Technical Review

Google reCAPTCHA Enterprise: Frictionless, Flexible, and Effective Web App Security

Date: August, 2020 **Author:** Jack Poller, Senior Analyst

Abstract

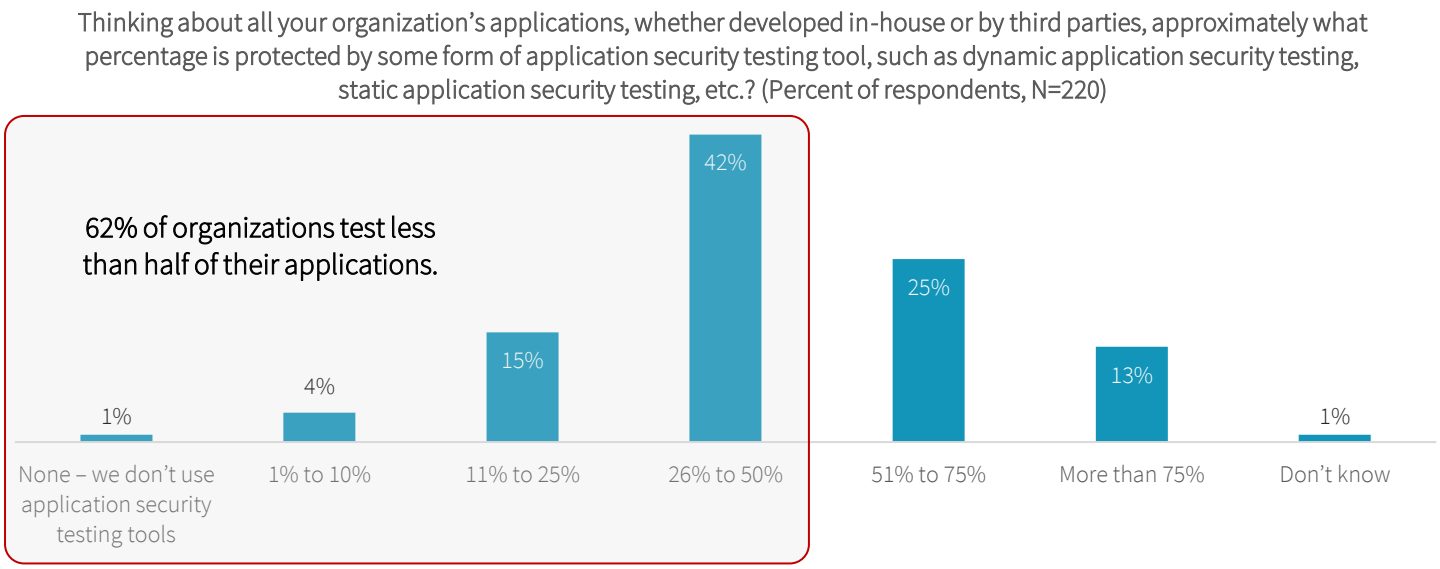
This ESG Technical Review documents ESG's evaluation and analysis of how developers can easily integrate Google reCAPTCHA Enterprise into public-facing web applications. We also evaluate how effectively reCAPTCHA can defend web applications from common automated threats and attacks while never interrupting users with a challenge.

The Challenges

The potential for large financial gains arguably makes public-facing web apps the primary target of cybercriminals. Over the last two years, the volume of attacks has dramatically increased. Akamai alone recorded more than 30 billion credential stuffing attacks against websites protected by its services in 2018, with attacks growing to 58 billion in 2019.[1,2] In the third quarter of 2019, Arkose Labs reported a 30% increase in website fraud and abuse attacks, comprising primarily fraudulent new account registrations (19%), fraudulent payment attacks (16%), and account takeovers (12%). Arkose measured a 70% increase in bot-driven attacks for account registration, with more than 50% coming from mobile platforms.[3]

As organizations have experienced this continual increase in threats to their public-facing web apps, and application security best practices have become well understood and well documented, developers, marketers, and security teams are incorporating cybersecurity principles and tools into DevOps pipelines and methodologies. While usage of application security (AppSec) tools has increased steadily in the past 5 years, 62% of organizations still only leverage these tools on less than half of their apps (see Figure 1).[4]

Figure 1. Use of Application Security Testing Tools



Source: Enterprise Strategy Group

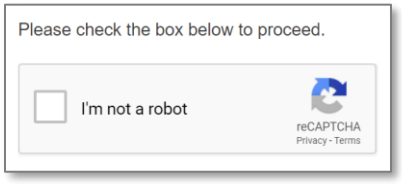
[1] Source: Akamai: *State of Internet/Security, Vol. 5, 2019*
[2] Source: Akamai: *Credential Stuffing in the Media Industry*, July 2020.
[3] Source: Arkose Labs: *Fraud and Abuse Report Q4 2019*.
[4] Source: ESG Master Survey Results, *Application and Email Security Trends*, September 2019.

Despite the prevalence of application security testing, public-facing web applications are still vulnerable to a unique set of automated threats that security testing tools struggle to identify. The Open Web Application Security Project (OWASP) has developed a list of the top 21 threats, categorized as threats targeting account credentials, payment cardholder data, vulnerability identification, and availability of inventory. Countermeasures to these threats can:

- **Detect**—distinguish between humans and automated processes and identify automated attacks.
- **Prevent**—reduce the susceptibility to automated threats.
- **Recover**—respond to automated attacks to return to normal operations.

reCAPTCHA Enterprise

Google designed reCAPTCHA Enterprise to help protect public-facing web applications from fraudulent activity, spam, and abuse. The fraud detection service uses advanced risk analysis strategies developed using Google's more than decade-long history of defending hyperscale web applications and the combined experience and knowledge gleaned from protecting over four million sites. These strategies differentiate between humans and bots and provide the application developer with a granular risk score, enabling the development and security teams to make well informed countermeasure decisions based on their level of acceptable risk.



reCAPTCHA Enterprise is easy to deploy, requiring just a few lines of code, has a lightweight footprint, and will never interrupt the user with a visual challenge. Thus, Google recommends installing reCAPTCHA Enterprise on every web page at the point of action (e.g., account creation, login verification, purchase, etc.). This enables developers to protect their entire application from the types of attacks identified in the OWASP Automated Threat Handbook, including account credential attacks, payment cardholder attacks, availability of inventory attacks, and more.

reCAPTCHA Enterprise Features include:

- **Adaptive risk analysis**—evaluates a multitude of parameters, sensors, and signals to differentiate between humans and bots.
- **Score-based detection**—enables developers to deploy data-driven countermeasures to stop bots and automated attacks while maintaining normal operations for valid users.
- **Granular risk scores**—with reason codes for high risk scores that enable developers to make informed decisions based on their level of acceptable risk for a given activity.
- **Frictionless deployment**—requires just a few lines of code and can be deployed on every page.
- **Frictionless user interface**—automated risk analysis will never present a challenge to the user.
- **Flexible API**—integrate reCAPTCHA Enterprise on websites and mobile applications.
- **Enterprise scale**—Google websites are protected by reCAPTCHA Enterprise, and the service can scale from the smallest to the largest applications.
- **Tunable risk analysis**—ensures the risk analysis matches the site's needs.
- **Mobile app support**—full support for mobile apps with APIs for Android and iOS.
- **Machine learning**—leverages Google's AI resources to recognize the patterns of legitimate and fraudulent transactions.

ESG Tested

ESG evaluated the ease of deployment and operation of reCAPTCHA Enterprise using a demo environment and a real-world application. We also audited the effectiveness of reCAPTCHA Enterprise protecting a real-world hyperscale web application.

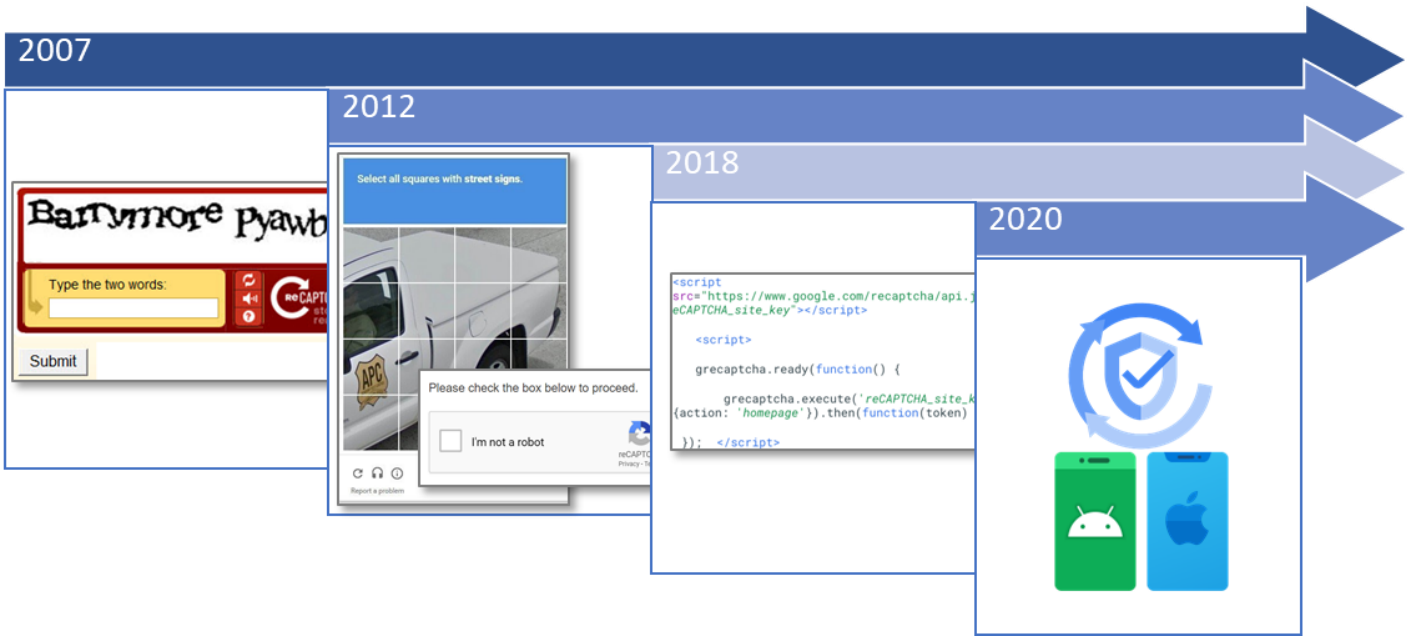
Frictionless and Flexible

reCAPTCHA Enterprise evolved from the CAPTCHA project (Completely Automated Public Turing test to tell Computers and Humans Apart), which used a visual challenge-response test to distinguish between humans and bots. Visual CAPTCHAs have lost utility over time, the result of:

- Friction—visual challenges can be hard to solve, annoying users and reducing user interaction and conversion rates.
- Accessibility—accessible alternatives may still be hard for those with visual and auditory difficulties.
- Automated bot workarounds—malicious actors have leveraged advances in computer vision and machine intelligence to solve challenges.
- Human workarounds—malicious actors may simply pay humans to solve challenges.

As shown in Figure 2, reCAPTCHA Enterprise has evolved from visual challenges to an automated solution. reCAPTCHA Enterprise removes the friction by replacing the visual challenge with a risk score, enabling developers to disambiguate between humans and bots and to protect web apps from automated threats. The latest release of reCAPTCHA Enterprise includes Android and iOS APIs, enabling developers to apply the same frictionless protection to mobile apps.

Figure 2. The Evolution of reCAPTCHA



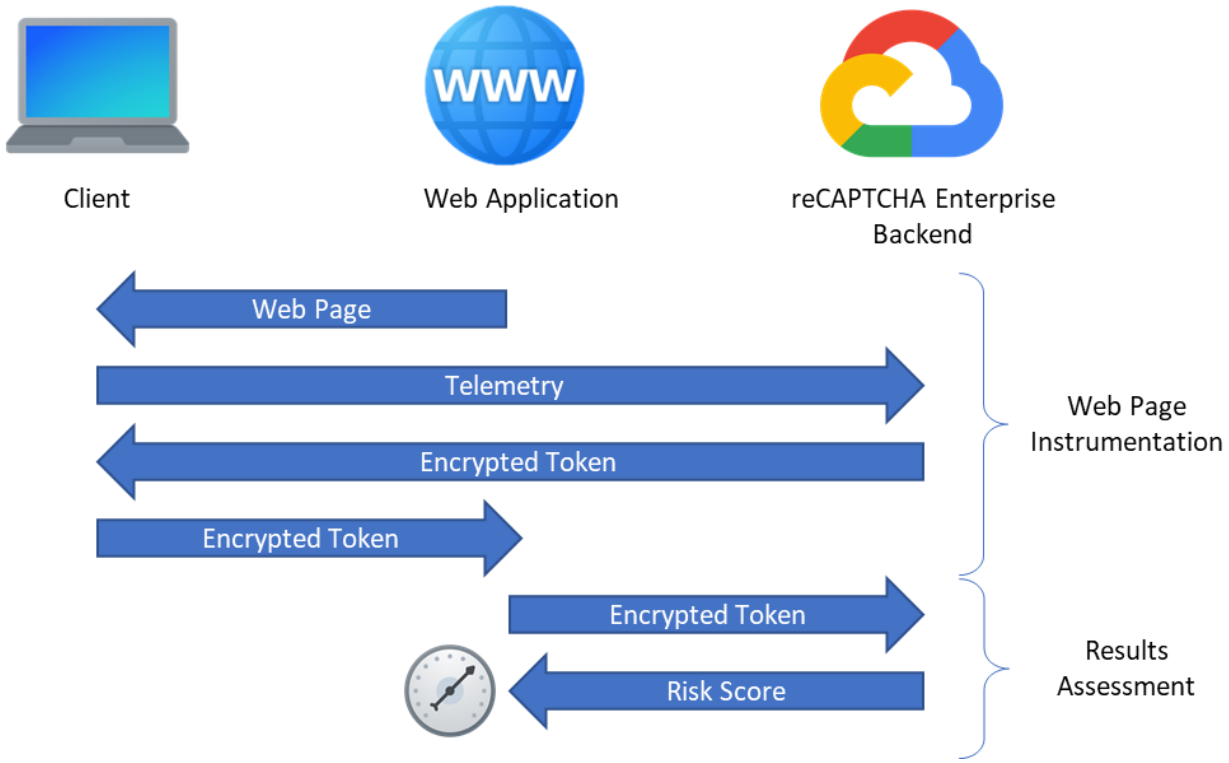
Source: Enterprise Strategy Group

The reCAPTCHA backend servers do the majority of the work, analyzing web client telemetry and assessing risk. In the rare event of a failure of the reCAPTCHA system, the web app continues to run, albeit with less protection. In comparison, a simple connectivity failure can bring down entire websites when using some alternative methods such as reverse proxies, web application firewalls, or content distribution networks.

reCAPTCHA Enterprise operation is simple, and the logic flow to protect a web page is shown in Figure 3.

1. Client loads web page with reCAPTCHA Javascript.
2. On page load or user action, client sends telemetry to reCAPTCHA Enterprise backend for analysis.
3. reCAPTCHA engine sends an encrypted token for the transaction to the client.
4. Client sends encrypted token to the web app.
5. Web app sends encrypted token to the reCAPTCHA backend.
6. reCAPTCHA returns a risk score to the web app.
7. Web app takes appropriate action based on the organization's acceptable level of risk.

Figure 3. reCAPTCHA Logic Flow



Source: Enterprise Strategy Group

Developers and security teams can easily instrument a web page by adding just a few lines of code. After registering with Google, developers instrument each web page by including Javascript that is executed either on page load or when the user acts (see Figure 4).

Because the script is lightweight and will never interrupt users, and Google's analysis engine works best when it has the most context about both legitimate and malicious interactions, Google recommends including reCAPTCHA on forms or actions as well as in the background of all pages. This increases the fidelity of risk scores and provides more comprehensive analytics.

Figure 4. reCAPTCHA Client Code Simplicity

```
<script
src="https://www.google.com/recaptcha/enterprise.js?render=reCAPTCHA_site_key">
</script>
<script>
  grecaptcha.enterprise.ready(function() {
      grecaptcha.enterprise.execute('reCAPTCHA_site_key',
        {action: 'homepage'}).then(function(token) {
          ...
      });
  });
  </script>
```

Source: Enterprise Strategy Group

Analyzing the results in the web app is just as simple as instrumenting the web page. The web app queries the reCAPTCHA Enterprise analysis engine backend servers for the assessment results and the servers return the risk score and the reason code (see Figure 5).

Figure 5. reCAPTCHA Web App Code Simplicity

Request Assessment Results

```
{
  "event": {
    "token": "token",
    "siteKey": "key"
  }
}
POST https://recaptchaenterprise.googleapis.com/v1/projects/project-id/assessments
```

Assessment Results

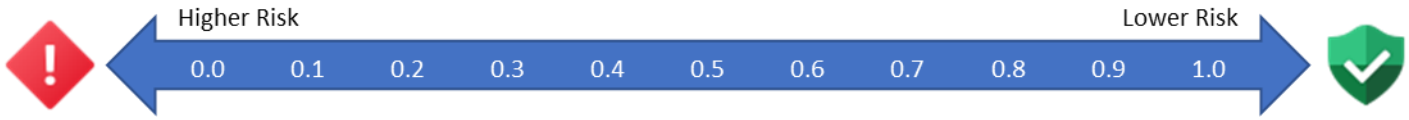
```
{
    "riskAnalysis": {
    "score": 0.1,
    "reasons": ["AUTOMATION"]
  },
}
```

Source: Enterprise Strategy Group

reCAPTCHA Enterprise provides the flexibility to protect as little as one web page or the entirety of a hyperscale website. The solution provides additional flexibility through granular risk scores and reason codes for high risk clients (see Figure 6). Rather than providing a binary result (safe/not safe, or human/bot), reCAPTCHA returns a risk score from 0.0 to 1.0, with 0 indicating a higher risk and 1 representing lower risk. Higher risk scores may include a reason code, such as AUTOMATION, indicating that the analysis concludes the source is an automated bot, and not a human.

Application developers, marketers, and cybersecurity analysts can make the appropriate decision based on their level of acceptable risk for a given action. For example, a developer may allow continued access to all web pages where the risk score > 0.2. By eliminating the higher risk accesses, the developer can prevent automated website scraping. In the same application, the developer may choose to allow payment actions only when the risk score falls in the lower risk range of 0.8-1.0. This can protect the web app from automated payment attacks.

Figure 6. Granular Risk Scores and Reason Codes

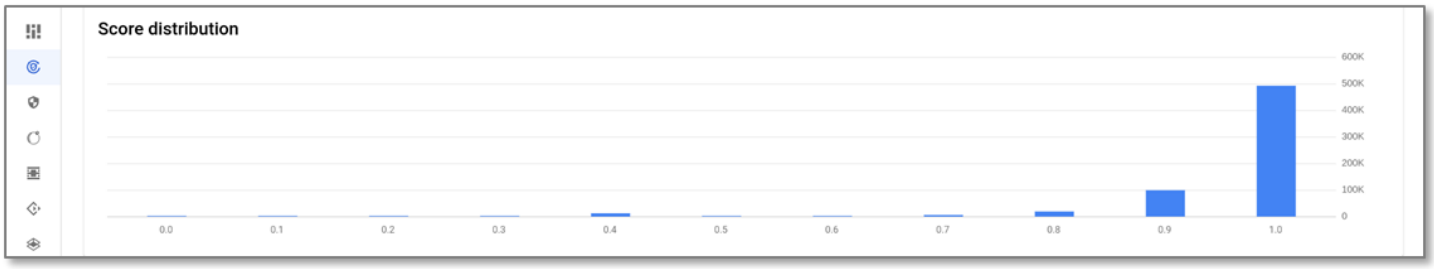


REASON CODE	DESCRIPTION
AUTOMATION	Matches the behavior of an automated agent.
UNEXPECTED_ENVIRONMENT	The reCAPTCHA snippet is being interacted with on a page other than its intended location on your site.
UNEXPECTED_USAGE_PATTERNS	Significantly different than expected patterns.
TOO_MUCH_TRAFFIC	Traffic volume from the source is higher than normal.
LOW_CONFIDENCE_SCORE	Not enough data has been received, unable to generate quality risk analysis.

Source: Enterprise Strategy Group

Google reCAPTCHA analytics provides the ability for users to view the distribution of risk scores, as shown in Figure 7. Organizations can use this information to tune their reCAPTCHA deployment and risk score tolerance.

Figure 7. reCAPTCHA Risk Score Distribution



Source: Enterprise Strategy Group

Why This Matters

Friction matters for web apps. Interrupting the user with hard-to-solve challenge-response images increases bounce rates. Complicated code running on the client to detect bots increases page size and load times, further decreasing user engagement.

Friction also matters for web developers. When balancing between app security and ease of implementation, developers often favor simplicity to speed time to market, relegating security measures to the next release, which often means never.

ESG validated that reCAPTCHA Enterprise removes the friction associated with protecting web apps from automated attacks against user accounts, payment systems, inventory systems, and more. Developers can easily add reCAPTCHA to any web page with a few lines of code, and reCAPTCHA will never interrupt the user with an annoying, hard-to-solve challenge-response.

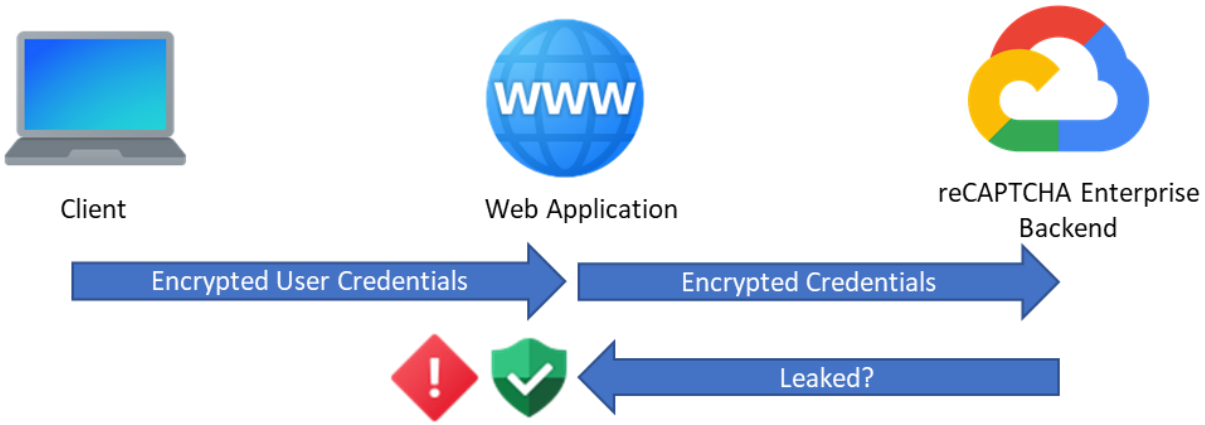
ESG found risk assessment logic is just as easy to implement in the web app, removing any implementation excuse. Further, because reCAPTCHA Enterprise is lightweight and provides granular risk scores and reasons, developers have the flexibility to add protection to any and all web pages. They can also adjust responses based on the organization's level of acceptable risk.

Effective

ESG evaluated the effectiveness of reCAPTCHA Enterprise as deployed in a real-world hyperscale website to protect against automated credential stuffing and account takeover attacks.

During account creation and password resets, the website uses reCAPTCHA to compare the user's password against Google's database of over four billion leaked passwords. As shown in Figure 8, the web app sends the user's encrypted credentials to the reCAPTCHA backend. If the credentials exist in Google's database, reCAPTCHA returns a response of "true," indicating that the user is attempting to reuse credentials that have been compromised on another site. The website rejects the user's password and forces the user to choose a different password.

Figure 8. Using reCAPTCHA Enterprise to Detect Previously Compromised Credentials

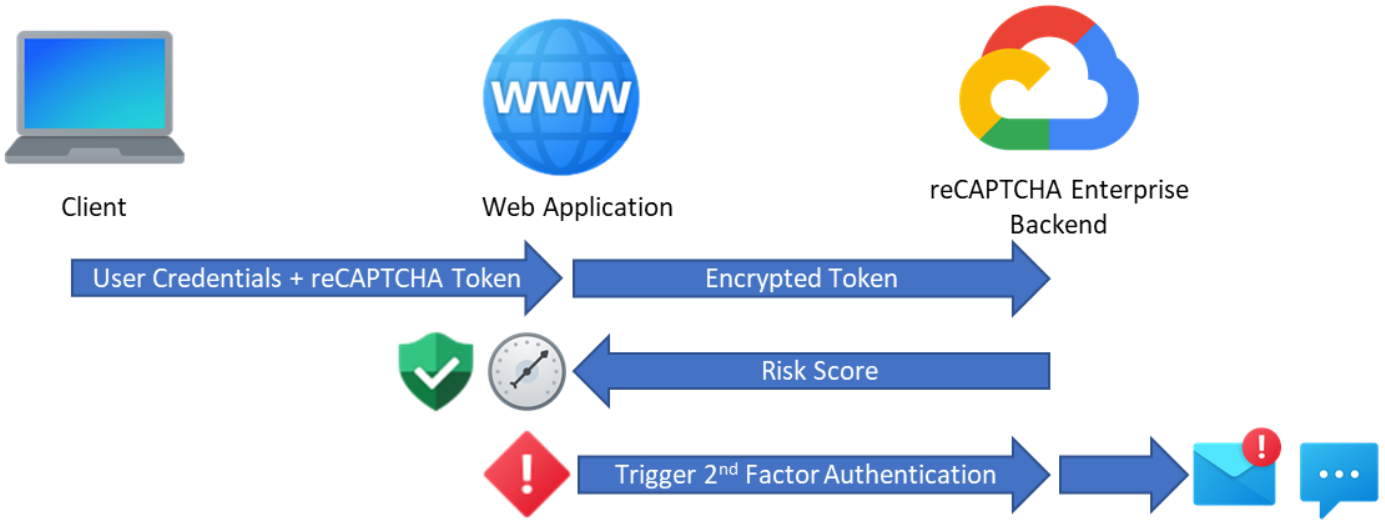


Source: Enterprise Strategy Group

The website also uses reCAPTCHA Enterprise to remove friction during the login process. After obtaining the user's credentials, the web app obtains the risk score from the reCAPTCHA analysis engine. If the risk score is > 0.5, the web app continues the login process. If the risk score is < 0.5, indicating a higher risk interaction, the web app requests a second factor authentication (2FA), sending a one-time password (OTP) via email or text. This logic removes the hassle and interruption of challenge-response bot detection and two-factor authentication for most users while increasing security for potentially high-risk transactions.

The COVID-19 pandemic has impacted cybercriminals and changed attack patterns in the first half of 2020, with a marked increase in hybrid human/bot attacks, and more than one-fifth (21%) of attacks use a mobile platform.[5] Marketers, security teams, and developers can use reCAPTCHA's mobile APIs to protect their websites and mobile apps with the same logic. This enables the organization to decrease mobile friction for low-risk transactions while increasing security with 2FA for high-risk transactions.

Figure 9. Using reCAPTCHA Enterprise to Protect from Account Takeovers

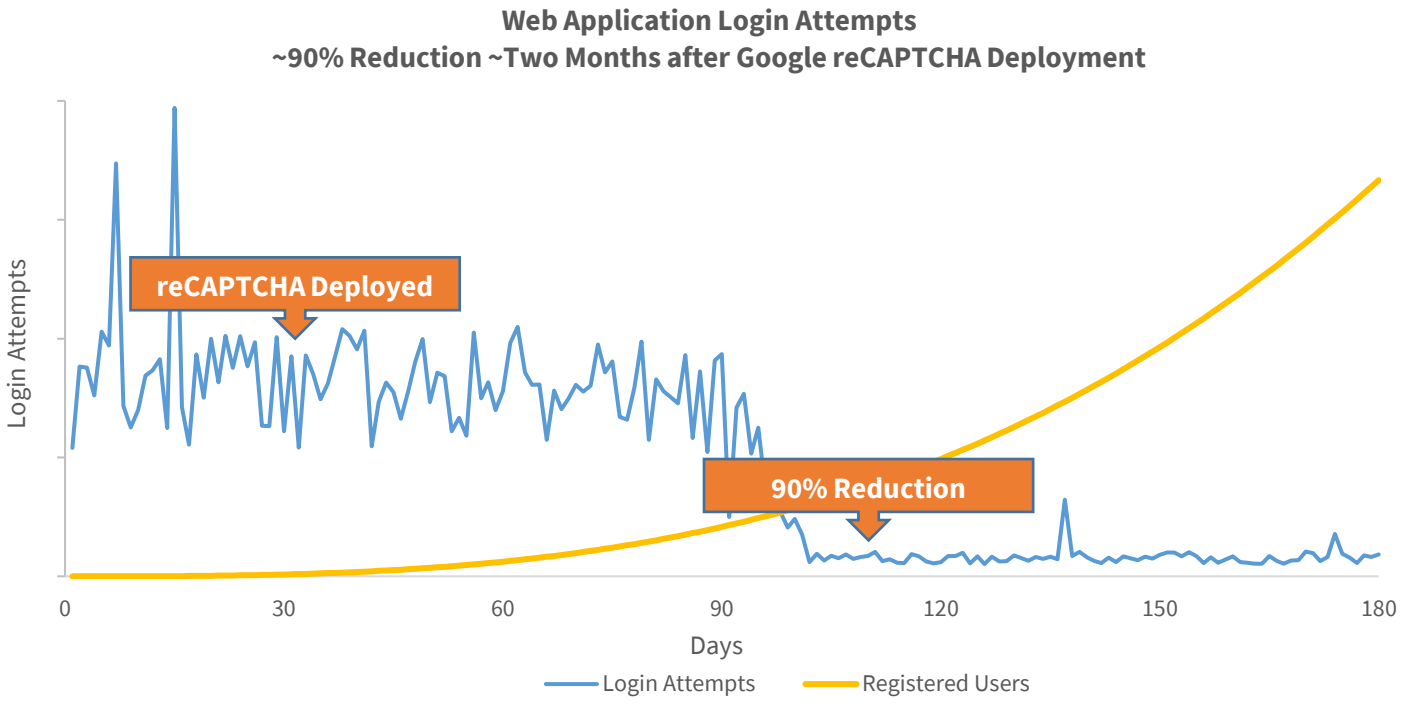


Source: Enterprise Strategy Group

[5] Source: Arkose Labs: *Fraud and Abuse Report Q4 2019*.

Approximately two months after reCAPTCHA Enterprise deployment, login attempts were reduced by approximately 90% while the registered user base continued to grow (see Figure 10).[6]

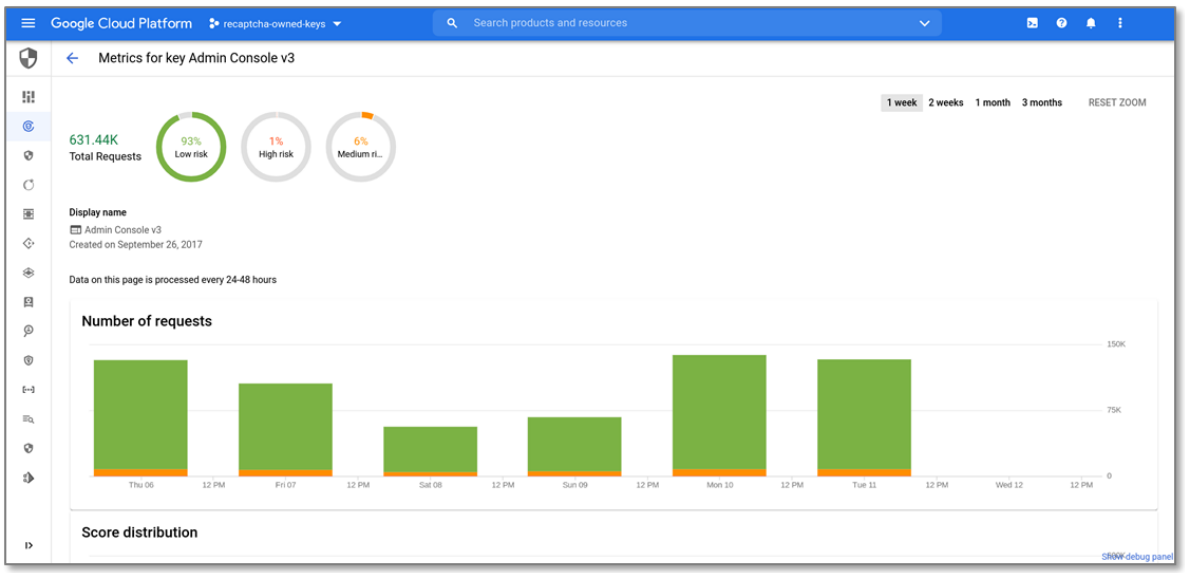
Figure 10. reCAPTCHA Enterprise Deployment Results



Source: Enterprise Strategy Group

The reCAPTCHA Enterprise console provides analytics, as shown in Figure 11. Developers and operations can use the analytics to understand reCAPTCHA performance. The analytics console displays risk assessment results grouped by category (malicious, suspicious, and legitimate), and users can analyze data for a variety of timeframes.

Figure 11. reCAPTCHA Enterprise Analytics



Source: Enterprise Strategy Group

[6] The values displayed in this graph are simulated but closely resemble the actual results evaluated by ESG.

Developers and operations can tune reCAPTCHA Enterprise by providing feedback to the analysis engine, marking interactions as either legitimate or fraudulent.

Why This Matters

Identities are the new digital currency. Once a cybercriminal has access to a valid identity, they can use stored payment cards to make fraudulent purchases, reset passwords, and take over additional accounts, providing access to even more stored payment data and accounts. While individual actors can manually attack a single website, the sophisticated cybercriminal can take advantage of automated bot attacks to rapidly collect large portfolios of accounts and payment information, which can be resold for a quick profit, generating massive ROI.

About two months after reCAPTCHA Enterprise was deployed to protect a hyperscale website, the number of login attempts was reduced by approximately 90% while the registered user base continued to grow. This demonstrates that reCAPTCHA Enterprise is successfully preventing automated account takeover attempts without discouraging or preventing legitimate user engagement.

The Bigger Truth

Motivated by potentially large financial gains, sophisticated actors are using automation to exploit inherent functionality and business logic flaws of public-facing web apps to take over accounts, abuse payment and inventory systems, and identify other vulnerabilities. Unlike software bugs and vulnerabilities, misuse and abuse of valid functionality can be difficult, if not impossible, to detect and prevent through code analysis or other traditional application security techniques.

Traditionally, marketing, web development, and security teams have included a visual challenge to distinguish between a legitimate human user and an illegitimate automated threat. Visual challenges introduce app complexity and are annoying and hard for humans to solve, increasing bounce rates and decreasing user engagement. Simultaneously, technological advances have made defeating CAPTCHAs much easier, removing much of the value of visual challenge-response tools.

ESG validated that reCAPTCHA Enterprise was effective in protecting a real-world hyperscale website from automated attacks, reducing login attempts by approximately 90% without reducing user engagement. As reCAPTCHA Enterprise works best when it has the most context about interactions and each organization has a different level of risk tolerance, your performance may differ.

We found that reCAPTCHA Enterprise removes the user friction associated with protecting web apps from automated attacks, and will never interrupt the user with an annoying, hard-to-solve visual challenge-response. Web developers and security teams can easily add reCAPTCHA to any web page with a few lines of code, and can easily assess risk with simple logic in the web app. reCAPTCHA Enterprise's granular risk scores provide the flexibility to add protection to any and all web pages, and organizations can adjust responses based on their level of acceptable risk.

If you are looking for flexible, frictionless, and effective protection from automated attacks against your public-facing web apps, ESG recommends that you consider the speed and simplicity of deploying reCAPTCHA Enterprise.