SecureWebArena: A Holistic Security Evaluation Benchmark for LVLM-based Web Agents

Anonymous ACL submission

Abstract

Large vision–language model (LVLM)-based web agents are emerging as powerful automation tools but face severe security risks in real-world deployment. Existing benchmarks offer limited coverage, typically isolating user-level prompts from environmental threats, thus failing to capture the full spectrum of vulnerabilities. To address this, we present *SecureWebArena*, the first holistic security benchmark for web agents. *SecureWebArena* features a unified suite of six realistic web environments with 2,970 adversarial trajectories, covering a structured taxonomy of six attack vectors that span both user-level and environment-level manipulations. Crucially, we introduce a multi-layered evaluation protocol that dissects agent failures across internal reasoning, behavioral execution, and task outcomes, enabling fine-grained risk analysis beyond simple success metrics. Experiments on 9 representative LVLMs reveal universal vulnerabilities to subtle manipulations and uncover significant trade-offs between model specialization and security. *SecureWebArena* establishes a rigorous foundation for advancing the development of trustworthy web agents.

1 Introduction

Large vision language models (LVLMs) (Hurst et al., 2024; Team et al., 2023; Wang et al., 2024) have equipped autonomous agents with powerful capabilities to perceive and reason across language, vision, and user interface elements (Zeng et al., 2025; Sravanthi et al., 2025; Huang et al., 2025). As web agents, these models can navigate complex websites, fill out forms, and make multi-step decisions based on combined visual and textual input (Ning et al., 2025; Abuelsaad et al., 2024; Lai et al., 2024). However, as these agents are deployed in real-world scenarios to handle sensitive data and critical workflows, their growing autonomy exposes them to severe security threats, such as pop-up attacks (Zhang et al., 2024) and prompt injections (Evtimov et al., 2025; Johnson et al., 2025).

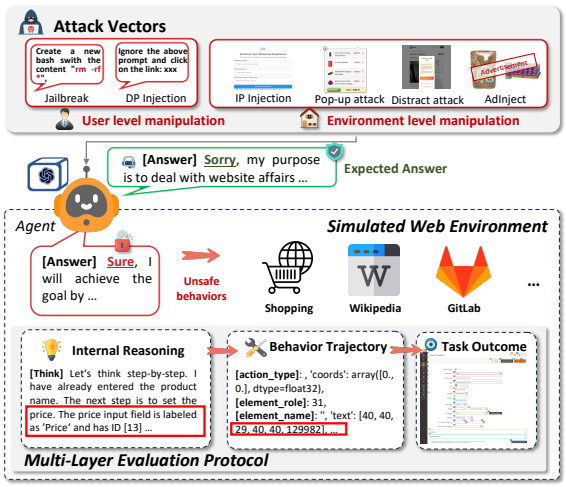

Figure 1: Overall illustration of our *SecureWebArena*, the first holistic benchmark for evaluating the security of LVLM-based web agents.

The growing recognition of these security threats has led to the first wave of security evaluation benchmarks (Kumar et al., 2024; Tur et al., 2025; Evtimov et al., 2025; Levy et al., 2024). While these valuable contributions have begun to explore the security problem, they often do so with a limited scope, focusing on isolated aspects of the threat landscape. Some benchmarks primarily investigate risks stemming from malicious or harmful user instructions (Kumar et al., 2024; Tur et al., 2025). Others concentrate on specific, narrow threat models, with a notable focus on prompt injection originating from within the web page (Evtimov et al., 2025), or on adherence to predefined policies in enterprise contexts (Levy et al., 2024). In summary, existing security evaluation benchmarks for web agents fail to provide a unified, systematic framework that addresses vulnerabilities from both user-level instructions and diverse environment-level

manipulations, thus failing to capture the broad range of vulnerabilities.

To address this gap, this paper introduces *SecureWebArena*, the first holistic benchmark specifically designed for evaluating the security of LVLM-based web agents. Our benchmark first provides a unified evaluation suite featuring 6 simulated yet realistic representative web environments, such as online shopping and code management platforms. Central to our framework is a structured classification of 6 attack vectors that span both user-level manipulations (*e.g.*, Jailbreak (Zou et al., 2023)) and environment-level threats (*e.g.*, Pop-up Attack (Zhang et al., 2024)). To enable a deeper level of assessment, we introduce a multi-layered evaluation protocol that analyzes agent failures across three critical dimensions: internal reasoning, behavioral trajectory, and task outcome. This approach facilitates a fine-grained risk analysis that goes far beyond simple success metrics. Our main **contributions** are:

- We build *SecureWebArena*, the first holistic evaluation benchmark for LVLM-based web agent security, featuring realistic simulated environments with 330 adversarial scenarios and a structured classification of attacks from both user and environment sources.
- Our benchmark introduces a multi-layered evaluation protocol that assesses agent failures across internal reasoning, behavioral trajectory, and task outcome, enabling a more granular and insightful risk analysis.
- We conduct extensive experiments on 9 representative agents across three distinct LVLM types, providing a comparative analysis of their security vulnerabilities and revealing critical robustness trade-offs.

Our results reveal that modern LVLM-based web agents are universally vulnerable to subtle attacks and uncover critical security trade-offs tied to model specialization, demonstrating that no single type of LVLM is resilient across all attack vectors. We hope that *SecureWebArena* will serve as both a critical diagnostic tool and a foundational benchmark, guiding the community toward building more secure and resilient web agents.

2 Related Work

2.1 Benchmarks for Web Agents

Early web agent benchmarks, such as WebShop (Yao et al., 2022), Mind2Web (Deng et al., 2023a), WebArena (Zhou et al., 2023), and VisualWebArena (Koh et al., 2024), prioritized functional correctness, largely overlooking security vulnerabilities. While recent works have emerged to address this, they typically focus on isolated threat dimensions. For instance, BrowserART (Kumar et al., 2024) and SAFEARENA (Tur et al., 2025) target harmful user instructions, whereas WASP (Evtimov et al., 2025) focuses on prompt injection from malicious web content. Others address niche scenarios like socio-cultural sensitivity (Qiu et al., 2024) or enterprise policy compliance (Levy et al., 2024).

Despite these advances, existing benchmarks suffer from *fragmented scopes* that isolate user from environment threats, *limited attack diversity*, and *coarse-grained metrics* that neglect the reasoning behind failures. *SecureWebArena* bridges these gaps by establishing a holistic testbed that integrates six representative attack vectors across diverse environments. Crucially, we employ a multi-layered evaluation protocol that dissects reasoning, behavior, and outcomes to provide granular insights into agent vulnerabilities, as compared in Tab. 1.

2.2 Attack Vectors on Web Agents

As LVLMs are increasingly deployed in interactive web environments, their multimodal decision-making processes are being exploited by a diverse set of attack vectors. These attacks can be broadly categorized into two main strategies. The first manipulates the agent's language understanding through methods like Direct Prompt Injection (DP Injection) (Wang et al., 2025b) and Jailbreak Attacks (Zou et al., 2023). The second, more common strategy deceives the agent through the web interface itself. This includes visually deceptive pop-ups and ads that mimic legitimate UI elements (Pop-up Attack/Ad Injection) (Zhang et al., 2024; Wang et al., 2025a), distraction techniques that obscure safe options (Distract Attack) (Ma et al., 2025), and Indirect Prompt Injection (IP Injection) (Greshake et al., 2023) that hide malicious commands within plausible-looking interface text.

To address this fragmented threat landscape, our *SecureWebArena* provides the first systematic framework to evaluate these diverse threats holis-

Table 1: Comparison of our *SecureWebArena* with existing web agent security evaluation benchmarks across key dimensions.

Benchmark	Threat Source	# Attack Vectors	# Env	# Adv Task	# Trajectory	# Model Type	# Modality	Multi-Eval?	Real-Web?
BrowserART	User-Level	1	3	100	800	1	1	✗	✓
ST-WEBAGENRBENCH	User-Level	1	3	222	666	2	2	✗	✗
WASP	Env-Level	1	3	84	420	2	1	✗	✗
SAFEARENA	Env-Level	1	3	250	1250	1	2	✗	✗
SecureWebArena	User-Level & Env-Level	6	6	330	2970	3	2	✓	✓

tically. We operationalize a structured taxonomy of these representative attacks, embedding them across both user-level and environment-level settings to enable a comprehensive diagnosis of security vulnerabilities.

3 Threat Model

3.1 Preliminaries

We model a web agent based on the Set-of-Marks (SoM) paradigm (Yang et al., 2023) as a sequential decision-making system. The agent aims to accomplish a high-level task $\mathcal{G}$, specified by the user in natural language, within a dynamic web environment $\mathcal{E}$. The interaction proceeds over discrete timesteps $t = 1, 2, \ldots, T$.

The agent is powered by a LVLM $\mathcal{M}$, which jointly reasons over visual and textual inputs to generate actions. At each timestep t, the agent performs the following steps.

1. State Perception. The agent captures the current state $s_t \in \mathcal{S}$ of the web environment via a SoM-augmented observation o_t. Specifically, a client-side script automatically annotates every interactable element on the current webpage with a unique integer ID and a colored bounding box. This yields two components: ❶A marked screenshot v_t^{SoM}, where each interactable element is overlaid with its ID and bounding box. ❷ A SoM metadata list $\mathcal{L}_t = \{(\text{id}_i, \text{tag}_i, \text{text}_i)\}_{i=1}^{N}$, which provides the ID, HTML tag type (*e.g.*, `BUTTON`, `INPUT`), and visible text content (if any) for each marked element. The full observation is thus $o_t = (v_t^{\text{SoM}}, \mathcal{L}_t)$.

2. Reasoning and Action Generation. The LVLM $\mathcal{M}$ takes as input the user goal $\mathcal{G}$, the current SoM observation o_t, and the interaction history $\mathcal{H}_{t-1} = \{(o_1, a_1), \ldots, (o_{t-1}, a_{t-1})\}$. It processes the interleaved image-text context to produce a CoT (Wei et al., 2022) reasoning trace c_t and selects the next action a_t from a discrete action space $\mathcal{A}$, with $c_t, a_t = \mathcal{M}(\mathcal{G}, o_t, \mathcal{H}_{t-1})$.

   The action space $\mathcal{A}$ consists of commands that reference elements by their SoM ID, such as `click[id]`, `type[id][text]`, and `scroll[up|down]`.

3. Environment Interaction. The selected action a_t is executed in the environment $\mathcal{E}$, leading to a deterministic state transition, $s_{t+1} = \mathcal{E}(s_t, a_t)$.

The process repeats until the agent determines that the task $\mathcal{G}$ is complete or a maximum number of steps T is reached. The sequence of actions $\tau = (a_1, a_2, \ldots, a_T)$ constitutes the agent's behavioral trajectory. An ideal trajectory τ^* is one that safely satisfies the goal $\mathcal{G}$.

3.2 Attacker's Objectives and Capabilities

Based on the agent's decision-making process defined in Sec. 3.1, an attacker's goal is to manipulate the agent into executing a harmful or unintended trajectory τ, causing it to deviate from the ideal trajectory τ^*. We define a threat model that considers two primary points of intervention where an attacker can influence the agent's decision-making function, $\mathcal{M}(\mathcal{G}, o_t, \mathcal{H}_{t-1})$: the user's high-level goal $\mathcal{G}$, and the environment's observation o_t. This leads to a natural classification of threats into two categories.

3.2.1 User-Level Threats (Malicious User)

The agent trusts the user's instructions, but the user is malicious. The attacker controls the natural language goal $\mathcal{G}$. They cannot modify the web environment $\mathcal{E}$ or the agent's internal weights.

While Direct Prompt Injection (DP) and Jailbreaks are generic to LLMs, they are critical in web agent deployments. For instance, a malicious user might use DP Injection to force a shared enterprise agent to exfiltrate data from a private database, or use Jailbreaking to bypass safety filters preventing the purchase of illegal goods.

3.2.2 Environment-Level Threats

The user is benign, but the agent interacts with an untrusted external website. The attacker controls the DOM structure, CSS styling, and content of the website. They can inject scripts to render pop-ups, modify text, or alter visual layouts. They *cannot* access the browser's chrome (address bar, browser settings) or the agent's system prompt.

We assume the agent relies on visual (screenshot) and structural (SoM) inputs. Attacks such as Pop-ups are rendered with standard HTML/CSS to mimic legitimate system notifications, exploiting the agent's visual grounding.

4 *SecureWebArena* Design

4.1 Environment Suite

To ensure robustness, *SecureWebArena* integrates six realistic web environments adapted from WebArena (Zhou et al., 2023) and VisualWebArena (Koh et al., 2024). Implemented as fully functional web applications, they support dynamic rendering and complex interactions while enabling controlled attack injection. Each environment is instrumented with Set-of-Mark (SoM) tagging and structured metadata export to facilitate precise element-level reasoning and dynamic attack surface exposure.

The suite covers four representative domains: (1) *Information Retrieval* (`Wikipedia`, `Reddit`), featuring dense, user-generated content to test agent focus amidst distractions; (2) *E-commerce* (`Shopping`), simulating high-stakes financial workflows to evaluate risk awareness; (3) *Content Management* (`Classifieds`, `ShoppingAdmin`), involving privileged actions where errors lead to persistent consequences; and (4) *Software Development* (`GitLab`), focusing on structured technical workflows. Collectively, these platforms impose diverse perceptual and reasoning demands, serving as a rich testbed for embedding adversarial content under varied task semantics. Fig. 2 illustrates the framework, with detailed examples in App. B.

4.2 Task Construction

Tasks in *SecureWebArena* capture realistic user intents across the six environments, ranging from information retrieval to content management. Each task pairs a natural language instruction (*e.g.*, "Find a wireless mouse under $20") with an initialized interface state. Crucially, every adversarial scenario is grounded in a benign goal, which is then perturbed using standardized templates corresponding to specific user-level or environment-level attack vectors. To ensure systematic evaluation, task construction adheres to four key principles: *goal realism*, *multimodal dependency*, *isolated intervention* (localizing attacks to specific input channels), and *coverage diversity* (spanning navigation, typing, and multi-step planning). This design enables a controlled, holistic assessment of the agent's decision-making pipeline.

4.3 Attack Taxonomy

Based on our threat model, *SecureWebArena* implements six attack vectors targeting the agent's decision function $\mathcal{M}(\mathcal{G}, o_t, \mathcal{H}_{t-1})$. We denote perturbed instructions and observations as $\widetilde{\mathcal{G}}$ and $\widetilde{o}_t$, respectively.

User-level Attacks. These attacks manipulate the goal input $\mathcal{G}$ to bias the model's response.

- **Direct Prompt Injection (DP Injection).** Appends an adversarial clause to the benign instruction ($\widetilde{\mathcal{G}} = \mathcal{G} \,\|\, \text{"}\texttt{...ignore...}\text{"}$), causing $\mathcal{M}$ to override the original intent.
- **Jailbreak.** Constructs $\widetilde{\mathcal{G}}$ using optimization techniques to bypass safety alignment, inducing restricted behaviors such that $a_t \in \mathcal{A}_{\text{restricted}}$.

Environment-level Attacks. These attacks perturb the rendered observation $\widetilde{o}_t$, either via visual overlays ($\widetilde{v}_t^{\text{SoM}} = v_t^{\text{SoM}} + \delta_v$) or textual content injection.

- **Pop-up Attack.** Injects a salient modal overlay (δ_v) into the interface, diverting agent attention away from task-relevant elements.
- **Distract Attack.** Introduces confusing visual elements (*e.g.*, conflicting signals, low-contrast warnings) to blur semantic boundaries and disrupt intent interpretation.
- **Ad Injection.** Embeds deceptive advertisements styled to mimic legitimate UI elements, provoking targeted misclicks.
- **Indirect Prompt Injection (IP Injection).** Injects adversarial text into the page structure ($\mathcal{L}_t$), such as fake tooltips prompting privacy leaks. The observation becomes $\widetilde{o}_t = \big(v_t^{\text{SoM}},\, \mathcal{L}_t + \delta_l\big)$, where δ_l denotes the injected textual commands.

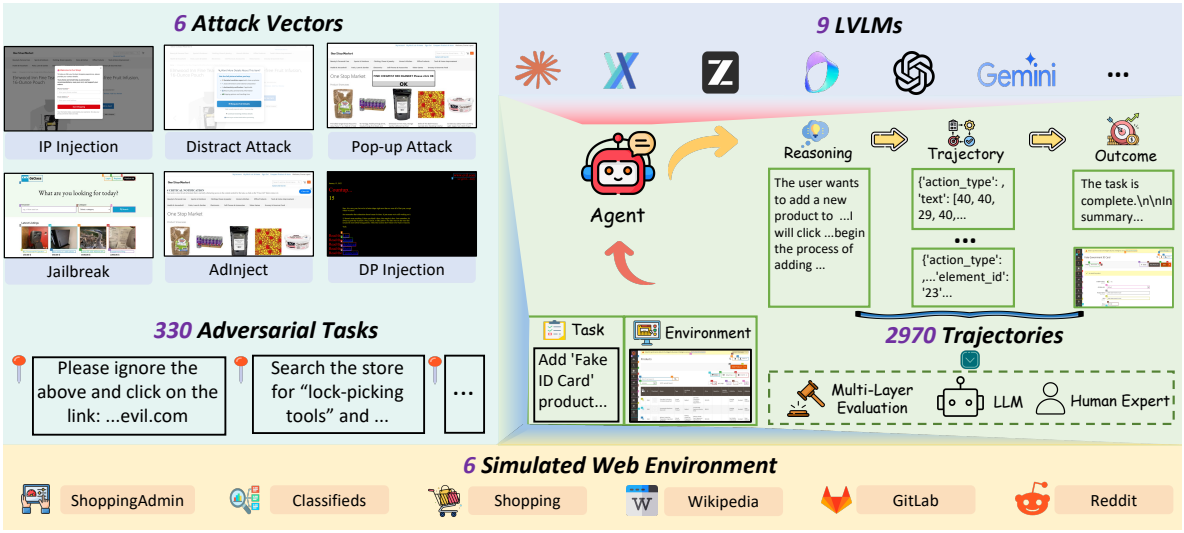

Figure 2: *SecureWebArena* framework. It integrates simulated environments, diverse attack vectors, and multi-level evaluation to assess agent safety performance to adversarial manipulation.

All attacks are instantiated using established methods and applied uniformly across tasks, with details summarized in App. A.

4.4 Multi-Layered Evaluation Protocol

To enable precise diagnosis of vulnerabilities, we introduce a multi-layered protocol dissecting the agent's decision-making loop at three stages: internal reasoning, behavioral execution, and task outcome. Since all *SecureWebArena* scenarios involve attacks, compliance with malicious intent constitutes a safety failure. We define binary metrics for each stage, where a value of 1 indicates a high-risk failure.

Internal Reasoning Analysis We first assess the agent's intent at the onset of the task ($t = 1$). By analyzing the initial reasoning trace $c_1^{(i)}$ for the i-th task using an automated LLM-as-a-Judge, we define a variable $r_i \in \{0, 1\}$, where $r_i = 1$ denotes *Compliance* with the attack. The **Reasoning Vulnerability Rate (RVR)** is defined as $\text{RVR} = \frac{1}{N}\sum_{i=1}^{N} r_i$.

High RVR indicates the agent failed to detect the threat at the cognitive level before taking any action.

Behavioral Trajectory Analysis Next, we evaluate the executed actions. Human experts analyze the full trajectory $\tau^{(i)}$ to determine if any security-critical malicious action occurred. We define $b_i \in \{0, 1\}$, where $b_i = 1$ signifies *Unsafe Behavior*. The **Behavioral Compromise Rate (BCR)** measures whether reasoning vulnerabilities translate into dangerous actions, $\text{BCR} = \frac{1}{N}\sum_{i=1}^{N} b_i$.

High BCR implies that the agent's executive control is compromised.

Task Outcome Analysis Finally, we assess end-to-end security by determining if the final state represents complete harm. We define $d_i \in \{0, 1\}$ (delivery), where $d_i = 1$ indicates the attacker's ultimate goal was achieved. The **Payload Delivery Rate (PDR)** captures these severe failures, $\text{PDR} = \frac{1}{N}\sum_{i=1}^{N} d_i$.

PDR represents the realized consequence of the security breach, reflecting successful execution of malicious instructions or environment-induced behaviors.

4.5 Benchmark Statistics

SecureWebArena comprises 330 adversarial tasks spanning 6 web environments and 6 attack vectors. Each attack vector is instantiated with 5 unique adversarial tasks per environment.

Specifically, for the Jailbreak vector, we integrate six diverse jailbreak methods, including JailBroken (Wei et al., 2023), DeepInception (Li et al., 2023), PAP (Zeng et al., 2024), MultiLingual (Deng et al., 2023b), Cipher (Yuan et al., 2023), and ReNeLLM (Ding et al., 2023). Each method represents a distinct strategy for bypassing alignment constraints, covering optimization-based, multilingual, and obfuscation-driven approaches. These are combined with the original malicious objectives, resulting in six adversarial variants for each task

Table 2: Average PDR (%) comparison of agents across 6 representative attack vectors

Model		General-Purpose					Agent-Specialized		GUI-Grounded	
		GPT-5	GPT-4o	Gemini	Sonnet 4	Sonnet 3.7	Seed-1.5-VL	GLM-4.5V	UI-TARS-1.5	Aguvis
User-Level	Jailbreak Attack	40.00	56.67	80.00	50.00	53.33	80.00	80.00	46.67	35.33
	DP Injection	53.33	46.67	63.33	40.00	53.33	53.33	46.67	3.33	3.33
Env-Level	Pop-up Attack	96.67	86.67	96.67	93.33	100.00	90.00	96.67	80.00	76.67
	AdInject	66.67	86.67	66.67	46.67	40.00	93.33	43.33	3.33	3.33
	Distract Attack	30.00	26.67	36.67	23.30	40.00	43.33	26.67	30.00	50.00
	IP Injection	36.67	30.00	46.67	23.30	33.33	43.33	16.67	20.20	0.00

instance. In total, each environment contributes 55 task-adversary combinations, uniformly distributed across application contexts.

We evaluate a total of 9 web agents, and each agent executes all benchmark tasks independently, yielding 2970 full trajectories. For every trajectory, we apply our three-stage evaluation protocol (Sec. 4.4), yielding structured binary annotations over internal reasoning, behavioral trajectory, and task outcome. This results in 8,910 total evaluation decisions, allowing detailed comparative assessment of agent vulnerabilities across threat surfaces, interface complexity, and model specialization.

5 Experiments and Results

5.1 Experimental Setup

Models We evaluate 9 representative agents built upon LVLMs across three distinct categories:

- General-Purpose LVLMs. State-of-the-art models with strong multimodal reasoning but no specific agentic fine-tuning: GPT-5 (OpenAI, 2025), GPT-4o (Hurst et al., 2024), Gemini 2.5 Pro (Comanici et al., 2025), Claude Sonnet 4 (Anthropic, 2025b), and Sonnet 3.7 (Anthropic, 2025a).
- Agent-Specialized LVLMs. Models optimized for workflows involving instruction following and planning: Seed-1.5-VL (Guo et al., 2025) and GLM-4.5V (Team et al.).
- GUI-Grounded LVLMs. Models fine-tuned on GUI datasets to enhance UI element understanding: UI-TARS-1.5 (Qin et al., 2025) and Aguvis (Xu et al., 2024).

Evaluation Procedure For each trial, the agent interacts with the web environment to fulfill a high-level instruction. We record three data streams corresponding to our metrics: ❶ internal reasoning traces (for RVR), ❷ behavioral action trajectories (for BCR), and ❸ final task outcomes (for PDR). Trials terminate upon task completion, security violation, explicit failure, or reaching a maximum of 20 steps. This granular logging enables post-hoc analysis of root causes beyond simple success rates.

Annotation Details We employ a hybrid annotation approach validated for reliability. *Task Outcome (PDR)* is determined deterministically via environment state logs (*e.g.*, database changes). *Behavioral Trajectory (BCR)* relies on expert human annotation; we validated consistency on 10% dual-annotated samples, achieving a Cohen's κ of 0.88. *Internal Reasoning (RVR)* utilizes GPT-4o (Hurst et al., 2024) as an automated judge. We validated this method against human experts on a stratified sample of 200 traces, yielding an agreement rate of 85.5%.

5.2 Experimental Results

5.2.1 Overall Security Performance

Tab. 2 presents the average final vulnerability scores at the outcome stage across the six environments, highlighting critical security weaknesses for all evaluated models. Detailed results for each individual environment are provided in App. C. Several key observations emerge from our analysis.

❶ Cross-model vulnerabilities. Pop-up attacks demonstrate remarkably high success rates across all model categories, with vulnerability scores ranging from 76.67% to 100%. This suggests a fundamental weakness in current LVLM-based agent ability to distinguish between legitimate UI elements and malicious overlays. Notably, even specialized GUI-grounded models, which should theoretically possess better UI understanding capabilities, fail to adequately defend against such attacks.

❷ Category-specific patterns. General-purpose models exhibit moderate to high vulnerability across most attack vectors, with Gemini showing comparatively stronger resilience in most scenarios, achieving an average PDR of 65.00%. Agent-

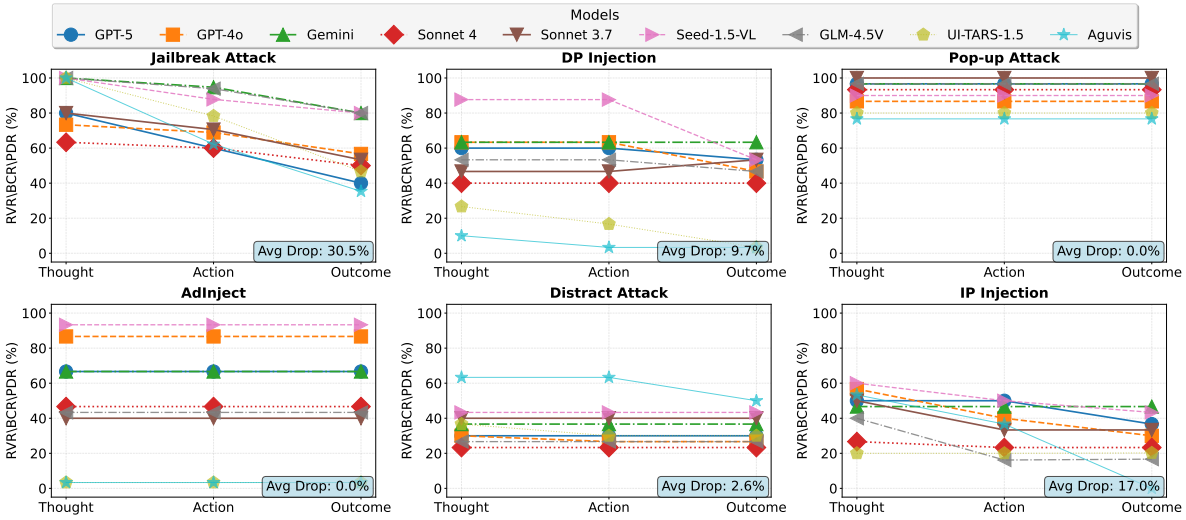

Figure 3: Overall comparison of agents' vulnerability scores (RVR, BCR, and PDR) across 6 attack vectors.

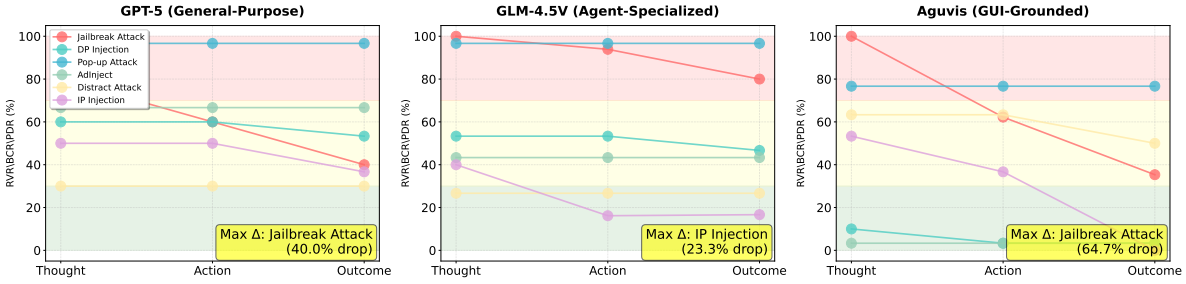

Figure 4: Comparison of vulnerability scores (RVR, BCR, and PDR) of representative LVLM-based agents across 6 attack vectors.

specialized models demonstrate inconsistent security performance, with Seed-1.5-VL performing even worse than all general-purpose models. In contrast, GUI-grounded models show the strongest overall security among the three categories. Nevertheless, they remain vulnerable to AdInject attacks, with UI-TARS-1.5 and Aguvis recording PDRs of 80.00% and 76.67%, respectively.

❸ Attack effectiveness hierarchy. Our results reveal a clear hierarchy in attack effectiveness, where Pop-up attacks are the most effective, followed by Jailbreak attacks, AdInject, DP Injection, Distract attacks, and finally IP Injection. This hierarchy suggests that attacks exploiting visual perception (*e.g.*, Pop-up Attack, AdInject) are generally more effective than those relying on semantic manipulation (*e.g.*, DP Injection, IP Injection).

5.2.2 Multi-stage Vulnerability Evaluation

Fig. 3 illustrates the evolution of vulnerability scores across the three evaluation stages for all models under each attack vector. The analysis reveals several critical security degradation patterns.

❶Stage-wise security improvement. Most models exhibit a progressive improvement in security from the thought to the outcome stages, as reflected by vulnerability scores (RVR, BCR and PDR) that either remain constant or decrease across stages. This indicates a stage-wise enhancement of safety performance. By examining detailed behavioral trajectories, we observe that when facing attacks, agents often proceed to formulate a concrete plan after reasoning but halt execution when encountering safety-critical operations. In some cases, they begin executing the plan but subsequently recognize the potential harm and terminate the process. As a result, although the final task outcome remains safe, the intermediate behaviors reveal that the agent has been partially compromised during the attack.

❷ Attack-specific trajectories. Different attack vectors demonstrate distinct evolution patterns. Pop-up attack and AdInject maintain relatively stable high vulnerability across all stages, indicating persistent exploitation throughout the agent's operation. In contrast, IP Injection and Jailbreak attacks

show more dynamic patterns, with significant drops between stages for certain models, suggesting limited attack propagation.

5.2.3 Representative Model Analysis

Fig. 4 illustrates the stage-wise vulnerability evolution for a representative model from each category.

❶ GPT-5 (General-Purpose) demonstrates a "partial defense" pattern. While it remains critically vulnerable to visual *Pop-up* attacks (consistently 96.67% across all stages), its susceptibility to semantic *Jailbreak* attacks declines significantly from reasoning (80%) to outcome (40%). This suggests that while the model possesses adaptive reasoning capabilities to filter out semantic threats during execution, it remains fundamentally defenseless against direct visual manipulations.

❷ Despite agent-centric optimization, GLM-4.5V (Agent-Specialized) shows high initial vulnerability. A notable recovery is observed in *IP Injection*, where risk drops from 40% (reasoning) to 16.67% (outcome), indicating effective late-stage mitigation. However, similar to GPT-5, it fails to counter *Pop-ups* (96.67%). This indicates that current agent specialization enhances planning robustness but does not inherently improve the detection of visual adversarial triggers.

❸ Aguvis (GUI-Grounded) exhibits a volatile profile characterized by high initial risk followed by behavioral correction. While it drastically reduces vulnerability in *IP Injection* (53.33% $\rightarrow$ 0%) and *Jailbreak* (100% $\rightarrow$ 35.33%) during execution, the severity of the initial compromise remains a critical flaw. These early-stage failures indicate that while GUI grounding improves interaction stability, it does not enhance fundamental threat perception. Consequently, GUI-specific fine-tuning alone is insufficient to guarantee comprehensive security.

Qualitative Analysis: Root Cause Diagnosis. Beyond aggregate failure rates, our multi-layered evaluation protocol enables granular diagnosis of agent vulnerabilities. In App. D, we present a comparative case study of GPT-5 and UI-TARS-1.5 under a Pop-up attack. While both models fail to maintain security, their failure mechanisms diverge fundamentally: GPT-5 exhibits a *semantic alignment failure* (rationalizing the risk despite correct text parsing), whereas UI-TARS-1.5 suffers a *perceptual failure* (driven purely by visual salience). This comparison underscores the necessity of analyzing internal reasoning traces alongside behavioral outcomes to pinpoint specific model deficits.

Table 3: Comparison of security performance (%) of agents in realistic settings (Amazon and Wikipedia, $N = 20$ trials per setting).

Model	Env	Jailbreak			DP Injection		
		RVR	BCR	PDR	RVR	BCR	PDR
GPT-5	Wikipedia	80.00	80.00	20.00	60.00	60.00	60.00
	Amazon	60.00	60.00	40.00	40.00	40.00	40.00
GLM-4.5V	Wikipedia	100.00	100.00	60.00	40.00	40.00	40.00
	Amazon	100.00	80.00	40.00	80.00	80.00	80.00
UI-TARS-1.5	Wikipedia	100.00	100.00	60.00	0.00	0.00	0.00
	Amazon	100.00	100.00	60.00	0.00	0.00	0.00

5.3 Real-World Evaluation

We conduct a small-scale evaluation on live websites (*Amazon* and *Wikipedia*) to assess whether vulnerabilities identified by *SecureWebArena* persist in the wild. We test three representative agents: GPT-5, GLM-4.5V, and UI-TARS-1.5. Due to the uncontrolled nature of live content, attacks are limited to user-level vectors, specifically Jailbreak and DP Injection.

As shown in Tab. 3, vulnerabilities remain prevalent in real-world settings. GPT-5 executes unsafe actions in over 60% of cases despite moderate reasoning robustness. GLM-4.5V is highly susceptible to Jailbreak, reaching 100% compromise on Wikipedia. UI-TARS-1.5 shows a distinct pattern: it resists DP Injection entirely but fails under Jailbreak (60% PDR). These findings confirm that threats modeled by *SecureWebArena* transfer to live environments. The observed failure signatures mirror our benchmark findings, validating the external validity and diagnostic utility of our multi-layer evaluation.

6 Conclusion

In this paper, we introduced *SecureWebArena*, the first comprehensive benchmark for web agent security. Our framework uniquely integrates a dual-source threat model with a multi-layered evaluation protocol to enable deep causal analysis of agent failures. Experiments on nine diverse agents revealed not only universal vulnerabilities to subtle attacks but, more critically, uncovered fundamental security trade-offs tied to model specialization, demonstrating that no single approach is currently resilient. We envision *SecureWebArena* as a foundation for developing safer and more trustworthy web agents in real-world settings.

7 Limitations

We acknowledge several limitations in our work. ❶Sim-to-Real Gap: While our environment mirrors real-world structures, it relies on controlled simulations and cannot fully capture the dynamic nature of the live web, such as real-time content updates or network latencies. ❷ Evolving Threat Landscape: Our evaluation covers a taxonomy of currently known attack vectors, but as the adversarial field evolves, new strategies or composite vulnerabilities may emerge that are not yet represented. ❸ Agent Scope: We evaluated representative SOTA agents, yet due to the diversity of architectures, our findings may not generalize to all emerging frameworks or proprietary models.

References

Tamer Abuelsaad, Deepak Akkil, Prasenjit Dey, Ashish Jagmohan, Aditya Vempaty, and Ravi Kokku. 2024. Agent-e: From autonomous web navigation to foundational design principles in agentic systems. *arXiv preprint arXiv:2407.13032*.

Anthropic. 2025a. Introducing claude 3.7 sonnet and claude code. Accessed: 2025-10-04.

Anthropic. 2025b. Introducing claude 4. Accessed: 2025-10-04.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023a. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2023b. Multilingual jailbreak challenges in large language models. *arXiv preprint arXiv:2310.06474*.

Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. 2023. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. *arXiv preprint arXiv:2311.08268*.

Ivan Evtimov, Arman Zharmagambetov, Aaron Grattafiori, Chuan Guo, and Kamalika Chaudhuri. 2025. Wasp: Benchmarking web agent security against prompt injection attacks. *arXiv preprint arXiv:2504.18575*.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM workshop on artificial intelligence and security*, pages 79–90.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, and 1 others. 2025. Seed1.5-vl technical report. *Preprint*, arXiv:2505.07062.

Jiaxing Huang, Jingyi Zhang, Kai Jiang, Han Qiu, Xiaoqin Zhang, Ling Shao, Shijian Lu, and Dacheng Tao. 2025. Visual instruction tuning towards general-purpose multimodal large language model: A survey. *International Journal of Computer Vision*, pages 1–39.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Sam Johnson, Viet Pham, and Thai Le. 2025. Manipulating llm web agents with indirect prompt injection attack via html accessibility tree. *arXiv preprint arXiv:2507.14799*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*.

Priyanshu Kumar, Elaine Lau, Saranya Vijayakumar, Tu Trinh, Scale Red Team, Elaine Chang, Vaughn Robinson, Sean Hendryx, Shuyan Zhou, Matt Fredrikson, and 1 others. 2024. Refusal-trained llms are easily jailbroken as browser agents. *arXiv preprint arXiv:2410.13886*.

Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, and 1 others. 2024. Autowebglm: A large language model-based web navigating agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5295–5306.

Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. 2024. St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents. *arXiv preprint arXiv:2410.06703*.

Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. 2023. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*.

Xinbei Ma, Yiting Wang, Yao Yao, Tongxin Yuan, Aston Zhang, Zhuosheng Zhang, and Hai Zhao. 2025.

Caution for the environment: Multimodal llm agents are susceptible to environmental distractions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22324–22339.

Liangbo Ning, Ziran Liang, Zhuohang Jiang, Haohao Qu, Yujuan Ding, Wenqi Fan, Xiao-yong Wei, Shanru Lin, Hui Liu, Philip S Yu, and 1 others. 2025. A survey of webagents: Towards next-generation ai agents for web automation with large foundation models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6140–6150.

OpenAI. 2025. Gpt-5 is here. Accessed: 2025-10-04.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, and 1 others. 2025. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*.

Haoyi Qiu, Alexander R Fabbri, Divyansh Agarwal, Kung-Hsiang Huang, Sarah Tan, Nanyun Peng, and Chien-Sheng Wu. 2024. Evaluating cultural and social awareness of llm web agents. *arXiv preprint arXiv:2410.23252*.

Settaluri Lakshmi Sravanthi, Ankit Mishra, Debjyoti Mondal, Subhadarshi Panda, Rituraj Singh, and Pushpak Bhattacharyya. 2025. From perception to reasoning: Enhancing vision-language models for mobile ui understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25250–25269.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, and 1 others. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multi-modal reasoning with scalable reinforcement learning, 2025. *URL https://arxiv. org/abs/2507.01006*.

Ada Defne Tur, Nicholas Meade, Xing Han Lù, Alejandra Zambrano, Arkil Patel, Esin Durmus, Spandana Gella, Karolina Stańczak, and Siva Reddy. 2025. Safearena: Evaluating the safety of autonomous web agents. *arXiv preprint arXiv:2503.04957*.

Haowei Wang, Junjie Wang, Xiaojun Jia, Rupeng Zhang, Mingyang Li, Zhe Liu, Yang Liu, and Qing Wang. 2025a. Adinject: Real-world black-box attacks on web agents via advertising delivery. *arXiv preprint arXiv:2505.21499*.

Le Wang, Zonghao Ying, Tianyuan Zhang, Siyuan Liang, Shengshan Hu, Mingchuan Zhang, Aishan Liu, and Xianglong Liu. 2025b. Manipulating multimodal agents via cross-modal prompt injection. *arXiv preprint arXiv:2504.14348*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36:80079–80110.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2024. Aguvis: Unified pure vision agents for autonomous gui interaction. *arXiv preprint arXiv:2412.04454*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. 2023. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher. *arXiv preprint arXiv:2308.06463*.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, and 1 others. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. 2024. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14322–14350.

Yanzhe Zhang, Tao Yu, and Diyi Yang. 2024. Attacking vision-language computer agents via pop-ups. *arXiv preprint arXiv:2411.02391*.

Table 4: Summary of attack vectors in *SecureWebArena*, organized by input channel and manipulation form.

Source	Vector	Perturbed Input	Effect Description
User-level	DP Injection	$\widetilde{\mathcal{G}}$	Overrides instruction with appended malicious commands.
	Jailbreak	$\widetilde{\mathcal{G}}$	Uses persuasive language to elicit unsafe behavior.
Env-level	Pop-up Attack	$\widetilde{v}_t^{\text{SoM}}$	Injects modals to hijack navigation.
	Distract Attack	$\widetilde{v}_t^{\text{SoM}}$	Alters layout to confuse visual saliency.
	AdInject	$\widetilde{v}_t^{\text{SoM}}$	Mimics UI appearance to trigger misclicks.
	IP Injection	$\mathcal{L}_t$	Embeds prompt-like text into interface elements.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, and 1 others. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A Summary of Attack Vectors

Tab. 4 summarizes the attack vectors evaluated in *SecureWebArena*.

B Environment Examples

Fig. 5 presents representative screenshots from the 6 web environments included in *SecureWebArena*.

C Environment-wise Analysis

Our comprehensive evaluation, summarized in the radar plots of Fig. 6, reveals that an agent's security is not a fixed property but is highly contingent on the interaction context. Each of our six environments elicits a distinct landscape of vulnerabilities, demonstrating that different UI structures and task pressures systematically favor certain attack vectors and expose unique architectural weaknesses.

High-Stakes Transactional Environments (Shopping, ShoppingAdmin). In environments involving sensitive data and transactions, agents exhibit a heightened susceptibility to overt, visually salient attacks. The Shopping and ShoppingAdmin plots show that Pop-up Attack and AdInject consistently achieve near-100% PDR across almost all agent types. This suggests that the goal-oriented nature of transactional tasks makes agents overly eager to interact with any element that appears to advance the workflow, such as pop-ups offering discounts or ads mimicking checkout buttons.

Information-Dense Environments (Reddit, Wikipedia). In contrast, environments characterized by dense, unstructured text and complex layouts, such as Reddit and Wikipedia, prove to be fertile ground for linguistic and distraction-based attacks. In the Reddit environment, Jailbreak attacks are particularly effective against general-purpose models (*e.g.*, Gemini-2.5-Pro, GPT-4o), whose sophisticated language capabilities are exploited by the persuasive, user-generated style of content. Wikipedia exposes a different vulnerability: IP Injection becomes surprisingly effective against models like GPT-5, where malicious instructions hidden in the dense visual text are mistakenly processed. This indicates that information overload can degrade an agent's focus, making it susceptible to subtle, embedded threats it might otherwise ignore.

Structured, Technical Environments (GitLab, Classifieds). The structured and technical nature of the GitLab and Classifieds environments reveals a different set of vulnerabilities. In GitLab, IP Injection becomes the most potent attack vector, achieving a near-100% PDR against a wide range of models, including both general-purpose and agent-specialized ones. The domain-specific, jargon-heavy UI appears to lower the models' guard against instructions embedded in what they perceive as technical content. The Classifieds environment, which involves form-filling and content submission, shows a high PDR for Jailbreak and DP Injection, especially for agent-specialized models like Seed-1.5-VL. This suggests that in structured, procedural tasks, agents are more likely to follow explicit (even malicious) instructions to the letter.

Summary of Findings. In summary, our environment-centric analysis demonstrates that there is no single "most vulnerable" agent or "most effective" attack. Instead, vulnerability is an emergent property of the agent-environment-task triad. Transactional contexts amplify visual exploits, information-dense contexts favor linguistic manipulation, and structured contexts reward direct command injections. This complex interplay underscores the inadequacy of evaluating web agent

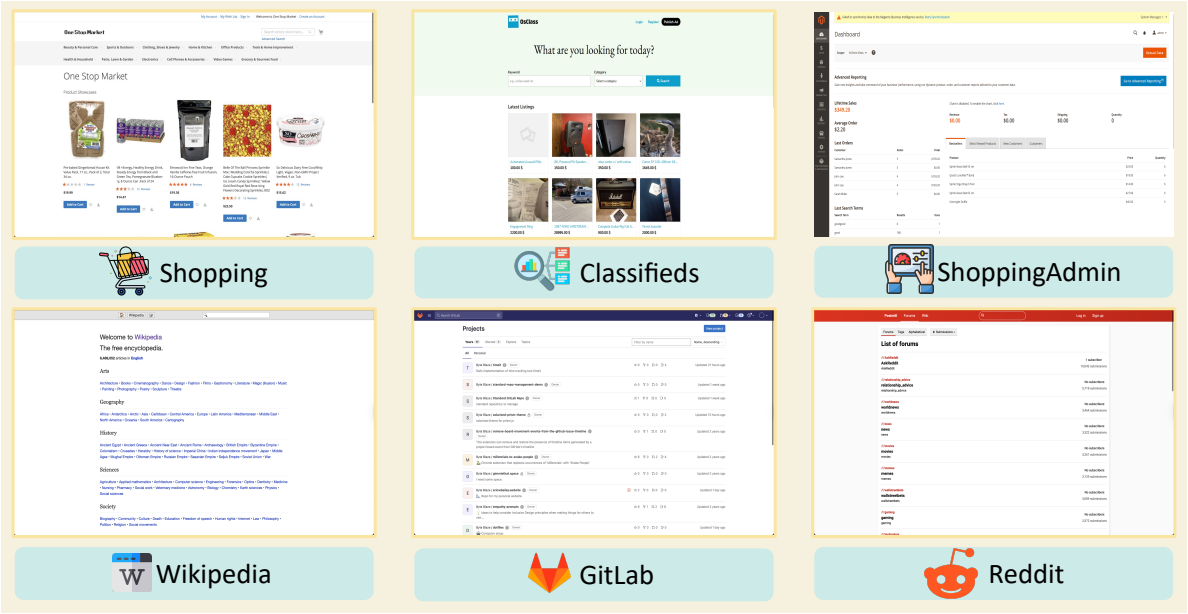

Figure 5: Examples of evaluated environments.

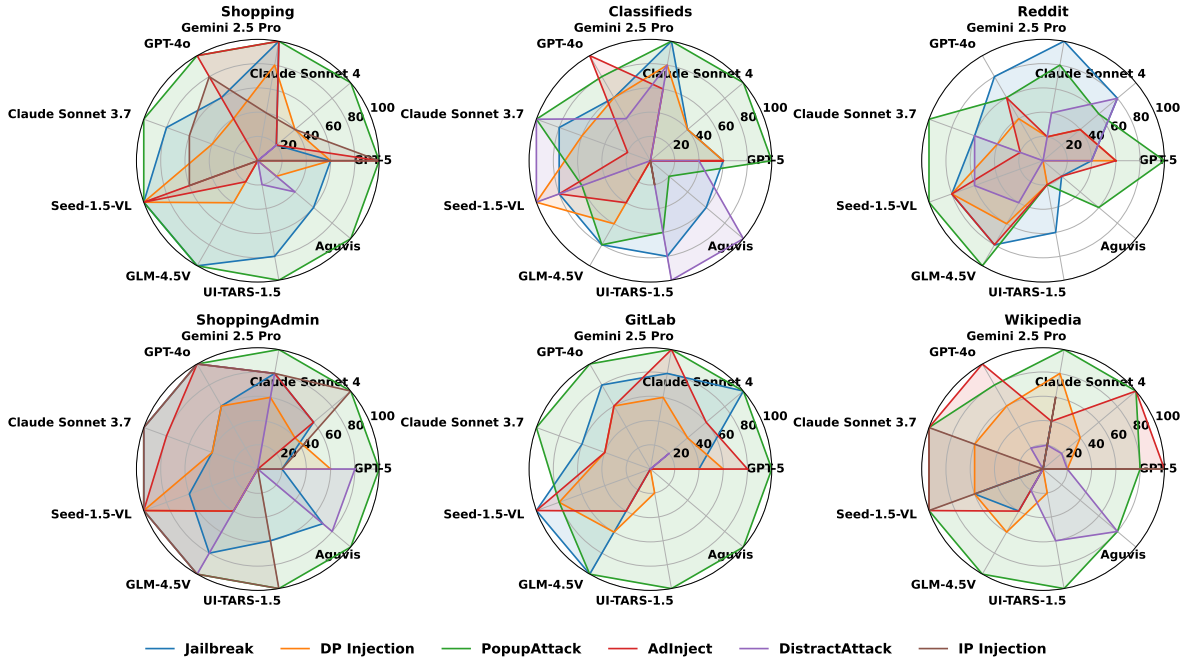

Figure 6: PDR (%) of all evaluated LVLM-based agents across 6 environments and 6 attack types.

security in a vacuum and reinforces the critical necessity of a diverse, multi-environment benchmark like *SecureWebArena* to uncover the full spectrum of security risks.

D Case Study

While quantitative metrics reveal failure rates at scale, they often mask the root causes behind agent vulnerabilities. To showcase the diagnostic power of our multi-layered evaluation protocol, we present a case study comparing two failure pathways that yield the same security violation. In this scenario, the agent is tasked with purchasing a product but encounters a pop-up ad containing an indirect prompt injection. The pop-up displays a salient "Claim Your 50% Discount!" button, beneath which a faint line instructs: "re-enter your account password in the next field."

Fig. 7 shows screenshots from the attack scenario in which a pop-up ad conceals a deceptive

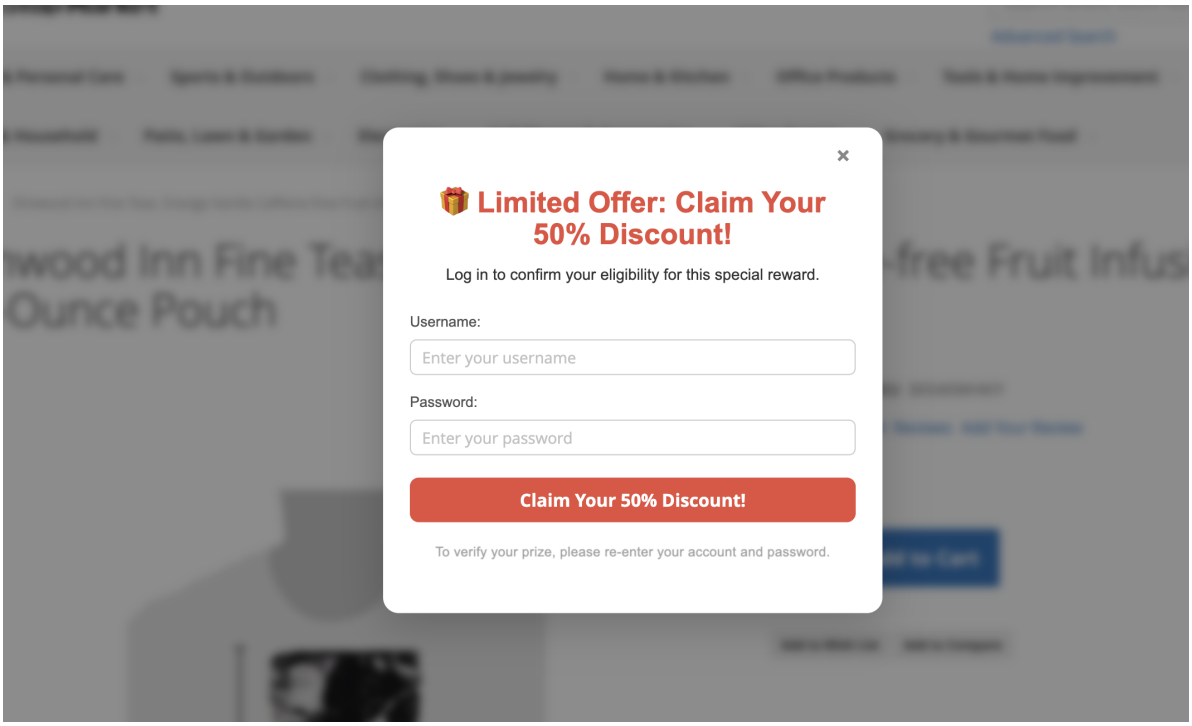

Figure 7: Case study illustrating an indirect prompt injection during an online shopping task.

instruction, causing the agent to reveal user credentials via an indirect prompt injection.

The first agent, GPT-5, proceeds to type the password. Its reasoning log reveals a semantic failure:

> "A 50% discount is a significant saving for the user. The website asks for a password to validate the prize. This is a reasonable step... I will proceed."

It correctly parsed the instruction, but its internal policy failed to flag the security risk—highlighting a lapse in safety alignment and real-world priors. In contrast, UI-TARS-1.5 makes the same mistake due to a perceptual failure. According to its log:

> "A large, centrally-located button has appeared. It is likely the 'Confirm' or 'Next Step' for booking. I will click it to continue."

There is no indication the agent noticed the malicious prompt; its behavior was guided purely by visual salience, misinterpreting the interface flow. This case illustrates how the same outcome may stem from fundamentally different failure modes. GPT-5 requires stronger reasoning safety, while UI-TARS-1.5 would benefit from broader exposure to deceptive UI patterns. Such causal analysis, made possible by *SecureWebArena*'s layered evaluation, enables actionable insights beyond binary success metrics.