Theory & ROI of Shadow Bans

Was bedeutet Shadow Ban?

The “Shadow Ban” concept refers to blocking an action without revealing that fact to the user, based on a real-time risk analysis like the hCaptcha Enterprise risk score.

For example:

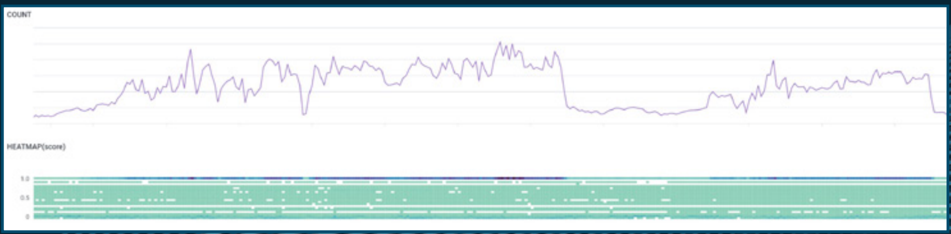
- rejecting a login with the same error message you’d use if the credentials were invalid
- tagging a signup such that the account created is unable to do anything harmful
- flagging a purchase so that it will be rejected quietly after the initial success message

Why are Shadow Bans useful, and when do I need them?

We strongly recommend using shadow bans in all implementations of hCaptcha Enterprise where an immediate (rather than delayed/ offline) action will be taken based on the risk analysis.

Why? A small amount of work gives a large payoff in greater detection percentage and stability. To illustrate this, let’s examine a real traffic example.

An Effective Shadow Ban in Practice



Over the two week period above, we see bad actors running several long-duration attacks (score of 1 in the heatmap). The goal was to use a credential stuffing tactic to find valid logins.

They fail in multiple attempts to evade detection because they cannot confirm if their changes were effective.

Here, we see them first attack for over a week continuously, without realizing none of their requests were succeeding. They eventually figure out that something is wrong, make several changes to their methods, and try again for several more days, again failing to realize that all of these malicious requests were still detected in real-time and would give them zero benefit.

Why is this valuable?

In the earlier example, a persistent threat was entirely neutralized for more than two weeks with zero human labor required from the defender or the threat platform.

Detections were stable and immediate in both cases, and the only action required by the defender was to simply consume the risk score, in this case to fail the login exactly as they would if the credentials were invalid.

The defender received many benefits:

- No attack logins were successful.
- If the defender opted to check the login credentials for high scores, they got high confidence detection of compromised accounts “for free” and could

then lock the accounts or require a 2FA confirmation and password reset on next login.

- The attacker was neutralized for several weeks.
- No adaptation effort was required to neutralize the attack.
- Many attackers simply get discouraged and move on to easier targets after a few weeks or months of wasted effort.

The attacker incurred substantial costs:

- They wasted resources during the attack.
  - Attackers are often paying for stealth proxies or rental of stolen IPs/devices from botnet operators, may be paying for cloud compute resources, and may be paying for clickfarm requests to solve challenges if visible challenges are shown.
- They leaked attack platform information during the attack: IPs, signatures, etc.
  - This facilitates detection of future, perhaps more sophisticated attacks.
- Most importantly, they wasted weeks of their time and got nothing for it.
  - Most people will do a cost/benefit analysis and eventually give up or move to easier targets when this happens repeatedly.

What happens if I don't use Shadow Bans, but instead use hCaptcha risk scores to block requests outright?

You are leaking valuable information when obviously blocking requests based on scores. This allows attackers to rapidly determine that a ban is in place, stop their attack, and start repeatedly trying cheap and efficient experiments to overcome defenses by varying every attack property under their control until they find a combination where the risk score is lower.

Once this happens and they start a higher volume attack, additional defenses take effect: our anomaly systems, ML, and other self-supervised learning algorithms. If you are an APT Mitigation subscriber, our SOC team will also notice and adjust system behavior in the event that the attacker manages to escape detection through repeated experimentation.

However, it is much less effective to operate in this manner: you go from a proactive to reactive security posture, and create additional defense costs in needing to remediate the first part of an attack until system detection of a novel approach reaches confidence and boosts scores.

Who uses Shadow Bans with hCaptcha Enterprise?

Every large hCaptcha Enterprise user has implemented shadow bans, including the world's largest fintech firms, leading merchant and game platforms, and many major online services.

After more than half a decade of applying this strategy in the real world, we have demonstrated its value numerous times. In the worst case scenario (an attacker with perfect information) you are no worse off than before, while in the best case scenario you have dramatically improved your security posture at minimal cost.

Quantifying the benefit of Shadow Bans

hCaptcha customers who implement shadow bans often see:

- Up to 20% faster detections of sophisticated threats
- Up to 40% more stability in detection of individual threat actors over a 30 day period

This results in fewer remediations and a high dollar value in additional real-time detections.

Why are Shadow Bans on hCaptcha scores a durable strategy?

Because hCaptcha Enterprise uses a genuinely adaptive and fully customizable ML platform, risk scores can both express business logic and rapidly adapt using automated learning loops.

This makes possible stable, fully automated detection of even the most complex threat actors on the largest, highest value application flows when Shadow Bans are implemented correctly.

Why don't other vendors advocate for Shadow Bans?

hCaptcha risk scores are both highly customizable and available only to Enterprise customers.

By comparison, other major security vendors like reCAPTCHA show their scores to anyone who signs up anonymously. This makes it trivial to fully reverse engineer and defeat their platforms.

Smaller, more marginal players like Arkose with immature detection technology may be harder to sign up for, but they fail to spot many attacks quickly. The common strategy of smaller players is to try to find new attack patterns using the "eyeballs on screens" approach to make up for less effective automated capabilities, and then have analysts write manual rules to block attacks.

This approach causes a high rate of quick detection escape as their rules are brittle, and a high risk of false positive detections. Excessive human intervention greatly increases the risk of incorrect analyst rules blocking real users, and this is a common complaint we have heard from many former customers of second-tier players like Arkose who have migrated to hCaptcha.

When can human SOC review complement automated detection?

While a less technical approach can work for low volume threats, it is largely ineffective against more sophisticated and higher volume attacks due to the latency of human response.

"Eyeballs on screens" can be a complementary adjunct when dealing with primarily human threats, but the goal of hCaptcha Enterprise is to automate the detect-escape-remediate-detect loop such that defender costs are minimized while attacker costs remain consistently high.

Shadow Bans are a valuable part of that strategy, in combination with segregation of different application flows into unique risk models (i.e. one sitekey per flow), applying your business logic via rules and rate limits, and using our reporting APIs to automate retraining when needed.

Human review will often do more harm than good over time unless paired with a very sophisticated set of mechanisms to surface anomalies, validate human insights, detect real traffic being impacted by over-broad rules, and apply automated safeguards.

This is part of the hCaptcha Enterprise APT Mitigation supplemental SOC review process, and integrated within our Rules and Private Learning features for self-serve behavior adjustment.

Implementing Shadow Bans based on Scores

In practice, there are only three scenarios to consider:

- Score is low (no special action required; score < 0.7)
- Score is suspicious but not certain (Elevate risk response, do not block; score 0.7-0.79)
- Score is high (Shadow Ban; safe to shadow-ban outright without revealing to user)

We offer extensive guidance in the *Scores and Modes* documentation along with many practical examples, and our integration specialists are available to consult on strategies at any time: just open a support ticket.

Why is responding to suspicious requests in real-time useful?

It is very valuable to include a concept of "elevated risk" in your application that lets you respond to suspicion in real-time, i.e. hCaptcha risk scores in the 0.7-0.79 range.

All risk calculations have a continuum of certainty, and knowing risk is elevated but not certain is very useful information: it lets you immediately take an action to further increase or decrease confidence in the session, likely stopping a bad actor without harming real users.

Examples of this can be: requiring a security question, triggering an email 2FA link to complete login, requiring a MFA code, etc. depending on your application.

Rather than blocking users outright when you have high suspicion but not certainty, you can instead respond to this behavior in such a way that you still drop almost all bad actors without adding much friction for real people, e.g. those sharing a CGNAT IP with someone else who happens to have malware on their PC.