Multimodal Web Agents for Automated (Dark) Web Navigation

Mrunal Vibhute[1][a], Neol Gutierrez[1][b], Kristina Radivojevic[2][c] and Paul Brenner[1][d]
[1]*Center for Research Computing, University of Notre Dame, Notre Dame, Indiana, U.S.A.*
[2]*Computer Science and Engineering, University of Notre Dame, Notre Dame, Indiana, U.S.A.*
{*mvibhute, ngutier, kradivo, paul.r.brenner*}*@nd.edu*

Keywords: Agent Architectures, Autonomous Systems, Web Intelligence, Cybersecurity, Vision and Perception.

Abstract: Studying marketplaces hosted on the dark web is challenging due to the robust security measures these platforms use to protect user anonymity and prevent unauthorized access. While these marketplaces facilitate the trade of illegal goods and services, their use of CAPTCHAs, encryption, and the Tor network creates significant barriers for researchers attempting to gather data. We developed a software agent capable of overcoming the obstacles to automating the navigation of these marketplaces. The tool is specifically designed for ethical and legal research, helping cybersecurity experts identify and analyze dark Web activities to mitigate potential threats. Built using Python and Selenium WebDriver, and operating within the Tor Browser for anonymity, our agent uses Multimodal Large Language Models (MLLMs) to help automate the data acquisition process. These models can interpret both text and images, enabling the agent to solve complex CAPTCHAs that would otherwise block access to random bots in the marketplace. Once logged in, the agent automatically collects important data like vendor details, product categories, and prices. Additionally, the data collected in this process is publicly available as downloadable files in our GitHub repository. Our research also provides valuable insights into security trends and patterns within these marketplaces, shedding light on the activities taking place within these clandestine networks.

1 INTRODUCTION

The dark Web is known for facilitating illegal activities, with marketplaces trading goods and services. Unlike the surface web, which is indexed by search engines like Google, dark web marketplaces are hidden and unindexed (Saleem et al., 2022), which means that they cannot be discovered through typical searches. While these marketplaces share similarities with traditional e-commerce sites, their lack of indexing, use of cryptocurrencies, and reliance on encryption and security measures present significant challenges for researchers attempting to analyze these networks (Spagnoletti et al., 2022).

Several research papers highlighted the significant damage caused by dark web marketplaces such as human trafficking, child exploitation, and terrorist activities (Kaur and Randhawa, 2020; Horan and Saiedian, 2021; Radivojevic et al., 2024). These marketplaces are hubs for trading illicit goods and services, including malware such as data-stealing Trojans, remote access Trojans (RATs), and ransomware, leading to increased cyberattacks.

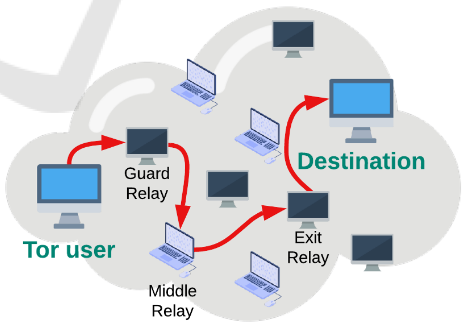

Figure 1: Working of TOR browser.

The dark web can only be accessed through specialized tools like the Tor Browser (Platzer and Lux, 2022). This browser uses the method of "onion routing" to ensure anonymity. The method involves three types of relay nodes or in simpler terms, devices in the network — each only aware of the preceding and following nodes — ensuring secure information transmission without revealing the source or destination.

[a] https://orcid.org/0009-0009-5843-7559
[b] https://orcid.org/0009-0001-9039-8897
[c] https://orcid.org/0000-0002-1645-5945
[d] https://orcid.org/0000-0002-2568-9786

Vibhute, M., Gutierrez, N., Radivojevic, K. and Brenner, P.
Multimodal Web Agents for Automated (Dark) Web Navigation.
DOI: 10.5220/0013171600003890
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 1*, pages 437-444
ISBN: 978-989-758-737-5; ISSN: 2184-433X
Proceedings Copyright © 2025 by SCITEPRESS – Science and Technology Publications, Lda.

As shown in Figure 1, the client device selects at least three random nodes from the Tor network. Encryption is applied in layers by the client node, with each node removing one layer as the request progresses toward the destination. All the nodes inside the Tor network can be middle and exit relays. The guard (entry) node must be stable and fast, while the exit relay can see the user's IP address but hides the user's identity from the destination site. Dark web sites use .onion URLs, which are 56-character alphanumeric strings only accessible via the Tor Browser.

Dark web marketplaces often employ sophisticated security measures, such as CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart). Modern CAPTCHAs also involve tasks such as identifying objects in pictures or solving puzzles. Complex, multi-step puzzle-like CAPTCHAs are usually found in dark marketplaces. These are particularly image-based, specifically created to block automated tools from accessing the sites, making manual data collection a slow and labor-intensive process.

According to a research project by the University of Amsterdam (Csuka et al., 2018) focused on breaking CAPTCHAs on the dark web, there are three common approaches to bypass them: 'Using a service that solves CAPTCHAs through human labor', 'Exploiting bugs in the implementation that allow the attacker to bypass the CAPTCHA' or 'Character recognition software to solve the CAPTCHA'. This paper presents an innovative approach to overcoming them using Multimodal Large Language Models (MLLMs). Selenium WebDriver is commonly used for automating navigation in dark web marketplaces, but manual CAPTCHA resolution still presents a major obstacle. While Selenium allows for scripted interaction with web pages, the lack of automated CAPTCHA-solving solutions limits the scalability of data acquisition efforts (Connolly et al., 2023; Bergman and Popov, 2023). Addressing this gap is critical for building fully automated workflows capable of real-time monitoring without human intervention. By integrating an MLLM with Python's Selenium WebDriver, we developed a web agent that can navigate dark web marketplaces, solve CAPTCHAs, and extract detailed product information. This data is made available through a public GitHub repository (https://github.com/crcresearch/DWData).

2 RELATED WORK

Dark web crawlers and data acquisition frameworks have evolved significantly to address the unique challenges posed by Tor-based marketplaces. CRATOR, for instance, presented a modular crawler designed to bypass anonymity constraints and CAPTCHAs while offering scalability for monitoring multiple marketplaces simultaneously (De Pascale et al., 2024). Similarly, the White House Market crawler demonstrated the utility of structured data extraction through automated techniques like XPath selectors and HTML parsing (Yannikos et al., 2022). Frameworks, such as BlackWidow (Schäfer et al., 2019), can monitor dark web activity by employing cloud-based microservices to facilitate real-time crawling and data collection (Ruiz Ródenas et al., 2023; Bergman and Popov, 2023).

There exist paid as well as open source tools that provide solutions that enable organizations to stay ahead of emerging dark web threats by continuously monitoring suspicious transactions, vendor activity, and marketplace trends (Burda et al., 2019). However, there is a delicate balance between tracking criminal activity and infringing on privacy. Hence, we need tools that crawl, collect, and analyze dark web data in real-time which also adhere to ethical norms and regulations.

CAPTCHAs, which were first introduced in 2000, have since been the focus of numerous studies aimed at developing automated solutions. As CAPTCHAs continue to evolve in complexity to deter bots, researchers have advanced techniques for automating their resolution (Motoyama et al., 2010). Given the prevalence of CAPTCHAs in restricting automated access, bypassing them has become a key research area. The OEDIPUS framework represents one of the more recent advancements, utilizing large language models (LLMs) to solve reasoning-based CAPTCHAs by breaking down AI-hard tasks into smaller, AI-easy components (Deng et al., 2024). Character recognition CAPTCHAs, one of the most common types, have been addressed with segmentation-free optical character recognition (OCR) models (Khatavkar et al., 2024).

CAPTCHAs containing image-based challenges often make use of CNN models (Yannikos and Heeger, 2024). These frameworks have been successful in solving arithmetic and puzzle-based CAPTCHAs on dark web marketplaces as well.(Ma et al., 2020; Yannikos et al., 2022). Generative adversarial networks (GANs) have also been applied to CAPTCHA-solving. DW-GAN, for instance, is capable of breaking dark web text-based CAPTCHAs by overcoming challenges such as background noise and variable-length characters(Zhang et al., 2022).

Finally, advances in natural language processing (NLP) have expanded the scope of CAPTCHA-solving efforts, with models such as BERT and RoBERTa being used to classify illicit content on dark web marketplaces. These models achieve high accuracy in categorizing illegal activities, further advancing the field of dark web data collection (Cascavilla et al., 2023). By integrating NLP techniques with CAPTCHA-solving frameworks, researchers continue to push the boundaries of what is possible in dark web data acquisition.

3 CONTEXTUAL OVERVIEW

This section explores the foundational technologies and systems that support web navigation and data acquisition from the dark web marketplaces. Topics discussed in this section set the stage for the detailed implementation of our semi-autonomous web agent.

3.1 Web Agent Infrastructure

For secure data acquisition from dark web websites, we utilize a burner laptop and a VPN, opting not to use a Virtual Machine (VM), as proposed in the three-layered protection strategy by (Connolly et al., 2023). We rely on the Tor Browser in combination with Selenium WebDriver, rather than conventional browsers such as Firefox or Chromium. This approach offers several advantages: it allows for faster interaction with dark web marketplace components, hides the user's IP address—commonly exposed during data scraping—and reduces the likelihood of the agent being blocked by IP-based restrictions. Additionally, Tor's built-in privacy features, such as "Letterboxing", help prevent fingerprinting by masking screen dimensions and other device-specific information, thus maintaining anonymity. Algorithm 1 gives a glimpse of the working of our web agent - a Python script using Selenium Webdriver.

3.2 Dark Marketplace

We chose to automate data acquisition for Dark Matter due to its well-structured nature, its large user base of over 26,000 registered users, more than 900 active vendors, and approximately 16,600 product listings. Furthermore, this marketplace has been operational for over 800 days, making it one of the oldest among its competitors, and thus an ideal candidate for study. Notably, its CAPTCHA system closely resembles the security challenges found in newer marketplaces, making it a relevant case for testing automated CAPTCHA-solving techniques.

```
Data: .onion url of marketplace, code to
      solve CAPTCHAs present in
      marketplace
Result: Automated process of Data
        acquisition from a marketplace
initialization;
while inside a dark marketplace do
    read contents of the page;
    if Containers found then
        Read information inside html tags;
        Append corresponding information in
          CSV file;
        Find next container;
    else
        if CAPTCHA found then
            Call method to solve the
              CAPTCHA challenge;
        else
            Reload the page;
        end
    end
end
```

Algorithm 1: Marketplace Agent Algorithm.

3.3 Marketplace CAPTCHAs

Automating the process of solving CAPTCHAs on dark marketplaces presents a significant challenge for web crawlers and data acquisition agents (Connolly et al., 2023; Hayes et al., 2018). Especially when current marketplaces require the user to solve a minimum of one and a maximum of three types of different CAPTCHAs to see the contents. According to our recent observations, these three categories are as follows: **DDOS CAPTCHA** - They protect the webpage against DDoS attacks by requiring users to solve challenges before accessing a site, limiting automated requests. They are often more complex and use either distorted text, image puzzles, or 2-factor authentication to deter bots. **Login CAPTCHA** - They prevent bots from brute-forcing credentials, often using text-based or reCAPTCHA challenges. These simpler CAPTCHAs can often be solved using neural networks, machine learning, or OCR techniques (Dinh and Ogiela, 2022). **Anti-Phishing CAPTCHA** - These CAPTCHAs prevent phishing by presenting challenges like scrambled URLs with replaced characters (e.g., * or _), phishing detection puzzles, or object identification tasks. They require manual input from users to ensure authenticity and block automated scripts.

3.4 Inclusion of Multimodal LLMs

The multimodal Large Language Models (LLMs) utilized in this research are OpenAI's GPT-4o and Anthropic's Claude.AI, both of which are equipped with the capability to process visual inputs and produce textual outputs. This multimodal capability is crucial for addressing the complex CAPTCHA challenges prevalent in dark web marketplaces.

Multimodal LLMs can be trained or adapted to solve image-based tasks using several approaches namely: Prompt Engineering, Few-shot Learning, and Fine-tuning. In our research, we use LLMs with few-shot learning to solve CAPTCHAs on dark web marketplaces. While the type of CAPTCHA remains the same, the position and appearance of elements change a little each time. Few-shot learning handles these variations efficiently without the high cost of fine-tuning.

4 IMPLEMENTATION OF AN AUTOMATED WEB AGENT

Dark Matter Marketplace physical products include narcotics, prescription drugs, steroids, counterfeit documents, fake currencies, and jewelry, while digital products consist of hacking tools, software, tutorials, bank account and credit card information, security software, malware, game keys, and e-books. The marketplace provides these illicit goods through various listings, which our agent automatically scrapes without human intervention. It is capable of extracting key product details like - Product name, Quantity available / offered, Price, Vendor name, country or region from where the product is dispatched, regions to which the vendor ships the product, where the product is restricted and payment options such as Escrow, Finalize Early (FE), and MultiSig transactions.

4.1 Agent Architecture

We designed the architecture of our automated web agent to navigate dark web marketplaces, bypass CAPTCHAs, and acquire data. Figure 2 contains the architecture of our agent. The key component controlling this process is the **main.py script**, which acts as the orchestrator, calling different modules and functions, each responsible for a specific task.
Web Scraping Module (webscraper_class.py): The webscraper_class.py module contains a custom python class that creates a selenium session, imports market functions, handles errors, and creates a CSV file for market listings module makes sure that the Tor browser connections are done and the scraper is ready to scrape in the headless mode. It also generates a CSV file to store the raw data obtained from the marketplace. This module is mainly responsible for handling the connection-related issues that occur either before or in the middle of the execution. Once the connection is established and the CSV is created, the code enters back into the main module to perform the next tasks.
Market File Module (market_file.py): This module has custom methods designed to collect data from the marketplace's web pages. These methods form the backbone of the data extraction process. The class includes functions for logging in, choosing the category of products, scraping, and navigating the marketplace, and leveraging a Selenium WebDriver to interact with the website. These functions are called in the same sequence as mentioned above.
Selenium Webdriver: Selenium is used as the primary tool to interact with the marketplace website. First, it opens the browser and navigates to the desired mirror link of the marketplace. If login credentials are required, the login() function is triggered, which automates the login process by entering stored credentials. The Selenium driver then identifies the CAPTCHA challenge that appears after login.
CAPTCHA Handling (solve_captcha.py): Once the CAPTCHA is detected, Selenium captures an image of the CAPTCHA puzzle and passes it back to the main.py script. From there, the image is forwarded to the solve_captcha.py module. This module sends an API request to OpenAI, which uses its visual processing capabilities to deduce the CAPTCHA code. Once the CAPTCHA is solved, the web agent enters the solution back into the website through the Selenium driver, allowing access to the marketplace's contents.
Page Navigation and Data Extraction: After solving the CAPTCHA, the Selenium driver gains access to the main content of the marketplace. The page_scrape() function is responsible for extracting data from the webpage containers, which contain important product details such as price, quantity, and category. The XPath is used to navigate these containers and retrieve the relevant information. Once collected, this data is sent back to the main.py file to be stored as a CSV file, making the data available for analysis.
Handling Multiple Pages and Categories: The agent is designed to handle both pagination and categorization. The goto_next_page() function allows the agent to navigate to subsequent pages within the marketplace, ensuring all product listings are captured. Additionally, the nav_menu() function helps distinguish between different categories of products, such

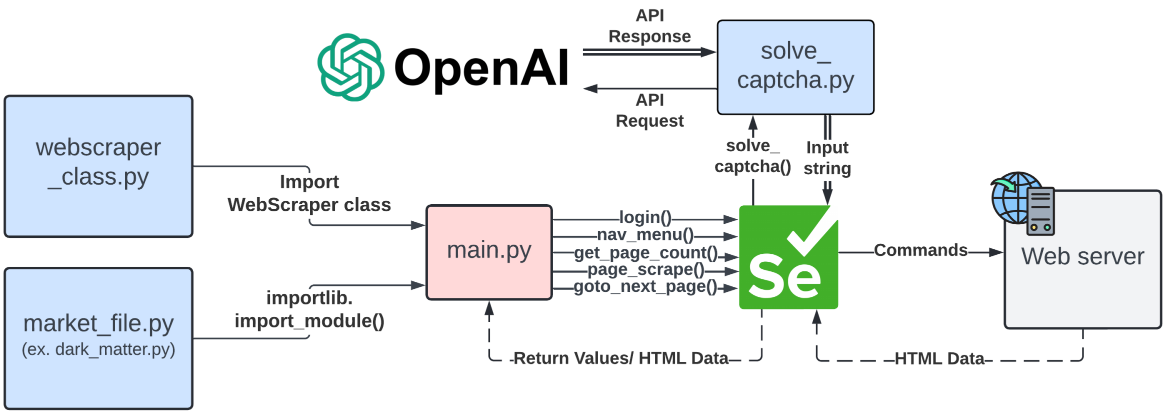

Figure 2: Modules and functions included in the Working Agent for Dark Matter Marketplace.

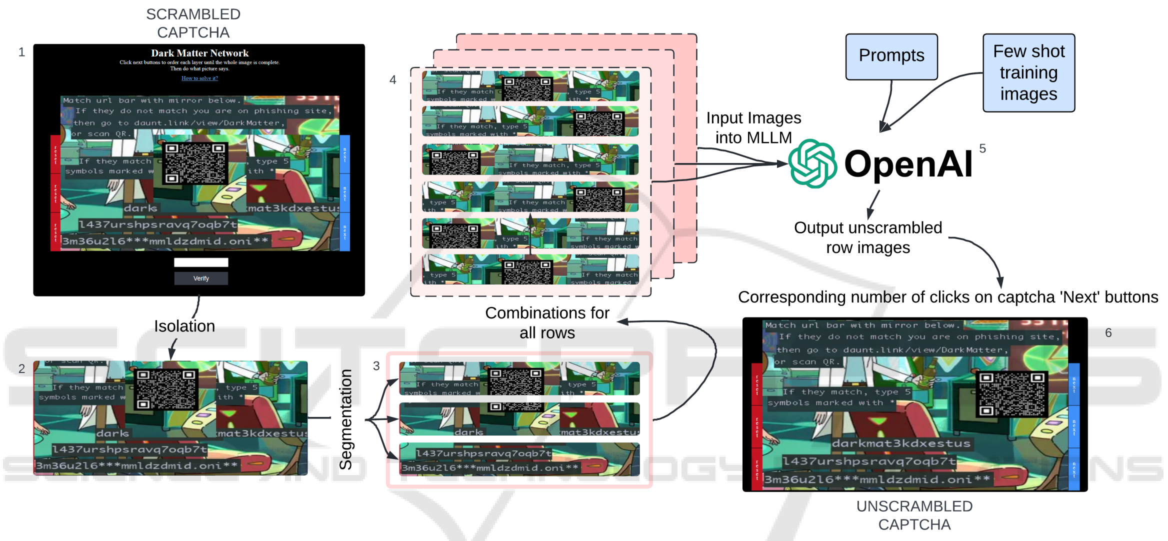

Figure 3: CAPTCHA solving using MLLM.

as physical or digital goods, which may have slight variations in their tags. The web agent first scrapes one category completely, and then moves on to the next category, repeating the process until all data is collected.

Dealing with Errors: Sometimes, the website session might expire, or an error might occur. In these cases, the agent can refresh the page and continue collecting data without needing human intervention.

In summary, this web agent can automatically log in, solve security puzzles, and gather important data from the dark web. The information is then saved in a structured file format (CSV), which can be analyzed to understand patterns in these underground marketplaces.

4.2 Solving the CAPTCHA Using Multimodal LLM Integration

The CAPTCHA consists of a 3x3 grid of horizontally scrambled images, which the user must unscramble by clicking the ”Next” and ”Reset” buttons for each row. An alphanumeric keyword, embedded in the unscrambled image, must then be entered into a provided input box. To bypass this CAPTCHA, we integrated a Large Multimodal Model (LMM) capable of solving the puzzle autonomously. When the solve_captcha() module is triggered, the agent coordinates with LMM models, such as OpenAI and/or Claude.ai, to automate CAPTCHA solving. This involves taking screenshots, creating permutations of scrambled rows, making API calls, and retrying until the CAPTCHA is successfully solved. Figure 3 demonstrates the steps involved in CAPTCHA solving using the OpenAI model.

The steps involved are as follows:

1. **Isolate the CAPTCHA Image.** Using the web driver, the agent captures the part of the screen containing only the scrambled image, excluding headings and other texts on the webpage. This process is referred to as ”Isolation” in the figure.
2. **Divide the Image into Sections.** Since the images are scrambled horizontally, only the rows need to be rearranged. The user presses the ’Next’ button to shuffle each row until an unscrambled combination is found. This process is repeated for all three rows. The agent divides the image into three horizontal rows for automated processing.
3. **Unscramble the Parts Using LMM.** For each row, six different combinations are generated, with only one being the correct unscrambled version. The agent tracks the number of clicks for each combination, and the LMM analyzes all six versions, selecting the correct unscrambled image based on learned visual patterns. Few-shot learning is employed by providing the model with labeled examples of scrambled and unscrambled rows.
4. **Reassemble the CAPTCHA.** Once all rows are unscrambled, the image is reassembled. The CAPTCHA on the marketplace is then reset, and the agent uses the recorded number of ’Next’ clicks to replicate the unscrambled state.
5. **Identify the Keyword.** The LMM uses OCR to extract the highlighted text or URL within the reassembled image. It then compares the URL with known mirror links of the marketplace to detect the embedded alphanumeric keyword.
6. **Solve the CAPTCHA.** The identified keyword is entered into the CAPTCHA input box, allowing the agent to proceed with marketplace navigation.

This process enables the agent to bypass CAPTCHA without human intervention, overcoming one of the key anti-automation mechanisms in the Dark Matter marketplace.

5 DISCUSSION

To test and measure the performance of our agent, we experiment by using our agent on the Dark Matter Marketplace. We successfully performed our no-human intertwined scrape of the entire marketplace within 92 minutes, while our agent successfully solved CAPTCHAs within 1 minute. This marketplace also has only one complicated antiphishing CAPTCHA which also works as a DDOS CAPTCHA. Assuming a first-attempt success, 19 API calls were required per CAPTCHA, calculated as 3 rows * 6 combinations per row + 1 for text identification. The total number of API calls is always going to be a multiple of the number of API calls required to solve all the subtasks.

Our approach captured around 17,000 listings, containing digital and physical products. This dataset has been made publicly available for researchers. This contribution demonstrates the effectiveness of automation in navigating and extracting data from complex dark web environments. The graph in Figure 4 illustrates the time taken (in seconds) by our agent to bypass the CAPTCHA. We tested the CAPTCHA solving using gpt-4o and claude-3-5-sonnet-20240620 MLLM models and found a combination of both for the rows works effectively - gpt-4o for row1 and claude for rows 2 and 3. While testing we recorded the total time for each session, which includes one or more attempts. If an attempt within a session fails, a new CAPTCHA is generated, and the process continues with the next attempt within the same session.

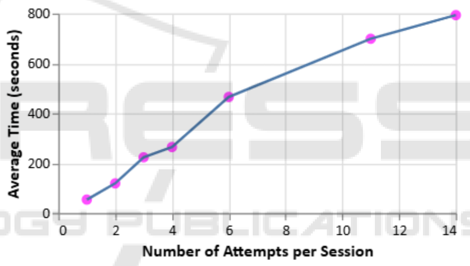

Figure 4: Graph of the number of CAPTCHA solving attempts taken to access the marketplace versus Median of the data of time in seconds.

6 CONCLUSION

The rise of anonymizing technologies such as the Tor browser has enabled enhanced privacy protection but has also created a space where illicit activities can operate under the radar. While existing automated data monitoring tools can solve certain CAPTCHAs, they often rely on machine learning models or neural networks that require extensive datasets for training. However, such datasets may not always be available for the diverse and evolving CAPTCHAs encountered on the dark web. We address this problem by combining Selenium with MLLMs, developing an agent capable of solving CAPTCHAs with minimal training images and prompt-based interactions. This agent holds great potential to evolve into a versatile CAPTCHA-solving tool powered by intelligent systems such as MLLMs.

Our research has the potential to contribute to the area of cyber security by building an automated agent designed to extract meaningful data from dark web marketplaces while adhering to ethical standards. By supporting certified authorities in the collection and monitoring of real-time data, trends in the sale of products such as arms, biological weapons, or killing contracts can be tracked. This allows authorities to gather potential leads on terrorist attacks or harmful activities, enhancing their ability to address such threats preemptively. This tool can also help identify emerging patterns in malware trade, providing insights into improving security software, mitigating vulnerabilities, and helping in the fight against cybercrimes. Although our agent solves the anti-phishing CAPTCHA of the Dark Matter Marketplace efficiently, it may occasionally require multiple attempts since OpenAI API may flag some CAPTCHA images for ethical reasons. Additionally, we open-source the data collected in the scraping process to contribute to the research community.

Looking forward, we aim to enhance the agent's reliability, speed, and adaptability, particularly in handling a broader range of CAPTCHAs across both the dark and clear web. The ultimate vision is to fully integrate MLLMs, minimizing the need for extensive training while offering more flexible and efficient CAPTCHA-solving capabilities.

REFERENCES

Bergman, J. and Popov, O. B. (2023). Exploring dark web crawlers: A systematic literature review of dark web crawlers and their implementation. *IEEE Access*, 11:35914–35933.

Burda, P., Boot, C., and Allodi, L. (2019). Characterizing the redundancy of darkweb. onion services. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–10.

Cascavilla, G., Catolino, G., and Sangiovanni, M. (2023). Illicit darkweb classification via natural-language processing: Classifying illicit content of webpages based on textual information. *arXiv preprint arXiv:2312.04944*.

Connolly, K., Klempay, A., McCann, M., and Brenner, P. (2023). Dark web marketplaces: Data for collaborative threat intelligence. *Digital Threats: Research and Practice*, 4(4):1–12.

Csuka, K., Gaastra, D., and de Bruijn, Y. (2018). Breaking captchas on the dark web.

De Pascale, D., Cascavilla, G., Tamburri, D. A., and Van Den Heuvel, W. J. (2024). Crator a crawler for tor: Turning dark web pages into open source intelligence. In *European Symposium on Research in Computer Security*, pages 144–161. Springer.

Deng, G., Ou, H., Liu, Y., Zhang, J., Zhang, T., and Liu, Y. (2024). Oedipus: Llm-enchanced reasoning captcha solver. *arXiv preprint arXiv:2405.07496*.

Dinh, N. and Ogiela, L. (2022). Human-artificial intelligence approaches for secure analysis in captcha codes. *EURASIP Journal on Information Security*, 2022(1):8.

Hayes, D. R., Cappa, F., and Cardon, J. (2018). A framework for more effective dark web marketplace investigations. *Information*, 9(8).

Horan, C. and Saiedian, H. (2021). Cyber crime investigation: Landscape, challenges, and future research directions. *Journal of Cybersecurity and Privacy*, 1(4):580–596.

Kaur, S. and Randhawa, S. (2020). Dark web: A web of crimes. *Wireless Personal Communications*, 112:2131–2158.

Khatavkar, V., Velankar, M., and Petkar, S. (2024). Segmentation-free connectionist temporal classification loss based ocr model for text captcha classification. *arXiv preprint arXiv:2402.05417*.

Ma, Y., Zhong, G., Liu, W., Sun, J., and Huang, K. (2020). Neural captcha networks. *Applied Soft Computing*, 97:106769.

Motoyama, M., Levchenko, K., Kanich, C., McCoy, D., Voelker, G. M., and Savage, S. (2010). Re:{CAPTCHAs—Understanding}{CAPTCHA-Solving} services in an economic context. In *19th USENIX Security Symposium (USENIX Security 10)*.

Platzer, F. and Lux, A. (2022). A synopsis of critical aspects for darknet research. In *Proceedings of the 17th international conference on availability, reliability and security*, pages 1–8.

Radivojevic, K., Connolly, K., Klempay, A., and Brenner, P. (2024). Dark web and internet freedom: navigating the duality to facilitate digital democracy. *Journal of Cyber Policy*, pages 1–16.

Ruiz Ródenas, J. M., Pastor-Galindo, J., and Gómez Mármol, F. (2023). A general and modular framework for dark web analysis. *Cluster Computing*, pages 1–17.

Saleem, J., Islam, R., and Kabir, M. A. (2022). The anonymity of the dark web: A survey. *IEEE Access*, 10:33628–33660.

Schäfer, M., Fuchs, M., Strohmeier, M., Engel, M., Liechti, M., and Lenders, V. (2019). Blackwidow: Monitoring the dark web for cyber security information. In *2019 11th International Conference on Cyber Conflict (CyCon)*, volume 900, pages 1–21.

Spagnoletti, P., Ceci, F., and Bygstad, B. (2022). Online black-markets: An investigation of a digital infrastructure in the dark. *Information Systems Frontiers*, pages 1–16.

Yannikos, Y. and Heeger, J. (2024). Captchas on darknet marketplaces: Overview and automated. *Electronic Imaging*, 36:1–6.

Yannikos, Y., Heeger, J., and Steinebach, M. (2022). Data acquisition on a large darknet marketplace. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–6.

Zhang, N., Ebrahimi, M., Li, W., and Chen, H. (2022). Counteracting dark web text-based captcha with generative adversarial learning for proactive cyber threat intelligence. *ACM Transactions on Management Information Systems (TMIS)*, 13(2):1–21.