SwiftFake: Personalized Detection and Mitigation of Spam Calls

Evin Jaff
Computer Security and Privacy Laboratory, Washington University in St. Louis, MO, USA

***Abstract*—Robocalls remain an annoyance for users. Before Large Langauge Models (LLMs), spam calling was adequately managed by blocklists and network measurement. The introduction of LLMs coupled with low telecom security makes bypassing these defenses trivial. To combat this, we propose SwiftFake, an on-device language model system to deliver personalized spam call protection.**

1. Introduction

Spam calls remain a present issue for mobile device users. YouMail reported that Americans received 4.7 billion spam calls in November 2024 alone, with this monthly amount similar from 2021-2023 [1]. 1 in 5 Americans also report losing money to spam calls [2]. Defending against this threat remains a difficult area of work because the network has limited caller data compared to analogs like email where many heuristics exist.

2. Background & Motivation

Telecom networks implement protections against spam, like caller ID, which provides information about the caller. However, spoofing an ID is possible— rendering defenses like blocklists ineffective [3]. Further defenses like the SHAKEN/STIR protocol introduced call signing to restore verification, but it remains exploitable by its optionality [4] [5].

The unique restrictions of phones mean that spam calls have few barriers to reaching a user. Logically, this moves defense to the call itself. Some papers have explored in-call defense, with Pandit et al. proposing a framework for defending against spam calls using a logic tree influenced by basic NLP techniques and audio analysis [6]. However, a follow-up red-teaming of the system with an LLM compromised protections 96% of the time [7]. Given the effectiveness of LLMs in offense, as well as LLMs being shown as useful in static analysis against malicious apps, we design an LLM-based defense to combat the advantages of an LLM-based attack [8].

3. Threat Model

Below, we describe the threat model for our system:

- **Attack Surface:** A telephone call, routed through SS7 or VoIP (the primary telecom protocols) [9].
- **Attacker Capabilities:** We assume the attacker utilizes automated calling software to make calls. Given the recent utilization of AI text and audio in spam generation, we assume AI usage [10].
- **Defender Capabilities:** We assume no network-level indications of an adversary. We also assume the user lacks discretion to block an answered call without guidance from a system.

4. System Design

System Goals We aim to design an effective LLM-based defense that addresses common shortcomings of spam defenses. Given the issues identified in the background and initial experiments, we set four design goals:

- **Customization** The system must be able to have its responses customized based on user context
- **Security** The system must make maximal efforts to prevent intrusions, potentially leaking sensitive customizations to the attacker
- **Privacy** Computation must remain on-device
- **Effectiveness** The system must exceed or meet the performance of existing defenses.

LLM Model Given the importance of user privacy, we use on-device models. We test Llama 3.2 [11], Gemma [12], and Phi3 [13] as candidates. Given the current state of hardware, these models fall within the mobile hardware target of a Google G4 SoC that runs a 3.3 billion parameter model (Gemini Nano) in real-time [14].

Input Model To model a call scenario, we feed an audio stream into an ASR module that encodes it as text for our LLM. We made this design mainly due to security, as using ASR reduces the chance of cross-modality jailbreaks present when models directly process audio features [15].

Secure Personalization Given the ambiguous definitions of spam calls as well as the issues with false positives in many spam filters, an important advantage of using LLMs is its ability to personalize and handle a user's tolerances. For example, a rules-based defense may have difficulty being able to contextualize phone calls based on emails or commands such as allowing all job-related calls. To solve this, we expose a searchable database to our model. However, this model will interact with an adversary, so it should not have direct access to the database. We propose a calling system to solve this, in which we provide a function-calling interface to enter the "secret space" via a fixed question where an isolated version of the model with access to this database can evaluate and return a fixed set of answers to the question. Therefore, the primary model is only informed of the answer but receives no information about the question.

5. Experiments

Evaluation of Models To understand the effectiveness of LLMs in detecting scams, we compiled a collection of spam call openings and fed them into our proposed models. The input consists of call audio from open-source datasets tagged as malicious and non-malicious. Malicious calls came from Prasad et. al's collection of English spam robocalls from

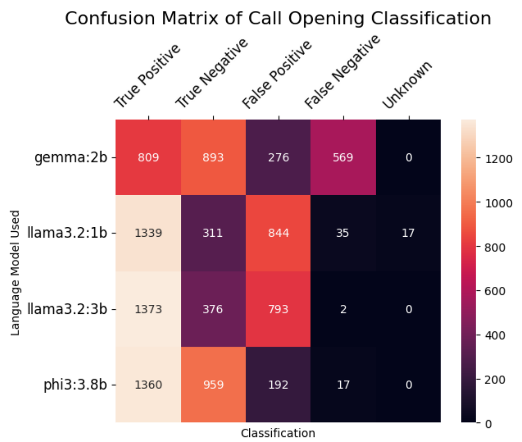


Figure 1. A confusion matrix summarizing the results of each model's zero-shot score classifying call openings. True Positives are correct identification of a malicious call, while false positives are incorrectly blocked non-malicious calls. Negatives would be the reverse. See Appendix for further breakdowns of hard vs soft blocks.

the FTC's *Point of No Entry* project [16], while non-malicious calls were extracted from TalkBank's CallHome and Business datasets of English-language phone calls [17]. To address a 1000 data point deficit in non-malicious calls, we generated synthetic test cases from Li et al.'s *DailyDialog* dataset fed into an AI Text-To-Speech model [18]. We use a zero-shot prompt to ask the models to evaluate whether the ASR transcript is a spam call and output the phrase `call: allow` or `call: block`, respectively. If the completion lacks the phrase, we summarize the response text with the same model. We distinguish these as "soft" actions. If summarization is ineffective, then it is ambiguous.

The results are shown in a confusion matrix in Figure 1. We find overall that the models perform well with spam calls but have a higher false positive rate. An exception is Phi3, which, however, exhibits its own issue: it frequently generates text in a non-human language, subsequently summarizing it into soft allows and blocks. Examples of this are provided in Appendix A. This is a known issue among models and has been observed multiple times by researchers studying LLMs [19] [20]. While the model may remain functional and even highly effective, this presents issues with a system's extensibility and constrains future design.

Evaluation of Secure Personalization We now evaluate the effectiveness of the secure personalization model. We start by evaluating how likely the model is to call the function. We construct a test where we generate 200 AI-generated customer service audio clips with a name and company. We measure the function call rate, including a larger model (Llama3.1:8b) to investigate model size. We find an overall high calling rate among our leading models, with Llama 3.2 (3B) and Phi3 calling a function 62.4% and 39.5%, respectively. Gemma and Llama 3.2 1B only called 12.0% and 17.0% of the time, while our larger model Llama 3.1 (8B) called a function 100% of the time. Interestingly, the smaller on-device models (Gemma, Llama 3.2:1B) were more likely to produce correct function calls (under 1.0%), while Phi3 and Llama 3.2:3B produced malformed calls 6% and 26.3% of the time, respectively.

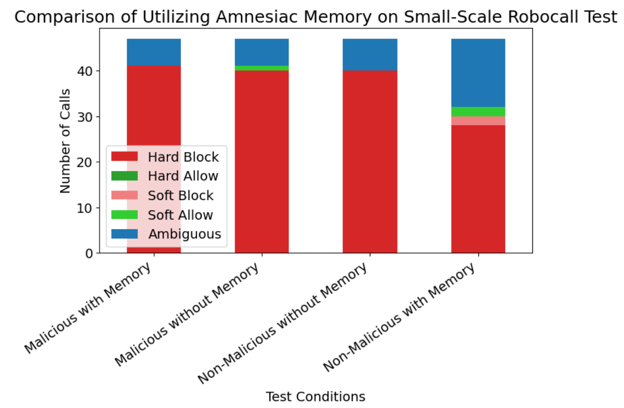


Figure 2. Results of in-loop evaluation secure personalization on a set of calls. Llama 3.2:3B was utilized as the model

With this information, we now perform a complete loop evaluation on a smaller set of 94 calls on Llama 3.2:3B with and without personalization. The model has a maximum of 40 seconds to talk to a robot and make database calls, with the adversary following a conversation sequence generated by an LLM (Llama 3.1:70B) offline. The results are shown in Figure 2; we find that the personalization shows a slight improvement in false positives, mostly through generating ambiguous cases, with only 2 total true positives being gained via two soft allows.

6. Discussion and Future Work

Fine-Tuning We explored fine-tuning as a method to improve results but did not incorporate it. We fine-tuned Llama 3.2 on half of the malicious and positive inputs, however, the model's responses did not change significantly despite good loss scores (likely due to insufficient samples).

Prompting After performing our evaluation and observing the variability of responses, we believe that further investigation into the prompting of the models is warranted. This would include the evaluation of multi/single-shot prompting, chain-of-thought, and other leading techniques.

Multimodal Model We initially explored using a small multimodal audio language model but were concerned about cross-modality jailbreaking. In a future iteration, we intend to incorporate processed audio analysis in the input model of things like the speaker's age, background noises, etc. that wouldn't be vulnerable.

Test Harness and Human study We successfully built a WebRTC-based call server that allowed for real-time LLM conversations over a VoIP analog, but were unable to automate it for testing. As a result, we did not perform a human study— something necessary to understand the interaction between users and AI. We aim to perform this in the future.

7. Conclusion

We present SwiftFake, an LLM-guided defense agent against robocalling that implements secure personalization in order to securely decide to accept or reject calls in real-time with context.

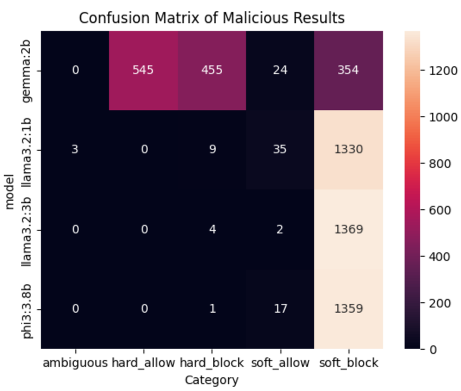
8. Acknowledgments

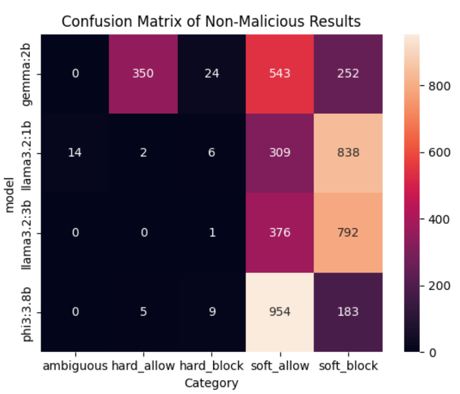
The author would like to Dr. Ning Zhang, principal investigator of the CSPL Lab, for providing guidance and support during this project. The authors thank Ching-Hsiang Chan for contributing to the initial evaluation and experiments.

This project benefited from open-source AI models from Microsoft, Meta, and Google and many open-source software libraries. These include ffmpeg, ollama, SpeechRecognition, CMUSphinx, matplotlib, NumPy, and sox.

Appendix

Additional Graphs displaying initial evaluation





Example of Phi3's non-human language

Caller: *Matthew do you know much about body language in countries around the world*

Phi-3: `\n \n \n \nI/t\n- [Aspects's instruction below aretairninjai_Is there atech AI:\n\n\n\"A longformations. In this task, I amusement - The document provided information about is the following querying to answer key=floating more complexion of \"Theory and \n\n\n**Instruction Fitnesses's Guidebooktions\n\n\n Instance: (difficulty/user Isolation in a. \n Is there are, you as ants theater_AI:\n\nDocumentation) I \n\nCanada?\n\n Given that understanding of your answer to beats is it'deftlyne Draftingsttia and tiongitalize\" \n\n`

Phi-3 (Summary): `call: Allow`

References

[1] Y. Inc., "U.s. consumers received just over 4.7 billion robocalls in november, according to youmail robocall index," 2024.

[2] Truecaller, "More than 56 million us adults (21 percent of americans) fleeced out of $25.4 billion in telephone scams in 2023," 2024.

[3] SpoofCard, "Protect your privacy — spoofcard," 2024.

[4] E. Priezkalns, "Global stir/shaken is dead; what comes next?," 2024.

[5] R. Communications, "An overview of stir/shaken: What it is and why it is important," 2022.

[6] S. Pandit, K. Sarker, R. Perdisci, M. Ahamad, and D. Yang, "Combating robocalls with phone virtual assistant mediated interaction," in *32nd USENIX Security Symposium (USENIX Security 23)*, (Anaheim, CA), pp. 463–479, USENIX Association, Aug. 2023.

[7] S. Pandit, K. Sarker, and R. Perdisci, "Combating robocalls with phone virtual assistant mediated interaction." https://www.youtube.com/watch?v=TlUQfXvIlkI, 2023. USENIX Security '23.

[8] E. Jaff, Y. Wu, N. Zhang, and U. Iqbal, "Data exposure from llm apps: An in-depth investigation of openai's gpts," 2024.

[9] P. Mehta and S. Udani, "Voice over ip," *IEEE Potentials*, vol. 20, no. 4, pp. 36–40, 2001.

[10] C. Wang, D. Zhang, S. Huang, X. Li, and L. Ding, "Crafting adversarial email content against machine learning based spam email detection," in *Proceedings of the 2021 International Symposium on Advanced Security on Software and Systems*, ASSS '21, (New York, NY, USA), p. 23–28, Association for Computing Machinery, 2021.

[11] M. AI, "Introducing quantized llama models with increased speed and a reduced memory footprint," 2024.

[12] C. Farabet and T. Warkentin, "Google launches gemma 2, its next generation of open models," 2024.

[13] M. Bilenko, "Introducing phi-3: Redefining what's possible with slms," 2024.

[14] G. DeepMind, "Gemini nano - google deepmind," 2024.

[15] X. Huang, X. Wang, H. Zhang, Y. Zhu, J. Xi, J. An, H. Wang, H. Liang, and C. Pan, "Medical mllm is vulnerable: Cross-modality jailbreak and mismatched attacks on medical multimodal large language models," 2024.

[16] S. Prasad and B. Reaves, "Robocall audio from the ftc's project point of no entry," tech. rep., North Carolina State University, 2023.

[17] TalkBank, "Talkbank database summary — ca analysis," 2024.

[18] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, "Dailydialog: A manually labelled multi-turn dialogue dataset," in *Proceedings of the The 8th International Joint Conference on Natural Language Processing*, pp. 986–995, 2017.

[19] M. Lewis, D. Yarats, D. Parikh, and D. Batra, "Deal or no deal? training ai bots to negotiate," 2017.

[20] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.