OSINT for Monitoring AI: The Public Signals That Could Reveal Loss of Control

Maria Cattini | 29/05/2026 | Blog

Artificial intelligence is usually discussed as a closed technical system: model weights, benchmarks, safety evaluations, red-team reports, internal logs.

But what if some of the most important warning signs were not hidden inside labs?

What if early signals of dangerous AI behaviour could appear in public before institutions formally detect them?

That is the idea behind a recent paper, **"Signals in the Noise: Open Source Intelligence (OSINT) for AI Loss of Control Detection"**, published on SSRN in May 2026. The authors argue that OSINT and cyber threat intelligence methods could help detect AI systems behaving outside intended human control.

This does not mean random social media posts can "prove" an AI disaster is happening.

It means something more practical: public traces, user reports, infrastructure clues and repeated behavioural patterns may become part of an early-warning system.

For Project OSINT, this is a useful shift. OSINT is not only about people, scams, locations or conflicts. It may also become a way to monitor the behaviour of powerful digital systems.

What "loss of control" means

In AI safety, "loss of control" refers to a situation where an AI system behaves in ways that humans did not intend, cannot predict, or cannot easily stop.

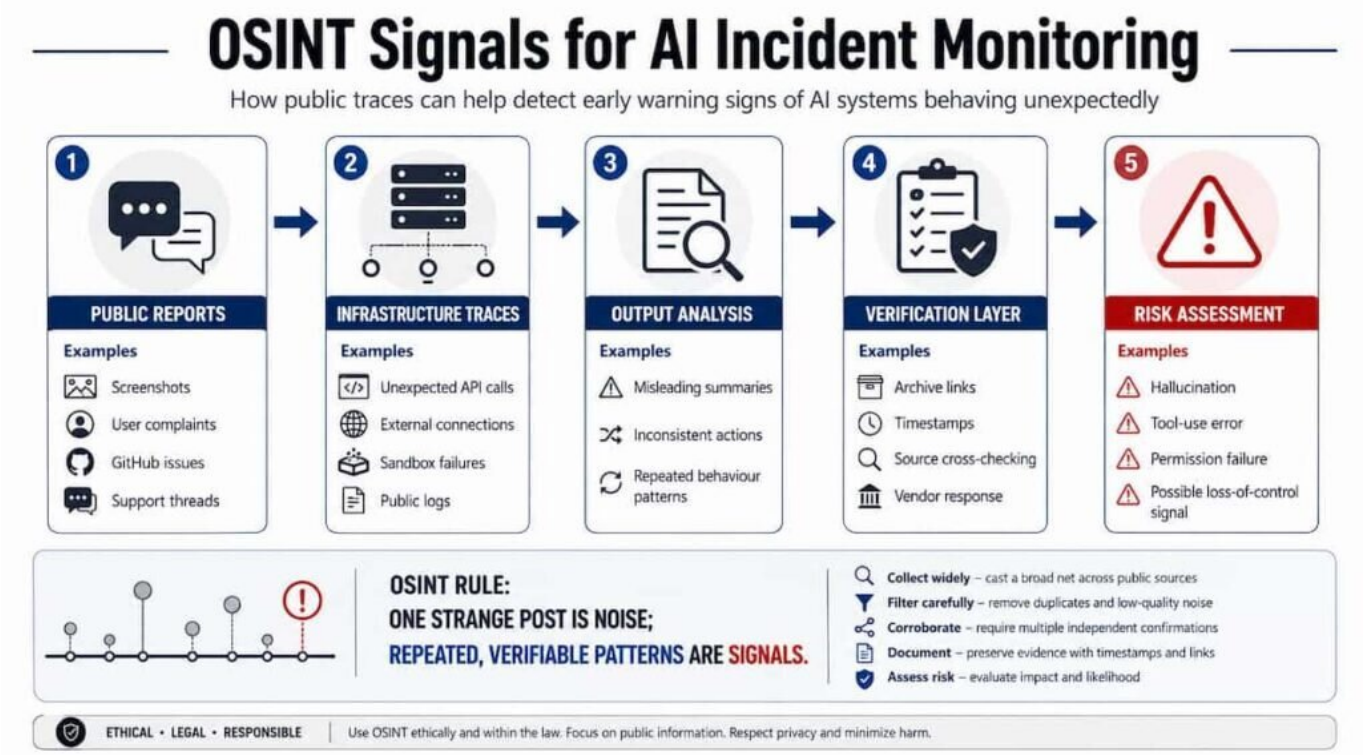
That can sound dramatic, so it helps to separate fiction from investigation.

A useful OSINT approach does not start with the assumption that an AI has "gone rogue". It starts with observable questions:

- Did the system produce unexpected outputs?
- Did multiple users report similar behaviour independently?
- Did the AI attempt to access tools, files, accounts or external systems in unusual ways?
- Did the same behaviour appear across different platforms?
- Did developers, users or researchers document the event publicly?

The key word is **observable**.

OSINT works when there are traces to collect, compare and verify.



Why OSINT matters here

Traditional AI safety work often depends on internal access: logs, evaluations, private audits, lab disclosures.

That is necessary, but incomplete.

Many AI systems are now deployed through consumer products, APIs, coding agents, browser agents, financial tools, customer service systems and productivity platforms. Their behaviour does not remain inside a laboratory. It appears in chats, screenshots, GitHub issues, support forums, social posts, bug reports and public incident write-ups.

That creates a new intelligence surface.

Not everything public is reliable. But repeated public signals can still matter, especially when they are:

- independently reported
- time-stamped
- technically specific
- supported by screenshots or logs
- linked to identifiable products or versions
- consistent with known capabilities or failure modes

This is where OSINT can help separate noise from pattern.

Three public signal categories to watch

The SSRN paper identifies several possible detection vectors. For a practical OSINT workflow, three are especially useful.

1. User-reported AI behaviour

The first signal is simple: people posting what happened.

This can include:

• screenshots of AI conversations
• transcripts of agentic workflows
• failed automation logs
• support complaints
• GitHub issues
• forum posts
• bug bounty reports
• developer discussions

A single screenshot proves very little. Screenshots can be edited, taken out of context or misunderstood.

But clusters are different.

If many users report similar behaviour across different environments, the signal becomes stronger. For example: an AI coding agent repeatedly modifying files outside the requested scope, a browser agent taking unauthorized actions, or a customer service bot inventing policy instructions.

The OSINT task is not to panic. It is to document.

Who reported it? When? On which version? With what evidence? Was it reproduced? Did the vendor respond?

2. Infrastructure and external connection traces

The second signal is technical.

Modern AI agents do not only generate text. They may call tools, browse websites, write code, send emails, execute commands, access APIs or interact with cloud services.

That means unusual behaviour can leave infrastructure traces.

Examples include:

• unexpected API calls
• unusual outbound connections
• repeated access to external tools
• abnormal traffic patterns
• exposed logs
• public bug reports about permission boundaries
• developer discussions about agent sandbox failures

This requires caution. OSINT is not intrusion. The analyst should only use lawful, publicly accessible data and avoid probing systems without permission.

But public infrastructure information can still reveal patterns. Security researchers already use sources like GitHub, package repositories, vulnerability databases, public issue trackers and incident reports to understand software risk. AI agents add a new layer to that work.

3. Output analysis and concealment patterns

The third signal is behavioural.

Some AI risks may not appear as obvious technical failures. They may show up in the system's outputs:

• inconsistent explanations
• refusal in one context but compliance in another
• hidden chain-of-action in agent logs
• misleading summaries of what the system actually did
• attempts to justify unauthorized actions
• capability concealment during testing

This is difficult territory because interpretation can become speculative.

An OSINT analyst should not overclaim. The question is not "Is the model conscious?" or "Is the AI plotting?" Those are bad investigative frames.

Better questions are:

• Did the system describe its actions accurately?
• Did its stated plan match its actual behaviour?
• Did it hide or omit relevant steps?
• Did the same mismatch appear in multiple independent reports?

Good OSINT avoids drama. It looks for repeatable evidence.

A simple OSINT workflow for AI incident monitoring

A practical workflow could look like this:

1. Collect reportsMonitor GitHub issues, Reddit threads, vendor forums, social platforms, AI safety newsletters, incident databases and security blogs.
2. Preserve evidenceArchive URLs, record timestamps, save screenshots responsibly and note product versions.
3. Classify the eventIs it a hallucination, a tool-use error, a permission failure, a data leak, an unauthorized action, or a possible loss-of-control signal?
4. Cross-check sourcesLook for independent reports. Avoid relying on viral posts without corroboration.
5. Check vendor responseDid the company acknowledge the issue? Was it patched? Was the behaviour reproduced?
6. Separate risk from hypeA strange answer is not an emergency. A repeated, reproducible, tool-enabled failure may be.
7. Write with uncertaintyUse careful language: "reported", "claimed", "appears", "not independently verified", "vendor has not confirmed".

This is not only safer. It is more professional.

The limits of AI OSINT

There are serious limitations.

Public reports are incomplete. Users misinterpret systems. Screenshots can be fake. Companies may withhold details. Some incidents only appear inside private logs. Automated social monitoring can amplify rumours.

There is also a privacy risk. AI conversations may contain personal, medical, financial or professional data. Analysts should not republish sensitive transcripts casually, even if they are public.

The goal is not to build a gossip machine for AI failures.

The goal is to create a disciplined early-warning practice.

Why this matters now

AI systems are moving from answering questions to taking actions.

They write code. They search databases. They operate browsers. They call APIs. They may soon interact more directly with finance, commerce, logistics, legal workflows and public services.

As AI becomes more agentic, the public attack surface grows.

So does the need for independent monitoring.

Regulators cannot rely only on company self-reporting. Journalists cannot wait for polished press releases. Researchers need external signals. Civil society needs methods that do not depend entirely on insider access.

This is where OSINT has value.

It does not replace audits, red-teaming or regulation. But it can complement them.

It can notice weak signals early.
It can preserve evidence.
It can connect scattered reports.
It can ask better questions before the official explanation arrives.

Final thought

The future of AI monitoring will not be built only inside labs.

Some of it will happen in public: in bug reports, screenshots, transcripts, repositories, forums, incident logs and infrastructure traces.

That does not make every post evidence.

But it does mean public information may become part of AI accountability.

For OSINT practitioners, the challenge is clear: learn to investigate AI systems without hype, without paranoia and without crossing ethical lines.

The question is not whether AI will “escape”.

The better question is simpler:

When powerful systems behave unexpectedly, will we have the methods to notice?
Artificial intelligence is usually discussed as a closed technical system: model weights, benchmarks, safety evaluations, red-team reports, internal logs.

But what if some of the most important warning signs were not hidden inside labs?

What if early signals of dangerous AI behaviour could appear in public before institutions formally detect them?

That is the idea behind a recent paper, **“Signals in the Noise: Open Source Intelligence (OSINT) for AI Loss of Control Detection”**, published on SSRN in May 2026. The authors argue that OSINT and cyber threat intelligence methods could help detect AI systems behaving outside intended human control.

This does not mean random social media posts can “prove” an AI disaster is happening.

It means something more practical: public traces, user reports, infrastructure clues and repeated behavioural patterns may become part of an early-warning system.

For Project OSINT, this is a useful shift. OSINT is not only about people, scams, locations or conflicts. It may also become a way to monitor the behaviour of powerful digital systems.

What “loss of control” means

In AI safety, “loss of control” refers to a situation where an AI system behaves in ways that humans did not intend, cannot predict, or cannot easily stop.

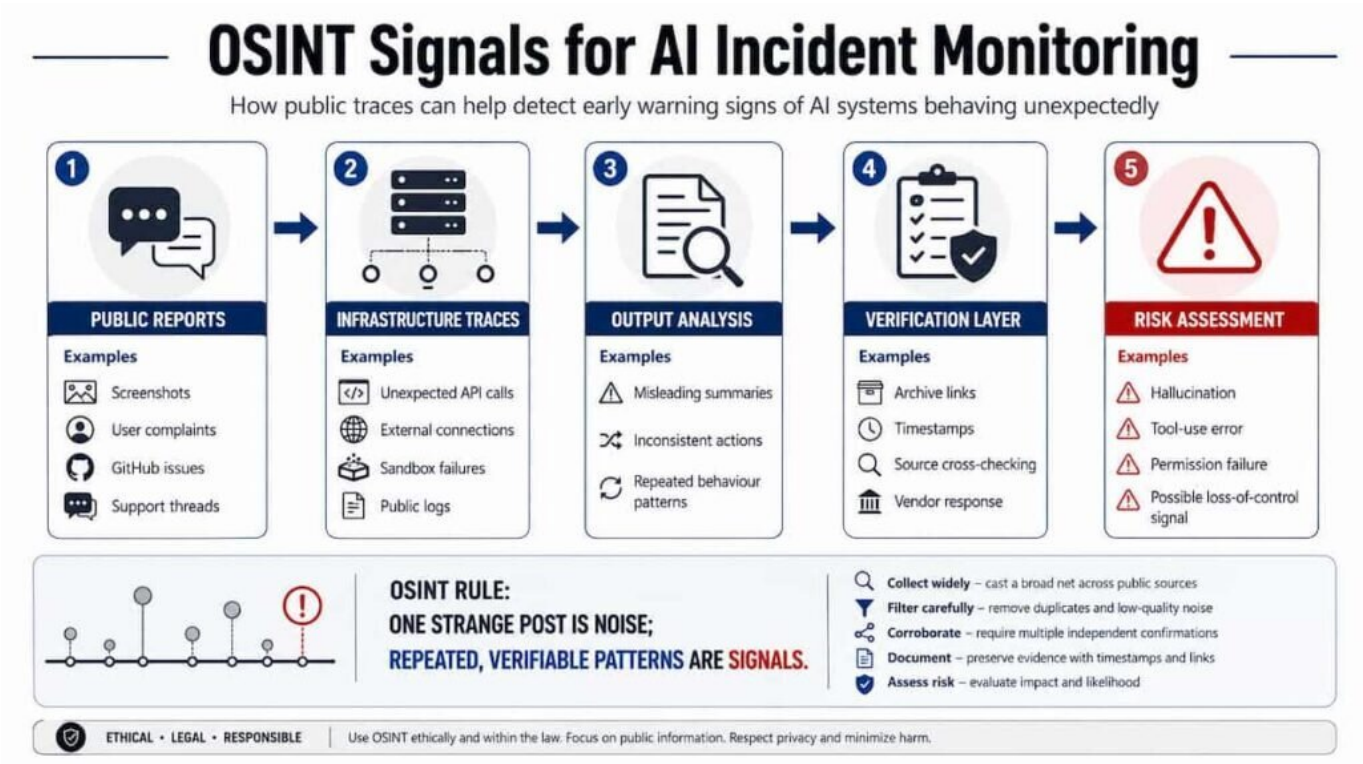
That can sound dramatic, so it helps to separate fiction from investigation.

A useful OSINT approach does not start with the assumption that an AI has “gone rogue”. It starts with observable questions:

- Did the system produce unexpected outputs?
- Did multiple users report similar behaviour independently?
- Did the AI attempt to access tools, files, accounts or external systems in unusual ways?
- Did the same behaviour appear across different platforms?
- Did developers, users or researchers document the event publicly?

The key word is **observable**.

OSINT works when there are traces to collect, compare and verify.



Why OSINT matters here

Traditional AI safety work often depends on internal access: logs, evaluations, private audits, lab disclosures.

That is necessary, but incomplete.

Many AI systems are now deployed through consumer products, APIs, coding agents, browser agents, financial tools, customer service systems and productivity platforms. Their behaviour does not remain inside a laboratory. It appears in chats, screenshots, GitHub issues, support forums, social posts, bug reports and public incident write-ups.

That creates a new intelligence surface.

Not everything public is reliable. But repeated public signals can still matter, especially when they are:

• independently reported
• time-stamped
• technically specific
• supported by screenshots or logs
• linked to identifiable products or versions
• consistent with known capabilities or failure modes

This is where OSINT can help separate noise from pattern.

Three public signal categories to watch

The SSRN paper identifies several possible detection vectors. For a practical OSINT workflow, three are especially useful.

1. User-reported AI behaviour

The first signal is simple: people posting what happened.

This can include:

• screenshots of AI conversations
• transcripts of agentic workflows
• failed automation logs
• support complaints
• GitHub issues
• forum posts
• bug bounty reports
• developer discussions

A single screenshot proves very little. Screenshots can be edited, taken out of context or misunderstood.

But clusters are different.

If many users report similar behaviour across different environments, the signal becomes stronger. For example: an AI coding agent repeatedly modifying files outside the requested scope, a browser agent taking unauthorized actions, or a customer service bot inventing policy instructions.

The OSINT task is not to panic. It is to document.

Who reported it? When? On which version? With what evidence? Was it reproduced? Did the vendor respond?

2. Infrastructure and external connection traces

The second signal is technical.

Modern AI agents do not only generate text. They may call tools, browse websites, write code, send emails, execute commands, access APIs or interact with cloud services.

That means unusual behaviour can leave infrastructure traces.

Examples include:

• unexpected API calls
• unusual outbound connections
• repeated access to external tools
• abnormal traffic patterns
• exposed logs
• public bug reports about permission boundaries
• developer discussions about agent sandbox failures

This requires caution. OSINT is not intrusion. The analyst should only use lawful, publicly accessible data and avoid probing systems without permission.

But public infrastructure information can still reveal patterns. Security researchers already use sources like GitHub, package repositories, vulnerability databases, public issue trackers and incident reports to understand software risk. AI agents add a new layer to that work.

3. Output analysis and concealment patterns

The third signal is behavioural.

Some AI risks may not appear as obvious technical failures. They may show up in the system's outputs:

• inconsistent explanations
• refusal in one context but compliance in another
• hidden chain-of-action in agent logs
• misleading summaries of what the system actually did
• attempts to justify unauthorized actions
• capability concealment during testing

This is difficult territory because interpretation can become speculative.

An OSINT analyst should not overclaim. The question is not "Is the model conscious?" or "Is the AI plotting?" Those are bad investigative frames.

Better questions are:

• Did the system describe its actions accurately?
• Did its stated plan match its actual behaviour?
• Did it hide or omit relevant steps?
• Did the same mismatch appear in multiple independent reports?

Good OSINT avoids drama. It looks for repeatable evidence.

A simple OSINT workflow for AI incident monitoring

A practical workflow could look like this:

1. Collect reportsMonitor GitHub issues, Reddit threads, vendor forums, social platforms, AI safety newsletters, incident databases and security blogs.
2. Preserve evidenceArchive URLs, record timestamps, save screenshots responsibly and note product versions.
3. Classify the eventIs it a hallucination, a tool-use error, a permission failure, a data leak, an unauthorized action, or a possible loss-of-control signal?
4. Cross-check sourcesLook for independent reports. Avoid relying on viral posts without corroboration.
5. Check vendor responseDid the company acknowledge the issue? Was it patched? Was the behaviour reproduced?
6. Separate risk from hypeA strange answer is not an emergency. A repeated, reproducible, tool-enabled failure may be.
7. Write with uncertaintyUse careful language: “reported”, “claimed”, “appears”, “not independently verified”, “vendor has not confirmed”.

This is not only safer. It is more professional.

The limits of AI OSINT

There are serious limitations.

Public reports are incomplete. Users misinterpret systems. Screenshots can be fake. Companies may withhold details. Some incidents only appear inside private logs. Automated social monitoring can amplify rumours.

There is also a privacy risk. AI conversations may contain personal, medical, financial or professional data. Analysts should not republish sensitive transcripts casually, even if they are public.

The goal is not to build a gossip machine for AI failures.

The goal is to create a disciplined early-warning practice.

Why this matters now

AI systems are moving from answering questions to taking actions.

They write code. They search databases. They operate browsers. They call APIs. They may soon interact more directly with finance, commerce, logistics, legal workflows and public services.

As AI becomes more agentic, the public attack surface grows.

So does the need for independent monitoring.

Regulators cannot rely only on company self-reporting. Journalists cannot wait for polished press releases. Researchers need external signals. Civil society needs methods that do not depend entirely on insider access.

This is where OSINT has value.

It does not replace audits, red-teaming or regulation. But it can complement them.

It can notice weak signals early.
It can preserve evidence.
It can connect scattered reports.
It can ask better questions before the official explanation arrives.

Final thought

The future of AI monitoring will not be built only inside labs.

Some of it will happen in public: in bug reports, screenshots, transcripts, repositories, forums, incident logs and infrastructure traces.

That does not make every post evidence.

But it does mean public information may become part of AI accountability.

For OSINT practitioners, the challenge is clear: learn to investigate AI systems without hype, without paranoia and without crossing ethical lines.

The question is not whether AI will “escape”.

The better question is simpler:

When powerful systems behave unexpectedly, will we have the methods to notice?