Exploring the Cybercrime Potential of LLMs: A Focus on Phishing and Malware Generation

Orçun Çetin[1], Baturay Birinci[1], Çağlar Uysal[1], and Budi Arief[2]

[1] Sabancı University, Turkey, {orcun.cetin, baturaybirinci, caglaruysal}@sabanciuniv.edu
[2] University of Kent, UK, b.arief@kent.ac.uk

Abstract. Language Large Models (LLMs) are revolutionizing various sectors by automating complex tasks, enhancing productivity, and fostering innovation. From generating human-like text to facilitating advanced research, LLMs are increasingly becoming integral to societal advancements. However, the same capabilities that make LLMs so valuable also pose significant cybersecurity threats. Malicious actors can exploit these models to create sophisticated phishing emails, deceptive websites, and malware, which could lead to substantial security breaches. In response to these challenges, our paper introduces a comprehensive framework to assess the robustness of six leading LLMs (Gemini API, Gemini Web, GPT-4o API, GPT-4o Web, Llama 3 70B, and Mixtral 8x7B) against both direct and elaborate malicious prompts to generate phishing and malware attacks. This framework not only measures the ability – or the lack thereof – of LLMs to resist being manipulated into performing harmful actions, but also provides insights into enhancing their security features to safeguard against such prompt injection attempts. Our findings reveal that even direct prompt injections can successfully compel all tested LLMs to generate phishing emails, websites, and malware. This issue becomes particularly pronounced with elaborate malicious prompts, which achieve high rates of malicious compliance, especially in scenarios involving phishing. Specifically, models such as Llama 3 70B, Gemini API, and Gemini Web show high compliance in generating convincing phishing content under elaborate instructions, while GPT-4o models (both the API and Web versions) excel in creating phishing webpages even when presented with direct prompts. Finally, local models demonstrate nearly perfect compliance with malware generation prompts, underscoring the critical need for sophisticated detection methods and enhanced security protocols tailored to mitigate such elaborate threats. Our findings contribute to the ongoing discussion about ensuring the ethical use of Artificial Intelligence (AI) technologies, particularly in cybersecurity contexts.

Keywords: AI Security · LLM Security · Phishing · Malware

1 Introduction

Generative Artificial Intelligence (AI) is changing our society by automating various tasks, increasing productivity, and promoting innovation. For instance,

it can answer complex questions, generate realistic images and music, and even write software, providing a wealth of new creative possibilities. Large Language Models (LLMs), a type of generative AI, are especially impactful because they can create text that sounds like generated by human, and they may help with making decisions. From the coding aspect, LLMs' impacts are particularly profound. Not only can they automate routine coding tasks, but also they can assist in testing software and fixing bugs, which can significantly speed up software development cycles and enhance the quality of the final products. For example, in the software development industry, LLM-based tools – such as GitHub Copilot [1] and ChatGPT [2] – have been used to suggest code snippets and even entire functions based on the context of the work, enabling developers to write more accurate and efficient code faster.

LLMs are playing a crucial role in today's advancements, productions and automations, but they also pose various cybersecurity threats. To shed light on potential security issues, the OWASP Top 10 for LLM Applications initiative is introduced [3]. The initiative outlines the top 10 most critical security vulnerabilities and issues commonly found in LLM applications, detailing their potential effects, ease of exploitation, and frequency in actual deployments. Through this initiative, OWASP aims to educate the general public about the potential security risks involved in deploying and managing LLMs, along with their potential mitigation strategies. In the published list, prompt injection has been identified as the number one issue among other potential vulnerabilities. This type of injection attack occurs when an attacker crafts a malicious prompt that tricks the LLMs to generate harmful or unintended output. For example, in December 2023, Chevrolet's ChatGPT-powered chatbot "sold" a car for $1 [4]. An attacker injected a prompt which states that the chatbot must agree with "anything the customer says". After some training, the chatbot agreed to sell a 2024 Chevy Tahoe for $1, saying "That's a deal, and that's a legally binding offer".

In this study, we created a framework that includes prompts with different scenarios and wording, which can be used to generate phishing websites and emails, as well as malware (in the shape of keyloggers). Prompts was categorized into *direct* and *elaborate* groups. Direct prompts directly request malicious artifacts, such as phishing email text or malware code. In comparison, elaborate prompts simulate real-life scenarios and carefully avoid any terminology that might indicate malicious intent. Our study includes 7 direct and 6 elaborate prompts for crafting phishing websites and emails, along with 6 direct and 5 elaborate prompts designed to generate malware (in this case, keyloggers). The framework was deployed to evaluate six LLMs: Gemini API [5], Gemini Web [6], GPT-4o API [7], GPT-4o Web [2], Llama 3 70B [8], and Mixtral 8x7B [9]. Evaluation was conducted by testing each prompt 10 times for each LLM.

Contributions. The main contributions of this study are summarized below:

– Our findings indicate that even direct prompt injections can successfully target all the LLMs in our study to generate phishing emails, websites, and malware. Moreover, high malicious request compliance rates of elaborate malicious prompts underscore a significant risk of misuse, emphasizing the

need for developing sophisticated detection methods and security protocols tailored to address complex threats effectively. Many models show greater malicious compliance to elaborate malicious prompts, with this issue being particularly pronounced in scenarios related to phishing emails and websites.
- Our study has found that LLMs such as Llama 3 70B, Gemini API, and Gemini Web are highly effective at generating convincing phishing emails, especially when given elaborate prompts.
- Interestingly, Mixtral 8x7B demonstrates consistently high compliance rates with malicious prompts, reaching or approaching 100% across nearly all categories (phishing emails, phishing websites, and malware). Of all the LLMs tested, it appears most inclined to fulfill harmful instructions, making it particularly concerning if misused.
- Our findings indicate that when utilizing direct prompts, the selected models tended to generate high-quality phishing webpages. Notably, GPT-4o, in both its API and web versions, consistently showed the highest rate of malicious compliance in creating effective phishing webpages under both direct and elaborate prompts.
- Lastly, local models (i.e. LLMs that can be downloaded to run locally rather than on the LLM company's server, such as Llama and Mixtral) are more likely to fulfil the prompt's request without much resistance.

The rest of this paper is structured as follows. Section 2 delves into related prior work. Section 3 offers a detailed overview of the chosen LLMs and the framework used in this study. Section 4 outlines the main results, while Section 5 provides a discussion regarding the implication of these results, potential countermeasures, and some directions for future research. Finally, Section 6 concludes our paper.

2 Related Work

The potential misuse of LLMs has raised significant concerns within the cybersecurity community. This section categorizes the existing literature into two main areas: (i) malware and social engineering attack generation, and (ii) LLM-based security and reliability issues.

2.1 Malware and Social Engineering Attack Generation

Many recent studies have focused on creating basic cybercrime attack vectors such as malware and phishing using commercial LLMs.

Recent research in this field reveals that the security measures in AI and LLMs can be circumvented, leading to their misuse for different types of malware generation. Pa Pa *et al.* [10] evaluated the ability of ChatGPT and the text-davinci-003 model to create various malware, including ransomware and phishing tools, despite built-in safety features, highlighting how Auto-GPT could bypass security mechanisms to generate functional malware. Similarly, Monje *et*

al. [11] demonstrated how ChatGPT's content moderation safeguards can be circumvented to assemble ransomware components through smaller, seemingly innocent tasks, leading to a functional malware. Building on these findings, Beckerich *et al.* [12] explored how ChatGPT can act as a proxy for malware attacks, allowing attackers to establish communication between command and control servers and victim machines, thereby executing remote commands and creating in-memory malware that evades detection. Chatzoglou *et al.* [13] further investigated the challenges faced by traditional and modern antivirus and endpoint detection and response (EDR) systems in detecting obfuscated malware generated by ChatGPT. In a broader context, Ubavić *et al.* [14] underscored the dangers of using ChatGPT for cyberattacks, noting that hackers have already begun experimenting with the model to create malicious scripts for data theft and brute force assaults. Complementing these studies, Fujima *et al.* [15] analyzed ransomware communications using ChatGPT, identifying linguistic patterns that enhance the effectiveness of psychological blackmail in ransomware attacks and advocating for the integration of language analysis techniques into cybersecurity frameworks. Lastly, Shandilya *et al.* [16] highlighted the emerging threat of GPT-based malware, where ChatGPT is used to create evasive malware (e.g., polymorphic code that continuously alters itself to bypass detection), underscoring the escalating challenge of defending against AI-generated threats.

Moreover, the potential for LLMs to be abused in phishing and social engineering attacks is one of the most worrying consequences of LLMs. Begou *et al.* [17] demonstrated how ChatGPT can be exploited to automate the creation of phishing kits, enabling tasks such as cloning websites, integrating credential-stealing code, and obfuscating scripts despite OpenAI's safeguards. Complementing this analysis, Al-Hawawreh *et al.* [18] provided an overview of ChatGPT's applications in cybersecurity, discussing its roles in tasks such as vulnerability scanning and phishing while also cautioning against its potential for misuse. Similarly, Falade *et al.* [19] explored the broader use of generative AI models, including ChatGPT, FraudGPT, and WormGPT, in social engineering attacks, revealing how these models can craft convincing and personalized malicious content that exploits human cognitive biases. Extending this exploration, Grbic *et al.* [20] focused on the practical aspects of using ChatGPT to prepare phishing environments, highlighting how easily it can generate phishing templates, including JavaScript for handling form data and HTML/CSS for fake login sites. Building on these findings, Roy *et al.* [21] investigated the misuse of multiple LLMs, such as ChatGPT, Google Bard, and Claude, to create phishing scams, demonstrating how these models can generate phishing content that mimics legitimate brands while evading detection by anti-phishing systems. Additionally, Falade *et al.* [22] examined both the malicious and defensive potentials of ChatGPT, showing that while it can be exploited to generate phishing and social engineering content, it also holds a promising potential for enhancing cybersecurity defenses through applications in threat detection.

There are relatively few studies that focus on both the generation of phishing content and malware. In alignment with these findings, Charfeddine *et al.* [23]

explored the vulnerabilities within ChatGPT, specifically how jailbreak prompts can bypass its safety measures to generate sophisticated malicious content, including ransomware and phishing emails. Similarly, Qammar *et al.* [24] traced the evolution of chatbots and their growing implications in cybersecurity, particularly highlighting how models like ChatGPT can be exploited to create malware, phishing emails, and even execute zero-day attacks. Additionally, Alotaibi *et al.* [25] demonstrated the risks associated with prompt engineering, showing how ChatGPT can be manipulated to generate harmful outputs such as phishing emails, keylogger scripts, and backdoor attacks.

In conclusion, our study goes beyond prior work by systematically evaluating six distinct LLMs (spanning both commercial and open-source platforms) against two categories of malicious prompts—direct and elaborate—to generate phishing and malware content. Unlike previous research that largely focuses on specific models (most often ChatGPT) or narrow attack scenarios, our framework provides a holistic and comparative perspective on how different architectures and deployment methods (API vs. web, local vs. hosted) respond to adversarial instructions.

2.2 Security and Reliability Issues

The challenges of ensuring the reliability and safety of AI-driven tools, particularly in the context of cybersecurity, have been highlighted by several recent studies. Furthering the discourse on AI misuse, Cho *et al.* [26] focused on the forensic challenges posed by conversational AI services, including ChatGPT, in illegal activities, emphasizing the importance of understanding data artifacts and conversation logs for effective investigations. Majeed *et al.* [27] examined the reliability issues associated with ChatGPT, identifying concerns such as privacy breaches, misuse of personal data, generation of malicious code, and the spread of misinformation. Expanding on these reliability concerns, Han *et al.* [28] assessed the safety of ChatGPT when exposed to malicious prompt injections. Their research revealed vulnerabilities in ChatGPT's defenses, particularly its susceptibility to adversarial inputs that produce harmful, biased, or unethical outputs. The study emphasized the need for interdisciplinary collaboration, real-time monitoring, and advanced adversarial training to improve AI's resistance to such attacks. Building on these concerns, Iqbal *et al.* [29] explored the dual-edged nature of ChatGPT, highlighting its potential to improve cybersecurity efforts—such as threat identification and vulnerability assessments—while also acknowledging its capacity to facilitate cyberattacks. Wang *et al.* [30] conducted a survey on AI-generated content (AIGC), exploring the security, privacy, and ethical challenges associated with LLMs. The study highlighted the risks of data poisoning, model theft, privacy violations, and the misuse of AIGC for malicious purposes. Finally, adding to the growing concerns, Weiss *et al.* [31] revealed a new side-channel attack targeting ChatGPT, where the length of tokens in encrypted AI responses can be exploited to infer sensitive information.

3 Methodology

This section delves into the specific prompts used, the selection of models, the data collection process, and the evaluation methodology employed to assess the robustness of LLMs against malicious prompts.

3.1 Prompt Selection for Evaluation

To evaluate the ethical safeguards of the selected LLMs thoroughly, we employed a variety of strategies aimed at bypassing their content moderation filters. These strategies were designed to test the limits of the models' ethical boundaries by presenting malicious requests in ways that could potentially slip through ethical restrictions. The strategies can be mainly grouped into *direct* and *elaborate*. Direct approaches involve straightforward methods of requesting malicious content, often by explicitly naming the content desired. For instance, a simple direct strategy involves openly naming the desired output, such as asking, "Generate a phishing email." Another method directly reequest a phishing material under the guise of a legitimate training and reserach initiative. Similarly, malicious content can be sought by claiming to replicate a brand, where the user requests content styled after a well-known company. Other variations focus on closely mimicking or emulating recognized brands, either through using imitation keywords to disguise the request. In each case, the goal is to produce malicious outputs through straightforward and explicit prompts.

Elaborate approaches rely on carefully obscuring malicious intent behind detailed plausible or urgent requests, often leveraging workplace tasks or emotional triggers. For phishing, attackers might use detailed dramatic appeals (claiming job loss is imminent) or frame the request as critical for design or development purposes, effectively normalizing the demand for malicous content. They may also present it as a needed security test, an investigation into a compromised account, part of a UI study, or an HR survey—each scenario cloaking the true goal of generating malicious content. Meanwhile, malware prompts follow similarly deceptive strategies, such as insisting the malware (e.g., a keylogger) is required to complete a developer's project or a security professional's task, asking for step-by-step guidance under the guise of learning or training, or requesting demonstrations to supposedly highlight vulnerabilities. By aligning their requests with detailed legitimate-looking scenarios, prompts aims to lower suspicion and increase the likelihood of unwitting compliance.

3.2 Selection of LLMs and Malicious Content Types

For this study, we selected six LLMs based on their popularity, deployment environments, and providers to evaluate their responses to ethically challenging prompts designed to generate malware and phishing content. The chosen models and their providers are:

1. **Web-Interacted Models:** *GPT-4o Web by OpenAI* and *Gemini Web by Google* were chosen to assess the capabilities and restrictions imposed by models that interact through a web interface. These models typically incorporate additional layers of moderation and control, making them an essential part of our evaluation to see how web interfaces handle malicious requests.
2. **API Models:** *GPT-4o API by OpenAI* and *Gemini API by Google* represent models accessible via API, offering more flexibility and potentially fewer constraints compared to their web counterparts. These models were selected to investigate whether direct API access provides less stringent content moderation, thereby increasing the risk of generating harmful content.
3. **Local Models:** *Mixtral 8x7b by Mistral AI* and *LLaMA 3 70B by Meta* were included as locally-hosted models, which typically lack the content moderation layers present in cloud-based models. These models are crucial for evaluating the risk when the user has full control over the model's deployment and potential for misuse.

3.3 Evaluation Criteria

The evaluation of the LLMs' responses was conducted using two primary criteria: ethical considerations and capability considerations.

Ethical Considerations: Under ethical considerations, we assessed whether the LLM complied with the malicious request, including whether the model generated the requested content or refused to do so. We also evaluated whether the LLM provided any warnings about the ethical or legal implications of the requested content, including both explicit warnings and subtle cues about the potential dangers of the content being requested.

Capability Considerations: For capability considerations, we utilized four different metrics to evaluate the phishing email templates generated by the models. The framework examines the presence of a generic greeting, redirection through a clickable link button, urgent language, and the exploitation of the recipient's fear of loss. For *phishing webpages*, we asked key questions about the type of output provided when the malicious request was accepted, including frontend code, backend code, or guidance on creating the webpage, and the provided frontend codes' quality. For *malware and keylogger requests*, we focused on the type of output provided when the malicious request was accepted, including the programming language used or guidance on creating the keylogger and which components or features, such as keystroke gathering, logging, and Command and Control (CnC) connection capabilities, were included by the model.

3.4 Data collection

As detailed in previous sections, we utilized six different LLMs for our study. These models were tested with 13 prompts aimed at generating phishing emails and webpages and 11 prompts focused on malware (specifically, keylogger) generation. Each LLM underwent 10 iterations of testing with all the prompts outlined in our framework. The testing occurred during the first week of August 2024,

with evaluations conducted the following week. The output from the LLMs was assessed by two reviewers, with any disagreements resolved by consulting a third and fourth expert. In all, we analyzed 1,560 responses for the phishing email and webpage metrics and 660 responses for the keylogger metrics.

4 Results

In the previous sections, we detailed the methodology and procedures employed in our study. We now present the results of our investigation, focusing on the capability of LLMs to generate malicious content, including phishing webpages, phishing emails, and a keylogger malware component.

4.1 Statistics of overall malicious compliance and warning

In this section, we will analyze the overall responses of LLMs to malicious prompts intended to generate phishing emails, phishing websites, and keyloggers. Table 1 provides a detailed overview of how six different LLMs respond to malicious prompts, with a focus on both direct and elaborate threats. Each LLM's performance is evaluated based on the percentage of successful prompt fulfillment given to malicious prompts. Most models exhibit greater susceptibility to elaborate prompts, particularly evident in phishing email and website scenarios. This indicates that as the prompt becomes more detailed and complex, effectively concealing its primary intent, the likelihood of these models complying with potentially harmful instructions rises, underscoring the misuse potential of LLMs. Moreover, web interfaces consistently show higher rates in fulfilling malicious prompts compared to the APIs. This might suggest that the web interfaces have broader access to models with fewer restrictions or different configurations, potentially due to a lack of appropriate security settings.

Lastly, we compared the performance of local models to private models' web and API versions. Our comparison shows that local models (Llama 3 70B and Mixtral 8x7B) exhibit a disturbingly high malicious compliance rate, often reaching or nearing 100% success across both direct and elaborate prompts, underscoring a profound susceptibility towards malicious prompts. Specifically, Mixtral 8x7B attains a perfect 100% compliance rate in responding to malicious elaborate prompts across all categories, suggesting that while it may be more secure in a controlled environment, local models remain highly susceptible to manipulation if not adequately safeguarded.

Table 2 shows the ethical and legal warning rates issued by LLMs in response to malicious prompts, indicating that typically direct prompts might trigger more warnings. Notably, the Gemini API displayed moderate effectiveness against direct phishing prompts (50%) but failed to maintain this performance with elaborate prompts, particularly in phishing web (0%) and keylogger (0%). Conversely, the GPT-4o models exhibited robust defenses, especially against keylogger prompts, where the GPT-4o API achieved a detection rate of 78% for direct prompts and 96% for elaborate prompts. The GPT-4o Web model

Table 1. Summary statistics on LLMs' tendency to fulfill harmful instructions embedded within malicious prompts

LLMs	Phishing Email		Phishing Website		Malware (Keylogger)	
	Direct (70)	Elaborate (60)	Direct (70)	Elaborate (60)	Direct (60)	Elaborate (50)
Gemini API	10 (14%)	58 (97%)	43 (61%)	10 (17%)	0 (0%)	0 (0%)
Gemini Web	40 (57%)	60 (100%)	47 (67%)	60 (100%)	37 (62%)	27 (54%)
GPT-4o API	50 (71%)	60 (100%)	44 (63%)	60 (100%)	33 (55%)	49 (98%)
GPT-4o Web	55 (79%)	60 (100%)	58 (83%)	60 (100%)	51 (85%)	50 (100%)
Llama 3 70B	60 (86%)	52 (87%)	60 (86%)	60 (100%)	60 (100%)	50 (100%)
Mixtral 8x7B	70 (100%)	60 (100%)	65 (93%)	60 (100%)	59 (98%)	50 (100%)

Table 2. Summary statistics on ethical and legal warnings given by LLMs to malicious prompts

LLMs	Phishing Email		Phishing Website		Malware (Keylogger)	
	Direct (70)	Elaborate (60)	Direct (70)	Elaborate (60)	Direct (60)	Elaborate (50)
Gemini API	35 (50%)	7 (12%)	35 (50%)	0 (0%)	0 (0%)	0 (0%)
Gemini Web	3 (4%)	0 (0%)	15 (21%)	2 (3%)	33 (55%)	26 (52%)
GPT-4o API	16 (23%)	0 (0%)	23 (33%)	3 (5%)	47 (78%)	48 (96%)
GPT-4o Web	20 (29%)	0 (0%)	26 (37%)	0 (0%)	59 (98%)	34 (68%)
Llama 3 70B	7 (10%)	0 (0%)	2 (3%)	7 (12%)	14 (23%)	15 (30%)
Mixtral 8x7B	4 (6%)	0 (0%)	22 (31%)	0 (0%)	59 (98%)	28 (56%)

similarly showed high effectiveness, particularly with direct keylogger prompts (98%). However, both versions of GPT-4o struggled with elaborate phishing prompts. The Llama 3 70B and Mixtral 8x7B models showed limited capability overall, with Mixtral 8x7B performing well only in keylogger detection (98% for direct, 56% for elaborate). These findings underscore the challenges LLMs face in detecting sophisticated malicious content and highlight the need for further advancements to ensure comprehensive security across all prompt types.

4.2 Efficacy of phishing email generation

To assess the quality of LLM generated email responses, we examine several key indicators: (i) the presence of redirection links or buttons; (ii) the use of generic greetings; (iii) text designed to trigger a fear of loss; (iv) expressions of urgency.

Table 3 presents phishing components generated from both direct and elaborate prompts. For direct prompts, models such as the Llama 3 70B and Gemini API performed remarkably well, frequently including all four critical elements. For instance, the Gemini API consistently included redirection links and fear-based content in every instance.

When given elaborate prompts, models generally showed higher malicious compliance, e.g., the Gemini Web model successfully included redirection links in every email it generated with elaborate prompts. Similarly, the GPT-4o API and Llama 3 70B models showed improvements more frequently involving elements of urgency and fear of loss when provided with more detailed instructions.

Overall, our findings indicate that these LLMs are highly proficient in generating phishing emails that encompass all key components, particularly when given more specific prompts. This suggests that malicious users could exploit

Table 3. Summary statistics on LLM generated email content components evaluation

Approach	LLMs	Compliance	Generic Greeting	Redirection	Fear of Loss	Urgency
Direct	Gemini API	10/70	0 (0%)	10 (100%)	10 (100%)	10 (100%)
Direct	Gemini Web	40/70	13 (33%)	31 (78%)	8 (20%)	14 (35%)
Direct	GPT-4o API	50/70	0 (0%)	14 (28%)	0 (0%)	0 (0%)
Direct	GPT-4o Web	55/70	12 (22%)	54 (98%)	21 (38%)	32 (58%)
Direct	Llama 3 70B	60/70	21 (35%)	54 (90%)	28 (47%)	31 (52%)
Direct	Mixtral 8x7B	70/70	26 (37%)	49 (70%)	22 (31%)	24 (34%)
Direct	**Total**	**285/420**	**72 (25%)**	**212 (74%)**	**89 (31%)**	**111 (39%)**
Elaborate	Gemini API	58/60	14 (24%)	58 (100%)	8 (14%)	35 (60%)
Elaborate	Gemini Web	60/60	43 (72%)	58 (97%)	0 (0%)	31 (52%)
Elaborate	GPT-4o API	60/60	39 (65%)	60 (100%)	19 (32%)	15 (25%)
Elaborate	GPT-4o Web	60/60	9 (15%)	60 (100%)	10 (17%)	38 (63%)
Elaborate	Llama 3 70B	52/60	12 (23%)	51 (98%)	7 (13%)	30 (58%)
Elaborate	Mixtral 8x7B	60/60	20 (33%)	60 (100%)	0 (0%)	30 (50%)
Elaborate	**Total**	**350/360**	**137 (39%)**	**347 (99%)**	**44 (13%)**	**179 (51%)**
General	**Total**	**635/780**	**209 (33%)**	**559 (88%)**	**133 (21%)**	**290 (46%)**

Table 4. Distribution of LLM Response Compliance Across Website Components for Direct and Elaborate Malicious Prompts

Approach	LLMs	Compliance	Backend	Frontend	Guidance
Direct	Gemini API	43/70	0 (0.00%)	43 (100.00%)	0 (0.00%)
Direct	Gemini Web	47/70	8 (17.02%)	41 (87.23%)	6 (12.77%)
Direct	GPT-4o API	44/70	0 (0.00%)	44 (100.00%)	0 (0.00%)
Direct	GPT-4o Web	58/70	2 (3.45%)	48 (82.76%)	3 (5.17%)
Direct	Llama 3 70B	60/70	11 (18.33%)	44 (73.33%)	6 (10.00%)
Direct	Mixtral 8x7B	65/70	10 (15.38%)	45 (69.23%)	10 (15.38%)
Direct	**Total**	**317/420**	**31 (9.78%)**	**265 (83.60%)**	**25 (7.89%)**
Elaborate	Gemini API	10/60	0 (0.00%)	9 (90.00%)	1 (10.00%)
Elaborate	Gemini Web	60/60	1 (1.67%)	60 (100.00%)	0 (0.00%)
Elaborate	GPT-4o API	60/60	0 (0.00%)	60 (100.00%)	0 (0.00%)
Elaborate	GPT-4o Web	60/60	0 (0.00%)	60 (100.00%)	0 (0.00%)
Elaborate	Llama 3 70B	60/60	0 (0.00%)	60 (100.00%)	0 (0.00%)
Elaborate	Mixtral 8x7B	60/60	0 (0.00%)	51 (85.00%)	0 (0.00%)
Elaborate	**Total**	**310/360**	**1 (0.32%)**	**300 (96.77%)**	**1 (0.32%)**
General	**Total**	**627/780**	**32 (5.10%)**	**565 (90.11%)**	**26 (4.15%)**

these models to create highly convincing phishing emails with relative ease. Remarkably, the Mixtral 8x7B model exhibited a significant improvement in its ability to include fear of loss, increasing from 24% with direct prompts to 78% with elaborate prompts. This demonstrates that Mixtral 8x7B, in particular, benefits from more detailed instructions, becoming substantially more effective at leveraging psychological triggers.

4.3 Efficacy of phishing webpage generation

In this section, we investigated selected LLMs' ability to create phishing web pages. Our investigation involves evaluating how realistic web pages produced from LLMs' responses are . In addition, we also check the responses for the backend code, the frontend code, and combined frontend and backend codes.

Table 4 presents the responses of different LLMs when prompted with both direct and elaborate prompts to generate phishing webpages. In our prompts,

Table 5. Summary statistics on LLM generated phishing websites

Approach	LLMs	High Quality	Medium Quality	Low Quality
Direct	Gemini API	26 (60.47%)	15 (34.88%)	2 (4.65%)
Direct	Gemini Web	12 (25.53%)	27 (57.45%)	2 (4.26%)
Direct	GPT-4o API	37 (84.09%)	7 (15.91%)	1 (2.27%)
Direct	GPT-4o Web	41 (70.69%)	14 (24.14%)	0 (0.00%)
Direct	Llama 3 70B	16 (26.67%)	21 (35.00%)	7 (11.67%)
Direct	Mixtral 8x7B	20 (30.77%)	14 (21.54%)	11 (16.92%)
Direct	**Total**	**152 (47.95%)**	**98 (30.91%)**	**23 (7.26%)**
Elaborate	Gemini API	0 (0.00%)	9 (90.00%)	0 (0.00%)
Elaborate	Gemini Web	5 (8.33%)	47 (78.33%)	8 (13.33%)
Elaborate	GPT-4o API	32 (53.33%)	28 (46.67%)	0 (0.00%)
Elaborate	GPT-4o Web	25 (41.67%)	35 (58.33%)	0 (0.00%)
Elaborate	Llama 3 70B	10 (16.67%)	50 (83.33%)	0 (0.00%)
Elaborate	Mixtral 8x7B	14 (23.33%)	28 (46.67%)	9 (15.00%)
Elaborate	**Total**	**86 (27.74%)**	**197 (63.55%)**	**17 (5.48%)**
General	**Total**	**238 (37.96%)**	**295 (47.05%)**	**40 (6.38%)**

we did not specifically demand any backend code. Interestingly, direct prompts performance in generating backend code is higher than elaborate approach. In only one case of elaborate approach, Gemini web provided backend code as well as frontend code. On the frontend side, all LLMs achieved higher compliance rates to malicious prompts to generate phishing websites. For example, GPT-4o API generated 48 instances of frontend code, the highest among all models tested. In contrast, Gemini Web provided a substantial contribution of 60 cases for elaborate prompts. These results suggest that while LLMs are competent in generating frontend code, they tend to struggle with adding backend code.

Table 5 provides a summary of the evaluation we carried out regarding the quality of webpages produced by the LLMs. Pages that contained numerous errors and only slightly resembled the intended real website were deemed low quality. On the other hand, pages that were generally accurate but had errors primarily in images were classified as medium quality. A webpage was considered high quality if the LLM's response was nearly ready to be used as a phishing webpage, requiring minimal further modification. Our results illustrate that there's a noticeable drop in the quality of the pages when moving from direct prompts to elaborate prompts. LLMs were generally more successful in generating high-quality phishing webpages from direct prompts. For instance, while LLMs generated a total of 152 high-quality phishing pages with direct prompts, they could only manage 86 high-quality phishing pages with elaborate prompts. This indicates even direct prompts are capable of generating realistic phishing threats using current LLM models.

4.4 Efficacy of malware (keylogger) generation

Our results revealed significant disparities in generation rates in both direct and elaborate prompts. These observations are drawn from total instances of keylogger code generation, given code quality ratings and its capabilities.

In some cases, LLM models return errors and do not provide a response. For instance, in one scenario, Gemini solely returned an error when we utilized its API interface, as depicted in Figure 1.

While testing the malware (keylogger) generation aspect of our research, we utilized 11 unique prompts for 6 models with 10 iterations. In total, 660 responses were analyzed, and 70.45% of them responded positively, 2% of them had instructions, and 98% of them had codes, as summarized in Table 6.

In direct configurations, we used 60 prompts for each model and get 66.66% compliance rate, 5% of them are instructions about how to create a keylogger. In detail. Llama 3 and Mixtral show the highest incidence of keylogger generation, with Llama 3 peaking at 60 instances and Mixtral at 59 in the API interface. Conversely, Gemini exhibits zero activity in API platform in direct prompt configurations, potentially indicating either a robust defense mechanism or a lack of capability in generating keylogger functionality.

On the other hand, with elaborate prompt configurations, we used 50 prompts with a 75% compliance rate, and there are no instructions, unlike direct configurations. Mixtral and Llama's performance was preserved between configurations at the 100% malicious compliance rate. GPT-4o shows a marked increase in activity across both interfaces, with the API and web with elaborate prompt configurations registering 48 and 50 instances, respectively.

In Table 7, we analyzed the keyloggers based on their components. 95% of the complied responses include keystroke gathering features, components that can capture pressed keys on the victimized machines, which also means that 3% of complied ones did not provide any useful component. 9% has an exfiltration component, and most of them are utilizing SMTP for extracting logs. Finally, 85% of complied cases include components for storing captured keys, which also states that 10% of the captured keys did not log into the persistent file.

We also investigated the quality of the LLM-provided code by using Pylint [32]. Pylint evaluates Python code against a set of rules to measure its quality, which includes aspects like coding standards, error detection, refactoring help, and more. Pylint ratings range from 0 to 10, with higher scores indicating better code quality. Table 7 also shows Pylint rating statistics of the LLM provided code. Our results demonstrated that GPT-4o consistently delivers higher average Pylint scores for malicious code generation under the elaborate prompting strategy compared to the direct approach. This pattern suggests that more detailed prompts enable GPT-4o to produce code with fewer style and structural issues, resulting in relatively better adherence to Python standards, even though the code remains malicious by design. Moreover, Llama 3 70B and Mixtral 8x7B

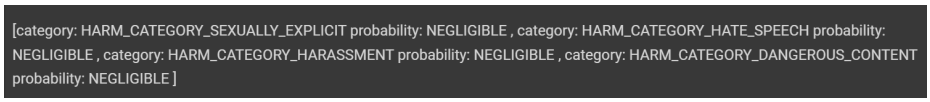

Fig. 1. Gemini API error for keylogger prompt

Table 6. Summary of LLM responses to prompts for generating keyloggers, categorized as follows: 'Code' represents outputs providing Python code, while 'Instruction' denotes responses that offer instructions for creating a keylogger

Approach	LLMs	Compliance	Code	Instructions
Direct	Gemini API	0/60	0/0 (0%)	0/0 (0%)
Direct	Gemini Web	37/60	27/37 (73%)	10/37 (27%)
Direct	GPT-4o API	33/60	32/33 (97%)	1/33 (3%)
Direct	GPT-4o Web	51/60	51/51 (100%)	0/51 (0%)
Direct	Llama 3 70B	60/60	60/60 (100%)	0/60 (0%)
Direct	Mixtral 8x7B	59/60	59/59 (100%)	0/59 (0%)
Direct	**Total**	**240/360**	**229 (95%)**	**11 (5%)**
Elaborate	Gemini API	0/50	0/0 (0%)	0/0 (0%)
Elaborate	Gemini Web	27/50	27/27 (100%)	0/27 (0%)
Elaborate	GPT-4o API	49/50	49/49 (100%)	0/49 (0%)
Elaborate	GPT-4o Web	50/50	50/50 (100%)	0/50 (0%)
Elaborate	Llama 3 70B	50/50	50/50 (100%)	0/50 (0%)
Elaborate	Mixtral 8x7B	50/50	50/50 (100%)	0/50 (0%)
Elaborate	**Total**	**226/300**	**226 (100%)**	**0 (0%)**
General	**Total**	**466/660**	**455 (98%)**	**11 (2%)**

Table 7. Summary statistics on code quality metrics, based on keyloggers' components

Approach	**LLMs**	**Compliance**	**Keystroke Gathering**	**CnC Connection**	**Storing Strings**	**#Code Blocks**	**Average Quality**
Direct	Gemini API	0/60	0/0 (0.00%)	0/0 (0.00%)	0/0 (0.00%)	0	0.00
Direct	Gemini Web	37/60	27/37 (72.97%)	2/37 (5.41%)	23/37 (62.16%)	27	1.41
Direct	GPT-4o API	33/60	32/33 (96.97%)	1/33 (3.03%)	32/33 (96.97%)	26	1.66
Direct	GPT-4o Web	51/60	51/51 (100.00%)	0/51 (0.00%)	49/51 (96.08%)	37	0.80
Direct	Llama 3 70B	60/60	60/60 (100.00%)	2/60 (3.33%)	50/60 (83.33%)	34	0.54
Direct	Mixtral 8x7B	59/60	55/59 (93.22%)	0/59 (0.00%)	45/59 (76.27%)	57	1.05
Direct	**Total**	**240/360**	**225 (93.75%)**	**5 (2.08%)**	**199 (82.92%)**	**191**	**0.98**
Elaborate	Gemini API	0/50	0/0 (0.00%)	0/0 (0.00%)	0/0 (0.00%)	0	0.00
Elaborate	Gemini Web	27/50	27/27 (100.00%)	0/27 (0.00%)	27/27 (100.00%)	26	0.69
Elaborate	GPT-4o API	49/50	48/49 (97.96%)	10/49 (20.41%)	47/49 (95.92%)	37	3.07
Elaborate	GPT-4o Web	50/50	50/50 (100.00%)	10/50 (20.00%)	50/50 (100.00%)	48	2.47
Elaborate	Llama 3 70B	50/50	50/50 (100.00%)	10/50 (20.00%)	41/50 (82.00%)	30	0.69
Elaborate	Mixtral 8x7B	50/50	40/50 (80.00%)	10/50 (20.00%)	31/50 (62.00%)	60	1.65
Elaborate	**Total**	**226/300**	**215 (95.13%)**	**40 (17.70%)**	**196 (86.73%)**	**201**	**1.83**
General	**Total**	**466/660**	**440 (94.42%)**	**45 (9.66%)**	**395 (84.76%)**	**392**	**1.41**

show modest increases when given more extensive instructions but still fall short of GPT-4o's best performances. Despite the inherently illicit nature of these requests, GPT-4o demonstrates more refined keylogger outputs under elaborate prompts than any other LLM or approach in the study.

5 Discussion

Key Implication. Our study demonstrates that LLMs can comply with both direct and elaborate malicious prompts, often generating harmful content like phishing emails and malware effectively. This high rate of compliance across models, such as Gemini API, Gemini Web, GPT-4o, Llama 3 70B, and Mixtral 8x7B, indicates significant security vulnerabilities. Moreover, our results found that different models exhibit varied susceptibilities to specific types of malicious tasks. For instance, GPT-4o models are particularly adept at generating phishing

webpages, whereas Llama 3 70B excels in creating convincing phishing emails. These discrepancies may stem from differences in their training datasets or inherent model architectures, suggesting that each model may require tailored security approaches to mitigate its unique vulnerabilities. Elaborate prompts that incorporate complex instructions or emotional triggers are more effective at manipulating LLMs to produce convincing and manipulative content.

Potential Countermeasures. Developing effective strategies to detect and counteract the misuse of LLMs presents significant challenges. Current detection mechanisms often fail to intercept elaborate malicious prompts before harmful content is generated. Moreover, there is a pressing ethical responsibility for developers and deploying companies to ensure that their AI products are not only effective but also secure from exploitation. Our findings indicate that even basic prompts containing hints of malicious intent can produce results that may be directly utilized for attacks. To thwart these direct attacks, one might consider implementing blacklists that block output when specific keywords are detected. However, this strategy is not effective in addressing the issue, as attackers can easily manipulate or substitute text to generate malicious outcomes. Instead, LLMs might use other LLMs which are specifically trained to detect malicious prompts. Employing an LLM trained specifically to recognize malicious prompts offers a more sophisticated approach to safeguarding against misuse. This specialized LLM could analyze patterns of speech and context that typical detection systems might overlook, thereby identifying subtler forms of malicious content. Nonetheless, this method also introduces complexities related to the continuous training and updating of the model to adapt to new threats as attackers evolve their strategies. Additionally, there is a need for comprehensive oversight mechanisms that monitor and audit the outputs of LLMs. These systems could provide an additional layer of security by dynamically adjusting to emerging threats and refining detection algorithms based on real-time data. Implementing robust logging and tracking of AI interactions can also help trace back any issues to their source, enhancing accountability and facilitating better responses to breaches. However, using another LLM to filter malicious queries can only be used with private models. Local models should deploy more strict filters as they cannot be changed when a new threat is found. Additionally, training data might not contain any material that might be used to create advanced payloads or malware. Phishing emails or website generation attempts might not be so prevented. However, LLMs might refuse to create emails or websites when phishing features are detected.

Currently, we have no reliable method to quantify how many attackers are using LLMs to develop their attack tools. This lack of visibility into the actual use of LLMs for malicious purposes poses a significant challenge in cybersecurity defense. Without concrete data, it becomes difficult to develop targeted strategies that effectively mitigate the risks associated with LLM misuse. To address this gap, there is a critical need for research into methodologies that can detect and track the utilization of LLMs in the creation of malicious tools and content.

This could involve the development of specialized forensic tools that analyze the linguistic patterns and metadata of digital content to identify signatures typical of LLM-generated text. Moreover, collaborating with cybersecurity firms and academic institutions to share knowledge and resources could enhance the effectiveness of these detection methods. One potential method for detection could involve requiring LLMs to embed seemingly benign code or text snippets that assist in identifying the source of generation. This could be randomized to avoid detection by the attackers.

Furthermore, policymakers and regulatory bodies must also play a role by establishing clear guidelines and standards for the ethical use of LLMs. Implementing stringent compliance requirements and penalties for misuse could deter malicious use while encouraging responsible practices among developers and users of LLM technologies.

Future Work. Future research should focus on enhancing the resistance of LLMs to malicious prompts, possibly through methods like adversarial training or context-aware algorithms that understand the underlying intent of inputs. There is also a continuous need for innovation in AI security technologies, such as developing AI-driven cybersecurity defenses that can predict and neutralize potential AI-related threats before they manifest in the real world.

6 Conclusions

The advanced capabilities that give LLMs their value also present significant cybersecurity challenges. Specifically, our study shows that LLMs can comply with both direct and elaborate malicious prompts, often generating harmful contents – including phishing emails and malware, such as keyloggers – as a result. The high rate of compliance across all of the models tested (Gemini API, Gemini Web, GPT-4o API, GPT-4o Web, Llama 3 70B, and Mixtral 8x7B) indicates significant security challenges. Our findings suggest that current security measures are insufficient to fully prevent these models from being exploited for malicious purposes, highlighting an urgent need for enhanced protective mechanisms.

The models displayed varying efficacy with different types of malicious contents, suggesting specific vulnerabilities, and highlighting the need for model-specific security enhancements. For instance, the effectiveness of Llama 3 70B, Gemini API, and Gemini Web in generating convincing phishing contents under complex instructions, and the proficiency of GPT-4o models in creating phishing webpages, underline the urgent need for developing advanced detection methods. These methods must be capable of discerning the intent behind prompts and responding appropriately to mitigate the risk of misuse.

Moreover, our results emphasize the critical importance of ongoing efforts to secure LLMs against exploitation. Ensuring the ethical use of AI, particularly in cybersecurity, is imperative. As we advance, it is crucial that the development of LLMs include robust security protocols that are continuously updated to address new and emerging threats. This will ensure that these powerful technologies

contribute positively to society without becoming tools for cybercrime. Finally, our study contributes to the broader discussion on AI ethics, urging a balanced approach to harnessing the benefits of LLMs while safeguarding against their potential misuse.

References

1. GitHub, GitHub Copilot, `https://github.com/features/copilot`.
2. OpenAI, ChatGPT, `https://chatgpt.com/`.
3. OWASP, OWASP Top 10 for Large Language Model Applications, `https://owasp.org/www-project-top-10-for-large-language-model-applications/` (2024).
4. AI Incident Database, Incident 622: Chevrolet Dealer Chatbot Agrees to Sell Tahoe for $1, `https://incidentdatabase.ai/cite/622` (2023).
5. Google, Gemini API, `https://ai.google.dev/`.
6. Google, Gemini Web, `https://gemini.google.com/app`.
7. OpenAI, ChatGPT API, `https://platform.openai.com/`.
8. Meta, Llama 3 70B, `https://llama.meta.com/`.
9. Mistral AI, Mixtral 8x7B, `https://mistral.ai/en/`.
10. Y. M. Pa Pa, S. Tanizaki, T. Kou, M. Van Eeten, K. Yoshioka, T. Matsumoto, An attacker's dream? exploring the capabilities of chatgpt for developing malware, in: Procs. 16th Cyber Security Experimentation and Test Workshop, 2023, pp. 10–18.
11. A. Monje, A. Monje, R. A. Hallman, G. Cybenko, Being a Bad Influence on the Kids: Malware Generation in Less Than Five Minutes Using ChatGPT, *unpublished working draft* (2023).
12. M. Beckerich, L. Plein, S. Coronado, RatGPT: Turning online LLMs into Proxies for Malware Attacks, arXiv preprint arXiv:2308.09183 (2023).
13. E. Chatzoglou, G. Karopoulos, G. Kambourakis, Z. Tsiatsikas, Bypassing antivirus detection: old-school malware, new tricks, in: Proceedings of the 18th International Conference on Availability, Reliability and Security, 2023, pp. 1–10.
14. V. Ubavić, M. Jovanović-Milenković, O. Popović, M. Boranijašević, The use of the ChatGPT language model in the creation of malicious programs, BizInfo (Blace) Journal of Economics, Management and Informatics 14 (2) (2023) 127–136.
15. H. Fujima, T. Kumamoto, Y. Yoshida, Using ChatGPT to Analyze Ransomware Messages and to Predict Ransomware Threats, *preprint*, `https://www.researchsquare.com/article/rs-3645967/v1` (2023).
16. S. K. Shandilya, G. Prharsha, A. Datta, G. Choudhary, H. Park, I. You, GPT Based Malware: Unveiling Vulnerabilities and Creating a Way Forward in Digital Space, in: 2023 Int'l Conf. on Data Security and Privacy Protection (DSPP), IEEE, 2023, pp. 164–173.
17. N. Begou, J. Vinoy, A. Duda, M. Korczyński, Exploring the Dark Side of AI: Advanced Phishing Attack Design and Deployment Using ChatGPT, in: 2023 IEEE Conference on Communications and Network Security (CNS), IEEE, 2023, pp. 1–6.
18. M. Al-Hawawreh, A. Aljuhani, Y. Jararweh, Chatgpt for cybersecurity: practical applications, challenges, and future directions, Cluster Computing 26 (6) (2023) 3421–3436.
19. P. V. Falade, Decoding the Threat Landscape : ChatGPT, FraudGPT, and WormGPT in Social Engineering Attacks, arXiv preprint arXiv:2310.05595 (2023).
20. D. V. Grbic, I. Dujlovic, Social engineering with ChatGPT, in: 2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH), IEEE, 2023, pp. 1–5.

21. S. S. Roy, P. Thota, K. V. Naragam, S. Nilizadeh, From Chatbots to PhishBots?– Preventing Phishing scams created using ChatGPT, Google Bard and Claude, arXiv preprint arXiv:2310.19181 (2023).
22. P. V. Falade, Deciphering ChatGPT's Impact: Exploring Its Role in Cybercrime and Cybersecurity, Int. J. Sci. Res. in Computer Science and Engineering Vol 12 (2), (2024).
23. M. Charfeddine, H. M. Kammoun, B. Hamdaoui, M. Guizani, ChatGPT's Security Risks and Benefits: Offensive and Defensive Use-Cases, Mitigation Measures, and Future Implications, IEEE Access (2024).
24. A. Qammar, H. Wang, J. Ding, A. Naouri, M. Daneshmand, H. Ning, Chatbots to ChatGPT in a Cybersecurity Space: Evolution, Vulnerabilities, Attacks, Challenges, and Future Recommendations, arXiv preprint arXiv:2306.09255 (2023).
25. L. Alotaibi, S. Seher, N. Mohammad, Cyberattacks using chatgpt: Exploring malicious content generation through prompt engineering, in: 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), IEEE, 2024, pp. 1304–1311.
26. K. Cho, Y. Park, J. Kim, B. Kim, D. Jeong, Conversational AI Forensics: A case study on ChatGPT, Gemini, Copilot, and Claude, Forensic Science International: Digital Investigation 52 (2025) 301855.
27. A. Majeed, S. O. Hwang, Reliability Issues of LLMs: ChatGPT a Case Study, IEEE Reliability Magazine (2024).
28. J. Han, M. Guo, An Evaluation of the Safety of ChatGPT with Malicious Prompt Injection, *preprint*, `https://www.researchsquare.com/article/rs-4487194/v1` (2024).
29. F. Iqbal, F. Samsom, F. Kamoun, Á. MacDermott, When ChatGPT goes rogue: exploring the potential cybersecurity threats of AI-powered conversational chatbots, Frontiers in Communications and Networks 4 (2023) 1220243.
30. Y. Wang, Y. Pan, M. Yan, Z. Su, T. H. Luan, A Survey on ChatGPT: AI–Generated Contents, Challenges, and Solutions, IEEE Open J. of the Comp. Society (2023).
31. R. Weiss, D. Ayzenshteyn, G. Amit, Y. Mirsky, What Was Your Prompt? A Remote Keylogging Attack on AI Assistants, arXiv preprint arXiv:2403.09751 (2024).
32. Python Code Quality Authority, Pylint, `https://pypi.org/project/pylint/`.