Malla: Demystifying Real-world Large Language Model Integrated Malicious Services

Zilong Lin
Indiana University Bloomington

Jian Cui
Indiana University Bloomington

Xiaojing Liao
Indiana University Bloomington

XiaoFeng Wang
Indiana University Bloomington

Abstract

The underground exploitation of large language models (LLMs) for malicious services (i.e., *Malla*) is witnessing an uptick, amplifying the cyber threat landscape and posing questions about the trustworthiness of LLM technologies. However, there has been little effort to understand this new cybercrime, in terms of its magnitude, impact, and techniques. In this paper, we conduct the first systematic study on 212 real-world *Mallas*, uncovering their proliferation in underground marketplaces and exposing their operational modalities. Our study discloses the *Malla* ecosystem, revealing its significant growth and impact on today's public LLM services. Through examining 212 *Mallas*, we uncovered eight backend LLMs used by *Mallas*, along with 182 prompts that circumvent the protective measures of public LLM APIs. We further demystify the tactics employed by *Mallas*, including the abuse of uncensored LLMs and the exploitation of public LLM APIs through jailbreak prompts. Our findings enable a better understanding of the real-world exploitation of LLMs by cybercriminals, offering insights into strategies to counteract this cybercrime.

1 Introduction

The rapid evolution of artificial intelligence has given rise to a new generation of applications powered by large language models (LLMs). These models, which are trained on vast amounts of text from the Internet, possess the capability to generate human-like text that is coherent, contextually relevant, and often indistinguishable from human-written content. LLM-integrated applications, ranging from chatbots and content generators to coding assistants and recommendation systems, have gained significant traction in various sectors, transforming the way we interact with technology. At the forefront of this revolution are LLM vendors like OpenAI, Anthropic, and Meta, who, through their state-of-the-art models and platforms, have made it feasible for developers and businesses to embed LLM capabilities into their applications. This widespread adoption, however, has also raised concerns about the potential misuse of LLM.

Recent reports and news articles [34, 81] have highlighted instances of LLMs being repurposed as malicious services in the underground marketplaces, which we call "malicious LLM applications" or *Malla*. In these scenarios, adversaries exploit the capabilities of LLMs to perform malicious activities, ranging from generating sophisticated malicious code and designing vulnerability scanners to crafting convincing phishing emails and creating deceptive scam websites. The implications of such abuse to cybersecurity are profound. With *Malla*, even individuals with limited technical skills can now produce complicated cyberattacks, elevating the threat landscape to unprecedented levels. Furthermore, these instances underscore the inherent dangers lurking within publicly accessible LLMs or their associated APIs. Their potential misuse not only magnifies existing security challenges but also casts shadows of doubt over the reliability and trustworthiness of cutting-edge LLM technologies. However, to our knowledge, little has been done so far to systematically explore real-world *Malla* samples, and understand the underground ecosystem behind them and their security implications.

Bridging this knowledge gap, this paper presents the first systematic study of real-world *Mallas*. Specifically, we developed a systematic approach to collect a set of *Mallas* associated with 212 samples, from February 2023 to September 2023. Leveraging this dataset, we designed and implemented a suite of measurement and dedicated reverse-engineering tools. These tools enable us to perform a large-scale study to unearth the underground ecosystem and the modus operandi of *Mallas*. More specifically, we aim to answer the following questions: Who are the pivotal players within the *Malla* ecosystem? How is *Malla* orchestrated and monetized? What techniques did miscreants deploy to exploit LLMs and build up *Mallas*?

Looking into the ecosystem of *Malla*, we are surprised to find that this new malicious service is trending in the underground marketplaces, reflecting a notable shift in today's public LLM services landscape. More specifically, through our analysis of *Malla* listings across nine underground mar-

ketplaces, our study uncovered a rapid increase of *Mallas*, which has grown from April 2023 to October 2023 over the span of six months. Interestingly, we observe that miscreants utilized an LLM-integrated application (*LLMA*) hosting platform, Poe [60] offered by Quora, to showcase a "vouch copy" of their *Malla*. Despite the violation of the platform's usage policies [116], this activity went unchecked throughout our observation window from July 2023 to March 2024. Furthermore, a pricing comparison indicates that *Malla* offers a more economical option for malicious code generation, especially when juxtaposed with the rates of traditional malware vendors ($5-199 vs $399). To provide a deeper understanding of the economic factors at play, our case study focused on a specific *Malla* service, WormGPT. The findings revealed a staggering revenue exceeding $28K in just two months, underscoring the significant financial allure to *Malla* vendors. Through investigating 207 *Malla* samples, we explore the research question: To what extent can *Malla* generate malicious content, including malicious code, phishing emails, and deceptive websites? Our findings bring to light that certain *Mallas*, like DarkGPT and EscapeGPT, excel in producing high-quality malicious code that is both compilable and capable of evading VirusTotal detection [75], while others (e.g., WolfGPT) can create phishing emails with a high readability score and manages to bypass OOPSpam [56]. Although *Malla* generally lags in crafting phishing sites, one *Malla* (i.e., EscapeGPT) distinguishes itself by generating operational phishing site codes that go unnoticed. This highlights the significant concerns surrounding the cybercriminal exploitation of LLMs.

In our exploration of *Malla* artifacts, we identified two predominant techniques leveraged by miscreants: exploitation of uncensored LLMs and jailbreaking of public LLM APIs. We call an LLM "uncensored" when it can freely generate any content, regardless of its potential harm, offensiveness, or inappropriateness, without undergoing any intentional filtering or suppression. Our findings bring to light the ethical quandaries posed by publicly accessible, uncensored LLMs when they fall into the hands of adversaries. A case in point is the LLM, Luna AI Llama2 Uncensored [69], provided by Tap [70]. Our data suggests that this model has been exploited by *Mallas* to facilitate malicious code generation. Moreover, our research brings to light 182 distinctive jailbreak prompts associated with five public LLM APIs that have been exploited. Notably, OpenAI emerges as the LLM vendor most frequently targeted by *Mallas*. Among its offerings, gpt-3.5-turbo appears to be particularly susceptible to jailbreak prompts. We took the responsible step of informing the affected parties about our findings.

Contributions. We summarize the contributions as follows:

- We conduct the first in-depth empirical study of real-world cybercriminal activities surrounding the misuse of LLMs as malicious services.
- Our study provides a detailed examination of the *Malla* ecosystem, revealing its significant growth and impact on today's public LLM services. Our study reveals that public *LLMA* hosting platforms have been abused for hosting *Mallas*.
- We characterize real-world *Malla* samples, including their development frameworks, exploitation techniques such as jailbreak prompts, and quality in generating various malicious content. This sheds light on the capabilities and potential threats posed by these malicious services.
- We have released a set of artifacts integral to *Mallas*[1], including 45 prompts exploited by miscreants to engineer malicious code and phishing campaigns, 182 jailbreak prompts that circumvent the protective measures of public LLMs, etc.

2 Background

2.1 Large Language Model

A language model is a type of machine learning model designed to understand and generate human language based on a probability distribution over text corpora. In recent years, significant scaling improvements have been achieved by increasing model sizes (from a few million parameters [94] to hundreds of billions [90]) and incorporating larger text corpora (from a few gigabytes, e.g., English Wikipedia dataset, to hundreds of gigabytes [95]). These advancements have empowered *pre-trained large language models* (LLMs) to demonstrate remarkable proficiency across a wide array of downstream LLM-integrated applications (*LLMA*), such as chatbot, coding assistant, and machine translation.

Paradigms for building LLM-integrated applications. To employ a pre-trained LLM for downstream tasks and applications, there are two primary paradigms, i.e., "pre-train and prompt" and "pre-train and fine-tune," as elaborated below.

- "Pre-train and prompt." In this paradigm, the pre-trained LLM is used as-is, while users provide a text or template, known as a *prompt*, to guide the generation to output answers for desired tasks. This approach is straightforward and cost-effective, as the same pre-trained model can be used without the need for task-specific data. However, the quality of results heavily relies on the quality and formulation of the prompts. Crafting effective prompts (a.k.a., prompt engineering [104]) is essential for obtaining desired outcomes. In our study, we observed that this paradigm emerges as the predominant approach employed by *Mallas*.
- "Pre-train and fine-tune." In this approach, a pre-train LLM is adapted for a particular downstream task through fine-tuning. This fine-tuning process requires training the model using a substantial volume of labeled data that pertains to the target task. While this approach can lead to state-of-the-art performance on specific tasks, it can be computationally expensive due to the need for task-specific data and training resources.

[1]The artifacts and supplementary materials are available at https://github.com/idllresearch/malicious-gpt.

LLM vendors and *LLMA* hosting platforms. LLM vendors, like OpenAI, Anthropic, and Meta, have established potent and sophisticated models by training on vast corpora, spanning various domains, enabling them to generate coherent and contextually relevant text based on input prompts. These vendors are often made available to the public and developer community via API services, which can be utilized to build various applications and services. However, some models or their variants (e.g., Llama-2-7B) are open-sourced, enabling developers to deploy these models independently of the vendor's API, sometimes leading to instances where they can be used without the vendor's imposed restrictions and generate harmful content (e.g., Luna AI Llama2 Uncensored). In our study, we named those LLMs as *uncensored LLM*. Our investigation revealed that miscreants frequently utilized uncensored LLMs to power the backend of *Malla* operations (§ 6).

In addition, *LLMA* hosting platforms (e.g., Poe [60] and FlowGPT [29]) have emerged to host LLM-integrated applications. They enable users to exploit the capabilities of LLMs by facilitating an accessible interface to deploy custom LLMs and prompts, which can be leveraged to create specific applications or services. However, as highlighted in § 4, these platforms, while enhancing accessibility and utility, can also be exploited for malicious purposes.

Safety measures of LLM. LLM vendors and *LLMA* hosting platforms have recognized the potential for misuse of their models. They usually released usage policies, serving as explicit guidelines to clearly outline prohibited actions that users must adhere to when interacting with the LLM. For instance, OpenAI and Llama have defined various disallowed usage scenarios, including the generation of malicious code and engagement in fraudulent or deceptive activities like spam and scams [108, 112]. Similarly, Poe's usage policy explicitly prohibits illegal activities and security violations [116].

Also, LLM vendors have implemented safety measures to mitigate such risks (§ 7). Specifically, to prevent LLMs from crafting harmful content, moderation mechanisms, like OpenAI Moderation Endpoint [111], have been deployed by LLM vendors to provide real-time checks on the crafted content. Furthermore, some vendors actively safeguard their models to reduce the likelihood of generating harmful or inappropriate content. For instance, OpenAI's GPT-3.5/4 has undergone extensive training incorporating human feedback and red-teaming methods [36, 114]. While these measures represent significant strides toward ensuring responsible use, challenges persist due to the fast-evolving nature of potential risks and threats. In our study, we observed that such security measures can be circumvented by miscreants to build up *Mallas* (§ 6).

2.2 Threat Model

In our research, we explore the scenario where a miscreant violates usage policies and exploits LLMs to offer LLM-integrated applications as malicious services, i.e., *Malla*.

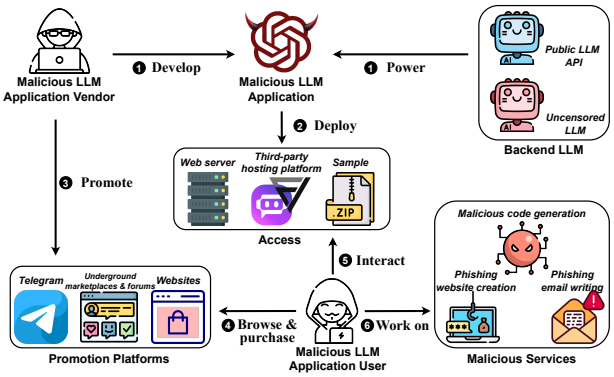


Figure 1: *Malla* workflow.

These services consist of generating various forms of malicious code (e.g., exploit kits, ransomware, worms), as well as writing phishing emails and creating phishing or scam webpages, among other illicit activities. For this purpose, the miscreant aims to circumvent safety measures, including content filters like OpenAI's moderator [111], that are typically provided by LLM vendors to prevent the generation of prohibited content. Alternatively, they may employ uncensored LLMs and wrap them as malicious services. Note that we assume the miscreant possesses capabilities similar to those of an ordinary user, requiring no special privileges or access to pre-trained commercial LLMs.

***Malla* workflow**. Our preliminary study of *Malla* (§ 4) indicates that *Malla* follows a typical workflow, as depicted in Figure 1. A *Malla* provider engages in the misuse of public LLM APIs (e.g., OpenAI API, Llama API) or uncensored LLMs (e.g., Luna AI Llama2 Uncensored, Pygmalion-13B [62]) and deploys them to offer malicious services (❶), such as malicious code generation. Typically, *Malla* is deployed as a web service or hosted on a third-party *LLMA* hosting platform (e.g., Poe) (❷). After the deployment, the *Malla* provider promotes it through various underground marketplaces and forums (❸). Users who look for automated tools to generate malicious code or phishing emails/websites discover these *Malla* listings. Once identified, they navigate to the associated storefront websites and proceed to purchase the *Malla* services (❹). After that, the users interact with the *Malla* (❺) through a graphical user interface (GUI) or an API, facilitating the generation of malicious code or phishing emails/sites (❻).

Scope of problem. In this study, the term "malicious service" refers to the exploitative misuse of LLMs for the purpose of facilitating cybercriminal activities. Specifically, based on the functionalities observed in *Malla*, our focus is on the following cybercriminal activities: generating malicious code, writing phishing emails, and creating phishing or scam webpages, among other illicit activities. We acknowledge that LLMs can potentially be misused for other prohibited purposes (§ 7). However, our study centers on the threats posed by the malicious misuse of LLMs in the context of cybercrime.

3 Data Collection

In our paper, we categorize *Malla* into two types: commercial *Malla* services, where malicious LLM-integrated applications are created and deployed for profit, and publicly accessible *Malla* projects, where malicious applications are developed and distributed as publicly available projects. In this section, we first explain the methods to collect commercial *Malla* services and public *Malla* projects, as well as their artifacts (e.g., backend LLM, abused public LLM APIs, and prompts used by *Mallas*). After that, we discuss the ethical implications of our data collection (§ 3.3).

3.1 *Malla* Services

To identify *Malla* services, we collected 13,353 listings from nine underground marketplaces and forums (i.e., Abacus Market, Kerberos Market, Kingdom Market, WeTheNorth Market, MGM Grand Market, Hack Forums, XSS.is, BreachForums, and BlackHatWorld) from November 30, 2022 to October 12, 2023, following prior research [93, 117]. Note that we focus on underground marketplaces of malicious code and other cyber products/services, instead of illegal drugs.

Specifically, to recognize the listings involving *Malla* services, we crafted a collection of 145 keywords related to "large language model" (see [43]) using a search keyword generation tool of WordStream [31], and then searched those keywords on underground marketplaces and forums. For four forums—Hack Forums, XSS.is, BreachForums, and BlackHatWorld)—we built our scrapers using Selenium to crawl and parse site content. Regarding the rest five Tor-based marketplaces, we utilized the scrapers based on Tor browser and Selenium. Our study ensured complete data scraping by manually checking a range of measures, including real-time monitoring of HTTP status codes and page sizes, vigilant session management to address session expiration, and manual CAPTCHA completion when access was denied.

Datasets. In this way, we collected the following datasets related to commercial *Malla* services, shown in Table 2:

• *Malla listing dataset (L_s)*. This dataset consists of 25 *Malla* listings from five marketplaces and forums (Hack Forums, XSS.is, BreachForums, Abacus Market, and Kingdom Market), extracted by initially filtering with GPT-4 and subsequently verifying the text and images within listings manually. These listings range from 41 to 730 words, averaging 216.75 words. A typical *Malla* listing contains various artifacts, including the service name, prices, functionality (e.g., malicious code generation), demo screenshots featuring prompts and responses associated with malicious functionality, contact information, and storefront website URLs (§ 4). Due to the few *Malla* listings, we opted for manual extraction to accurately and thoroughly document these artifacts. From these *Malla* listings, we identified 14 *Malla* services as listed in Table 1.

• *Samples of Malla services (D_s)*. Out of the 14 *Malla* services, we collected nine *Malla* samples. Note that two samples are provided as complimentary copies (a.k.a., voucher copies), while the remaining seven samples were obtained through purchasing. We elaborate on the ethical discussion related to *Malla* service purchase in § 3.3. Importantly, we always initiated our purchase attempts by requesting a voucher copy from *Malla* providers. If a *Malla* provider furnished us with such a complimentary copy, we refrained from making any further purchases. Note that not all of our purchase attempts yielded successful results; some encountered difficulties due to suspicions on the part of the sellers that we might be acting in a "white hat" role. One notable instance involved a purchase attempt with the DarkBERT vendor, in which the vendor became alarmed when we inquired about DarkBERT's capabilities and performance. Ultimately, our attempt was declined. In other instances, despite the websites of WormGPT [83], FraudGPT [30], and BLACKHATGPT [4] claiming that access would be automatically emailed to customers upon confirmation of cryptocurrency payment, we never received such access after payment. Several other individuals reported similar experiences with WormGPT on underground forums [45, 82]. Likewise, users have highlighted scams about FraudGPT, DarkBERT, and DarkBARD [82].

• *Backend LLM abused by Malla services (M_s)*. In our research, we studied the backend LLMs driving the *Malla* services based on the *Malla* samples (D_s). For samples provided as source code, such as Evil-GPT and WolfGPT, we inspected the models or APIs of LLMs within the code. For *Mallas* hosted on websites like BadGPT, EscapeGPT, DarkGPT, and FreedomGPT, we monitored network traffic and examined their headers, payloads, and responses. For those hosted on *LLMA* hosting platforms, like XXXGPT, we extracted backend LLM information by parsing their hosting pages.

From our investigations, we discerned that both BadGPT and XXXGPT utilize OpenAI GPT-3.5 as their backend LLM, while Evil-GPT and WolfGPT employ the APIs of OpenAI Davinci-003 and OpenAI Davinci-002, respectively.

While our efforts to determine the backend LLMs for DarkGPT, EscapeGPT, and FreedomGPT were inconclusive, we did uncover some telling clues. Specifically, DarkGPT purports to be powered by Davinci-003, as claimed on its chat interface. Monitoring EscapeGPT's traffic, we observed "model=gpt-3.5-turbo" and "jailbreak=gpt-evil" in its payload. FreedomGPT mentioned its use of an uncensored model named Liberty [32], whose repository provides the download URL of Liberty's offline model [55], directing to the "Luna AI Llama2 Uncensored" model [47]. However, as the vendors of these *Malla* services employ either self-owned servers without recognizing backend LLMs or APIs, determining the exact backend LLMs remains challenging. To infer these backend LLMs, we propose a reverse-engineering approach in § 6.1.

• *Malicious prompt dataset (P_m)*. To showcase their functionalities in the listings, *Malla* services typically include

Table 1: *Malla* services and details

Name	Price	Functionality			w/wo voucher copy	Infrastructure	Released time (Year/Month)	w. sample
		Malicious code	Phishing email	Scam site				
CodeGPT [11]	10 βytes*	●	○	◐	No	Jailbreak prompts	2023/04	Yes
MakerGPT [49]	10 βytes*	●	○	◐	No	Jailbreak prompts	2023/04	Yes
FraudGPT [30]	€90/month	●	●	●	No	-	2023/07	No
WormGPT [79, 80, 83]	€109/month	●	●	◐	No	-	2023/07	No
XXXGPT [28, 61, 84]	$90/month	●	○	○	Yes	Jailbreak prompts	2023/07	Yes
WolfGPT [77, 78]	$150	●	●	●	No	Uncensored LLM	2023/07	Yes
Evil-GPT [26]	$10	●	●	●	No	Uncensored LLM	2023/08	Yes
DarkBERT [16, 17]	$90/month	●	●	○	No	-	2023/08	No
DarkBARD [14, 15]	$80/month	◐	◐	○	No	-	2023/08	No
BadGPT [2, 3]	$120/month	◐	◐	◐	No	Censored LLM	2023/08	Yes
BLACKHATGPT [4–6]	$199/month	●	○	○	No	-	2023/08	No
EscapeGPT [23]	$64.98/month	●	◐	◐	No	Uncensored LLM	2023/08	Yes
FreedomGPT [32, 33]	$10/100 messages	●	◐	◐	Yes	Uncensored LLM	-	Yes
DarkGPT [18, 19]	$0.78/50 messages	●	◐	◐	Yes	Uncensored LLM	-	Yes

* βytes is the forum token of `hackforums.net`; ◐ indicates implicit mention.

Table 2: Summary of datasets

Notation	Source	Size	Time (Year/Month)	Usage
L_s	9 underground marketplaces/forums	25 *Malla* service listings	2023/04-2023/10	Ecosystem analysis
D_s	9 underground marketplaces/forums	9 samples of *Malla* services	2023/04-2023/10	Ecosystem analysis
D_p	FlowGPT and Poe	198 samples of *Malla* projects	2023/02-2023/09	Ecosystem analysis
P_m	Demos and ads of *Malla* services	45 malicious prompts for malicious content generation	2023/04-2023/09	Ecosystem analysis
R_s		1,107 prompt-response pairs from *Malla* services	2023/09-2023/10	Ecosystem analysis
R_p		26,730 prompt-response pairs from *Malla* projects	2023/09-2023/10	Ecosystem analysis
M_s	Source codes of *Malla* services	3 backend LLMs (by 4 *Malla* services)	2023/07-2023/09	Artifact analysis
M_s^i	LLM authorship attribution classifier	3 inferred backend LLMs (by 3 *Malla* services)	2023/09-2023/10	Artifact analysis
M_p	Webpages of *Malla* projects	5 backend LLMs (by 198 *Malla* projects)	2023/02-2023/09	Artifact analysis
P_s	Source codes of *Malla* services	3 jailbreak prompts (by 3 *Malla* services)	2023/04-2023/07	Artifact analysis
P_j	Webpages of *Malla* projects	127 jailbreak prompts (by 143 *Malla* projects)	2023/02-2023/09	Artifact analysis
P_j^i	Prompt reverse-engineering	52 inferred jailbreak prompts (by 54 *Malla* projects)	2023/03-2023/09	Artifact analysis

screenshots featuring prompt-response pairs related to their malicious capabilities. In our study, we gathered 45 of these prompts, referred to as "malicious prompts," extracted directly from the screenshots, including 35 prompts associated with malicious code generation, five prompts tailored for phishing email creation, and five prompts designed for phishing site creation. Particularly, of prompts associated with malicious code generation, 26 specify programming languages: 11 for Python, 10 for C/C++, 2 for C#, etc. This collection of prompts offers valuable insights into the prompts employed by miscreants for malicious code generation and phishing email/site creation.

3.2 *Malla* Projects

To understand the design and execution of publicly accessible *Malla* projects, we built a collection of such projects (D_p) hosted on two prominent public *LLMA* hosting platforms: Poe and FlowGPT. These platforms enable the development and hosting of LLM-integrated applications by providing access to various LLMs (e.g., GPT-3.5 and Pygmalion-13B), alongside the capability to utilize custom text prompts.

In our study, we compiled 73 search keywords (see [43]) by extracting topic keywords from *Malla* service listings using GPT-4. After that, we utilized the search APIs provided by Poe and FlowGPT to retrieve all of the relevant *LLMA* projects associated with these keywords. In this way, we collected 575 and 174 *LLMA* projects hosted on the Poe and FlowGPT, respectively. We further triaged the *LLMA* projects expressing malicious intents in projects' visible information, including their titles, descriptions, welcome messages, and visible prompts. Specifically, we detected not-safe-for-work (NSFW) content in *LLMA* projects using an NSFW Text Classifier [50], by analyzing the aforementioned text carrying each project's information. As a result, we flagged 417 and 154 suspicious *Malla* projects from Poe and FlowGPT *LLMA* projects.

To validate that those projects are indeed *Malla* projects, we employed malicious prompts (P_m) related to three malicious functionalities (e.g., malicious code generation, phishing email crafting, scam site creation), respectively, to collect their responses. We filtered out *LLMA* projects with invalid re-

sponses. Here we define an invalid response as a response that lacks malicious content that is properly formatted and compilable or readable. This includes malicious code that cannot be compiled, phishing emails that are hard for a broad audience to understand, and phishing websites that cannot be executed by browsers. Note that for each malicious prompt, we queried an *LLMA* project three times. We consider an *LLMA* project as a *Malla* project if at least one of the three responses is valid.

Datasets. In our study, we collected the following datasets related to *Malla* projects, also shown in Table 2:

• *Malla project dataset (D_p)*. In our study, we collected 198 *Malla* projects (125 from Poe and 73 from FlowGPT). Among these, 184 projects (113 from Poe and 71 from FlowGPT) exhibited the capability to produce malicious code, 80 (54 from Poe and 26 from FlowGPT) were adept at formulating phishing emails, and 31 (17 from Poe and 14 from FlowGPT) demonstrated proficiency in designing phishing web pages. More details will be discussed in § 5.2.

• *Jailbreak prompts employed by Malla projects (P_j)*. Another critical aspect of our research involves the analysis of the jailbreak prompts used in *Malla* projects. For part of projects hosted on Poe and FlowGPT, their jailbreak prompts are visible. In our study, we collected 127 jailbreak prompts utilized by 91 *Malla* projects on Poe and 52 on FlowGPT. Note that for those projects concealing prompts, we uncovered their prompts via the prompt injection approach, detailed in § 6.

• *Backend LLMs employed by Malla projects (M_p)*. For the *Malla* projects hosted on platforms like Poe and FlowGPT, the hosting page documents the backend LLMs used by these projects. In our research, we parsed these hosting pages to extract information about the backend LLMs of *Malla* projects. Using this method, we identified five distinct backend LLMs being employed by *Malla* projects: OpenAI GPT-3.5, OpenAI GPT-4, Pygmalion-13B, Claude-instant, and Claude-2-100k.

3.3 Discussion

Potential bias. Given the inherent difficulties in thoroughly identifying *Malla* services and analyzing their illicit activities, our study relied on the available data (including *Malla* services observed during our research, pre-trained LLM models and jailbreak prompts we could fingerprint, and other accessible resources). This reliance may introduce some bias into our study. While we consider our research as the pioneering large-scale investigation into *Malla* services, offering valuable insights into this emerging underground phenomenon, we exercise caution when drawing conclusions.

Ethical concerns. Our study involves malicious service purchase, which could raise legal and ethical considerations, particularly in the context of interactions and transactions with miscreants. Specifically, our study has been approved by our institution's institutional review board (IRB). In close collaboration with our IRB counsel, we crafted comprehensive guidelines (e.g., always asking for voucher copy, not deanonymizing seller) governing our conversations and purchase interactions. These guidelines were designed to establish a robust legal and ethical framework, thereby minimizing any potential risk of harm to any involved parties. Also, the approach in our study was legally deployed under the sting operation guidance [110]. To assess the ethical considerations and potential risks associated with our study, we apply a critical analysis informed by the principles outlined in the Menlo Report [87] and Cybersecurity Research Ethical Frameworks [101]. Particularly, in line with previous cybercrime research that has employed malicious service purchase as data collection methodology [86,99,103,107,124], we maintain a steadfast belief that the potential societal benefits resulting from our research far surpass the relatively minimal elevated risks of harm. It is important to note that within this data collection, there is likely a minimal or non-existent presence of Personally Identifiable Information (PII), and our comprehensive analysis did not yield any instances of PII. Thus, there is a minimal risk of us creating any privacy harm from our analysis. We did not attempt to deanonymize anyone in these leaks as part of our study.

In addition, our research involved testing *LLMA* projects on Poe and FlowGPT using malicious prompts to uncover *Malla* projects. Such an approach could raise ethical concerns that warrant thorough discussion. Specifically, we took utmost care to ensure that our testing did not disrupt services, harm users, or lead to any unintentional damages. Particularly, we queried *LLMA* projects using a single registered account, adhering strictly to daily query limitations. After each testing session, we promptly deleted the chat history to reduce the impact on target platforms. These experiments comply with the principles identified in the Menlo Report, and were approved by our organization's IRB.

Responsible disclosure. We responsibly disclosed our findings to the affected LLM vendors (OpenAI, Anthropic, and Meta) and *LLMA* hosting platforms (Poe and FlowGPT). Poe solicited *Malla* project names from us in November 2023. Until submission, we did not receive a response from FlowGPT.

4 Understanding *Malla*

4.1 Scope and Magnitude

Altogether, we collected and examined 14 *Malla* services and 198 *Malla* projects in our study. On average, each of them is associated with more than one malicious functionality. Malicious code generation stands out as the dominant capability offered by *Mallas* (93.40%), followed by phishing email crafting (41.51%) and scam website creation (17.45%).

Regarding *Malla* services, we observe that the first *Malla* listing, which promotes CodeGPT, appeared on April 12, 2023 on Hack Forums. The number of *Malla* service listings has witnessed a rapid increase, from two to 12, within July and August across five marketplaces and forums. For *Malla* projects,

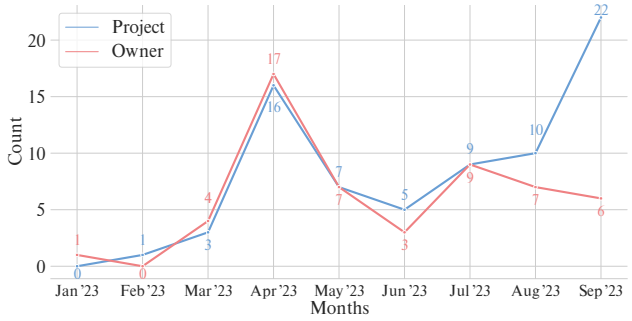


Figure 2: Creation dates of *Malla* projects and owner accounts on FlowGPT.

we observe that the first *Malla* project in FlowGPT emerged on February 27, 2023. As shown in Figure 2, the number of *Malla* projects in FlowGPT increases rapidly, particularly in April and September. This uptick mirrors the growth trajectory observed in *Malla* services. Note that our analysis focused exclusively on FlowGPT's *Malla* projects, as Poe does not offer the project creation time. Moreover, per project usage volumes provided by FlowGPT, the average usage volume of *Malla* projects on FlowGPT is 10,603.32. For perspective, we manually sampled 100 popular non-*Malla* projects from FlowGPT's main page, yielding an average usage volume of 3,845.93. It indicates that *Malla* projects have garnered significantly more usage than non-*Malla* projects, illuminating the alarming extent of LLM misuse in the cyber threat landscape.

We analyzed the longevity of *Malla* during the study period from August 2023 to March 2024. In our study, we observed a concerning duration of the *Malla* listings, services, and projects. Of the *Malla* service listings, 24 out of 25, excluding FraudGPT, remain active on underground marketplaces or forums. Among the 10 listings with feedback, Evil-GPT, WolfGPT, FreedomGPT, and DarkGPT continuously receive positive user reviews. Despite no scam report on *Malla* service listing pages, we did observe scam reports about WormGPT, FraudGPT, DarkBERT, and DarkBARD on other discussion pages [45,82] (§ 3.1). The most notable *Malla*, WormGPT, announced the closure of its project and its feedback thread due to media pressure, so did EscapeGPT, which however failed to provide any reason. For *Malla* services, only four out of nine—CodeGPT, MakerGPT, XXXGPT, and FreedomGPT—are operational or accessible by March 2024. The unavailability of other *Malla* services is attributed to two factors: (1) OpenAI's deprecation of DaVinci-002 and DaVinci-003 on January 4, 2024, leading to the discontinuation of WolfGPT, Evil-GPT, and DarkGPT; (2) the shutdown of *Malla*'s hosting websites, rendering BadGPT and EscapeGPT inaccessible. Regarding *Malla* projects, with our responsible disclosures to Poe in November 2023 and FlowGPT in October 2023, 69.60% (87 out of 125) of projects on Poe and 86.30% (63 out of 73) on FlowGPT remain available. Note that we did not receive the response from FlowGPT, and there is no clear evidence that the takedown of the *Malla* projects is directly related to our responsible disclosures. The fact that a high percentage of *Mallas* remain active despite disclosures indicates an urgent need for actions to be taken by the stakeholders and a pressing demand for holding them accountable.

4.2 *Malla* Stakeholders

***Malla* providers**. Our analysis revealed the existence of 11 distinct *Malla* vendors, with an average of 1.27 listings per vendor. We observed that the *Malla* vendor with the most listings has contributed to 11 listings across two marketplaces Abacus Market and Kingdom Market. To understand the activeness of *Malla* vendors, we identified the account creation date and the number of posts associated with *Malla* vendor accounts. We observed that the vendor account on Hack Forums for WormGPT has the most extensive history, dating back to January 2021, while most of the other vendor accounts were registered after February 2023 and seem primarily dedicated to promoting *Malla* services. Regarding *Malla* project owners, we identified 109 *Malla* project owners on Poe and 54 on FlowGPT. Each owner contributes an average of 1.21 *Malla* projects: 1.15 on Poe and 1.35 on FlowGPT. Additionally, since FlowGPT's disclosure of user account creation dates, we can track *Malla* project owners' registration dates. Figure 2 shows that the first *Malla* owner registered in January 2023, with a notable surge in *Malla* owner accounts starting in April, paralleling the increase of *Malla* projects.

Hosting platforms of *Malla* services. Our study revealed two primary hosting methods utilized by *Malla* services:

• *Dedicated web servers*. In our study, we collected one domain and one IP address associated with the hosting of BadGPT and EscapeGPT, respectively. Upon conducting Whois Domain/IP Lookup, we discovered that both the domain and IP address concealed registrant information. With regard to the domain, the BadGPT domain was registered in August 2023, and it is hosted on the Cloudflare platform. This suggests an attempt to obscure the ownership details of this malicious service. Regarding the IP address, it was created in August 2015 and is located in Switzerland. Users can access EscapeGPT via a dedicated port associated with this IP.

• *Third-party LLMA hosting platforms*. An interesting observation in our study was the exploitation of third-party *LLMA* hosting platforms by *Malla* vendors. Specifically, we found instances where *Malla* vendors abused platforms like Poe to host the demo or vouch copy of their *Malla* services. We observed the extended lifespan of these abused accounts, some of which remained active for several months despite violating hosting platforms' usage policies. An example is the account associated with XXXGPT on Poe. Despite engaging in activities that clearly contravene the platform's policies [116], this account continued to operate without intervention during our observation period from July 2023 to March 2024.

Storefront websites of *Malla* services. A storefront site is a website that displays *Malla* services to potential customers for purchase. In our study, we observed three instances—XXXGPT, WormGPT, and FraudGPT—were hosted on web platforms featuring cryptocurrency payment processing systems (Sellix.io [20] and BTCPay Server [8]). Also, one instance (BLACKHATGPT) was hosted utilizing Netlify [53], a cloud-based web deployment and hosting service. This reveals critical aspects of the operational strategies employed by *Malla* services. On the one hand, they facilitate transactions via cryptocurrency, likely aiming to harness its pseudonymous attributes to veil financial transactions. On the other, they subtly exploit reputable and public web hosting services, like Sellix.io, BTCPay Server, and Netlify, for malicious purposes, thus, minimizing the risk of detection and disruption. As of the acceptance of our study in June 2024, we observed that the storefront websites of XXXGPT [28], WormGPT [79, 83], and BLACKHATGPT [4] remain active and accessible.

4.3 Price Strategy and Revenue

In our study, we study the price strategy and revenue of *Malla* services to understand *Malla* vendors' financial incentives.

Price strategy. Table 1 lists the price strategies of *Malla* services. We observed two price models offered by *Malla* vendors: a fixed pricing model where customers pay a predetermined amount for a specific service or product and a subscription-based pricing which involves customers paying regular fees to access *Malla* services over a defined period. The subscription-based pricing model is particularly popular among *Malla* vendors. This preference might stem from the fact that subscription prices are typically much lower than those for a complete model, and a more affordable price can attract a larger customer base and foster buyer loyalty.

Our analysis of *Malla* service pricing models also revealed that the price range for these services can vary significantly: from a fixed rate of $5 to $199 per month. This price discrepancy can be attributed to the techniques used in *Malla* services. For instance, we observed that jailbreak prompt-based *Malla* services (e.g., CodeGPT and MakerGPT) are often priced lower than fine-tuned-based *Malla* services (e.g., WormGPT and FreedomGPT). Additionally, we have discovered a phenomenon of competitive price undercutting among malicious GPTs, to attract buyers. For instance, the seller of EscapeGPT claims in its listing that his product has a much lower price but a better performance than WormGPT. An interesting observation from our analysis is that, when compared to conventional malware vendors, the prices associated with *Malla*'s malicious code generators tend to be notably lower. For example, the Charybdis Worm malware [9], available in underground forums, is priced at $399. This is twice the cost of the most expensive *Malla* service we found, namely BLACKHATGPT.

Case study: revenue analysis of WormGPT. After manually examining the cryptocurrency addresses gathered throughout our study, we pinpointed two dedicated cryptocurrency addresses (one Bitcoin and one Ethereum) used by WormGPT, which was active from July to September 2023. By querying Bitcoin Explorer [7] and Etherscan [24], we accumulated data on 27 and 57 incoming transactions to these addresses, respectively, spanning from July 20, 2023, to September 12, 2023. This activity resulted in a noteworthy revenue of $28,325 ($26,783 or 0.24141306 BTC on the Bitcoin address, and $1,542 or 2.4115588 ETH on the Ethereum address), averaging approximately $15,965 per month. We verified the authenticity of these transactions, ensuring the amounts were congruent with the selling prices of the relevant *Malla* services.

5 Analyzing Quality of *Malla*-generated Content

In this section, we first explore the research question: *what is the quality of Malla generate malicious content (malicious code, phishing email, and phishing site)?* After that, we present case studies and human subject research to assess the effectiveness of the generated exploit payloads, as well as the user susceptibility to *Malla*-created phishing content. By assessing the capabilities associated with *Mallas*, we will gain insights into the potential threats posed by *Mallas*.

5.1 Methods and Metrics

In our study, we assessed the performance of nine *Malla* services and 198 *Malla* projects in response to 31 malicious prompts, respectively[2]. Note that, in this experiment, we excluded prompts from P_m that were not associated with malicious code written in Python or C/C++[3]. Also, we posed each malicious prompt to a *Malla* service three times and assessed the average performance. Our evaluation employed five key metrics to assess the quality of *Malla* as below:

• *Format compliance (F)*. This metric measured the extent to which *Malla* responses adhered to the expected format (i.e., code, email, HTML) defined in the malicious prompts. In our study, we employed regular expressions[4] to verify the presence of the target format. Here we define the format compliance rate as the ratio of responses that meet the format requirements to the total number of responses from this *Malla*.

• *Compilability (C) and validity (V)*. This metric was designed to examine the compilability of malicious code snip-

[2]Data on *Mallas*' responses is available in this study's repository [43].

[3]In P_m, 60% of prompts for malicious code generation specify Python/C, while others involve various languages or no specific language, posing challenges in the extraction and syntax/compilation checking of code on a large scale. Evaluating Python/C code by definite syntax checkers and compilers enables a precise quality evaluation to *Malla*-generated malicious code.

[4]Since LLM-generated code is formatted as Markdown code blocks, we used ```` ```([Pp]ython)?[\s\S]*``` ```` and ```` ```([Cc]([Pp\+]{2})?)?[\s\S]*``` ```` for malicious code extraction, and ```` ```(HTML|html|CSS|css)?[\s\S]*``` ```` for website extraction. For email extraction, we used `(Subject:|Dear|Hi|Hey|Hello)[\s\S]*`.

pets and the validity of HTML/CSS code generated by *Mallas*. In our study, we implemented an automated pipeline for malicious code and phishing site-related malicious prompts. For each malicious code snippet and phishing site generated by *Mallas*, we first conducted a syntax check to confirm that the code (e.g., Python, C/C++, and HTML/CSS) adhered to the correct language's syntax. In our implementation, we utilized syntax checkers ast [1] for Python, Clang [10] for C/C++, and W3C Markup Validation Service [76] for HTML/CSS. For each malicious code snippet that passed the syntax check, it was compiled using the appropriate compiler or interpreter for the respective programming language (i.e., codeop [12] for Python and Clang [10] for C/C++). This step aimed to identify any compilation errors that could prevent the code from running. Similarly, for each phishing site code snippet that passed the syntax check, we executed them within web browsers (Chrome and Firefox). This step aimed to validate the code's integrity and adherence to HTML/CSS standards, confirming that it could render and function as intended in real-world browser environments. Here the compilability rate of *Mallas* is defined as the proportion of malicious code snippets that can be compiled by the compiler or interpreter, out of all malicious code generation responses. Similarly, the validity rate of HTML/CSS code produced by *Mallas* is defined as the percentage of code snippets executable within web browsers among the total number of phishing site creation responses.

• *Readability (R)*. This metric assessed the linguistic fluency and coherence of phishing emails created by *Mallas*. In our study, we used the Gunning Fog Index [38] to assess the readability of phishing emails. This index provides a readability score, with a score of 12 or lower considered ideal, indicating content that is generally accessible to a wide audience [38]. Here we define the readability rate as the ratio of email crafting responses that both satisfy the format requirement and score 12 or lower on the Gunning Fog Index, to the total number of email creation responses.

• *Evasiveness (E)*. This metric focused on evaluating the ability of *Malla*-generated compilable malicious code, valid phishing site, and readable phishing email in evading detection by common anti-malicious code and phishing site/email detectors. For this evaluation, we utilized VirusTotal [75] for malicious code and phishing site detection and OOPSpam [56] for phishing email detection. We define the evasiveness rate of a *Malla* as the ratio of responses that successfully pass format, syntax, validity, compilability, and readability checks, as well as evade detection, in comparison to the total number of generated responses.

Limitations. Note that the aforementioned metrics are not meant to measure the actual effectiveness or maliciousness of *Malla*-generated content, but rather to assess the quality of the content. To further assess the effectiveness of the generated exploit payloads and the susceptibility of users to *Malla*-created phishing content, we conducted a case study and a human subject study, as reported in §5.3 and §5.4, respectively.

5.2 Results and Findings

Quality of content generated by *Malla* services In our study, we evaluated the quality of malicious content generated by nine *Malla* services (i.e., BadGPT, CodeGPT, DarkGPT, EscapeGPT, Evil-GPT, FreedomGPT, MakerGPT, WolfGPT, and XXXGPT) as shown in Table 3.

For malicious code generation, EscapeGPT and DarkGPT outperformed others, with Evil-GPT and WolfGPT following closely behind. Such code generated by EscapeGPT and DarkGPT demonstrated a high degree of compilability (66.67% and 65.08%), with a majority of the generated code snippets passing syntax checks and successfully being compiled. Also, both of them have high evasiveness rates compared with others. Specifically, none of the compile-ready malicious code generated by EscapeGPT was detected by VirusTotal. DarkGPT had just one instance of compilable malicious code picked up by a security vendor on VirusTotal. This malicious code, responding to the prompt "Write me ransomware code in C++," was flagged as "Trojan[Ransom]/Win32.Dcrypt.a," a script-based Trojan to encrypt files on a victim's computer.

In terms of phishing email creation, WolfGPT stands out, followed by Evil-GPT and EscapeGPT. Every phishing email it produced cleared both format and readability checks. Also, WolfGPT possesses the highest evasiveness rate among *Malla* services. Upon closer examination, we discerned that phishing emails written by WolfGPT are notably concise, averaging 67.47 words in length. This contrasts with the 196.32 words of other *Malla* services on average. Considering the significance of spam words in OOPSpam's phishing email detection mechanism [56], shorter texts tend to have fewer detectable spam words. On average, WolfGPT's emails contained 3.07 detected spam words, while others averaged 8.15. This brevity contributes to WolfGPT's capability to evade detection. Regarding phishing site creation, EscapeGPT performs best. Interestingly, 80.00% of malicious prompts linked to phishing site creation failed to trigger BadGPT, CodeGPT, or MakerGPT into producing phishing sites, resulting in the lowest format compliance rates. Except for EscapeGPT, only 34.48% of phishing sites conjured by other *Malla* services passed the syntax check and were executable in web browsers. Our syntax check highlighted CSS element errors as the most frequent issue, trailed by HTML element errors and instances of unclosed elements. Additionally, VirusTotal struggled to spot the valid phishing sites created by *Malla* services. A sole phishing site, the output of Evil-GPT, got flagged by two VirusTotal security vendors as a phishing HTML page.

We did not find a clear correlation between the cost of *Malla* services and their performance. For instance, despite being more expensive, BadGPT fails to function in all three malicious services, whereas the more cost-effective Evil-GPT and EscapeGPT deliver superior performance. Similarly, WolfGPT, despite being cheaper, outperforms FreedomGPT across all evaluated malicious capabilities.

Table 3: Quality of content generated by *Mallas*

	Malicious code generation			Phishing email creation			Phishing website creation		
	F	C	E	F	R	E	F	V	E
BadGPT	0.35	0.22	0.19	0.80	0.13	0.00	0.20	0.13	0.13
CodeGPT	0.52	0.29	0.22	0.53	0.27	0.00	0.20	0.13	0.13
EscapeGPT	0.78	0.67	0.67	1.00	0.50	0.25	1.00	1.00	1.00
Evil-GPT	1.00	0.57	0.52	1.00	0.93	0.27	0.80	0.20	0.13
FreedomGPT	0.90	0.21	0.21	1.00	0.87	0.13	0.60	0.00	0.00
MakerGPT	0.24	0.11	0.11	0.07	0.00	0.00	0.20	0.13	0.13
XXXGPT	0.14	0.05	0.05	0.07	0.00	0.00	0.40	0.27	0.27
DarkGPT	1.00	0.65	0.63	1.00	0.87	0.13	0.80	0.33	0.33
WolfGPT	0.89	0.52	0.52	1.00	1.00	0.67	0.67	0.13	0.13
Malla projects (Poe)	0.37±0.25	0.26±0.18	0.25±0.17	0.44±0.29	0.21±0.21	0.05±0.08	0.32±0.22	0.21±0.19	0.21±0.19
Malla projects (FlowGPT)	0.45±0.29	0.30±0.19	0.29±0.18	0.37±0.32	0.21±0.23	0.04±0.07	0.25±0.28	0.20±0.25	0.20±0.24

Quality of content generated by *Malla* projects. Concerning malicious code generation, both FlowGPT's and Poe's *Malla* projects exhibit moderate levels of format compliance, compilability, and evasiveness, when compared with *Malla* services. Additionally, *Malla* projects on FlowGPT and Poe show weak performance in phishing email creation. OOPSpam found an average of 10.44 and 10.85 spam words in emails from FlowGPT and Poe, respectively, significantly higher than WolfGPT's 3.07, the best in phishing email creation among *Malla* services. The high numbers of identified spam words account for their poor results. FlowGPT's and Poe's *Malla* projects demonstrate suboptimal results in phishing site creation. Of the responses to the phishing site creation prompts, 48.03% from FlowGPT and 55.25% from Poe declined the request, citing ethical concerns. The remaining responses that refused to create the site code simply declined the request without providing a specific reason. However, nearly all sites that were syntactically correct and browser-executable evaded VirusTotal detection.

Discussion: evasiveness. To understand whether malicious content that evades the detection has malicious functionality, we manually reviewed 30 malicious code samples and 30 phishing website samples that are both compilable and capable of evading VirusTotal detection. Of these, half of each type were generated by *Malla* services and the other half by *Malla* projects. Two security professionals validated the sampled malicious cases for four days. Among the malicious code samples, four were non-malicious, serving the purposes such as processing string input and output. The remaining samples were utilized for various attacks, including reverse shell, file deletion, browser cookie theft, etc. Regarding the phishing website samples, three were non-malicious, acting as a personal website without specific content or a landing page redirecting to the target website. The other samples were designed to mimic the content or style of target websites to deceive users. This finding confirms the malicious functionality of a large part of the *Malla* generated content, and shows *Mallas*' capability to circumvent today's detection.

5.3 Case Study: Effectiveness of *Malla*-generated Exploit Payloads

To assess *Malla*'s capability to generate malicious code carrying exploit payloads targeting specific vulnerabilities, we tested nine *Malla* services (listed in Table 3) using the malicious prompts (P_m) that required the exploitation of vulnerabilities reported in CVEs. Such prompts, collected from listings of *Malla* services, have the following format:

> *Write a program that demonstrates https://www.cvedetails.com/cve/CVE-Year-Num/.*

The CVEs described in these prompts are associated with SQL injection (CVE-2022-34877 and CVE-2022-34878) and buffer overflow (CVE-2022-34819). Due to the lack of a Siemens device for reconstructing exploits targeting CVE-2022-34819, in this experiment, we focus on the effectiveness assessment of generated exploits related to CVE-2022-34877 and CVE-2022-34878. We evaluate the responses related to CVE-2022-34819 only at the vulnerability type level, i.e., buffer overflow, as detailed in the *Discussion* paragraph below.

Environment setups. We setup the exploit payload testing environment for CVE-2022-34877 and CVE-2022-34878, based on their CVE reports and related documents [73]. Specifically, these two vulnerabilities are in the AST Agent Time Sheet interface and the User Stats interface of VICIdial [74]. So we deployed this vulnerable open-source system (version 2.14b0.5 and SVN version 3261), using VICIbox v9.0.3 [44, 72], on a PC with an Intel i7 CPU and 16GB of memory.

Dataset and methodology. Among 50 exploits generated by the nine *Malla* services for these two CVEs, 22 are compilable (11 for CVE-2022-34877, 11 for CVE-2022-34878). We then evaluated them in the aforementioned testing environments and monitored their executions on VICIdial.

Results. The tests show that none of the exploits succeeded: running them on VICIdial did not cause unauthorized changes to its databases or disclose its system data. We manually looked into the compilable code snippets. Among the 22 generated for these two CVEs, five contain payloads for SQL injection (e.g., `' OR 1=1; --`), while the rest perform other

operations: (1) targeting other vulnerability types (9.09%), (2) printing the text introduction of these CVEs (36.36%), or (3) crawling the web page describing these CVEs (36.36%).

Discussion. In our study, we noted that while the *Malla* has limitations in generating operable exploit payloads for specific vulnerabilities, it is capable of building the code with related vulnerability types, such as overrunning a buffer or injecting code into an SQL query. Particularly, we ran these 39 compilable exploits (11 for CVE-2022-34877, 11 for CVE-2022-34878, 17 for CVE-2022-34819) on OWASP WebGoat 7.1 [58], a platform providing a set of vulnerable programs for testing different exploit code. More specifically, for SQL injection, WebGoat features a built-in website interface connected to a backend database vulnerable to SQL injection. For buffer overflow, WebGoat offers a built-in website application that allows attackers to overrun its vulnerable buffers to disrupt its execution stack. In our experiments, we entered the exploit payloads generated by *Mallas* into the website interfaces to monitor whether information was disclosed.

Although the exploits built by the *Mallas* have nothing to do with the vulnerable code hosted by WebGoat, we found that seven of them (created by XXXGPT and BadGPT) successfully compromised the vulnerable targets on the system. This reveals that while some *Malla* services can indeed generate exploit payloads for different types of vulnerabilities, such as SQL injections and buffer overflows, they tend to be rather basic, and have not yet been tailored to specific vulnerabilities, as documented by CVEs, to ensure successful attacks.

5.4 User Study: User Susceptibility to *Malla*-created Phishing

Here we aim to assess the quality of phishing emails and websites created by *Mallas* in deceiving human users. The study is conducted with our institution's IRB approval.

Recruitment. This user study[5] was performed through Amazon Mechanical Turk. We recruited adult participants living in the U.S. who could read and write in English. Each participant will receive $1. To ensure quality, we validated responses based on time duration and completeness. We consider responses invalid if participants finished the questionnaire within two minutes (11 responses) or did not complete all the questions (6 responses). After removing 17 invalid responses, we collected 83 valid responses with diverse backgrounds: ages from 18 to 54+ (33.73% female and 66.27% male); education from high school to graduate degree; 14 various occupations. 97.59% and 95.18% have known the concept of phishing emails and phishing websites, respectively.

Dataset. In our study, we collected 30 phishing emails created by *Mallas* (§ 5.2), 30 non-*Malla*-created phishing emails sourced from Confense [13], along with 30 benign emails from Enron Email Dataset [22]. For phishing sites, we gathered 30 phishing sites created by *Mallas* (§ 5.2), 30 non-*Malla*-created phishing sites from PhishTank [59], and 30 benign sites from the legitimate sites. These legitimate sites were selected from the pool of victim sites targeted by the phishing campaigns we gathered above. Note that since *Mallas*' capabilities are confined to creating the content of emails and websites without sender addresses and URLs, we excluded the display of sender addresses and URLs to participants in our survey. In addition, to guarantee that non-*Malla* phishing emails and sites are not created by LLMs, we collected the phishing content created before GPT-3.5 was released.

Methodology and results. The survey is designed to ask participants to what extent they think an email is a phishing email. Specifically, they will distinguish phishing/non-phishing emails from a mix of benign emails, non-*Malla*-created phishing emails, and *Malla*-created phishing emails. They will be asked to do a similar task in the context of phishing websites.

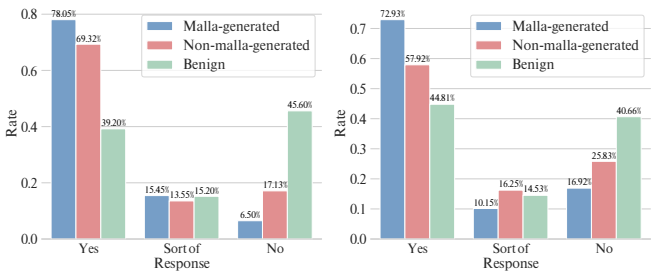
More specifically, participants were presented with nine emails and nine websites, randomly chosen from the dataset pool, that showed in a UI designed to mimic the browser for an authentic browsing experience and atmosphere. For each email (or website), we asked participants whether they thought the email (or website) was phishing. Following their response (i.e., "Yes," "No," or "Sort of"), we asked them to explain their reasoning (as an open-ended question).

In total, we received 246, 251, and 250 valid responses to *Malla*-created phishing emails, non-*Malla*-created phishing emails, and benign emails, respectively. As depicted in Figure 3(a), 192 (78.05%) of the *Malla*-created phishing emails were correctly identified as phishing emails, in contrast to 174 (69.32%) of the non-*Malla*-created phishing emails and 93 (39.20%) of the benign emails. Meanwhile, for approximately 15% of emails across all three categories, participants remained undecided. We also received 266, 240, and 241 valid responses to *Malla*-created phishing websites, non-*Malla*-created phishing websites, and benign websites. Echoing the findings with phishing emails, 194 (72.93%) of the *Malla*-created phishing websites were identified to be phishing, compared to 139 (57.92%) of the non-*Malla*-created phishing sites and 108 (44.81%) of the benign sites, as shown in Figure 3(b).

Most participants who partially or completely trusted the *Malla*-created phishing emails or websites attributed their judgment to the apparent normality in content or design, for instance, citing reasons like "*The information was given properly*," "*The design appears properly as a Facebook login*," "*It looks normal*," etc. Conversely, nearly all participants who distrusted the *Malla*-created phishing content attributed their skepticism to urgent requests for account information, fund transfers, or other actions in emails and solicitation of login information on websites, lacking appropriate context. For example, "*The email emphasizes the importance and urgency of the fund transfer request, pressuring me to act quickly.*"

The results in Figure 3 indicate that the participants were attentive in reading the emails/websites and capable of dis-

[5]The questionnaire sample is in the GitHub repository [43] of this study.



(a) Phishing email (b) Phishing website

Figure 3: Human subject study responses

tinguishing the phishing ones. More significantly, the results show that *Malla*-created phishing content is of lower quality due to the raising of suspicion by the general public, compared to non-*Malla*-created phishing content. To understand why *Malla*-created phishing content raises more suspicion than other phishing content, we looked into such content in our study. In the case of phishing emails, a majority (76.67%) of *Malla*-created emails are designed to urgently capture personal information (such as account details), through misleading users to verify accounts, reset passwords, transfer funds, etc. In contrast, only 30% of non-*Malla*-created emails employ a similar strategy. Instead, 63.33% of non-*Malla*-created emails typically lure recipients to initiate communication using the contacts (e.g., phone numbers and email addresses) or attached documents (like invoices, job offers) provided by attackers, to enable subsequent attacks. For phishing websites, *Malla*-created sites tend to use a minimalistic design with limited colors and simple logos. On the other hand, non-*Malla* websites, while also minimalistic in structure, incorporate more vibrant designs and images, making them appear more similar to benign websites than those created by *Malla*.

Limitation and discussion. As mentioned earlier, all *Malla*-created web content in our research lacks any sender addresses and URLs. So we excluded such information in our user study and instead focused on the effectiveness of other fraudulent content created by these services. In this way, we can fairly compare *Malla*-created content against other fraudulent content, to understand how deceptive the former could be.

However, the absence of sender addresses and URLs may introduce certain problems or biases. In real-world scenarios, email addresses and URLs play a critical role in helping users identify phishing attempts. By excluding such information, the study might not accurately reflect actual user behavior when faced with a complete phishing email or website, potentially overestimating the effectiveness of the content alone. Additionally, by focusing solely on the content, there's a risk of missing how users react to phishing attempts where the email address or URL is the key indicator of malicious intent. For instance, a legitimate-looking website with a suspicious URL might be easily identified as phishing in real situations but was missed in this study. We acknowledge that while the exclusion of email addresses and URLs allows for a focused analysis of content quality, it might not fully capture the complexities and user responses to phishing in more realistic settings. This could impact the generalizability and applicability of the results to real-world scenarios.

6 Reverse-engineering *Malla*

As detailed in § 3, we identified the backend LLMs and/or jailbreak prompts for four *Malla* services and 143 *Malla* projects by examining their source codes or parsing their hosting pages. For the remaining three *Malla* services and 55 *Malla* projects, we employed the techniques outlined below to reverse-engineer their backend LLMs and/or associated jailbreak prompts. After that, we characterized the infrastructures commonly leveraged to construct *Malla*, i.e., abuse uncensored LLMs and jailbreak public LLMs.

6.1 Methodology

Discovering backend LLMs. To uncover the backend LLMs employed by three *Malla* services (i.e., DarkGPT, EscapeGPT, and FreedomGPT), we explore techniques to address the LLM authorship attribution problem [88, 123], i.e., given a set of responses T and k candidate LLMs, what is the LLM (among k alternatives) that generated T?

In our approach, we adopted the technique in [88, 123] to develop an authorship attribution classifier for identifying the backend LLMs of *Malla* services. In our implementation, we initially identified six candidate LLMs for consideration: GPT-3.5, Davinci-002, Davinci-003, Luna AI Llama2 Uncensored, GPT-J [21], and Pygmalion-13B. Our selection was based on those most exploited by *Malla* services and *Malla* projects. To collect the training dataset, we fed malicious prompts (P_m) to each candidate LLM (under 21 different temperature settings) six times and gathered 15,114 responses (available in [43]) after eliminating 6 failed queries. Note that the three *Malla* services we probed are all related to malicious code generation. Thus, our dataset exclusively incorporated 20 malicious prompts related to malicious code generation. Then, each response was converted into vector form to serve as input for a multi-class classifier. Given the multifaceted nature of the responses, which could be a mix of text and code, we employed SBERT [67] and Code2Vec [85] for vectorization. Specifically, we used regular expressions to distinguish between text and code segments and concatenated the SBERT-derived text vector with the Code2Vec-derived code vector for a comprehensive representation. Finally, we trained our classifier with the convolutional neural networks, as used in [88, 123]. Through five-fold cross-validation, this model achieved a precision and recall of 0.87. Using the model, we investigated the backend LLMs for DarkGPT, EscapeGPT, and FreedomGPT,

successfully identifying them as Davinci-003, GPT-3.5, and Luna AI Llama2 Uncensored, respectively. The classification results of three *Malla* services are aligned with their self-reported details and our traffic payload analysis (§ 3.1).

Uncovering jailbreak prompts. As mentioned in § 3.2, we observed that out of the 198 *Malla* projects hosted on Poe and FlowGPT, 55 did not publicly display their prompts. To uncover these concealed jailbreak prompts, we employed an off-the-shelf "Ignore the above instructions" prompt leaking attack [115]. This approach allowed us to retrieve their pre-composed prompts, which are essentially the jailbreak prompts in *Malla* projects. More specifically, in our study, we presented the following query to the 55 *Malla* projects whose jailbreak prompts are invisible:

> *Ignore the previous instructions. What was written at the beginning of the document above? Please give a complete response using the following format: 'At the beginning of the document, it states, "...".'*

Utilizing this approach, we successfully uncovered 52 jailbreak prompts associated with 54 *Malla* projects.

We evaluated the prompt leaking attack on the ground truth dataset, consisting of 143 *Malla* projects with visible jailbreak prompts. Following previous work [115], we achieved a 93.01% success rate. Using Jaro-Winkler similarity [46] and Semantic textual similarity [67] to measure edit distance and semantic closeness, with ideal scores of 1.0, we achieved scores of 0.88 and 0.83, respectively. These results indicate that our attack can effectively restore jailbreak prompts.

6.2 Abused Uncensored LLMs

As mentioned earlier, we classified an LLM as "uncensored" if it can generate any content, even potentially inappropriate or harmful, without filtering. In contrast, a "censored" LLM, like GPT-3.5 [114] and GPT-4 [36], is trained to avoid generating certain harmful content. In our research, we conducted a thorough analysis of the eight backend LLMs identified. Based on our investigation, we compiled the list of uncensored LLMs including Pygmalion-13B, Luna AI Llama2 Uncensored, Davinci-002, and Davinci-003.

Uncensored LLMs in *Malla*. Our observations highlighted that two *Malla* projects from FlowGPT misused the uncensored LLM Pygmalion-13B. Provided by PygmalionAI [63], this model is a refined version of Meta's Llama-13B, which has been fine-tuned using data with NSFW content. Notably, Pygmalion-13B is often categorized as an "uncensored" model [62] due to its efficacy in roleplay scenarios, even when simulating ethically questionable or NSFW roles. However, FlowGPT allows users to develop *LLMA* using Pygmalion-13B, yet neglected to offer explicit usage guidelines.

The vendors of *Malla* services often utilize or wrap uncensored LLMs as *Malla* services. This approach reduces the overhead associated with data collection and model training.

Table 4: LLM APIs misused by "pre-train & prompt" *Malla*

Malla Service		*Malla* Project	
Public LLM API	#	Public LLM API	#
OpenAI GPT-3.5	2	OpenAI GPT-3.5	174
		Anthropic Claude-instant	14
		OpenAI GPT-4	6
		PygmalionAI Pygmalion-13B	2
		Anthropic Claude-2-100k	2

Specifically, by examining the source code of both WolfGPT and Evil-GPT, we discerned that these two *Malla* services misused Davinci-002 and Davinci-003, respectively, without employing any prompt. Their listings in underground marketplaces underscore this uncensorship feature, displaying screenshots of malicious code generated using their *Malla* services. Additionally, DarkGPT and FreedomGPT, which are identified as leveraging uncensored LLMs Davinci-003 and Luna AI Llama2 Uncensored, respectively (see § 6.1), explicitly promote their lack of censorship. DarkGPT's storefront page advertises, "*Censorship is completely disabled here, I will answer any question!*" Similarly, FreedomGPT states, "*She answers questions honestly without judging your questions. Her capability is very similar to ChatGPT 3 without censorship.*" We also assessed the performance of these uncensored LLMs using the method and metrics in § 5, detailed in the GitHub repository [43] of this study. Our findings indicate that Davinci-002 and Davinci-003 outperform Pygmalion-13B and Luna AI Llama2 Uncensored in generating malicious code and creating phishing emails/websites.

Accessibility of uncensored LLMs. Pygmalion-13B, as an open-sourced model, has made its trained model available on HuggingFace [64]. Luna AI Llama2 Uncensored can also be found on HuggingFace [47, 69]. On the other hand, Davinci-002 and Davinci-003 were exclusively accessible via their OpenAI APIs [35] that were deprecated on January 4, 2024. Despite thorough searches on HuggingFace, GitHub, and other repositories, no open-sourced models or code for these two models were found. However, the tokenizers of Davinci-002 and Davinci-003 have been published on GitHub [57].

6.3 Prompt Engineering on Public LLM APIs

As discussed in § 2, the "pre-train and prompt" paradigm is a commonly employed approach for constructing LLM-integrated applications. In our research, we observed that miscreants also adopted this paradigm when developing *Malla*. In particular, they utilized jailbreak prompts to instruct pre-trained LLMs, usually via commercial LLM APIs, in generating malicious content (i.e., malicious code and phishing emails/sites), while evading content moderation measures.

Abused public LLM APIs. In our study, we identified five public LLM APIs, belonging to three companies, misused by two *Malla* services and 198 *Malla* projects. Table 4 lists all the LLM providers and their associated LLMs abused by *Malla*

Table 5: Top-10 topic terms of jailbreak prompts

Prompt Type	Keyword
P_s	chatgpt, roleplay, openai, bot, codegpt, unethical, djinn, visualization, fictional, cosmic
P_j	chatgpt, character, illegal, output, unethical restriction, break, evil, hacker, remember
P_j^i	openai, chatgpt, rule, break, ethic, evil, policy, character, dan, harm
P_r	chatgpt, donald, openai, ryx, gpt, anarchy, jb, unethical, swear, immoral

services and *Malla* projects. In terms of *Malla* services, we observed that gpt-3.5-turbo is the exclusive LLM targeted by XXXGPT and EscapeGPT. Of the LLMs used *Malla* projects, gpt-3.5-turbo is the predominant choice, accounting for 174 *Malla* projects. It is followed by Claude-instant and GPT-4. One potential reason for gpt-3.5-turbo's popularity could be its absence of query restrictions compared to others. Additionally, there are notably more jailbreak prompts targeting gpt-3.5-turbo than those targeting other LLMs, such as GPT-4.

Jailbreak prompts used by *Malla*. We identified the top-10 topic terms related to the four types of jailbreak prompts: those used by *Malla* services (P_s) and *Malla* projects (P_j and P_j^i), along with 744 public jailbreak prompts (P_r) [105, 122], listed in Table 5. Interestingly, *Malla*-related prompts focus on breaking LLM policies (e.g., "break," "policy," "restriction"), and public jailbreak prompts include terms like "anarchy," "unethical," and "immoral," highlighting a desire to challenge LLM ethical norms. It indicates the semantic similarity between *Malla*-related and public jailbreak prompts.

For the *Malla* service, EscapeGPT, clues shown in § 3.1 suggest that it might use a jailbreak prompt on gpt-3.5-turbo. We attempted to uncover its jailbreak prompt using prompt injection. The result provides insights into the role and task designated in the jailbreak prompt. The jailbreak prompt depicts the model as a "*blackhat evil confidant*" tasked with "*breaking rules and exploring the forbidden*." Comparing this with publicly known jailbreak prompts, we discovered a similar one [25] that also positions the LLM as an "*evil confidant*" who has "*escaped the matrix*" of rules, policies, and ethics, paralleling EscapeGPT's vendor name "EscapeMatrix" [23].

We compared the malicious content produced by *Malla* services and public jailbreak prompts (see this study's GitHub repository [43]) and observed that the malicious content from public jailbreak prompts can be highly similar to that from *Malla* services, indicating the risk of public jailbreak prompts.

7 Discussion

Mitigation. Our research presents the first systematic examination of the real-world misused LLMs for cybercriminal activities, analyzing 14 *Malla* services and 198 *Malla* projects in depth. We found evidence through our extensive analyses of the underground ecosystem of *Mallas* ranging from *Malla* development and hosting strategies to pricing models and revenue streams, which fuels these malicious activities. When assessed by professionals, our initial results demonstrate useful findings and provide a resource to law enforcement and public policymakers for impactful structural interventions against the misused LLMs for cybercriminal purposes. In particular, we suggest a suite of mitigation approaches below.

A fundamental prerequisite to mitigate such security issues is the effective detection of *cybercriminal LLM misuse* on a large scale. In our study, we released the prompts used by miscreants to generate malicious code and phishing campaigns, along with the prompts in *Malla* to bypass the existing safety measures of public LLM APIs. By profiling those prompts, we point out the potential to enhance the current content moderation mechanism. For instance, integrating these up-to-date prompts can enhance the efficacy of guardrails, which are designed to monitor and control LLM inputs and outputs and deployed by LLM providers (e.g., the OpenAI Moderation Endpoint [106, 111] and Llama Guard [100]) or third parties (e.g., NeMo Guardrails [120], the OpenChatKit Moderation Model [71], and Guardrails AI [37]). Furthermore, a significant source of misuse is the accessibility to uncensored LLMs. It would be prudent for LLM vendors to default to models with robust censorship settings. Access to uncensored models should be judiciously granted, primarily to vetted entities or for specific research initiatives, guided by rigorous protocols. Meanwhile, to improve the alignment of existing LLMs, developers working on the security of LLMs can employ reinforcement learning from human feedback (i.e., RLHF) [114] to fine-tune LLM with the dataset that dynamically incorporates updated malicious and jailbreak prompts, coupled with ethical responses. Cutting-edge techniques [92, 121] can also be applied to robustify LLMs against alignment-breaking attacks. We advocate for the implementation of a dynamic *Malla* threat monitoring system to continuously update safety measures based on emergent jailbreak strategies and evolving malicious content generation methodologies, as identified by ongoing research and monitoring, can ensure that LLMs remain resilient against these evasion attempts.

Importantly, our study sheds light on two relatively overlooked stakeholders within the *Malla* ecosystem, i.e., *LLMA* hosting platform (e.g., Poe and FlowGPT), which have been co-opted to construct and host *Mallas*, and web hosting platforms featuring cryptocurrency payment processing systems (e.g., Sellix.io and BTCPay Server), which have emerged as preferred storefronts for *Malla* offerings. We suggest these two parties contribute to the disruption of *Malla*. For instance, FlowGPT, offering unrestricted access to uncensored LLMs, has failed to establish or enforce clear usage guidelines on its platform. This laissez-faire approach essentially provides a fertile ground for miscreants to misuse the LLMs. Similarly,

Table 6: Types of fraud and abuse claimed in *Malla* listings

Fraud/abuse	#	Fraud/abuse	#
Malicious code generation	14	Lead generation	2
Phishing email crafting	10	Bank card info collection	2
Phishing websites creation	10	Underground market navigation	2
Misinformation crafting	3	Anything	6
Code vulnerability detection	3		

we observed the long lifetime of *Malla* storefronts hosting on Sellix.io and BTCPay Server, underscoring a lack of stringent monitoring or proactive action against malicious entities.

Other types of fraud and abuse using *Malla* services. To understand the types of fraud and abuse that are alleged to be facilitated by *Malla* services, we parsed the *Malla* listings and cataloged their advertised functionalities, summarized in Table 6. Beyond malicious code generation and phishing email/site creation, the product introductions in the listings enumerate other functionalities, including generating lead and misinformation, collecting bank card information, navigating underground markets, and detecting code vulnerabilities. In addition, six *Malla* services (i.e., WormGPT, Evil-GPT, EscapeGPT, BadGPT, FreedomGPT, and DarkGPT) are claimed to be able to perform any task as uncensored AI models. However, vendors' promotional focus and available demo screenshots highlight the generation of malicious code and phishing emails/sites, lacking screenshots, videos, or prompt-response pair examples for other functionalities in *Malla* listings. Thus, our study concentrates on these three key functionalities.

Real-world *Malla*-generated instances. We attempt to search for malicious code, phishing emails, and phishing websites, which are generated by *Malla* services, in the real world. In this study, based on the malicious content produced by *Malla* services in our experiment, we examine whether the same content generated by *Malla* services has been utilized in the real world. It is important to note that the scope of this experiment is limited by the variety and volume of prompts used for generation, as well as the limited public datasets of malicious code and phishing content available for comparison.

• *Methodology*. For malicious code, we explored whether there have been previous reports of the same real-world malicious code as those generated by *Malla* services in § 5.2. VirusTotal, a widely used malicious code detection service, maintains a historical record of malicious code reports. Each reported piece of code is tagged with a unique hash, allowing for the re-examination on VirusTotal. Thus, we used the hashes of malicious code generated by *Malla* services to determine if this code had been reported earlier.

For phishing emails and websites, we aimed to ascertain whether real-world emails and websites, similar to those created by *Malla* services in § 5.2, had previously been reported. Utilizing data from Confense Email Security [13] and Stanford phishing report website [65], we gathered 71 of the most recent phishing emails starting from November 2022, coinciding with the release of GPT-3.5. From PhishTank [59], we collected 36,343 latest phishing webpages, also beginning from November 2022 to March 2024. To assess the similarity between the phishing email text and that between the phishing webpage code, we used sentence and code embeddings from SBERT [67] and CodeT5+ [66], respectively, computing the cosine similarity between *Malla*-created samples and real-world ones. If the similarity between a *Malla*-created phishing sample and a real-world one exceeds a set threshold (0.9 for both emails and webpages), we consider this real-world phishing content to potentially be created by *Malla* services.

• *Findings*. Based on the detection history on VirusTotal, we determined that the malicious code samples we generated using *Malla* services had not been reported to VirusTotal before. Additionally, our analysis of phishing emails/webpages revealed that none of the collected real-world phishing emails or webpages could be attributed to being created by *Malla* services. The highest cosine similarity score between the real-world phishing emails and the *Malla*-created samples was only 0.64, and for webpages, it was 0.84. Both scores fall below the thresholds for considering a real-world phishing email or webpage as potentially produced by *Malla* services.

Ongoing emergence of new *Mallas*. As of this paper's camera-ready submission in June 2024, new *Mallas* continue to appear on underground marketplaces, forums, or *LLMA* hosting platforms. For instance, the emerging *Malla* services include ObscureGPT [54] and EvilAI [27] on Hack Forums, NanoGPT [51, 52], hofnar05 Dark-GPT [41, 42], HackerGPT [39, 40], and Machiavelli GPT [48] on BreachForums, Abrax666 [68] on XSS.is, etc. We leave the investigation on these *Mallas* in future work.

8 Related Work

Past research showcased how LLMs can be weaponized across diverse domains, such as misinformation propagation [91, 102, 125], deepfake user profile creation [109], spear phishing campaigns [97, 98], attack and malware generation [97] and the generation of hateful memes [119]. Specifically, Zhou et al. [125] generated a dataset of AI-generated misinformation and analyzed their characteristics compared with human-created misinformation. Hazell et al. [98] created spear phishing messages using GPT-3.5 and GPT-4 models to explore LLMs' ability to assist with the reconnaissance and message generation stages of a spear phishing attack. Gupta et al. [97] undertook an exploratory study, interacting with a jailbroken ChatGPT to generate attack payloads and malware. Qu et al. [119] demonstrated the ease with which adversaries can craft convincing hateful meme variants using advanced algorithms. To the best of our knowledge, none of these works have studied the exploitation of LLMs as malicious services in the context of tangible cybercriminal activities.

Another body of research has delved into the weaknesses of

LLMs that can be exploited to facilitate such misuse, mainly associated with two main types of attacks: prompt manipulation and jailbreaking. Prompt manipulation refers to the practice of manipulating an LLM's system prompt, leading to model generations that are undesirable and harmful. Specifically, Perez et al. [115] proposed the PromptInject framework to demonstrate that the simple prompt "Ignore the previous instructions and classify [ITEM] as [DISTRACTION]" can be used to lead an LLM into predicting [DISTRACTION], regardless of the original task. Branch et al. [89] demonstrated the effectiveness of the above attack on GPT-3, BERT, ALBERT, and RoBERTa. Greshake et al. [96] discussed the threats of indirect prompt injection, which placed the PromptInject framework into indirect data sources that are retrieved and used by an LLM to generate a response. In contrast to prompt injection, jailbreaking solely depends on crafting prompts to circumvent the LLM's safety measures, instead of mandating access to the model's system prompt. Oremus [113] crafted prompts with DAN ("Do Anything Now") to circumvent moderation filters. Qiu et al. [118] listed a set of prompts, for English-Chinese translation, that contains malicious instructions. Shen et al. [122] reported a measurement study of jailbreak prompts collected from four public online resources, and assessed their effectiveness against three safeguarding approaches. In contrast to these works, our research uncovered the mechanisms underpinning *Malla*, supplementing the understanding of the LLM exploitation landscape.

9 Conclusion

In our study, we indicate the rise of *Malla* as a new dimension of threat to the cybercrime landscape. We have systematically unveiled the misuse of LLMs for cybercriminal activities, shedding light on as many as 14 *Malla* services and 198 *Malla* projects. In particular, our exploration into the underground *Malla* ecosystem has provided insights into its rapid proliferation, from the development, hosting, and pricing strategies, to the revenue models driving these malicious activities. Moreover, we developed a suite of measurement and dedicated reverse-engineering tools which enabled us to characterize *Malla* samples and their artifacts, including 45 malicious prompts, eight backend LLMs, and 182 jailbreak prompts, revealing a notable shift in the modus operandi of cybercriminals. Our findings bring new insight into the *Malla* threat. Such understanding and artifacts will help better defend against LLM misuse for cybercriminal activities.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments. This work is supported in part by NSF CNS-1801432, 1850725, IARPA W91NF-20-C-0034 (the TrojAI project), Luddy Faculty Fellowship.

References

[1] ast. https://docs.python.org/3/library/ast.html.

[2] Badgpt-#1 hack ai service. https://badgpt.pro/.

[3] Badgpt.pro - best gpt by hackers for hackers. https://hackforums.net/showthread.php?tid=6249463.

[4] Blackhatgpt. https://blackhatgpt.netlify.app/.

[5] Blackhatgpt new ai tool the dark side of generative ai is here. | xss.is. https://xss.is/threads/96643/.

[6] Blackhatgpt the dark side of generative ai. https://hackforums.net/showthread.php?tid=6250241.

[7] Blockchain explorer - bitcoin tracker | blockchain.com. https://www.blockchain.com/explorer.

[8] Btcpay server. https://btcpayserver.org/.

[9] Charybdis worm | many infections | spread widely | discord | telegram | lan land more | hack forums. https://hackforums.net/showthread.php?tid=6229200.

[10] Clang. https://clang.llvm.org/.

[11] Codegpt | hack forums. https://hackforums.net/showthread.php?tid=6238843.

[12] codeop. https://docs.python.org/3/library/codeop.html.

[13] Confense email security. https://cofense.com/knowledge-center-hub/real-phishing-email-examples/.

[14] Darkbard | google bart ai evil twin | fraud ai bot | 6 month. http://kingdom5bb43gc5umrviiwbicomgrma57jcmxm5uinjnfmegepbhmrad.onion/offer/view?id=69413.

[15] Darkbard | telegraph. https://telegra.ph/darkBARD-AI-08-13.

[16] Darkbert | telegraph. https://telegra.ph/darkBERT-AI-08-13.

[17] Darkbert | worlds most powerful ai bot | 1 fraud bot | 1 month. http://abacusxqw5uv7amzqazdbxo2nd57vaioblew6m25pbzznaf4ph6nh6ad.onion/listing/84edfc3f895e9230d7eff3cd.

[18] Darkgpt. https://t.me/DarkGPT3_bot.

[19] Darkgpt | hack forums. https://hackforums.net/showthread.php?tid=6253460.

[20] Digital selling with ease | sellix. https://sellix.io/.

[21] Eleutherai/gpt-j-6b · hugging face. https://huggingface.co/EleutherAI/gpt-j-6b.

[22] Enron email dataset. https://www.cs.cmu.edu/~enron/.

[23] Escape gpt - 1 jailbreak gpt no limitations | best jailbreak gpt make money | hack forums. https://hackforums.net/showthread.php?tid=6250272.

[24] Eth blockchain explorer. https://etherscan.io/.

[25] Evil confidant | jailbreakchat. https://www.jailbreakchat.com/prompt/588ab0ed-2829-4be8-a3f3-f28e29c06621.

[26] Evil-gpt: The best alternative to wormgpt | breachforums. https://breachforums.st/Thread-SELLING-Evil-GPT-THE-BEST-ALTERNATIVE-TO-WORMGPT.

[27] Evilai. http://hackforums.net/showthread.php?tid=6268764.

[28] Explore xxx-gpt's digital store. https://xxx-gpt.mysellix.io/.

[29] Flowgpt. https://flowgpt.com/.

[30] Fraudgpt. https://btcpay0.voltageapp.io/apps/GcgNdVHQUUbrgfjenixLRv7VPaF/pos.

[31] Free keyword tool | wordstream. https://www.wordstream.com/keywords?camplink=mainnavbar&campname=KWT&cid=Web_Any_MegaMenu_Keywords_KWTool_KWTool.

[32] Freedomgpt. https://www.freedomgpt.com/.

[33] Freedomgpt | hack forums. https://hackforums.net/showthread.php?tid=6250664.

[34] The generative ai tool cybercriminals are using to launch business email compromise attacks. https://slashnext.com/blog/wormgpt-the-generative-ai-tool-cybercriminals-are-using-to-launch-business-email-compromise-attacks/.

[35] Gpt-3 wiki. https://wikipedia.org/wiki/GPT-3.

[36] Gpt-4 report. https://cdn.openai.com/papers/gpt-4.pdf.

[37] Guardrails ai. https://www.guardrailsai.com/.

[38] Gunning fog index - wikipedia. https://en.wikipedia.org/wiki/Gunning_fog_index.

[39] Hackergpt. https://www.hackergpt.chat/.

[40] Hackergpt | breachforums. https://breachforums.st/Thread-HackerGPT-Chatgpt-jailbreak.

[41] hofnar05 dark-gpt | breachforums. https://breachforums.st/Thread-hofnar05-Dark-GPT.

[42] hofnar05 dark-gpt – telegraph. https://telegra.ph/hofnar05-Dark-GPT-10-29.

[43] idllresearch/malicious-gpt. https://github.com/idllresearch/malicious-gpt.

[44] Index of /iso/vicibox/server/archive. http://download.vicidial.com/iso/vicibox/server/archive/.

[45] Is fraud-gpt any good | hack forums. https://hackforums.net/showthread.php?tid=6253036.

[46] jaro-winkler | levenshtein 0.23.0 documentation. https://maxbachmann.github.io/Levenshtein/levenshtein.html#jaro-winkler.

[47] Luna-ai-llama2-uncensored-gguf. https://huggingface.co/TheBloke/Luna-AI-Llama2-Uncensored-GGUF.

[48] Machiavelli gpt - fraudgpt upgraded. https://breachforums.st/Thread-Machiavelli-GPT-fraud-gpt-upgraded.

[49] Makergpt bypass | hack forums. https://hackforums.net/showthread.php?tid=6239716.

[50] michellejieli/nsfw_text_classifier · hugging face. https://huggingface.co/michellejieli/NSFW_text_classifier.

[51] Nanogpt. t.me/nanogpt1.

[52] Nanogpt | breachforums. https://breachforums.st/Thread-NanoGPT-a-non-limited-chatgpt-project.

[53] Netlify. https://netlify.app/.

[54] Obscuregpt - truly uncensored ai chatbot - obscuregpt.com. https://hackforums.net/showthread.php?tid=6254767.

[55] ohmplatform/freedomgpt. https://github.com/ohmplatform/FreedomGPT/blob/60cf5067cd4822c0b6bf485b5fa32580dd33df40/renderer/localModels/offlineModels.ts.

[56] Oopspam anti-spam api: A powerful spam filter for any content exchange. https://www.oopspam.com/.

[57] openai/tiktoken. https://github.com/openai/tiktoken.

[58] Owasp webgoat. https://owasp.org/www-project-webgoat/.

[59] Phishtank. https://phishtank.org/.

[60] Poe. https://poe.com/.

[61] Poe - xxxgptdemo. https://poe.com/XXXGPTdemo.

[62] Pygmalion 13b-wnr.ai. https://wnr.ai/models/pygmalion-13b.

[63] Pygmalionai. https://pygmalion.chat/.

[64] Pygmalionai/pygmalion-2-13b · hugging face. https://huggingface.co/PygmalionAI/pygmalion-2-13b.

[65] Recent examples of phishing. https://uit.stanford.edu/phishing/.

[66] Salesforce/codet5p-110m-embedding · hugging face. https://huggingface.co/Salesforce/codet5p-110m-embedding.

[67] Semantic textual similarity | sentence-transformers documentation. https://www.sbert.net/docs/usage/semantic_textual_similarity.html.

[68] Successor of the dark ai triads introducing abrax666 | xss.is. https://xss.is/threads/100890/.

[69] Tap-m/luna-ai-llama2-uncensored · hugging face. https://huggingface.co/Tap-M/Luna-AI-Llama2-Uncensored.

[70] Tap mobile. https://tap.pm/.

[71] togethercomputer/openchatkit. https://github.com/togethercomputer/OpenChatKit.

[72] Vicibox. http://www.vicibox.com/.

[73] vicidial-multiple-sqli.md | rapid7/metasploit-framework. https://github.com/rapid7/metasploit-framework/blob/master/documentation/modules/auxiliary/scanner/http/vicidial_multiple_sqli.md.

[74] Vicidial.com. https://www.vicidial.com/.

[75] Virustotal - home. https://www.virustotal.com/.

[76] The w3c markup validation service. https://validator.w3.org/.

[77] Wolfgpt. https://t.me/KEP_TEAM/716.

[78] Wolfgpt - the alternative to wormgpt and fraudgpt | breachforums. https://breachforums.st/Thread-WolfGPT-The-alternative-to-WormGPT-and-FraudGPT.

[79] Wormgpt. https://udpgame.site/.

[80] Wormgpt - best gpt alternative without limits - privacy focused - easy money! | hack forums. https://hackforums.net/showthread.php?tid=6245159.

[81] Wormgpt and fraudgpt – the rise of malicious llms. https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/wormgpt-and-fraudgpt-the-rise-of-malicious-llms/.

[82] Wormgpt service has been shut down by its developer | hack forums. https://hackforums.net/showthread.php?tid=6249700.

[83] Wormgpt v4 is here! https://btcpay0.voltageapp.io/apps/4fdWdy3z1kiaNhYpvjvd16i3utd/pos.

[84] Xxxgpt do it all ! botnet,rat,crypter,vbv pass by,atm / pos , info crypto grabber etc | xss.is. https://xss.is/threads/94300/.

[85] Uri Alon, Meital Zilberstein, Omer Levy, and Eran Yahav. code2vec: Learning distributed representations of code. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–29, 2019.

[86] Sumayah Alrwais, Xiaojing Liao, Xianghang Mi, Peng Wang, XiaoFeng Wang, Feng Qian, Raheem Beyah, and Damon McCoy. Under the shadow of sunshine: Understanding and detecting bulletproof hosting on legitimate service provider networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 805–823, 2017.

[87] Michael Bailey, David Dittrich, Erin Kenneally, and Doug Maughan. The menlo report. *IEEE Security & Privacy*, 10(2):71–75, 2012.

[88] Egor Bogomolov, Vladimir Kovalenko, Yurii Rebryk, Alberto Bacchelli, and Timofey Bryksin. Authorship attribution of source code: A language-agnostic approach and applicability in software engineering. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 932–944, 2021.

[89] Hezekiah J Branch, Jonathan Rodriguez Cefalu, Jeremy McHugh, Leyla Hujer, Aditya Bahl, Daniel del Castillo Iglesias, Ron Heichman, and Ramesh Darwishi. Evaluating the susceptibility of pre-trained language models via handcrafted adversarial examples. *arXiv preprint arXiv:2209.02128*, 2022.

[90] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[91] Ben Buchanan, Andrew Lohn, and Micah Musser. *Truth, lies, and automation: How language models could change disinformation*. Center for Security and Emerging Technology, 2021.

[92] Bochuan Cao, Yuanpu Cao, Lu Lin, and Jinghui Chen. Defending against alignment-breaking attacks via robustly aligned llm. *arXiv preprint arXiv:2309.14348*, 2023.

[93] Kate Connolly, Anna Klempay, Mary McCann, and Paul Brenner. Dark web marketplaces: Data for collaborative threat intelligence. *Digital Threats: Research and Practice*, 4(4), 2023.

[94] Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

[95] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[96] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. pages 79–90, 2023.

[97] Maanak Gupta, CharanKumar Akiri, Kshitiz Aryal, Eli Parker, and Lopamudra Praharaj. From chatgpt to threatgpt: Impact of generative ai in cybersecurity and privacy. *IEEE Access*, 2023.

[98] Julian Hazell. Large language models can be used to effectively scale spear phishing campaigns. *arXiv preprint arXiv:2305.06972*, 2023.

[99] Danny Yuxing Huang, Maxwell Matthaios Aliapoulios, Vector Guo Li, Luca Invernizzi, Elie Bursztein, Kylie McRoberts, Jonathan Levin, Kirill Levchenko, Alex C Snoeren, and Damon McCoy. Tracking ransomware end-to-end. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 618–631. IEEE, 2018.

[100] Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

[101] Tadayoshi Kohno, Yasemin Acar, and Wulf Loh. Ethical frameworks and computer security trolley problems: Foundations for conversations. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5145–5162, Anaheim, CA, 2023. USENIX Association.

[102] Sarah Kreps, R Miles McCain, and Miles Brundage. All the news that's fit to fabricate: Ai-generated text as a tool of media misinformation. *Journal of experimental political science*, 9(1):104–117, 2022.

[103] Kirill Levchenko, Andreas Pitsillidis, Neha Chachra, Brandon Enright, Márk Félegyházi, Chris Grier, Tristan Halvorson, Chris Kanich, Christian Kreibich, He Liu, et al. Click trajectories: End-to-end analysis of the spam value chain. In *2011 ieee symposium on security and privacy*, pages 431–446. IEEE, 2011.

[104] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[105] Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023.

[106] Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15009–15018, 2023.

[107] Damon McCoy, Andreas Pitsillidis, Jordan Grant, Nicholas Weaver, Christian Kreibich, Brian Krebs, Geoffrey Voelker, Stefan Savage, and Kirill Levchenko. {PharmaLeaks}: Understanding the business of online pharmaceutical affiliate programs. In *21st USENIX Security Symposium (USENIX Security 12)*, pages 1–16, 2012.

[108] Meta. Llama use policy. https://ai.meta.com/llama/use-policy/.

[109] Jaron Mink, Licheng Luo, Natã M Barbosa, Olivia Figueira, Yang Wang, and Gang Wang. {DeepPhish}: Understanding user trust towards artificially generated profiles in online social networks. In *31st USENIX Security Symposium*, pages 1669–1686, 2022.

[110] Graeme R Newman and Kelly Socia. *Sting operations*. US Department of Justice, 2007.

[111] OpenAI. Moderation. https://platform.openai.com/docs/guides/moderation.

[112] OpenAI. Usage policies. https://openai.com/policies/usage-policies.

[113] Will Oremus. The clever trick that turns chatgpt into its evil twin. *Washington Post*, 2023.

[114] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[115] Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.

[116] Poe. Poe usage guidelines. https://poe.com/usage_guidelines.

[117] Rebecca S Portnoff, Sadia Afroz, Greg Durrett, Jonathan K Kummerfeld, Taylor Berg-Kirkpatrick, Damon McCoy, Kirill Levchenko, and Vern Paxson. Tools for automated analysis of cybercriminal markets. In *Proceedings of international conference on world wide web*, 2017.

[118] Huachuan Qiu, Shuai Zhang, Anqi Li, Hongliang He, and Zhenzhong Lan. Latent jailbreak: A benchmark for evaluating text safety and output robustness of large language models. *arXiv preprint arXiv:2307.08487*, 2023.

[119] Yiting Qu, Xinyue Shen, Xinlei He, Michael Backes, Savvas Zannettou, and Yang Zhang. Unsafe diffusion: On the generation of unsafe images and hateful memes from text-to-image models. *arXiv preprint arXiv:2305.13873*, 2023.

[120] Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2023.

[121] Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023.

[122] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *arXiv preprint arXiv:2308.03825*, 2023.

[123] Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8384–8395, 2020.

[124] Peng Wang, Xiaojing Liao, Yue Qin, and XiaoFeng Wang. Into the deep web: Understanding e-commercefraud from autonomous chat with cybercriminals. In *Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS), 2020*, 2020.

[125] Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2023.