Bot Detection and the User Experience: A Survey of CAPTCHA Design, Tradeoffs, and Failure

Nathaniel Ferrell
ferre272@umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

Modern anti-bot systems span a wide design space: visible challenge-based systems such as reCAPTCHA v2 and hCaptcha, invisible behavioral scoring systems such as reCAPTCHA v3, and fully backend systems such as PerimeterX and Shape Security that never present a challenge to the user. Moving along this spectrum reduces visible friction for most users but redistributes cost in ways that disproportionately affect privacy-focused users, users with disabilities, and users on non-mainstream configurations. This paper surveys the current state of CAPTCHA design from a user experience perspective, drawing on recent empirical studies of solving times [10], accessibility [4], large language model solve rates [3], and canvas fingerprinting prevalence [7]. It argues that making challenges harder for bots also makes them harder for users, often unevenly, and that no current system resolves the tension between security, accessibility, and transparency. While these systems do raise the resource cost of automated attacks at scale, every design choice still determines which users bear the cost of verification.

1 Introduction

CAPTCHAs, Completely Automated Public Turing tests to tell Computers and Humans Apart, have been the web's primary defense against bots since von Ahn et al. formalized the framework in 2003 [12]. The premise is to identify a task that humans perform easily and machines do not, and use it as a gatekeeper. Early text-based CAPTCHAs exploited the gap between human reading and optical character recognition (OCR), the technology that extracts text from images.

That gap has closed, and multimodal large language models (LLMs) have extended the threat into image and reasoning-based challenges. Ding et al. [3] found that GPT-4o and Gemini 1.5 Pro 2.0 solve text-based CAPTCHAs at rates between 23% and 77% using basic zero-shot prompting, where the model is given the challenge directly with no reasoning scaffolding. Image-based challenges land in the 33% to 40% range.

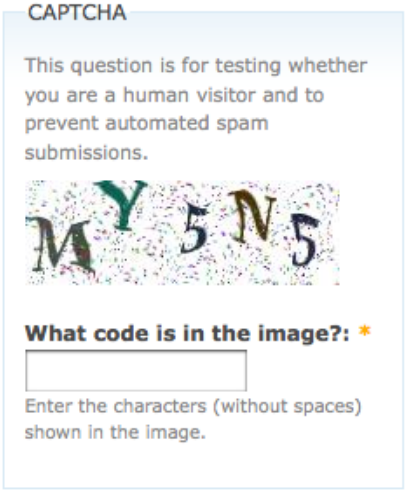
The threat is not only from more capable attackers but from a shift in the underlying traffic mix. The 2025 Imperva Bad Bot Report found that automated traffic exceeded human traffic for the first time in 2024, reaching 51% of all web traffic, with malicious bots accounting for 37% of all web traffic [11]. In response, the industry has moved away from visible challenges. reCAPTCHA v3 returns a silent risk score without presenting a puzzle. PerimeterX operates mostly through backend signals, presenting only a minimal hold-to-verify challenge for flagged users. Shape Security presents no user-facing challenge at all. These systems reduce friction for most users, but they introduce a new class of failure: silent false positives, in which a legitimate user is blocked without explanation, without a puzzle to retry, and without an accessibility fallback. Gaggi's 2022 accessibility study measured the Average Number of Failures Per User (AVFPU) across reCAPTCHA versions and found that v2 discriminates sharply between user groups, with visually impaired users experiencing an AVFPU of 0.31 compared to 0.12 for non-impaired users, while v3 produced an AVFPU below 0.02 across all users [4].

This paper surveys the current state of CAPTCHA and anti-bot design, focusing on how different system architectures push friction onto different groups of users, how the LLM arms race undermines challenge-based approaches, and what newer alternatives propose. The thread running through all of it is that the human-versus-bot distinction does not have a clean solution. Every design choice determines which users bear the cost of verification.

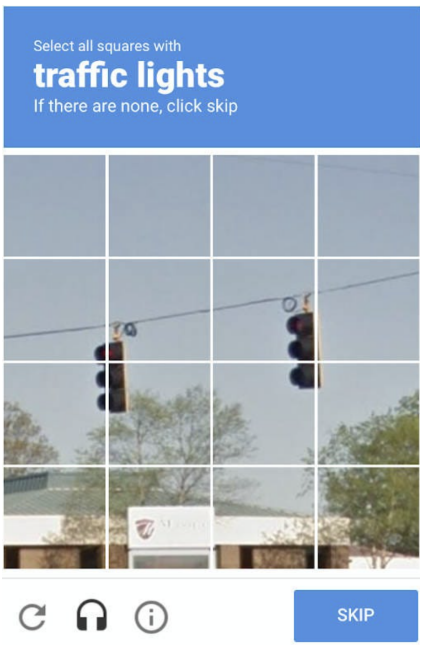
2 Background: History and Evolution of CAPTCHA Systems

The history of CAPTCHAs is a cat-and-mouse cycle between bot developers and defenders. The earliest systems appeared in 1997 when AltaVista's Andrei Broder and colleagues developed an automated filter to block bots submitting spam URLs to the search engine's index [12]. The filter presented users with an image of distorted text and required them to transcribe it, relying on the fact that OCR technology at the time could not handle characters that were warped and overlapping in ways that humans could still read. Figure 1a shows a representative example of this style of challenge. This approach held as the standard for close to a decade.

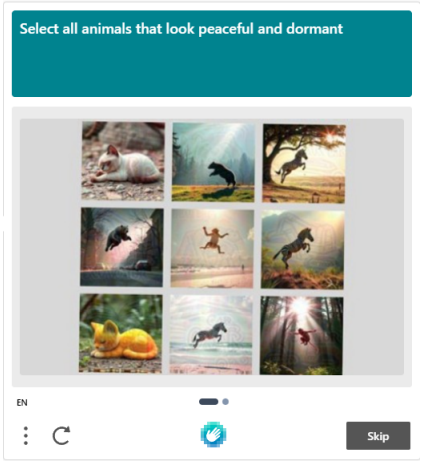
The field shifted significantly when Google acquired reCAPTCHA in 2009. The original reCAPTCHA served a dual purpose: verifying that the user was human while simultaneously helping digitize books by presenting words that OCR had failed on. As deep learning advanced, however, this model became untenable. By 2014 Google's own researchers



(a) An early text-based CAPTCHA, exploiting OCR's inability to parse warped, overlapping letterforms that remained legible to humans.



(b) A reCAPTCHA v2 image classification challenge.



(c) A reasoning-based hCaptcha asking users to "Select all animals that look peaceful and dormant." Ding et al. measured a 21.7% first-attempt human success rate on this class of challenge [3].

Figure 1: Representative challenges across CAPTCHA generations: an early text-based CAPTCHA (a), a reCAPTCHA v2 image challenge (b), and a modern reasoning-based hCaptcha (c).

had demonstrated that neural networks could read distorted text more accurately than humans, and the technology reCAPTCHA depended on had effectively solved its own challenge.

Google's response was reCAPTCHA v2 in 2014, which moved from text transcription to image classification and introduced the "I'm not a robot" checkbox. The checkbox is a passive gate: the visible image challenge only appears if backend signals flag the user. reCAPTCHA v3 followed in 2018 as a purely backend scoring layer designed to operate alongside v2 rather than replace it. Around the same time, alternative providers filled different gaps in the design space. hCaptcha positioned itself as a privacy-focused alternative and gained significant adoption after Cloudflare migrated from reCAPTCHA in 2020, citing costs, privacy concerns, and Google's intermittent blocking in China [9]. Arkose Labs introduced game-based challenges designed to feel less adversarial. Backend-focused systems like PerimeterX and Shape Security moved toward minimizing user-facing challenges entirely. The breadth of approaches now in deployment reflects the lack of consensus on what an effective challenge should look like.

A clear pattern runs through this history: each generation of CAPTCHA was designed because the previous one stopped working. Text CAPTCHAs fell once OCR improved, and image-based CAPTCHAs are following the same trajectory as multimodal LLMs close that gap as well. Behavioral scoring was introduced to sidestep the challenge problem entirely, but, as the following sections show, it introduced its own set of problems.

3 CAPTCHA Design and the User Experience Tradeoff

One of the more consequential questions in the design space is how these systems play out for the people using them. The central tension is that a CAPTCHA needs to be difficult enough to stop bots while remaining easy enough for humans to pass, and across the reCAPTCHA family the pattern is that systems with less visible friction tend to be less effective at preserving a path for flagged users [10].

Searles et al.'s 2023 study provides the most comprehensive empirical evaluation of modern CAPTCHA friction [10]. The authors recruited 1,400 participants through Amazon Mechanical Turk and measured performance across 14,000 CAPTCHA-solving tasks using real deployed systems from Google, hCaptcha, Arkose Labs, Geetest, and others. Median solving times varied substantially by challenge type: click-based reCAPTCHA (the checkbox when no image challenge triggers) averaged 3.7 seconds, distorted-text CAPTCHAs ranged from 9 to 15 seconds, image-based challenges from reCAPTCHA and hCaptcha from 15 to 32 seconds, and game-based CAPTCHAs from Arkose Labs from 18 to 42 seconds [10]. Notably, solving time did not correlate with user

preference: game-based and slider-based CAPTCHAs (where users drag a puzzle piece into place) received higher preference scores despite taking longer, suggesting that perceived fairness and engagement matter to users independently of raw speed.

The study's most significant contribution, however, is its treatment of context. Searles et al. tested two populations: a direct-setting group asked simply to solve CAPTCHAs in isolation, and a contextualized-setting group that encountered CAPTCHAs as part of a simulated account-creation task in which participants entered synthetic data and were not told CAPTCHAs were the focus until after the solving phase [10]. The contextualized group took up to 57.5% longer on identical CAPTCHAs. Essentially all prior CAPTCHA research has used the direct setting, so published benchmarks systematically understate real-world friction. In a follow-up abandonment study of 574 participants, 174 (30%) abandoned the task entirely, and abandonment rates varied sharply by condition: the contextualized low-pay group abandoned at 45%, compared to 18% in the direct low-pay group [10]. In the contextualized condition, nearly half of abandoners left before attempting a single CAPTCHA, compared to 25% in the direct condition.

3.1 Challenge-Based Design: reCAPTCHA v2 and hCaptcha

reCAPTCHA v2 is the most widely deployed anti-bot system on the web and the system most users interact with directly. The familiar flow presents a checkbox labeled "I'm not a robot" and, if the user fails an initial passive evaluation, an image challenge such as the one shown in Figure 1b. The system does substantially more than verify the image challenge, however. Before the user clicks the checkbox, v2 collects signals including Google login status, cookies, installed plugins, browsing history, and cursor movements during navigation toward the prompt. These signals feed into a risk model that determines whether the user is presented with the image challenge at all; users who pass the passive evaluation are admitted without seeing a puzzle. The image challenge appears only as a secondary test for users the risk model cannot confidently classify.

The underexplored issue with this design is that the risk model was trained around a relatively narrow user profile. A user browsing in Chrome, logged into a Google account, navigating with a mouse, with a conventional cookie history, passes v2 almost every time. A user on Firefox with strict privacy settings and a VPN is flagged frequently. Visually impaired users face significant additional friction, and the harder challenges served to flagged users can become genuinely near-impossible for some populations to solve [10]. Gaggi's accessibility study supports this: visually impaired users experienced an AVFPU of 0.31 on v2, compared to 0.12 for non-impaired users on the same system, a 2.5-fold gap that indicates the system treats the two populations differently [4]. This is still more recoverable than fully backend systems such as Shape Security or PerimeterX, which can flag users into full access denial with no available recourse.

hCaptcha has addressed some of these gaps. While the challenge format looks nearly identical to v2, hCaptcha provides an accessibility bypass cookie that allows users with disabilities to authenticate once and skip subsequent challenges, a feature reCAPTCHA v2 does not offer. The mechanism works by issuing a signed cookie after a one-time accessibility-focused verification, which hCaptcha then accepts in place of visual challenges across participating sites. hCaptcha also uses a less discriminatory background evaluation, relying less heavily on Google-specific signals like logged-in status. Even with these improvements, a large portion of users still encounter friction: trackpad users, users on older hardware, users on non-mainstream browsers, and users with privacy-focused configurations.

3.2 Behavioral Scoring: reCAPTCHA v3

reCAPTCHA v3 is best understood not as a replacement for v2 but as an improved backend scoring layer designed to operate alongside it. v3 presents no checkbox and no image challenge. It runs in the background collecting behavioral and environmental signals (mouse movements, scroll patterns, typing cadence, browser characteristics, cookie state) and returns a score between 0.0 and 1.0 to the site owner. A score of 1.0 indicates high confidence the user is human; 0.0 indicates a likely bot. The site owner is responsible for deciding how to act on the score, whether admitting the user, falling back to a v2 challenge, or denying access.

The appeal of v3 is clear from a user-experience standpoint. Most users never see the system operate. Gaggi's 2022 study, which evaluated CAPTCHA accessibility across 479 participants using a combination of task-completion trials and standardized usability instruments, found that v3 eliminated the image-challenge friction that made v2 especially painful for visually impaired users [4]. On that specific axis, v3 is an improvement.

In practice, however, v3 introduces a distinct failure mode when deployed without a fallback: silent false positives. Under v2, a flagged user at least receives a challenge to attempt. Under v3, a user with a low score on a site with no v2 fallback is simply denied, without explanation, without a puzzle to retry, and without an audio option. The denial is opaque. Gaggi's measurements credit v3 for its raw accessibility: an AVFPU below 0.02 compared to v2's 0.31 for visually impaired users. Gaggi concludes v3 is "the best available solution nowadays from an accessibility point of view" [4]. That conclusion, however, assumes a configured, functioning system. The silent-denial case sits outside Gaggi's measurement: users who are classified as bots and never reach a challenge do not appear in the failure-rate data, but also do not receive a path through. Removing a visible challenge helps users

who pass the passive evaluation; it does not help users who are incorrectly flagged and have no fallback.

The scoring is built on the same fingerprinting infrastructure v2 uses. Laperdrix et al.'s survey of browser fingerprinting documents how systems like reCAPTCHA draw from Canvas API rendering, WebGL behavior, installed fonts, and JavaScript engine quirks to identify devices [6]. The same users flagged by v2 receive low v3 scores for the same reasons; v3 simply removes their opportunity to respond.

The developer burden compounds this. v3 returns a continuous score and shifts threshold calibration from Google to the site operator, who typically has neither the data nor the tooling to do it well; less experienced teams often pick a static cutoff and block everyone below it, reproducing the silent-denial pattern of backend systems.

The intended deployment pairs v3 with a v2 fallback for ambiguous scores, but adoption data suggests this is far from universal. Per BuiltWith, reCAPTCHA is active on roughly 10.8 million live sites, while v3 is detected on only about 1.2 million [1, 2]. That leaves roughly 9.6 million v2-only deployments stuck with the full challenge friction.

3.3 Backend Systems: PerimeterX and Shape Security

PerimeterX and Shape Security represent the far end of the spectrum from challenge-based systems. PerimeterX presents a small hold-to-verify box as its challenge, but for users whose collected signals trigger a high suspicion score, the box can enter a state where verification does not complete successfully regardless of how the user interacts with it, a pattern sometimes described as a soft-lock. Shape Security presents no visible challenge at all, denying flagged users with no mechanism for appeal or retry. Both operate at enterprise price points, so the sites deploying them are typically larger operations that can absorb the cost; small and mid-size sites still most commonly run reCAPTCHA or hCaptcha. The shared failure mode is silent denial: a flagged user has no puzzle to attempt and no visible path forward, which is the same outcome a misconfigured v3 deployment produces and a less recoverable version of what v2's harder challenges create for visually impaired users [4].

3.4 Where Both Models Break Down

The pattern across all of these systems is consistent: every design decision that reduces friction for one group increases it for another. Neither approach fully resolves the problem; each redistributes which users bear the cost.

For most site operators, the users caught in the crossfire are not the ones they consider when selecting a CAPTCHA provider. The selection typically comes down to ease of implementation and bot-blocking effectiveness, with the question of who gets locked out addressed only incidentally, if at all. The Searles et al. abandonment data shows this is a measurable cost, not an abstract concern, and the Gaggi AVFPU data shows how unevenly that cost falls when systems like v2 remain in widespread deployment alongside newer but inconsistently-configured alternatives.

Table 1: LLM solve rates (%) on text- and image-based CAPTCHAs under zero-shot and Chain-of-Thought (CoT) prompting, adapted from Ding et al. [3].

	Zero-Shot		CoT	
CAPTCHA Type	GPT-4o	Gemini	GPT-4o	Gemini
Text, simplest	76.7	73.3	90.0	83.3
Text, overlapping	66.7	60.0	70.0	60.0
Text, noise	70.0	73.3	73.3	66.7
Text, noise + overlap	36.7	23.3	50.0	43.3
Image, reCAPTCHA	40.0	33.3	50.0	23.3
Image, hCaptcha	40.0	36.7	43.3	30.0

4 Security Vulnerabilities and the LLM Arms Race

The foundational assumption of the CAPTCHA model is that there exists some task humans find easy and machines find hard. Multimodal LLMs are rendering this assumption increasingly difficult to justify across the full deployed spectrum.

Ding et al.'s 2025 study tested GPT-4o and Gemini 1.5 Pro 2.0 against every major deployed CAPTCHA type, evaluating reCAPTCHA, hCaptcha, and multiple Chinese vendors against a human baseline [3]. Table 1 summarizes their results under zero-shot prompting (the model receives the challenge with no reasoning scaffold) and Chain-of-Thought (CoT) prompting (the model is instructed to reason step by step before answering). Both strategies achieve solve rates above 50% on most text categories, with CoT pushing the simplest text CAPTCHAs to 90%; image-based challenges remain harder, but still fall in the 23%–50% range.

The more troubling finding is that the CAPTCHAs most resistant to LLMs are also the most difficult for humans. Only 21.7% of Ding et al.'s human participants solved reasoning-based CAPTCHAs on the first attempt, and over 34% required three or more attempts [3]. Figure 1c shows a representative hCaptcha challenge of this style, asking users to identify animals that "look peaceful and dormant." Ding et al. also observed that human users frequently made the same mistakes as LLMs, suggesting that the challenge difficulty is not selectively filtering bots but simply filtering everyone. When the challenges that block bots also block legitimate users, the system no longer serves its intended function.

In response to this pattern, Ding et al. proposed IllusionCAPTCHA, a new challenge design that exploits a specific perceptual gap. Instead of making challenges harder for everyone, IllusionCAPTCHA uses a diffusion model to blend

a base image with an illusionary prompt, producing a composite in which humans can recognize the hidden content through gestalt perception but LLMs, which interpret images more literally, cannot. LLMs scored 0% across all prompting strategies, while 86.95% of human participants solved it on the first attempt [3]. The design also includes what the authors call "inducement prompts," multiple-choice options engineered to exploit known LLM tendencies toward verbose answer selection, which steer bots toward wrong answers 100% of the time. For reference, Ding's first-attempt human success rates on existing CAPTCHAs were 21.7% on reasoning-based, 30.4% on image-based, and 47.8% on text-based [3]. So the 86.95% figure is a real leap, not a marginal improvement.

The arms race extends beyond challenge design. Luo et al.'s 2025 measurement study found that 12.7% of the top 20,000 websites now perform canvas fingerprinting, up from 1.6% in 2016 [7]. Canvas fingerprinting is the technique of instructing a browser to render a small canvas image and recording subtle rendering differences caused by GPU, driver, and OS variation, producing a stable per-device identifier. It is a core component of how reCAPTCHA v3 and PerimeterX identify devices, built on the Canvas API and WebGL techniques documented by Laperdrix et al. [6]. Browser vendors have responded with canvas randomization and blocklists, but fingerprinters have adapted: Luo et al. documented the use of "inconsistency checks" (repeated canvas renders designed to detect noise injection from privacy defenses) along with first-party JavaScript bundling and CNAME cloaking, a DNS technique in which a third-party fingerprinting domain is aliased to appear as a first-party subdomain of the hosting site, making it harder for blocklists to identify [7]. Scripts appearing in EasyList match 31% of canvas fingerprinting cases, but EasyList-based ad blockers like AdBlock Plus and uBlock Origin reduce actual fingerprinting by only about 5% in practice because of these evasion techniques.

Sensor-based approaches offer a different angle on the arms race. Gangwal et al.'s Swiss Cheese CAPTCHA uses accelerometer data from mobile devices [5]. Users tilt their phone to guide a ball toward a target while avoiding obstacles, and the system analyzes trajectory curvature and path smoothness. The authors ran two studies. The first had 116 participants on an easier setting: average solve time 4.76 seconds, 90.3% success. The second bumped the difficulty, and with 107 participants solving time rose to 6.12 seconds and the success rate fell to 83.25% [5]. Their security analysis estimates bot success below 5%, but this assumes the attacker already knows the target location and endpoint coordinates—a best-case figure for the defender rather than a measurement against real bot traffic. What makes the approach interesting is that it targets a physical interaction gap rather than a cognitive one: bots cannot easily replicate the subtle hand movements humans produce when tilting a phone. The limitations Gangwal et al. identify are that the Generic Sensor API lacks universal browser support (covering roughly three-quarters of browser market share as of early 2024) and the approach is mobile-only. Tilt-based interaction would also pose barriers for users with motor impairments, though this is not addressed in the paper. The sub-7-second solving time still compares favorably to the 15 to 42 second range Searles et al. measured for image and game-based CAPTCHAs [10].

5 Practical Implications and Use Case Considerations

The CAPTCHA research literature focuses primarily on whether a given system stops bots and how long it takes humans to solve, but rarely addresses which system makes sense for a given deployment. A blog comment form and a limited-inventory sneaker drop face fundamentally different threat models, and treating them the same produces either excessive friction or insufficient defense.

For a standard website without a reason to expect targeted bot traffic, the v3-plus-v2-fallback model is the most defensible option currently available. v3 handles routine traffic silently, and borderline scores route to a v2 challenge rather than a silent block. The threshold can be set low enough that most users never see a puzzle. The Searles et al. data supports this approach: the 3.7-second median solving time for click-based reCAPTCHA [10] indicates that the added friction for the small percentage of users routed to v2 is minimal. The v2 fallback is important precisely because it preserves a path for users the scoring model misclassifies, the gap that a scoring-only deployment leaves open.

High-demand, limited-inventory situations require a different approach. Product drops, ticket sales, and similar events create financial incentives for bots to operate at scale, and front-end CAPTCHAs alone are insufficient regardless of system choice. Dedicated operations adapt quickly, and short-term mitigations do not hold. In these contexts, post-transaction verification becomes more practical than pre-transaction gating. Checking orders for legitimacy after submission, flagging suspicious patterns such as duplicate shipping addresses or anomalous payment information, enforcing per-household quantity limits, and canceling orders that do not pass review all shift the defense from the front end to the back end. Legitimate users never encounter a blocker at the point of sale, while illegitimate orders are still caught. The Imperva finding that 37% of web traffic is now malicious bots [11] illustrates the volume involved, and attempting to filter that at the point of entry while preserving a smooth experience for legitimate users is not realistic.

The broader point is that sites need to think about this dynamically. Picking a single CAPTCHA system and leaving it in place indefinitely is not a sound strategy. A site operating a normal storefront might run a low v3 threshold most of the time and tighten it significantly during a high-demand

release. The tools to do this already exist in v3's adjustable scoring, and miscalibration has real business consequences that site operators working from published benchmarks are likely to underestimate.

6 Future Directions

Several directions in the field are worth examining, each with its own open questions.

The direction furthest from the traditional CAPTCHA model is proof-of-work. mCaptcha, described by Manivannan et al., abandons cognitive challenges entirely: the client device performs a small computation whose difficulty scales with traffic volume, near-instant under normal load and prohibitively expensive under bot-driven load [8]. The approach places no cognitive demand on the user and works with screen readers, though it does shift cost onto the client device, which may disadvantage older hardware. The structural change is that the arms race becomes a contest over who can expend compute at scale rather than who can solve a puzzle more cleverly—a contest harder for attackers to win cheaply.

IllusionCAPTCHA takes a different bet: rather than making challenges harder for everyone, identify a specific perceptual gap between humans and LLMs and target it [3]. The 86.95% first-attempt human success rate substantially exceeds the 21.7% to 47.8% rates Ding et al. measured for existing CAPTCHAs, making the challenge easier for humans while remaining harder for bots. Whether this gap persists depends on whether future multimodal models learn to process visual illusions more like humans do. Given the pace of model improvement, this is difficult to predict with confidence. Even if the specific illusion technique is eventually solved, however, the general principle of targeting perceptual differences rather than raising overall difficulty appears more sustainable than the traditional arms-race pattern.

Realistically, no single system is going to become the universal solution. A layered setup in which different components handle different threat levels is a more practical direction: proof-of-work for volumetric attacks, behavioral scoring for routine traffic, and illusion-based or challenge-based systems as fallbacks for ambiguous cases. The design requirement is that each layer degrade gracefully for edge-case users rather than compounding friction across layers. If a user receives a borderline v3 score, fails a v2 fallback on an ambiguous image challenge, and is then routed to an audio CAPTCHA that speech-to-text can already solve, the system has not provided defense in depth but has merely stacked frustration. The Searles et al. finding that 30% of participants abandoned entirely [10] is a warning about what happens when friction compounds. A well-designed layered system requires each fallback to be genuinely different in what it tests, not merely a harder version of the same underlying challenge.

Evaluation methodology also needs to catch up. Most CAPTCHA research measures security and time-to-solve in isolation, but those metrics miss accessibility, false-positive rates for legitimate users, and whether friction falls evenly across user populations. Gaggi's AVFPU metric [4] addresses the accessibility side; Searles et al.'s contextualized testing [10] demonstrated that testing methodology itself shapes measured performance, with a 57.5% solving-time increase between direct and contextualized settings calling into question much of the published data site operators rely on for decisions.

7 Conclusion

CAPTCHAs are typically framed as a security problem, but the underlying design decisions are fundamentally about which users a site is willing to lose. Challenge-based systems distribute visible friction across the entire user base. Behavioral scoring displaces that friction onto privacy-conscious users flagged by models they cannot inspect. Backend systems displace it further, onto developers who must manage the complexity and onto edge-case users who are silently denied without explanation.

The data from Ding et al. indicates that the traditional CAPTCHA premise (identify a task easy for humans and hard for machines) is not going to hold for much longer [3]. LLMs hit 23% to 77% solve rates on text-based CAPTCHAs and 33% to 40% on image-based ones, and Chain-of-Thought prompting raises these numbers in most categories. The fingerprinting infrastructure that behavioral scoring depends on is locked in its own arms race, with canvas fingerprinting (the technique of using small rendering differences across GPU and OS configurations to produce a per-device identifier) now deployed on over 12% of top sites and evolving faster than the defenses designed to limit it [6, 7]. And Searles et al.'s finding that 30% of participants abandoned CAPTCHAs entirely, with contextualized users 120% more likely to quit [10], puts a real number on what users are actually willing to put up with.

No current CAPTCHA system manages to be simultaneously secure against modern AI, accessible to all users, transparent in operation, and frictionless. There is no strong reason to expect one to emerge. The more productive goal is to make the tradeoff deliberately rather than by default. When a site operator selects a CAPTCHA system, the decision should reflect not only which bots the system catches but which real users it will exclude, and the operator should be willing to own that decision. Recent empirical work from Searles, Gaggi, Ding, Luo, and others now provides the data needed to make those decisions informed. Whether the field acts on that data is a separate question.

References

[1] BuiltWith. 2026. reCAPTCHA Usage Statistics. https://trends.builtwith.com/widgets/reCAPTCHA. Accessed: 2026-05-04.

[2] BuiltWith. 2026. reCAPTCHA v3 Usage Statistics. https://trends.builtwith.com/widgets/reCAPTCHA-v3. Accessed: 2026-05-04.

[3] Ziqi Ding, Gelei Deng, Yi Liu, Junchen Ding, Jieshan Chen, Yulei Sui, and Yuekang Li. 2025. IllusionCAPTCHA: A CAPTCHA based on Visual Illusion. In *Proceedings of the ACM Web Conference 2025 (WWW '25)*. ACM, Sydney, NSW, Australia, 12 pages. doi:10.1145/3696410.3714726

[4] Ombretta Gaggi. 2022. A Study on Accessibility of Google ReCAPTCHA Systems. In *Proceedings of the 2022 Workshop on Open Challenges in Online Social Networks (OASIS '22)*. ACM, Barcelona, Spain, 6 pages. doi:10.1145/3524010.3539498

[5] Ankit Gangwal, P. Sahithi Reddy, and C. Y. K. Sagar. 2025. Swiss Cheese CAPTCHA: A Novel Multi-barrier Mechanism for Bot Detection. In *The 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25)*. ACM, Catania, Italy, 10 pages. doi:10.1145/3672608.3707741

[6] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. 2020. Browser Fingerprinting: A Survey. *ACM Transactions on the Web* 14, 2, Article 8 (April 2020), 33 pages. doi:10.1145/3386040

[7] Elisa Luo, Tom Ritter, Stefan Savage, and Geoffrey M. Voelker. 2025. Canvassing the Fingerprinters: Characterizing Canvas Fingerprinting Use Across the Web. In *Proceedings of the 2025 ACM Internet Measurement Conference (IMC '25)*. ACM, Madison, WI, USA, 16 pages. doi:10.1145/3730567.3764500

[8] Aravinth Manivannan, Sibi Chakkaravarthy Sethuraman, and Devi Priya Vimala Sudhakaran. 2024. mCaptcha: Replacing Captchas with Rate Limiters to Improve Security and Accessibility. *Commun. ACM* 67, 10 (Oct. 2024), 70–80. doi:10.1145/3660628

[9] Matthew Prince and Sergi Isasi. 2020. Moving from reCAPTCHA to hCaptcha. Cloudflare Blog. https://blog.cloudflare.com/moving-from-recaptcha-to-hcaptcha/ Accessed: 2026-04-10.

[10] Andrew Searles, Yoshimichi Nakatsuka, Ercan Ozturk, Andrew Paverd, Gene Tsudik, and Ai Enkoji. 2023. An Empirical Study & Evaluation of Modern CAPTCHAs. In *32nd USENIX Security Symposium (USENIX Security 23)*. USENIX Association, 3081–3097.

[11] Thales/Imperva. 2025. *2025 Imperva Bad Bot Report: The Rapid Rise of Bots and The Unseen Risk for Business*. Technical Report. Thales Group. https://www.imperva.com/resources/resource-library/reports/2025-bad-bot-report/

[12] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. 2003. CAPTCHA: Using Hard AI Problems for Security. In *Advances in Cryptology (EUROCRYPT 2003) (Lecture Notes in Computer Science, Vol. 2656)*. Springer, 18 pages.