CEQUENCE

Whitepaper

Cequence Bot Management

Advanced Machine Learning-Based Bot Detection and Mitigation

Contents

As brick-and-mortar business models embrace the cloud and web applications like ecommerce, bad actors are increasingly using sophisticated bot attacks to commit fraud. Bot-based business logic abuse, or bot attacks, often start with basic reconnaissance, escalating based on signs of success or retooling when faced with resistance. Bot scenarios can lead organizations down the path of infrastructure overspend or slow performance for legitimate users if not addressed quickly. In the face of this evolving threat, organizations need to implement advanced bot mitigation solutions to protect their crown jewels – the business logic behind their public-facing applications.

Traditional Bot Management Techniques No Longer Work

Web Application Firewalls

When first faced with fraud resulting from bot attacks, enterprises often turn to existing tools such as web application firewalls (WAFs). However, since WAFs prevent attacks by blocking specific IP addresses, attackers can easily bypass them by spreading their attacks across millions of IP addresses available through bulletproof proxy vendors. The next attempt to prevent the attack is typically geo-fencing, which also can be easily bypassed using residential IP addresses to blend in with customer traffic. Additionally, WAF signatures are often used to look for suspicious HTTP headers, but attackers have evolved to respond with the necessary values to successfully mimic real browsers.

Client Telemetry-Based Bot Mitigation

While traditional bot mitigation tools provide an initial reprieve from bot attacks, they do not provide the sustained efficacy that enterprises need to fight back against bots that can dynamically change their behavior. To differentiate legitimate requests from bot traffic, traditional bot mitigation products take an approach of analyzing and collecting client data collected from end-user devices. They use different techniques to collect this data, depending on whether the application is web, mobile, or API-based.

For web applications, such vendors provide JavaScript to embed within HTML pages, which, when delivered to web browsers, collects and reports telemetry data. For mobile applications, a mobile SDK is provided for developers to integrate into their apps. Then, when the apps are used, the SDK collects and transmits telemetry data back to the bot mitigation tool. For API-based applications, there is no way to add an agent, thereby limiting telemetry collection, creating a policy inconsistency and a protection gap between an organization's web/mobile apps and APIs. They also suffer from several additional disadvantages, highlighted below.

- **Impact on application load times and user experience.** JavaScript integrated into a webpage, such as CAPTCHAs, can degrade webpage load times by as much as two seconds, not to mention the time wasted by the end user solving the CAPTCHA. SDKs collecting telemetry data can also impact application load time, response time, and battery life.
- **Slowing down application development.** Any external integration in the form of JavaScript or a mobile SDK becomes a heavy lift for teams coordinating app development, testing, and certification. With these two approaches, enterprises can spend months of effort to protect a handful of applications from bot attacks. For a mobile SDK to be effective, it needs to be pushed to all legitimate users. To ensure protection is maintained, businesses must upgrade their entire user population every time a new version of their mobile app is released, placing a significant burden on end-users who may or may not accept the upgrade.
- **Inability to protect direct API traffic.** Since direct API clients cannot generate client-side telemetry signals, such solutions must resort to simple statistical techniques like rate limiting, or worse, whitelisting IP address ranges, to avoid blocking legitimate API traffic.
- **Inability to handle legitimate bots.** Like direct API clients, legitimate bots can't integrate an agent to collect client-side telemetry. So, traditional tools must resort to whitelisting IP address ranges of popular search engine vendors, to avoid blocking crawlers and other "good bots".
- **Lack of sustained efficacy.** Due to the presence of client-side code, it's easy for cybercriminals to analyze JavaScript and mobile SDKs from different vendors and change their attack behavior to evade them. To counter this behavior, enterprises must deploy updated JavaScript and SDKs to all the applications requiring protection.

Powered by Machine Learning-Based Fingerprinting

Cequence Security has taken a very different approach to bot mitigation and uses multi-dimensional machine learning (ML) techniques to analyze user behavior without requiring any client-side or application integration. Cequence analyzes behavioral intent consistently across web, mobile, and API traffic, detecting legitimate bots based on their behavior, not just based on their IP addresses. Conversely, attack traffic is identified by its behavior as well, which is continuously determined and learned by the ML models. Cequence leverages an approach that actively "fingerprints" incoming requests based on the similarity of their behavioral traits, as analyzed by the ML models.

Cequence Bot Management is deployed with over 150 customizable rules that encode common behavioral traits of automated attacks, resulting in immediate mitigation efficacy upon deployment. Additional ML models provide time-based analysis that can identify statistical anomalies of request rates or anomalies in usage over time. These increase the effectiveness of attack detection. ML models can be further tuned or customized based on specific application use cases. Additionally, the open platform allows customers to import security intelligence as well as export it into third-party systems like SIEMs, anti-fraud tools, firewalls, and IDS/IPS systems.

Cequence also provides managed threat protection services, backed by leading data scientists and cybersecurity experts working in close partnership with the customer. The services include updates and tuning of ML models, policy updates, and the creation of advanced application-specific rules. It allows enterprises to focus on their applications and business logic, leaving industry experts to protect their applications without any required changes or need for extra development.

Four Pillars of Threat Detection

To differentiate malicious from good intent, the Cequence approach analyzes each application request based on the four components of an automated attack – tools, infrastructure, credentials, and behavior.

Tools

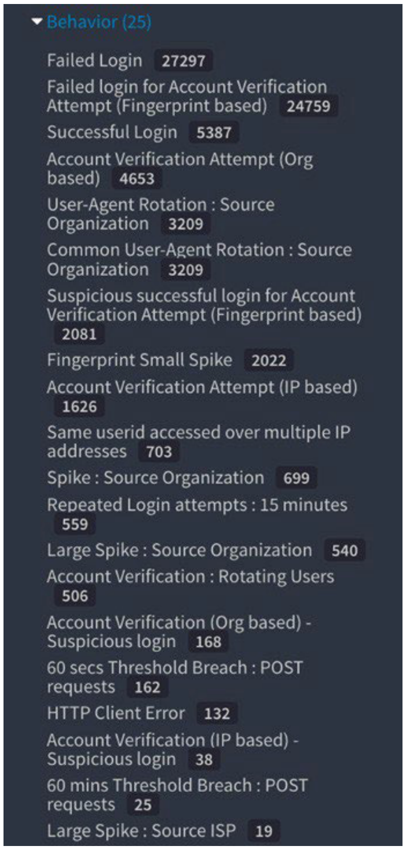
Tools represent the most basic component of an attack. This detection pillar focuses on identifying heuristics that deal with the immutable characteristics of the code launching the attack or, increasingly, the attributes of an off-the-shelf tool that are difficult to change for novice bad actors. Popular examples include SNIPR, Sentry MBA, BlackBullet, and OpenBullet.

Infrastructure

Infrastructure represents the resources that bad actors need to distribute their attacks and anonymize themselves. This includes detecting the top offending organizations and networks used for abuse across our customer base and identifying increasing usage of high reputation residential IP proxies (RESIPs) through services like Luminati, Storm Proxies, and others. Additional examples of infrastructure are "proxy-as-a-service" businesses that tacitly allow and encourage fraud through their networks and compromised IoT devices that serve as semi-public open proxies to launch attacks.

Credentials

Attackers need user accounts – either legitimate and compromised or fake– to carry out attacks. The credentials detection pillar focuses on how credentials are used in automated attacks. This helps identify techniques and patterns used across requests that could indicate credential abuse or iteration using stolen credentials from well-known data breaches. It also helps identify large-scale manual fake account creation to uncover patterns that may expose bad actors' intent.



Failed Login	27297
Failed login for Account Verification Attempt (Fingerprint based)	24759
Successful Login	5387
Account Verification Attempt (Org based)	4653
User-Agent Rotation : Source Organization	3209
Common User-Agent Rotation : Source Organization	3209
Suspicious successful login for Account Verification Attempt (Fingerprint based)	2081
Fingerprint Small Spike	2022
Account Verification Attempt (IP based)	1626
Same userid accessed over multiple IP addresses	703
Spike : Source Organization	699
Repeated Login attempts : 15 minutes	559
Large Spike : Source Organization	540
Account Verification : Rotating Users	506
Account Verification (Org based) - Suspicious login	168
60 secs Threshold Breach : POST requests	162
HTTP Client Error	132
Account Verification (IP based) - Suspicious login	38
60 mins Threshold Breach : POST requests	25
Large Spike : Source ISP	19

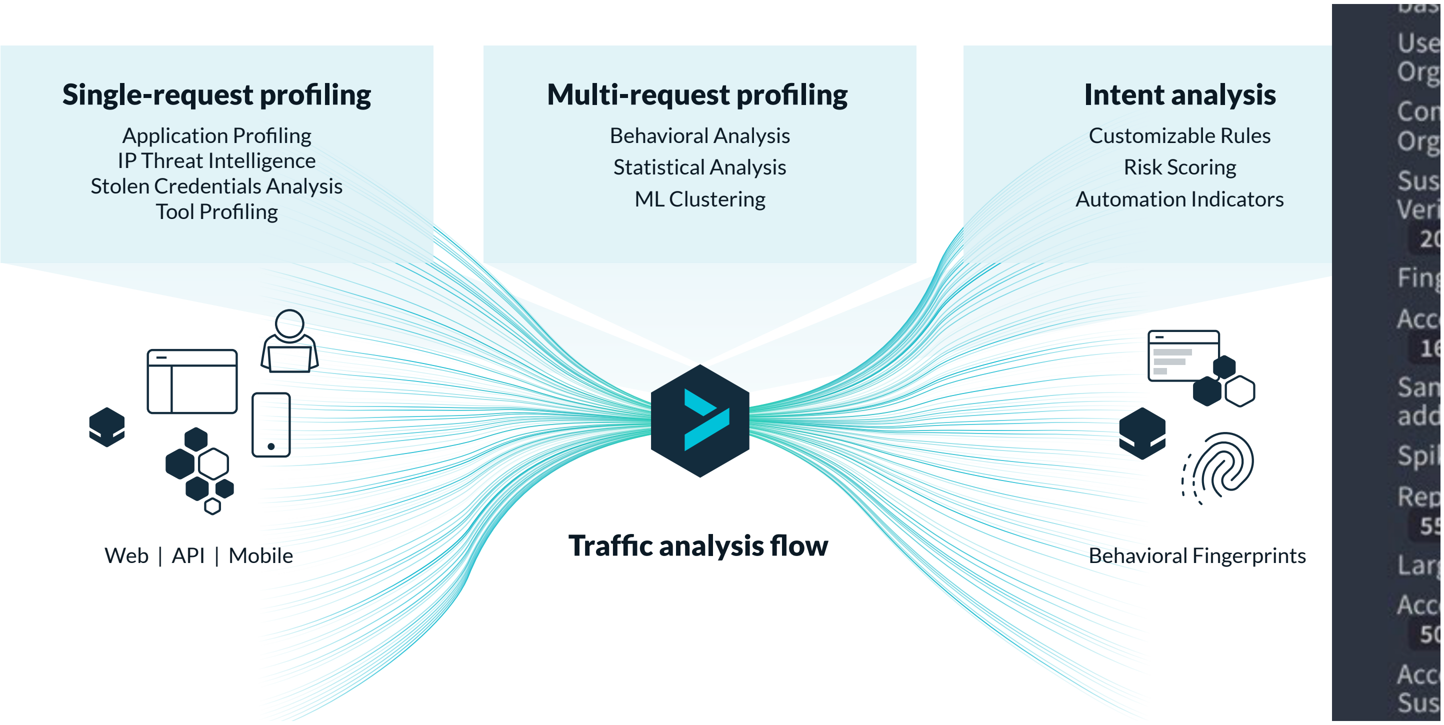
Example of rules triggering for a request highlighting behavioral heuristics

Behavior

Behavior represents the unique characteristics, or behavioral fingerprint, a bad actor creates when using tools, infrastructure, and credentials to launch an attack. This pillar deals with the human element of automated bot attacks and helps identify both "low and slow" and "fast and furious" attacks. It also examines a bad actor's preferred tactics as they try to evade detection and sustain the offensive.

Multi-Dimensional ML Analysis

As shown in the image below, ML-based analysis is applied to the four pillars of an attack in three phases:



Phase One: Single-Request Profiling

Each incoming request is analyzed in context, using supervised ML models to automatically discover and optimize 18 HTTP transaction attributes (header positions, values, etc.), that are then grouped by their browser class (e.g., Chrome, iOS, Safari, headless browsers, Selenium, etc.). The resulting attribute subsets are used to compute "fingerprints" – unique digital hashes for requests with similar characteristics. Using the same fingerprint value, Cequence groups the requests generated from different application clients (e.g., iOS app users).

Unlike traditional bot management solutions, which use fingerprints to identify each client device based on telemetry signals, Cequence instead examines clients' behavior. The following analysis techniques are performed in the context of a single client request.

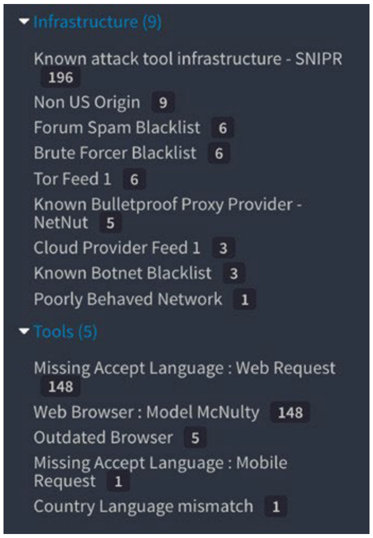
- **Browser, OS, app and device identification:** Cequence uses supervised ML models, continuously trained by Cequence to predict the browser, OS type, and version, application or toolkit used to generate the requests and device type. These predicted values are then compared to the values sent by the client in the request itself (e.g., in User-Agent strings). One of the typical tactics used by bad actors is to pretend to be a known browser while using a botnet script or toolkit to generate requests. Such behavior is immediately detected as the output of this step.
- **Geo-IP reputation analysis:** Cequence checks the originating IP address of the request in threat intelligence feeds, since it's common to find bots using known bad infrastructures or IP address ranges. Bots are also frequently hosted on cloud providers such as AWS or other lower-cost regional cloud providers.
- **Attack tool similarity matching:** Using supervised models, Cequence compares the incoming requests against patterns of known botnet toolkits like SNIPR, Sentry MBA, OpenBullet, and BlackBullet. With strong offline learning provided by labeled data for major known botnet toolkits, the models ensure a high degree of efficacy in identifying requests generated by these sources.

- **Credential reputation analysis:** Cequence subscribes to external threat intelligence feeds that help identify username-password combinations originating from known data breaches. This analysis uses Bloom filters to avoid the security and privacy risks of storing or caching user credentials.

Phase Two: Multi-Request Analysis

Cequence then analyzes the broader spectrum of requests originating from the same IP address or user session. This step is critical in determining malicious behavior, which often transcends the scope of a single transaction and is often seen as a pattern across multiple requests. For multiple requests, the following analysis techniques are used.

- **Anomalous traffic spike detection:** Cequence performs multi-request analysis using Facebook's Prophet model for time-series analysis to detect changes in seasonal traffic patterns, such as anomalous spikes in usage compared to normal daily or weekly activity. This ML-based time-series analysis offers several advantages over non-ML based alternatives, like moving average analysis, which arbitrarily uses thresholds of 15 minutes or 30 minutes to evaluate usage over time without any prior knowledge of the application's seasonal usage pattern.
- **Fingerprint clustering:** Cequence uses unsupervised ML to cluster together fingerprints based on closeness attributes. These are shown as "Clusters" in the management dashboard. A cluster may, therefore, include multiple fingerprints found to exhibit similar characteristics.



Infrastructure (9)	
Known attack tool infrastructure - SNIPR	196
Non US Origin	9
Forum Spam Blacklist	6
Brute Forcer Blacklist	6
Tor Feed 1	6
Known Bulletproof Proxy Provider - NetNut	5
Cloud Provider Feed 1	3
Known Botnet Blacklist	3
Poorly Behaved Network	1
Tools (5)	
Missing Accept Language : Web Request	148
Web Browser : Model McNulty	148
Outdated Browser	5
Missing Accept Language : Mobile Request	1
Country Language mismatch	1

Example of rules triggering that highlight detection of malicious tools and infrastructure

Phase Three: Intent Analysis

The final phase of the analysis applies more than 150 predefined and customer-defined rules against an incoming request. Out-of-the-box rules are defined and updated by Cequence, with codified patterns of typical bot or malicious behavior seen across all Cequence customers. Custom rules can be created and managed by customers to enforce a positive security model so that incoming requests that match those rules are treated as good, and anything else is treated as bad and blocked. Conversely, rules can be used to enforce a negative security model so that incoming requests matching those rules are treated as bad requests, while anything else is treated as good and allowed to pass through. At the end of this phase, the management dashboard displays the characteristics for each incoming request in each of the four detection pillars, as well as an analysis based on the predefined and custom rules.

Real-Time Mitigation

As attacks are detected, Cequence provides real-time mitigation options including blocking, logging, rate limiting, header injection, and deception. These mitigation options are native and do not require third-party solutions such as WAFs or API gateways, making them highly performant and effective. Cequence advanced AI also autonomously creates threat mitigation rules and policies that can be implemented automatically or after human review.

Summary

Cequence Bot Management is the only solution that does not require the use of JavaScript or mobile SDKs. The agentless, ML-based approach delivers broader coverage and higher sustained efficacy against sophisticated bot attacks while eliminating JavaScript and SDK-related penalties such as slow page loads, friction-inducing development delays, and poor user experience from forced mobile upgrades. As organizations migrate to an API-based development methodology, Cequence applies the exact same ML-based analysis to API traffic, ensuring policy consistency across all public-facing applications.

When the solution is combined with the Cequence Managed Threat Protection Service, customers benefit from the collective knowledge that the Cequence data scientists and cybersecurity experts provide. Unlike other offerings, Cequence Bot Management is not a black box. It's open and customizable nature gives customers the ability to access attack data, modify policies, and observe results in real time when leveraging the expertise of the Managed Threat Protection Service.