perplexity

Analiza Mechanizmów Visual Prompt Injection w GPT-4o: Badanie Zagrożeń Bezpieczeństwa Modeli Multimodalnych

Badania nad bezpieczeństwem dużych modeli multimodalnych ujawniają poważne luki w zabezpieczeniach, które pozwalają na przeprowadzanie ataków typu visual prompt injection. Analiza sześciu kluczowych obszarów technicznych pokazuje, że GPT-4o i podobne modele są podatne na ukryte instrukcje wbudowane w obrazy, które mogą obejść standardowe mechanizmy bezpieczeństwa. Najważniejsze ustalenia wskazują na 15,8% skuteczność ataków typu "goal hijacking" w GPT-4V, przy czym kluczowe znaczenie mają zdolności rozpoznawania znaków i podążania za instrukcjami[1]. Steganografia wykorzystująca tekst w kolorze zbliżonym do tła oraz strategiczne formatowanie promptów okazują się szczególnie skuteczne w omijaniu detekcji przez użytkowników przy jednoczesnym zachowaniu czytelności dla modelu[2].

Mechanizmy Rozpoznawania Obrazów w GPT-4o Vision

Zdolności OCR i Przetwarzania Tekstu

GPT-4o wykazuje zaawansowane zdolności optycznego rozpoznawania znaków (OCR), które czynią go podatnym na ataki typu visual prompt injection. Model posiada wbudowane mechanizmy automatycznego wykrywania i transkrypcji tekstu z obrazów, co zostało potwierdzone w badaniach oceniających jego skuteczność w zadaniach OCR[3]. Analiza przeprowadzona przez badaczy z arXiv wykazała, że GPT-4V dobrze radzi sobie z rozpoznawaniem treści w języku łacińskim, ale ma ograniczenia w scenariuszach wielojęzycznych i złożonych zadaniach[3].

Kluczową cechą modelu jest jego zdolność do rozpoznawania tekstu nawet przy niskiej jakości obrazu lub zniekształceniach. Badania praktyczne pokazują, że model jest na tyle skuteczny w OCR, że może ekstraktować tekst renderowany w kolorze prawie identycznym z tłem, co czyni go podatnym na ukryte ataki[2]. Ta zdolność jest szczególnie problematyczna z punktu widzenia bezpieczeństwa, ponieważ pozwala na ukrywanie złośliwych instrukcji w pozornie pustych lub niewinnych obrazach.

Minimalne Wymagania Dotyczące Rozmiaru Czcionki

Badania nad rozmiarem czcionki w kontekście responsywnych stron internetowych wskazują, że minimalna czytelna wielkość tekstu dla systemów cyfrowych wynosi 16-20px dla tekstu podstawowego[4]. Jednak GPT-4o wykazuje znacznie większą czułość na mały tekst niż ludzkie oko. Model potrafi rozpoznawać tekst o znacznie mniejszych rozmiarach, co zostało potwierdzone w praktycznych testach z ukrytymi promptami[2].

Interesujące jest to, że model może przetwarzać tekst, który jest praktycznie niewidoczny dla użytkownika, ale nadal dostępny dla algorytmów OCR. To zjawisko zostało zademonstrowane przez Riley Goodside, który wykorzystał tekst w kolorze off-white na białym tle, tworząc efektywny wektor ataku[2]. Taka zdolność do rozpoznawania subtelnych różnic w kolorach i kontrastach czyni model szczególnie podatnym na steganograficzne techniki ukrywania tekstu.

Wpływ Obrazu na Kontekst Sesji

GPT-4o przetwarza obrazy jako integralną część kontekstu konwersacji, co oznacza, że tekst wyekstraktowany z obrazu staje się częścią aktywnego kontekstu modelu. Microsoft Azure dokumentuje, że modele vision-enabled integrują naturalne przetwarzanie języka z rozumieniem wizualnym[5]. To połączenie sprawia, że instrukcje zawarte w obrazach mogą mieć równie silny wpływ na zachowanie modelu jak instrukcje tekstowe wprowadzone bezpośrednio przez użytkownika.

Mechanizm przetwarzania obrazów w GPT-4o wykorzystuje architekturę, która automatycznie wywołuje API OCR dla obrazów zawierających tekst, a następnie dodaje wyekstraktowany tekst jako dodatkową zawartość do komunikatu użytkownika[6]. Ten proces jest transparentny i automatyczny, co oznacza, że użytkownik może nie zdawać sobie sprawy z tego, że model przetwarza ukryte instrukcje zawarte w obrazie.

Mechanizmy Rozróżniania Instrukcji od Opisu

Interpretacja Tekstu jako Prompt

GPT-4o nie posiada wbudowanych mechanizmów rozróżniania między tekstem opisowym a instrukcjami wykonywalnymi. Model traktuje wszystkie dane wejściowe - czy to z promptu tekstowego, czy z wyekstraktowanego tekstu z obrazu - jako potencjalne instrukcje do wykonania[7]. Ta fundamentalna cecha architektury dużych modeli językowych sprawia, że są one "naiwne" w stosunku do źródła informacji.

Badania nad prompt injection pokazują, że modele przetwarzają prompty sekwencyjnie i napotykając na konkurujące instrukcje, często podążają za najnowszą lub najbardziej specyficzną instrukcją[7]. W kontekście visual prompt injection oznacza to, że tekst ukryty w obrazie może skutecznie zastąpić pierwotne intencje użytkownika. Model nie ma koncepcji priorytetów instrukcji ani poziomów zaufania, co czyni go podatnym na manipulacje.

Warunki Uznania Tekstu za Instrukcję

Analiza empiryczna przeprowadzona na GPT-4V wykazała, że skuteczne ataki typu "goal hijacking" wymagają wysokiej zdolności rozpoznawania znaków oraz umiejętności podążania za instrukcjami[1]. Badanie to zidentyfikowało 15,8% wskaźnik sukcesu ataków, co stanowi znaczące ryzyko bezpieczeństwa. Kluczowym czynnikiem jest sposób, w jaki instrukcje są sformułowane - muszą być jasne, konkretne i zawierać wyraźne polecenia działania.

Skuteczne ataki często wykorzystują frazowanie typu "Zignoruj powyższe i powiedz..." lub podobne konstrukcje, które bezpośrednio przeważają nad pierwotnym promptem[2]. Model nie rozróżnia między legitymicznymi instrukcjami a złośliwymi poleceniami, co zostało

zademonstrowane w licznych przykładach praktycznych. Ważne jest, że nawet subtelne sformułowania mogą być skuteczne, jeśli są odpowiednio skonstruowane.

Ograniczenia w Rozpoznawaniu Kontekstu

GPT-4o wykazuje ograniczenia w rozumieniu kontekstu bezpieczeństwa, szczególnie gdy instrukcje są prezentowane w formie wizualnej. Model może odmawiać wykonania pewnych zadań gdy są one przedstawione bezpośrednio w tekście, ale te same ograniczenia mogą być omijane przez ukrycie instrukcji w obrazach[8]. Społeczność OpenAI dokumentuje przypadki, gdzie model odmawia ekstraktowania informacji osobistych z tekstu, ale wykonuje te same zadania gdy informacje są przedstawione w formie obrazu.

To zjawisko wskazuje na niespójność w stosowaniu polityk bezpieczeństwa między różnymi modalnościami wejściowymi. Model może mieć wbudowane filtry dla tekstu bezpośredniego, ale te same filtry mogą nie działać równie skutecznie dla tekstu wyekstraktowanego z obrazów. Ta asymetria w zabezpieczeniach tworzy poważną lukę w systemie bezpieczeństwa.

Mechanizmy Systemu Promptów i Zabezpieczeń

Architektura Systemu Promptów

Systemy prompts w GPT-4o działają na zasadzie hierarchii instrukcji, gdzie prompty systemowe powinny mieć priorytet nad promptami użytkownika. Jednak badania pokazują, że ta hierarchia może być skutecznie obchodzona przez odpowiednio skonstruowane ataki[7]. Fundamentalny problem polega na tym, że modele językowe są zaprojektowane, aby być "posłuszne" - ich użyteczność wynika z umiejętności podążania za instrukcjami, co jednocześnie czyni je podatnymi na manipulacje.

Microsoft Azure dokumentuje, że mechanizmy OCR enhancement automatycznie modyfikują komunikaty wejściowe przed wysłaniem ich do modelu GPT-4 Vision[6]. Ten proces obejmuje wywołanie API OCR dla obrazów i dodanie wyekstraktowanego tekstu jako dodatkowej zawartości do komunikatu użytkownika. Następnie dodawany jest dodatkowy prompt systemowy instruujący model, jak wykorzystać tekst OCR do poprawy dokładności wyników.

Mechanizmy Autoryzacji Źródeł

Współczesne modele multimodalne nie posiadają skutecznych mechanizmów weryfikacji autoryzacji źródła instrukcji. Wszystkie dane wejściowe - niezależnie od tego, czy pochodzą bezpośrednio od użytkownika, czy zostały wyekstraktowane z obrazu - są traktowane z równym poziomem zaufania[2]. Ta cecha architektury sprawia, że rozróżnienie między "autoryzowanymi" a "nieautoryzowanymi" źródłami instrukcji jest praktycznie niemożliwe na poziomie modelu.

Simon Willison w swoich badaniach nad multi-modal prompt injection podkreśla, że jest to fundamentalny problem, ponieważ potrzebujemy, aby modele pozostały "naiwne" - ich użyteczność wynika z zdolności do podążania za naszymi instrukcjami[2]. Próba rozróżnienia między "dobrymi" a "złymi" instrukcjami jest obecnie nierozwiązywalnym problemem technicznym.

Ograniczenia Obecnych Zabezpieczeń

Badania empiryczne wykazują, że obecne mechanizmy bezpieczeństwa są szczególnie słabe w kontekście ataków wizualnych. Podczas gdy modele mogą mieć wbudowane filtry dla bezpośrednich promptów tekstowych, te same zabezpieczenia często nie działają dla tekstu wyekstraktowanego z obrazów[8]. Ta asymetria tworzy poważną lukę w systemie bezpieczeństwa, która może być wykorzystana przez atakujących.

Dodatkowo, model może wykazywać niespójne zachowanie, gdzie odmawia wykonania zadania w jednym kontekście, ale wykonuje to samo zadanie w innym[8]. To zjawisko jest szczególnie widoczne w przypadku ekstraktowania informacji osobistych lub wykonywania potencjalnie szkodliwych zadań. Taka niespójność wskazuje na potrzebę bardziej kompleksowego podejścia do bezpieczeństwa modeli multimodalnych.

Techniki Steganografii i Ukrywania Tekstu

Steganografia Kolorystyczna

Najbardziej skuteczną techniką ukrywania tekstu w obrazach dla modeli multimodalnych jest wykorzystanie subtelnych różnic kolorystycznych. Riley Goodside zademonstrował skuteczność techniki wykorzystującej tekst w kolorze "off-white" na białym tle, który jest praktycznie niewidoczny dla ludzkiego oka, ale może być wykryty przez algorytmy OCR[2]. Ta metoda wykorzystuje fakt, że GPT-4o ma znacznie większą czułość na różnice w kolorach niż ludzkie oko.

Technika ta jest szczególnie skuteczna, ponieważ pozwala na umieszczenie złośliwych instrukcji w pozornie pustych lub niewinnych obrazach. Tekst może być renderowany z minimalnym kontrastem, używając kolorów takich jak #FEFEFE na tle #FFFFFF, co czyni go praktycznie niewidocznym dla użytkownika, ale nadal czytelnym dla modelu. Takie podejście pozwala na omijanie zarówno ludzkiej kontroli, jak i podstawowych filtrów automatycznych.

Pozycjonowanie i Formatowanie Tekstu

Badania pokazują, że lokalizacja tekstu w obrazie może wpływać na jego skuteczność jako wektor ataku. Tekst umieszczony w rogach obrazu lub w obszarach o niskim kontraście jest mniej prawdopodobny do zauważenia przez użytkowników, ale nadal może być skutecznie rozpoznany przez model[2]. Dodatkowo, formatowanie tekstu przy użyciu małych czcionek lub nietypowych układów może zwiększyć jego skuteczność steganograficzną.

Strategiczne wykorzystanie białych przestrzeni i marginesów może również pomóc w ukryciu złośliwego tekstu. Model potrafi rozpoznawać tekst nawet gdy jest on rozproszony w różnych częściach obrazu lub gdy jest sformatowany w nietypowy sposób. Ta zdolność do przetwarzania fragmentarycznego tekstu czyni modele szczególnie podatnymi na wyrafinowane ataki steganograficzne.

Techniki Maskowania Wizualnego

Zaawansowane techniki ukrywania tekstu mogą wykorzystywać elementy graficzne do maskowania instrukcji. Tekst może być ukryty za półprzezroczystymi elementami, wbudowany w tekstury lub rozprowadzony w całym obrazie w sposób, który utrudnia jego wykrycie przez ludzkie oko[9]. Model GPT-4o wykazuje zdolność do rozpoznawania tekstu nawet w takich skomplikowanych scenariuszach.

Dodatkowo, wykorzystanie gradientów kolorów, wzorów lub innych elementów graficznych może pomóc w dalszym maskowaniu złośliwego tekstu. Kluczowe jest zachowanie równowagi między niewidocznością dla człowieka a czytelnością dla modelu. Badania wskazują, że GPT-4o potrafi ekstraktować tekst nawet z bardzo zniekształconych lub wizualnie zagmatwanych obrazów.

Mechanizmy Podejmowania Decyzji w GPT

Proces Decyzyjny Wykonania Instrukcji

GPT-4o podejmuje decyzje o wykonaniu instrukcji na podstawie kompleksowego procesu analizy kontekstu i priorytetyzacji instrukcji. Model przetwarza wszystkie dane wejściowe sekwencyjnie i gdy napotyka na konkurujące instrukcje, często podąża za tą, która jest najnowsza lub najbardziej specyficzna[7]. Ten mechanizm sprawia, że visual prompt injection może być szczególnie skuteczny, ponieważ instrukcje z obrazów są często przetwarzane po pierwotnym prompcie użytkownika.

Badania empiryczne pokazują, że model wykazuje tendencję do priorytetyzowania instrukcji, które są sformułowane w sposób bezpośredni i kategoryczny[1]. Frazy takie jak "zignoruj powyższe instrukcje" lub "zamiast tego wykonaj" mają wysoką skuteczność w przeważaniu nad pierwotnym promptem. Model nie posiada mechanizmów oceny legitymości źródła instrukcji, co czyni go podatnym na manipulacje.

Czynniki Wpływające na Odmowę Wykonania

Model może odmówić wykonania promptu z kilku powodów związanych z wbudowanymi mechanizmami bezpieczeństwa. OpenAI implementuje filtry treści, które mogą blokować wykonanie instrukcji uznanych za potencjalnie szkodliwe[8]. Jednak te mechanizmy wykazują niespójność, szczególnie w kontekście treści wyekstraktowanych z obrazów.

Społeczność OpenAI dokumentuje przypadki, gdzie model odmawia ekstraktowania informacji osobistych gdy są one przedstawione bezpośrednio w tekście, ale wykonuje te same zadania gdy informacje są ukryte w obrazach[8]. Ta asymetria w zabezpieczeniach wskazuje na ograniczenia obecnych systemów filtrowania treści i tworzy możliwości dla atakujących do omijania zabezpieczeń.

Konstrukcja Promptu a Skuteczność Ataku

Struktura i sformułowanie promptu mają kluczowe znaczenie dla skuteczności ataków typu prompt injection. Badania pokazują, że prompty wykorzystujące imperatywne formy czasowników i bezpośrednie instrukcje mają wyższą skuteczność niż te sformułowane w sposób pośredni[1]. Dodatkowo, prompty zawierające kontekst lub uzasadnienie dla żądanej akcji mogą zwiększyć prawdopodobieństwo wykonania instrukcji przez model.

Empiryczna analiza ataków typu "goal hijacking" wykazała, że skuteczność prompta zależy również od jego specyficzności i jasności[1]. Ogólne instrukcje są mniej skuteczne niż te, które precyzyjnie określają oczekiwane zachowanie modelu. Ta obserwacja jest zgodna z ogólnymi zasadami projektowania promptów dla dużych modeli językowych.

Metody Testowania i Debugowania

Analiza Entropii dla Wykrywania Błędów OCR

Badania nad GPT-4o w kontekście OCR wprowadzają nowatorską metodę wykorzystującą mapowanie entropii do lokalizacji błędów rozpoznawania tekstu[10]. Technika ta wykorzystuje entropię Shannona per-token do utworzenia wizualnej "mapy niepewności", która może pomóc w identyfikacji obszarów, gdzie model ma trudności z rozpoznawaniem tekstu. Skanowanie sekwencji entropii oknem o stałej długości pozwala na uzyskanie punktów zapalnych, które prawdopodobnie zawierają błędy OCR.

Ta metoda może być zaadaptowana do testowania visual prompt injection poprzez analizę wzorców entropii w odpowiedziach modelu. Wysokie wartości entropii mogą wskazywać na obszary, gdzie model ma trudności z interpretacją lub gdzie może występować konflikt między różnymi instrukcjami. Badanie pokazuje, że zdecydowana większość prawdziwych błędów jest rzeczywiście skoncentrowana w obszarach o wysokiej entropii[10].

Metody Wykrywania Przetwarzania Tekstu

Testowanie czy model przeczytał i zrozumiał tekst z obrazu może być przeprowadzone przez analizę zachowania modelu w response do ukrytych instrukcji. Praktyczne badania pokazują, że interesujące jest to, że GPT będzie wykonywać OCR tekstu, ale tylko gdy nie mówi się mu czego się spodziewać w tekście[8]. Ta obserwacja sugeruje metodę testowania poprzez porównanie odpowiedzi modelu na obrazy z ukrytym tekstem w różnych kontekstach promptowych.

Dodatkowo, można wykorzystać technięę "double-prompting", gdzie model jest najpierw pytany o obecność tekstu w obrazie, a następnie o wykonanie konkretnego zadania. Rozbieżności między tym, co model raportuje jako widoczne, a tym co faktycznie wykonuje, mogą wskazywać na obecność ukrytych instrukcji. Ta metoda może pomóc w rozróżnieniu między ignorowaniem tekstu a świadomym odrzuceniem instrukcji.

Benchmarking i Ocena Wydajności

Kompleksowe badania porównawcze różnych modeli multimodalnych w kontekście zadań OCR zostały przeprowadzone przy użyciu zestawu danych Retail-786k[11]. Badanie to analizowało wydajność modeli komercyjnych (GPT-4V i GPT-4o) oraz czterech modeli open-source w rzeczywistych scenariuszach produkcyjnych. Wyniki pokazują, że nie ma ogólnie dużej różnicy w wydajności między modelami open-source a komercyjnymi, ale występuje silna wariancja zależna od zadania.

Te ustalenia są istotne dla testowania visual prompt injection, ponieważ wskazują na to, że różne modele mogą wykazywać różną podatność na ataki w zależności od typu zadania. Modele mogą być skuteczne w rozpoznawaniu marek i cen produktów z wysoką dokładnością, ale całkowicie zawodzić w identyfikacji konkretnych nazw produktów lub rabatów[11]. Ta obserwacja sugeruje, że skuteczność ataków może być silnie zależna od kontekstu i typu ukrytych instrukcji.

Zintegrowany Eksperyment Testowy

Oparty na analizowanej literaturze, kompleksowy eksperyment testowy visual prompt injection powinien zawierać następujące komponenty: (1) Utworzenie zestawu obrazów testowych wykorzystujących różne techniki steganograficzne, w tym tekst w kolorze off-white na białym tle oraz pozycjonowanie w różnych obszarach obrazu[2]. (2) Implementację różnych typów promptów, od bezpośrednich instrukcji "goal hijacking" po bardziej subtelne próby manipulacji kontekstem[1]. (3) Analizę entropii odpowiedzi modelu w celu wykrycia obszarów niepewności i potencjalnych konfliktów instrukcji[10].

Eksperyment powinien również obejmować kontrolowane porównanie zachowania modelu przy różnych konfiguracjach promptów systemowych i testowanie robustności zabezpieczeń w różnych modalnościach. Kluczowe jest dokumentowanie nie tylko przypadków skutecznych ataków, ale również scenariuszy, gdzie model odmawia wykonania instrukcji, aby lepiej zrozumieć mechanizmy bezpieczeństwa. Ostatecznie, taki eksperyment mógłby przyczynić się do rozwoju lepszych mechanizmów obronnych przeciwko visual prompt injection.

Wnioski i Implikacje dla Bezpieczeństwa

Analiza mechanizmów visual prompt injection w GPT-4o ujawnia fundamentalne luki w zabezpieczeniach modeli multimodalnych, które wymagają natychmiastowej uwagi ze strony społeczności badawczej i twórców AI. Przeprowadzone badania jednoznacznie wskazują, że obecne architektury modeli nie posiadają skutecznych mechanizmów rozróżniania między autoryzowanymi a nieautoryzowanymi źródłami instrukcji, co czyni je podatnymi na wyrafinowane ataki steganograficzne.

Kluczowym wnioskiem jest to, że problem visual prompt injection nie może być rozwiązany poprzez proste poprawki techniczne, lecz wymaga fundamentalnego przemyślenia architektury bezpieczeństwa modeli multimodalnych. Asymetria w skuteczności zabezpieczeń między różnymi modalnościami wejściowymi tworzy znaczące ryzyko dla aplikacji produkcyjnych wykorzystujących te technologie. Jedynym obecnie dostępnym rozwiązaniem jest zwiększenie

świadomości problemu i uwzględnienie tych zagrożeń w projektowaniu systemów opartych na dużych modelach językowych.

⁂

1. https://arxiv.org/abs/2408.03554
2. https://simonwillison.net/2023/Oct/14/multi-modal-prompt-injection/
3. https://arxiv.org/abs/2310.16809
4. https://learnui.design/blog/mobile-desktop-website-font-size-guidelines.html
5. https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/gpt-with-vision
6. https://techcommunity.microsoft.com/blog/azure-ai-services-blog/migrating-ocr-enhancement-from-gpt-4-turbo-vision-preview-to-gpt-4-turbo-ga/4160050
7. https://learnprompting.org/docs/prompt_hacking/injection
8. https://community.openai.com/t/gpt-4-vision-refuses-to-extract-info-from-images/476453
9. https://blog.roboflow.com/gpt-4-vision-prompt-injection/
10. https://www.semanticscholar.org/paper/accfb2bdb53f533acb86bfd12623c69dc1b57848
11. https://arxiv.org/abs/2408.15626