International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

PRIVACY LEAKAGE IN LLM-BASED LOG ANALYSIS

Ms. Manasa Reddy Vippala[1], **Mr. Moosaramtota Abhinav**[2]

[1]Department of Computer Science, Vishwa Vishwani Institute of Systems and Management, Hyderabad, Telangana, India
[2]Department of Computer Science and Engineering, KLH University, Hyderabad, Telangana, India
Emails: 1manasa1reddy1@gmail.com, abhimoosara1105@gmail.com

ABSTRACT

Software logs often contain sensitive information, such as email addresses, passwords, and API keys. As developers increasingly use Large Language Model (LLM) tools to diagnose software errors, log content may be shared with cloud-based AI services. Depending on the provider's data-handling practices, this information may be retained or appear in generated responses, creating privacy risks that are easy to overlook. Although existing research has examined privacy leakage in LLM systems, the allocation of accountability for privacy risks in LLM-based log analysis warrants further investigation.

This paper adopts a conceptual and descriptive approach based on secondary sources to examine these risks. It argues that responsibility is shared among three parties: the organisation that owns the logs, the engineers who share them, and the AI provider that processes them. However, accountability remains difficult to define because these parties have different roles, levels of control, and responsibilities in handling sensitive information. The paper also examines consent, transparency, and trust, with reference to India's Digital Personal Data Protection Act, 2023. Redaction is discussed as one possible safeguard, while acknowledging its limitations. The paper argues that reducing privacy risks requires clearer accountability among the parties involved, alongside complementary safeguards such as redaction, data minimisation, and human oversight. These measures can help organisations adopt LLM-based log analysis more responsibly while reducing the risk of exposing sensitive information.

Keywords: Privacy Leakage, Large Language Models, Log Analysis, Data Protection, Accountability, Responsible AI

1. INTRODUCTION

Every computer system keeps a diary. When a website loads, a payment goes through, or a login fails, the system writes down what happened. These records are called logs, and they are the first place an engineer looks when something breaks.

Reading logs by hand is slow. A busy system writes far more than any person can read, so developers have started using AI tools built on large language models (LLMs) to help understand them. The model reads the text, spots the error and explains the likely cause in plain language. It saves time, and it is very easy to do.

The catch is that logs are written for machines, not for privacy. They often carry email addresses and IP addresses, and sometimes passwords or API keys that slipped in by accident. When logs are sent to an external AI service, their contents are shared with another party. Depending on the provider's data-handling practices, the information may be stored or used in ways that create privacy risks.

An obvious fix is to hide the sensitive parts before the log is shared. But hiding too much can erase the very clues the model needs to find the fault. In our view, a filter that protects privacy but breaks diagnosis is not one a developer would keep using.

Behind this trade-off sits a harder question: when sensitive information leaks through LLM-based log analysis, who should be held accountable? The organisation that owns the logs, the engineers who share them, and the AI provider that processes them all play a part, yet none of them alone seems clearly answerable. This paper examines that question. It describes the privacy risks of LLM-based log analysis, discusses how responsibility is shared among the three parties, and considers safeguards such as redaction that can reduce the risk.

2. REVIEW OF LITERATURE

2.1 AI-based log analysis

Ma et al. (2026) and Akhtar et al. (2025) both review how large language models are being used to study logs. The picture they paint is similar. Models are used to turn messy log lines into structured records, to spot unusual behaviour and to help trace faults back to their cause.

Because logs are mostly free text, LLMs often cope better than older rule-based tools. The authors also admit the practical hurdles: cost, reliability and how hard it is to judge whether a diagnosis is right.

Although these reviews discuss practical challenges in LLM-based log analysis, the accountability implications of sharing logs containing personal information are not their main focus. This leaves room to examine how responsibility should be shared when sensitive information is exposed during log analysis.

2.2 Privacy leakage from AI

Carlini et al. (2021) showed that language models can memorise pieces of their training data and repeat them when prompted in the right way, including personal information. The lesson is simple: what goes into a model may not always stay there. El Yagoubi et al. (2026), in the AgentLeak study, widen the view to systems where several AI agents work together, and show that private data can slip out through the messages the agents pass to each other, not only through the final answer.

On the protective side, tools for finding and masking personal data in text exist, and the idea of cleaning a prompt before sending it is not new. Still, this work mostly concerns general text. It says little about software logs, which are full of timestamps, codes and odd formats, and it rarely considers who is responsible when sensitive details still slip through.

2.3 Responsibility and accountability

Goetze (2022) discusses the responsibility gap: when an autonomous system causes harm, it can seem that nobody is clearly to blame. The argument is that the people who build and use such systems cannot step back from what those systems do. Custers et al. (2025) describe responsibility as shared between several parties, with extra duties of care for anyone who handles data on behalf of others. Raji et al. (2020) offer an end-to-end auditing framework, so that an AI system is checked at each stage of its life rather than only after something goes wrong.

These ideas matter here because LLM-based log analysis involves several parties and a flow of sensitive data between them. This paper applies their shared-responsibility thinking to the

specific case of logs: the organisation, the engineers and the AI provider each handle the data at a different point, and each has a role in keeping it safe. The papers themselves, however, are not about logs or log analysis.

3. RESEARCH GAP

Existing work shows how LLMs analyse logs and warns that privacy is at risk. Tools for hiding personal data in general text also exist. But to the best of our review, little research connects these threads to ask who should be held accountable when personal data leaks through LLM-based log analysis. How responsibility is shared between the organisation, its engineers and the AI provider, and what safeguards support that accountability, is still loosely defined.

4. OBJECTIVES

- To explain how private data can leak when software logs are shared with LLM-based tools.
- To compare the benefits of LLM-based log analysis with its privacy risks.
- To discuss how responsibility is shared among the organisation, its engineers and the AI provider.
- To examine the role of consent, transparency and trust, with reference to India's Digital Personal Data Protection Act, 2023.
- To describe safeguards, including redaction, that can reduce privacy risks.

5. RESEARCH METHODOLOGY

This paper uses a conceptual and descriptive approach based on secondary sources. It draws on academic papers and India's Digital Personal Data Protection Act, 2023. No experiment was conducted and no primary data were collected. The paper analyses the privacy risks of LLM-based log analysis, discusses the responsibilities of the parties involved, and considers safeguards in light of the reviewed literature.

6. CONCEPTUAL BACKGROUND

Logs. Logs are records of what a computer system did and when. They help engineers find the cause of errors, but they can also capture personal details by accident.

Large Language Models (LLMs). LLMs are AI tools that read and write text. Trained on huge amounts of writing, they can read a log and explain a likely cause in plain language.

Private or personal data (PII). This is information about a real person, such as an email address or IP address. If it is exposed, it can be used to identify, contact or track that person.

AI ethics. AI ethics is about using AI fairly, safely and responsibly. It asks not only whether a tool works, but who could be harmed and who is accountable.

Redaction. Redaction means hiding or masking private details before data is shared, for example swapping an email for a placeholder. It lets the useful part of a log travel while the sensitive part stays behind.

Accountability. Accountability means being answerable for what happens. In LLM-based log analysis, it asks who must explain, correct and take responsibility when sensitive information is exposed.

DPDP Act 2023. India's Digital Personal Data Protection Act, 2023, establishes a legal framework for processing digital personal data. It addresses consent, certain legitimate uses, the duties of Data Fiduciaries, and the rights of individuals.

7. ANALYSIS AND DISCUSSION

PART A: THE CHALLENGE

7.1 How Private Data Leaks

Software logs are detailed records of the events that occur within a computer system, including user logins, file access, network activity, and error messages. Because these records capture

day-to-day technical activity, they often contain identifiers and sometimes personal data (Ma et al., 2026). Typical examples include email addresses, IP addresses, usernames, session tokens, passwords, and API keys. Although such details are often necessary for troubleshooting, they are not always removed before logs are shared with external tools.

When developers use external AI services for log analysis, log content is transmitted to the service for processing. The provider's policies and the applicable data-handling arrangements determine how submitted information is stored, accessed, and used. Organisations should review these practices before sharing logs, rather than assume that every service handles data in the same way.

Research has shown that large language models can, in certain circumstances, reproduce fragments of their training data (Carlini et al., 2021). This does not establish that logs submitted as prompts will be memorised or revealed. However, it highlights a possible privacy risk. El Yagoubi et al. (2026) also examine privacy leakage in multi-agent LLM systems. These studies illustrate possible risks, but they do not prove that every risk occurs in every AI service.

Consider a developer who submits a log containing user@email.com to an AI tool to investigate a login failure. The email address is transmitted to the service as part of the log. Whether it is stored or used later depends on the provider's data-handling practices. This illustrates why organisations should check how a service handles submitted information before sharing logs.

What makes this problem difficult is that data exposure may go unnoticed. A system can function normally even when sensitive data is being transmitted or stored. This makes LLM-based log analysis an ethical concern as well as a technical one.

7.2 Benefit vs. Risk

The appeal of using LLMs for log analysis is easy to understand. These tools can read large volumes of technical text quickly, identify patterns that might take a human much longer to notice, and suggest possible causes for an error in plain language (Akhtar et al., 2025; Ma et al., 2026). For developers working under time pressure, this can reduce the effort needed to diagnose problems and speed up routine maintenance tasks. Some organisations have begun to use such tools as part of their diagnostic workflow, although rigorous validation of their performance at production scale remains limited (Ma et al., 2026).

However, the privacy cost of this convenience is not always visible at the moment the log is submitted. Unlike a clear security breach, the movement of sensitive data to an external AI provider happens quietly, with no immediate warning or visible consequence. Because nothing appears to go wrong, teams may continue sharing logs without pausing to consider what personal or confidential information is being sent.

There is also a practical tension on the other side. If an organisation hides or removes too much information before sending logs to an AI tool, the model may no longer have enough context to identify the real cause of an error. Heavy redaction can therefore reduce the usefulness of the analysis. This creates a trade-off: protecting privacy more strongly often means accepting a less precise diagnosis, while preserving full diagnostic detail means accepting greater privacy exposure. Managing this balance is a key part of using LLMs responsibly for log analysis.

PART B: THE RESPONSIBILITY

7.3 Who Is Responsible

When sensitive information from a log reaches an external AI service, it is rarely clear who should be held accountable if something goes wrong. Three parties are typically involved: the organisation that owns the logs, the engineers or developers who choose to share them, and the AI provider that receives and processes them. Each party can reasonably point to another as the one who made the key decision, which makes it difficult to assign responsibility in a straightforward way.

The organisation that owns the logs is usually best placed to decide what data may be shared and with whom. It sets internal policies, chooses which AI tools are approved, and is ultimately responsible for the data it controls. However, a developer working under time pressure might paste a log into an AI assistant without first checking whether it contains personal details or credentials. In such cases, the organisation may argue that the leak resulted from an employee's action rather than a failure of its own policy.

The engineers who handle the logs are the ones who directly interact with the AI tool. They decide what to copy, what to send, and whether to remove sensitive details first. While they act within the rules set by their organisation, they also carry practical responsibility for the

immediate decision to share data. If an engineer sends a log containing a password or an API key without redacting it, that choice has direct consequences regardless of broader company policy.

The AI provider, meanwhile, receives and processes the data after it has been submitted. Its responsibilities depend on the terms of service and the data-processing agreements in place. The extent to which providers may retain or reuse submitted data varies, and users may not always be aware of these differences.

Table 1: Responsibilities in LLM-based log analysis

Party	Role in the workflow	Potential Responsibility
Company / organisation	Owns or controls the logs, sets policies, and approves AI tools	Establishes sharing rules and safeguards
Engineers	Handle logs and decide what to submit under organisational procedures	Check logs and follow approved sharing practices
AI Provider	Receives and processes submitted data	Handle data according to applicable requirements and agreed terms

Note. The legal roles of the organisation and AI provider depend on the actual data-processing arrangement and applicable legal requirements. The roles and responsibilities shown here are illustrative, not automatic legal classifications (Government of India, 2023).

AI governance research suggests that responsibility in complex technological systems is often shared across multiple actors (Custers et al., 2025; Goetze, 2022). Although this research does not focus specifically on log analysis, it highlights the accountability challenge when responsibilities overlap. The central problem in LLM-based log analysis is not that no one is responsible, but that responsibility is distributed without being clearly defined.

7.4 Permission, Openness, and Trust

A further ethical concern is that the people whose information appears in software logs — such as employees, customers, or system users — are rarely aware that their data may be read by an AI tool. In most cases, they did not give explicit permission for their email address, IP address, or activity records to be shared with an external AI provider. This raises a question of fairness: data collected for one purpose, such as system monitoring, is being used for another purpose, namely AI-based error diagnosis, without the knowledge of the person it concerns.

At the same time, asking every individual for permission before a log is analysed may not be realistic. Logs can contain information about many people at once, and obtaining consent from each person would be slow and often impractical. This does not mean the concern disappears; rather, it means organisations need to find other ways to handle the situation responsibly, such as limiting what is shared and being clear about how AI tools are used.

Transparency plays an important role here. When organisations are open about the fact that logs may be processed by AI services, and about what kinds of data are involved, employees and users are better able to understand the risks. Openness also makes it easier to build trust, since people are more willing to accept data practices when they know what is happening.

In India, the Digital Personal Data Protection Act, 2023 also addresses the handling of personal data, including consent, the duties of Data Fiduciaries, and the rights of individuals whose data is processed (Government of India, 2023). While a full legal analysis is beyond the scope of this paper, the Act provides a relevant backdrop for thinking about when and how personal information in logs may be shared with AI services.

PART C: THE WAY FORWARD

7.5 Safeguards

Redaction

One of the most direct ways to reduce privacy risk is redaction — the practice of hiding or masking sensitive details before a log is shared with an AI tool. For example, an email address can be replaced with a label such as [EMAIL], or an IP address with [IP_ADDRESS]. This

prevents the AI provider from seeing the actual value while still allowing the model to understand the general structure of the log.

Redaction is useful because it reduces the amount of personal data that leaves the organisation. However, it is not a complete solution. Automatic tools can miss unusual formats or context-dependent details, giving a false sense of safety. Removing too much information can also make the log harder for the model to interpret, weakening the diagnosis. Importantly, someone must still decide what counts as sensitive and what should be masked — and that person or team remains accountable for the choice.

Other Safeguards

Beyond redaction, several other practices can help reduce risk. Organisations can share only the portion of a log that is needed to solve the problem, rather than the entire file. Keeping a human in the loop — where a person reviews the AI's suggestion before any action is taken — adds a layer of oversight. Regular checks of the organisation's own process, such as reviewing which logs are being sent and what data they contain, help catch problems early; this is similar in spirit to the end-to-end auditing approaches proposed for AI systems (Raji et al., 2020). Finally, being open with employees and users about how AI tools are used builds trust and aligns with the transparency discussed earlier.

These measures are best understood as ideas to consider rather than fixed technical steps. Their value lies in combining them: redaction alone is helpful, but it works best alongside data minimisation, human oversight, and openness about AI use.

8. MAJOR FINDINGS

This review leads to several key observations. First, software logs often contain identifiers and sometimes personal or confidential details, and sending them to LLM-based tools moves this information beyond the organisation's direct control. Second, data exposure may go unnoticed, making the privacy risk easy to overlook. Third, responsibility is shared among the company, its engineers, and the AI provider, yet it remains poorly defined in practice. Fourth, the people whose data appears in logs are rarely informed or asked for consent. Finally, redaction and

sharing less data reduce risk, but neither is sufficient on its own; a combination of safeguards is needed.

9. SUGGESTIONS

Based on the discussion above, organisations should redact sensitive details before logs are shared with AI tools and limit sharing to only what is necessary for the task. Engineers should be encouraged to check logs for personal information before submitting them. Keeping a human in the review process, periodically checking internal practices, and being open about AI use can further reduce risk. To make accountability clearer, organisations should decide in advance who is responsible for approving what is shared with an AI provider, record that decision, and agree with the provider in writing on how submitted data will be handled. These steps work best together, since no single measure fully addresses the privacy concerns raised by LLM-based log analysis.

10. CONCLUSION

This paper has examined the privacy risks that arise when software logs containing sensitive information are shared with LLM-based tools. It has argued that responsibility for protecting this data is shared among organisations, engineers, and AI providers, but remains poorly defined in practice. Redaction, data minimisation, human oversight, and transparency offer practical ways to reduce risk, though none is sufficient alone. As AI tools become more common in everyday technical work, addressing these concerns is important for using AI responsibly. This study is based on existing sources, and future work can build on it through empirical testing of specific safeguards.

REFERENCES

Akhtar, S., Khan, S., & Parkinson, S. (2025). LLM-based event log analysis techniques: A survey. *arXiv*. https://arxiv.org/abs/2502.00677

Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, Ú., Oprea, A., & Raffel, C. (2021). Extracting training data from large language models. In *Proceedings of the 30th USENIX Security Symposium* (pp. 2633–2650). USENIX Association. https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting

Custers, B., Lahmann, H., & Scott, B. I. (2025). From liability gaps to liability overlaps: Shared responsibilities and fiduciary duties in AI and other complex technologies. *AI & Society, 40*(5), 4035–4050. https://doi.org/10.1007/s00146-024-02137-1

El Yagoubi, F., Badu-Marfo, G., & Al Mallah, R. (2026). AgentLeak: A benchmark for internal-channel privacy leakage in multi-agent LLM systems. *IEEE Access, 14*, 94960–94978. https://doi.org/10.1109/ACCESS.2026.3704541

Goetze, T. S. (2022). Mind the gap: Autonomous systems, the responsibility gap, and moral entanglement. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency* (pp. 972–983). ACM. https://doi.org/10.1145/3531146.3533106

Government of India. (2023). *The Digital Personal Data Protection Act, 2023*. Ministry of Law and Justice. https://www.meity.gov.in/static/uploads/2024/06/2bf1f0e9f04e6fb4f8fef35e82c42aa5.pdf

Ma, Z., Yang, J., & Chen, T.-H. (2026). LLM4Log: A systematic review of large language model-based log analysis. *ACM Transactions on Software Engineering and Methodology*. Advance online publication. https://doi.org/10.1145/3842759

Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., & Barnes, P. (2020). Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 33–44). ACM. https://doi.org/10.1145/3351095.3372873