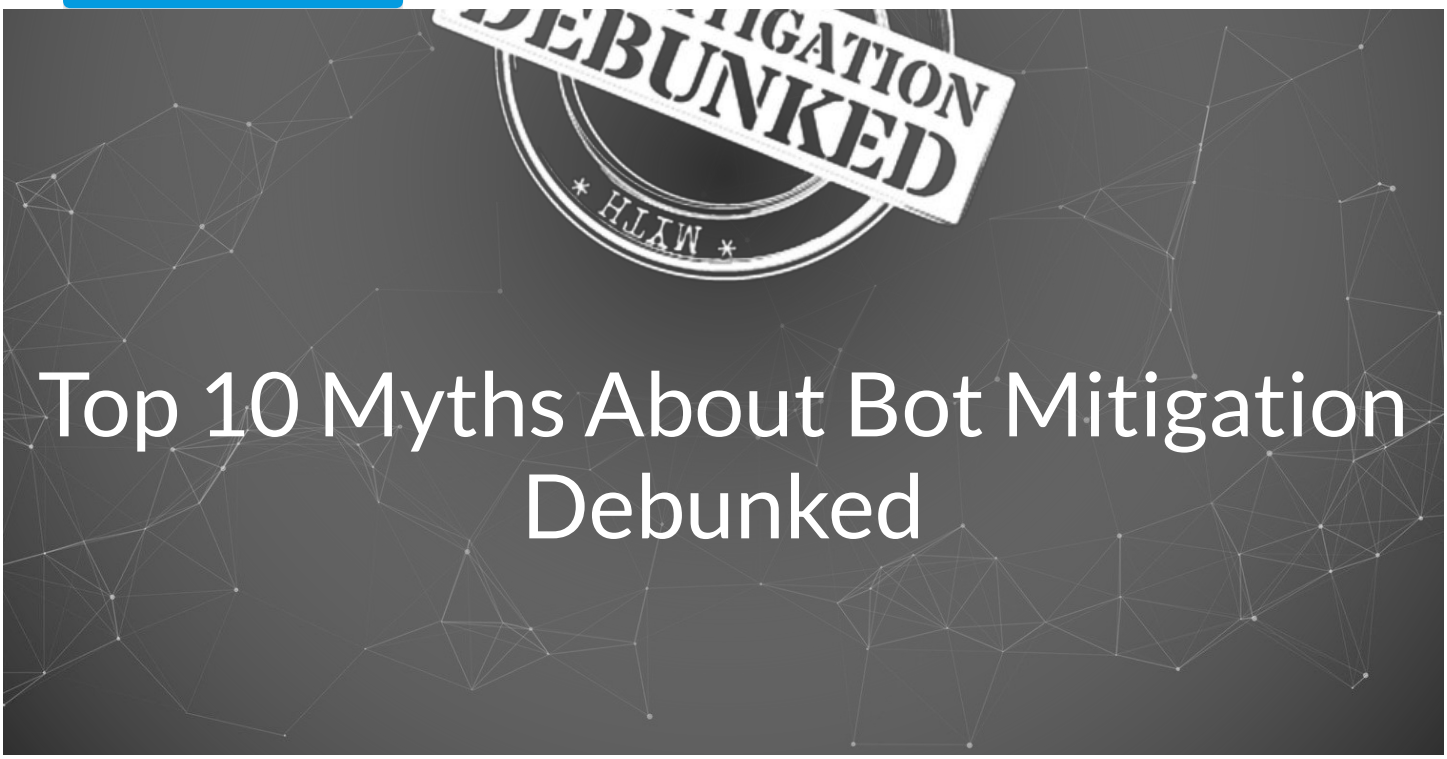
kasada

REQUEST A DEMO




# Top 10 Myths About Bot Mitigation Debunked

There's a lot of innovation and contrasting architectures in the bot mitigation and detection industry, so it can be difficult to distinguish between them and determine fact from fic There's no silver bullet when it co stopping bots. Solutions must be and accommodate for a broad ra detection methods, mitigative actions,

and understanding of adversarial methods to remain effective.

Here's 10 myths we often hear within the industry that we'd like to shed some light on:

# Myth #1: Client-side JavaScript isn't effective for detecting modern bots

It's true that client-side JavaScript bot detection done poorly isn't effective. If bot mitigation providers don't obfuscate their code properly, then attackers will be able to reverse engineer and bypass your anti-bot solution. But when done correctly, it is highly effective. Unfortunately, there are still prominent

solutions in the marketplace that give client-side detection a bad rap; they either don't secure their code at all, or they use open-source obfuscation methods that are easy to reverse engineer. The answer, however, isn't to abandon client-side JavaScript and shift solely to server-side detection. Client-side JavaScript exposes an important data dimension that allows for real-time detection of modern bots on the first request, by identifying the strong indicators of bot-driven automation, without having to monitor behaviors server-side in advance of making decisions.

**REALITY**: *Combining client-side and server-side detection yields the maximum benefit to stopping modern, automated attacks before they're allowed to enter your infrastructure – each able to detect different attributes and complement one another. Dynamic polymorphic techniques with encrypted communications are effective at providing long-term resilience for preventing the reverse engineering of JavaScript code.*

# Myth #2: Machine Learning is the best way for stopping modern bots

Machine learning (ML) trains models based on historical data. By nature, it takes time to look over new data before it can make a semi-accurate decision. The time to make a decision can range from minutes to days depending on the solution, during which time bot operators can send thousands of undetected requests. This leaves a sizable window where a threat actor can successfully exploit ML-based defenses. An approach that solely relies on ML will be slow to respond to evolving automated attacks, and an attacker only has to change their attack behavior slightly to fool it again. In addition, attackers are increasingly skilled at tricking ML models into producing faulty outputs by feeding models with fake data and spoofing request headers using readily available services such as residential proxy

networks.

**REALITY**: *Machine learning requires analyzing behaviors to identify anomalies. While undoubtedly an important tool for bot detection, it relies on historical data, cannot stop bots on the first request, and is vulnerable to bots that fly beneath the radar by acting just like humans and those able to train and trick the model. Server-side data analytics, coupled with real-time client-side detection, provide a combination of proactive and reactive detection techniques to add layers of depth that a bot operator must now overcome to be successful.*

# Myth #3: Bot mitigation is most effective implemented at the edge

There are tradeoffs unique to organizations that must be considered when it comes to deciding where to implement bot mitigation. Those who

place minimizing latency at a premium will benefit from bot mitigation at the edge, but not as much as you'd think. Studies show about a 30 millisecond decrease in page load times when bot detection decisions are made from the ISP edge instead of a nearby cloud data-center. Meanwhile, implementing bot mitigation at the edge likely locks you into a CDN that may be prohibitive for cost, multi-CDN implementation or other design considerations. On the other hand, those who place security at a premium benefit by implementing bot mitigation near the edge, but not at the edge. One benefit to this is having defenses "hide" behind a CDN, making it far more difficult for adversaries to know what they're up against. Another major security benefit is having a more centralized view of bot activity with the ability to adapt quickly. Highly distributed edge security platforms ironically have difficulty detecting highly distributed bot attacks. This is because edge nodes aren't working together, making distributed attacks such as those that come in low and slow easy to sail

through. Highly distributed platforms are also slow to react to new threats, as they are unable to rollout new configurations instantaneously and unlikely designed to inject new defense scripts from the edge into the client.

**REALITY**: *The imperceptible latency improvement implementing bot mitigation at the edge is often offset by the defense and cost improvements gained by implementing a more agile and resilient architecture from a nearby cloud data-center, behind a CDN. Also, CDN scale for delivering static content doesn't necessarily translate to bot mitigation scale. Deciding which is better for you depends on your priorities and the tradeoffs you wish to make.*

# Myth #4: Manage bad bots, but don't mitigate them

There's a fallacy that if you manage bots – such as serving them fake content, or tarpit them – you will fool their operators into thinking they've been successful in

their methods. Whereby mitigating bots by blocking them outright would make bot operators aware of your defenses and have them come back even more determined than ever. The reality is bot operators are smart and highly motivated. They have means to determine when bot management techniques are employed and attempt to deceive them. They will run health checks, test other URLs/APIs, leverage published bypasses, in order to find vulnerable entries into your online business and return even more determined than ever. What is effective is to respect the adversary, whether mitigating or maganging (or some combination of the two), and develop an effective plan to stop them by slowing down their development, iteration, testing and compute cycles – thereby making the target harder to beat. While also taking the economics away by making attacks too computationally expensive for them to conduct at-scale. By doing so, attackers will move on by finding easier targets to generate a profit.

**REALITY**: *Adversaries aren't easily fooled*

*by the range of bot management actions that can be served to a bot request. Making the target difficult to develop and iterate new bots against, along with striking back with increasingly difficult challenges, has shown to be highly effective at deterring persistent bot attacks over the long-term.*

# Myth #5: Petabytes of threat intel data is a competitive advantage

IP address reputation data offer very little advantage to a bot mitigation solution. The dynamic nature of residential proxy services means that at best you will be only blocking data-center services, and at worst you will be blocking legitimate users. Also, the value of data expires quickly in the anti-bot industry. A minute is a lifetime for bot mitigation services, and most anti-bot detection services aren't agile enough to ingest this data and update rules in order to take action quickly enough. Leveraging

large data sets is great as an out of band detection framework; however, these systems are rarely fast enough to effectively block bad bots.

**FACT**: *Attackers hide behind residential proxy networks which makes the value of threat intel data based on the deemed reputation of an IP address suspect and short lived. Data must be acted upon quickly before it becomes useless, and most anti-bot solutions aren't able to act upon such data quick enough to invalidate requests based on IP-based threat intelligence. So while threat intel can be a competitive advantage, understand how it can be actioned in real-time to detect and stop bad bots.*

# Myth #6: Advanced device fingerprinting creates a unique human identifier

Device fingerprinting has become an increasingly outdated method for

identifying humans. One reason is because adversaries harvest digital fingerprints using browser or device malware. Fraudsters silently collect real cookies and other browser information, such as mouse clicks and then resell this data, usually on the dark web. These copies of real user sessions and browser data can be loaded into bot frameworks to imitate a real user who is using a browser and fools the data collection and classification process of bot mitigation vendors. In addition, it has become more difficult to obtain “hi-res” device fingerprints for humans as they increasingly disguise themselves for web privacy considerations. The anti-tracking movement and privacy advocates resulting in the increased use of residential proxy networks, adoption of privacy browsers & modes, and plans to eliminate 3rd party cookie tracking – are altogether making device fingerprints less unique and more difficult to distinguish between bots and humans.

**REALITY**: *It is becoming more difficult to collect device fingerprints and adversaries have learned to harvest them to trick the*

*systems which rely upon them. Instead of trying to uniquely identify each request as a human with fingerprinting, other client-side data must be collected to interrogate any request and ensure requests present themselves in the context of a legitimate browser. Instead of attempting to identify whether a request is from a human, instead determine whether a request demonstrates the immutable evidence associated whenever bots interact with online businesses.*

# Myth #7: CAPTCHA is good to use some of the time, when you are unsure

CAPTCHAs add friction to the user

experience and consistently lead to conversion loss. Yet bot operators like it when organizations use them, as they have built bots that can solve CAPTCHAs efficiently and cost-effectively. They use "CAPTCHA farms" such as 2CAPTCHA, which outsource CAPTCHA validation to areas of the world where labor is cheap, ensuring that CAPTCHAs present no obstacles to well-funded, semi-technical bot builders. Attackers can validate CAPTCHAs for less than $1 per 1,000 solved CAPTCHAs – so now the cheap and easy security control is frustrating your paying customers, but not the fraudsters. In addition, applying image processing and ML has been proven effective for bot builders to automate the solving of sophisticated and specialized CAPTCHAs.

**REALITY**: *The rationale for using CAPTCHAs, including those which are more advanced and/or introduce less customer friction, is a decision avoidance solution. You are taxing potential customers and risking lost sales, but not really impacting bot builders, as CAPTCHAs of all types are*

*capable of being bypassed. A bot mitigation provider serving a CAPTCHA is like saying – "I don't know, I give up and here's the key to access the castle if you solve this puzzle".*

# Myth #8: Human cognitive science distinguishes bots from humans

The use of cognitive science is an attempt to distinguish between requests from bots and humans by attempting to measure typing speed, mouse movement, accelerometer data on mobile phones, and other gestures. In theory, this makes sense as there are clear differences that distinguish between the two by identifying the imperfections associated with humans. In practice, however, bot operators have wisened up to this approach and have developed request bots aimed to insert legitimate recorded sessions that present the bot just like a human.

**REALITY**: *human gesture data is easy to*

*record and widely available for inexpensive purchases on the web, which is used to bypass defenses based on cognitive science.*

# Myth #9: False positives are an issue, but false negatives aren't

Whenever an anti-bot solution is based on risk scoring – most of which are – there's an inherent balancing act. Tightening your tolerance for risk results in blocking more legitimate users (false positives) while loosening too much results in the opposite. So false negatives are always an issue when making tradeoffs against your risk tolerance. What makes matters worse is that most anti-bot reporting doesn't let you see what requests get through their system, only those that get blocked. So how are you best able to understand the risk tradeoff decision you are being forced to make? Furthermore, false negatives are exacerbated not only because of risk

tolerance, but also because modern bots, such as those generated using Puppeteer Extra Stealth hidden behind residential proxy networks, are extraordinarily difficult to detect using risk scores and behavioral analysis. And without portal visibility of what requests are blocked vs. what get through, you are none the wiser until it's too late. Our experience has shown that anywhere from 30-70% of bad bots legacy anti-bot systems should block, aren't.

**REALITY**: *false negatives are a cumulative result of risk tolerance decisions and also from bots that go undetected. Without proper visibility, it makes reconnaissance difficult while putting your business at additional risk of online fraud and other automated attacks.*

# Myth #10: Bot detection accuracy gets better with new release updates

While most anti-bot providers

systematically provide release updates based on new enhancements to detection capabilities, these benefits are often for nought if the adversaries have figured-out how to bypass detection – making your R&D essentially worthless. There are many means to bypass anti-bot solutions: reverse engineering poorly obfuscated JavaScript, weak encryption of data in-flight, static detection logic, and static script structure to name a few. While adding new detection capabilities to any anti-bot solution should be helpful, request bots that leverage published bypasses easily accessible on the Internet give adversaries a "back door" into your systems which can greatly detriment the benefits gained by improving other aspects of the system.

**REALITY**: *the best way to determine whether your anti-bot solution is subject to bypass methods is to search online for recent exploits using Google, GitHub and within Discord groups. You should be concerned about new updates if they don't address the root cause of an available bypass which are widely available and*

*shared across the bot-building community.*

# What to do?

As mentioned, there's no silver bullet when it comes to stopping bots. Anything that was built or engineered can be deconstructed or reverse engineered when there's the correct motivation to do so. Attackers will look to exploit the weakest element of your system. At Kasada, we take the approach of thinking like an adversary. Our architecture and product direction accounts for how the system might be exploited to build layers of depth and resilience to our anti-bot platform.

To be effective at stopping modern bots, you need as many tools in the toolbox as possible, with an architecture that is able to put the right tool for the job to work instantly. Our experience interacting with billions of bots tells us there are certain occasions when client-side detection is necessary, and other times when data learned server-side is essential. There's often occasions where

what's been learned server-side needs to be implemented immediately on the client-side to get the maximum benefit and desired outcome. We continue to overcome the tricks bot operators leverage to bypass detection and reverse engineer solutions – in a way that requires no rules to manage, no risk scores to assign, and no CATPCHAs to validate. At the same time, we are fully transparent with our customers by providing visibility into all requests we process – not just those which are blocked.

Our flexible, adaptive architecture coupled with our expert bot hunting team, enables customers to protect themselves against modern bot attacks, including those never seen before.

Request a demo now to see why…

- more than 85% of our customers were using other bot mitigation customers prior to contacting us
- we stop more than 5 billion requests every month that were inadvertently missed by the legacy

anti-bot systems in front of us

- we are trusted to protect more than 2 billion in eCommerce transactions every month

… while stopping the bot attacks that others can’t.

Request Demo

Request Demo

Sign up now to receive the latest on automated threats and bot activity from Kasada.

Email*

Submit

**CYBERSECURITY THREATS**

- Cyber Threats
- Denial of Service
- Contact Scraping
- Account Takeover
- System Takeover
- Bot Mitigation
- Bot Management Solutions

**HELPFUL LINKS**

- Product
- About
- Resources
- Careers
- Compliance
- Contact

**CONTACT US**

125 Park Ave
New York, NY
10017
877-473-5073

822 George St,
Chippendale
NSW 2008
1300-768-601