offer the same level of protection. Choose mature, trusted sandbox technologies that are designed specifically to isolate potentially malicious code and activities.

The following maturity model will help you assess your approach:

- **Level 1 (lowest)**: no compute isolation. The AI agent runs on, or has remote access to, the host platform alongside other workloads and data
- **Level 2**: the agent runs on the same host as other workloads, but is constrained using kernel primitives such as process separation and OCI containers *which are properly configured*. A residual risk of kernel exploit breakout remains
- **Level 3**: the agent runs on the same host as other workloads but is isolated using virtualisation technologies
- **Level 4 (highest)**: the AI agent runs on dedicated hardware that is separate from other workloads

e. Assign and limit an agent's access to credentials

All agents should be assigned their own unique identity in a class which differentiates them from human or individual systems.

The credentials available to an AI agent form part of its potential 'blast radius' if it behaves unexpectedly.

Credentials include API keys, OAUTH grants, SSH keys and any authenticated sessions the agent can access or use. These credentials allow the agent to perform actions using the identities and permissions associated with them.

Where an agent has access to a host environment, such as a level 1 above (no compute isolation), it may also be able to use credentials that are available to the user account under which it is running.

Where possible you should restrict the credentials available to the agent. Ensure it has only the permissions it needs for the task being performed and use credentials with the shortest possible lifetime. For some use cases, a proxy can help by injecting credentials into requests without exposing the credentials directly to the agent, which ensures the agent cannot use the credentials through an unexpected endpoint it finds access to.

5 Observability: log, audit and monitor agentic AI activity as part of security operations

While an AI agent is operating, you need a reliable way to access and analyse telemetry about the activities it's carrying out, both in near real time and afterwards.

This should include:

- chain of thought traces and transcripts from the AI agent
- logging events from the wider sandbox environment - such as access logs, proxies and network traffic

This observability telemetry is critical for actively overseeing an AI agent's activities in real time, and for investigating any incidents.

Make sure you save and protect these logs appropriately. They may contain sensitive data so you should ensure they are protected from modification or deletion. Where possible, logs should be immutable so you can trust them during an investigation.

You should also consider the attack surface created by your log collection infrastructure and whether an AI agent could abuse it to escape from its sandbox environment.

Agentic AI activity should be treated as a form of user activity. It should therefore be included in 24/7 security operational monitoring and incident response responses, with appropriate action taken when events arise. This may include running initial experiments in office hours, when more human oversight is available. Once you are confident that all controls are effective, you can expand to overnight or weekend autonomous execution.

For more information, follow our guidance on logging and monitoring, and incident management.

6 Make your AI activity easy to attribute

If your AI agent communicates with third-party systems, make it as easy as possible for those organisations to identify that the activity originates from you.

Industry methods for doing this are still emerging, but examples include:

- ensuring traffic originates from IP addresses that support reverse lookups
- adding identifying headers to outbound communications as a form of watermarking, such as HTTP headers

If you receive abuse reports or other communications about your AI agent's activity, manage them in the same way as a vulnerability or abuse report, and follow an incident management playbook.

7 Emergency shutdown: maintain the ability to 'pull the plug'

If an incident is detected or reported, you should always be able to 'pull the plug' and halt autonomous AI agent activity immediately.

This may mean more than stopping the agentic AI processes. Your controls should cover the wider system, allowing you to rapidly restrict network access to the agentic AI infrastructure and interrupt the communications between selected AI agents and the AI model inference infrastructure.

Further reading

The advice in this blog should form part of a wider approach to securing AI models and systems. Further information is available in:

- The ETSI standard, EN 304 223. Baseline Cyber Security Requirements for AI Models and Systems
- Thinking carefully before adopting agentic AI
- Careful adoption of agentic AI services

AI security is evolving rapidly. As capabilities, use cases and standards develop, organisations should continue to review their controls and assumptions, and ensure that the level of autonomy they grant AI agents remains proportionate to the risk appetite of the organisation.

WRITTEN BY

Toby W

Principal Security Architect, NCSC

PUBLISHED

20 August 2026

WRITTEN FOR

Cyber security professionals Large organisations Public sector Small & medium sized organisations

PART OF BLOG

NCSC publications