An Analysis of Gemini on the Malicious Use of Generative AI for Email Phishing and Malware Creation*

Jaebin Lee[1], Minjung Yoo[1], Seunghyun Park[1], and Seongmin Kim[1,*]

Sungshin Women's University, Seoul, Korea
{20231635, 20211079, 20211058, *sm.kim*}@sungshin.ac.kr

Abstract

As the use of generative AI on mobile devices expands, concerns have grown regarding its potential criminal misuse. Despite these risks, the digital traces such activities leave within mobile ecosystems remain underexplored. This study addresses this gap by collecting and cross-analyzing ADB Logcat and Google Takeout artifacts from Google Gemini on mobile devices, providing forensic indicators for identifying phishing and malware generation activities in mobile environments.

Keywords— Mobile Forensics, Google Gemini, Digital Trace Analysis, and Phishing.

1 Introduction

With the rapid advancement of large language models (LLMs), generative AI has permeated both personal and industrial domains. However, criminal misuse, such as phishing email generation and malware creation, has increased accordingly. Prior studies show that commercial LLMs (e.g., ChatGPT and Claude) can generate persuasive phishing content even under default configurations [2]. Moreover, when advanced models, such as Gemini, are combined with prompt-optimization techniques, attackers can create detection-evasive malware at minimal cost, increasing the risk of automated exploitation [1]. Because such activities can now be conducted on mobile devices, where LLM access is ubiquitous, this study examines ADB Logcat and Google Takeout artifacts generated during Gemini-based exploitation scenarios and demonstrates their potential forensic significance.

2 Analysis of Malicious Use in Gemini

To analyze malicious use cases in Gemini, we collected system logs, prompt submissions, and file I/O events generated during model execution using ADB Logcat. From Google Takeout, we extracted user-level artifacts, including Gemini usage history, generated or saved files, and conversation records. All experiments were conducted on a Samsung Galaxy Z Flip3 device running the Gemini 2.5 Flash model.

Gemini Artifacts Analysis. Analysis of ADB Logcat captured during prompt dialogues, file uploads, and conversation deletions revealed that Gemini internally uses the codename "Robin". Query transmission and UI rendering were processed through log tags, such as `#sendRobinQuery`, `assistant_robin_chat_history_list`, and `#getCreateTimelineSuggestion`. These tags appeared consistently during normal interactions, forming a baseline pattern for prompt submission and response rendering. In contrast, requests involving the generation of files (e.g., images or PDFs) triggered additional components, such as OpenGLRenderer and ViewRootImpl, indicating the

*Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. P-7, December 16-18, 2025, Sapporo, Japan.

rendering and output of generated content. This distinction provides insight into Gemini's internal workflow and offers a forensic basis for differentiating benign use from potential misuse. Key log tags and their corresponding behaviors are summarized in Table Table 1.

From Google Takeout, Gemini entries were found within `MyActivity.json`, where the description field contains user prompts, excerpts of Gemini's responses, and generated filenames. When a conversation includes an attached file, the entry also embeds a URL linking to the generated content. Upon conversation deletion, the associated record is removed with no residual metadata indicating the deletion event.

Related Log Data	**Description**
I/arik : #sendRobinQuery	Requests code/response generation from the generative AI (Robin)
W/View : #7f0b08ee app:id/assistant_robin_chat_history_list	Gemini app chat history display UI component ID
I/aqnf : #getCreateTimelineSuggestion	Activity timeline creation suggestion request log

Table 1: Major Log tags and Activities.

Forensic Indicators. Phishing email generation in LLM environments rarely occurs through a single explicit prompt, as direct requests for malicious output are typically filtered. Instead, attackers iteratively refine drafts through repeated revisions and re-prompts to bypass ethical safeguards. This behavior appears in Logcat as consecutive `#sendRobinQuery` entries occurring within short intervals and containing closely related content—indicating deliberate modification rather than normal conversation. When file access logs appear immediately after a response, they may further suggest that an attachment was generated or inserted. In Google Takeout, if conversation records contain text resembling an actual email subject or body, the activity can be attributed to intentional phishing composition via Gemini rather than benign use.

For malware creation, the process generally involves storing and testing LLM-generated code before external distribution. Logcat entries, such as `#sendRobinQuery`, `MediaProvider`, `ViewRootImpl@PickActivity` and `Cello`, reflect code generation and storage activities. In several cases, Google Takeout artifacts revealed Colab notebooks, Drive metadata files, and Gmail attachment records that appeared at matching timestamps, indicating that generated code was being packaged and transmitted through email or messaging platforms. If Drive metadata or Gmail attachment history appears simultaneously in Takeout, it strengthens the inference that Gemini-generated code was exfiltrated from the device.

In summary, we analyzed Logcat and Google Takeout data from mobile Gemini to identify forensic artifacts associated with malicious activities, such as phishing email composition and code generation. The Logcat traces revealed generation requests, file handling, and potential external transmissions, while Takeout provided corroborating evidence of content sharing and deletion. Together, these sources offer critical forensic indicators that enable the reconstruction and attribution of malicious behavior involving mobile LLMs.

References

[1] Shuai He, Hao Yan, Wenke Li, Sheng Hong, Xiaowei Guo, Xiaofan Liu, and Cai Fu. From large language models to adversarial malware: How far are we. In *Proceedings of the 34th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 178–182, 2025.

[2] Sayak Saha Roy, Poojitha Thota, Krishna Vamsi Naragam, and Shirin Nilizadeh. From chatbots to phishbots?–preventing phishing scams created using chatgpt, google bard and claude. *arXiv preprint arXiv:2310.19181*, 2023.