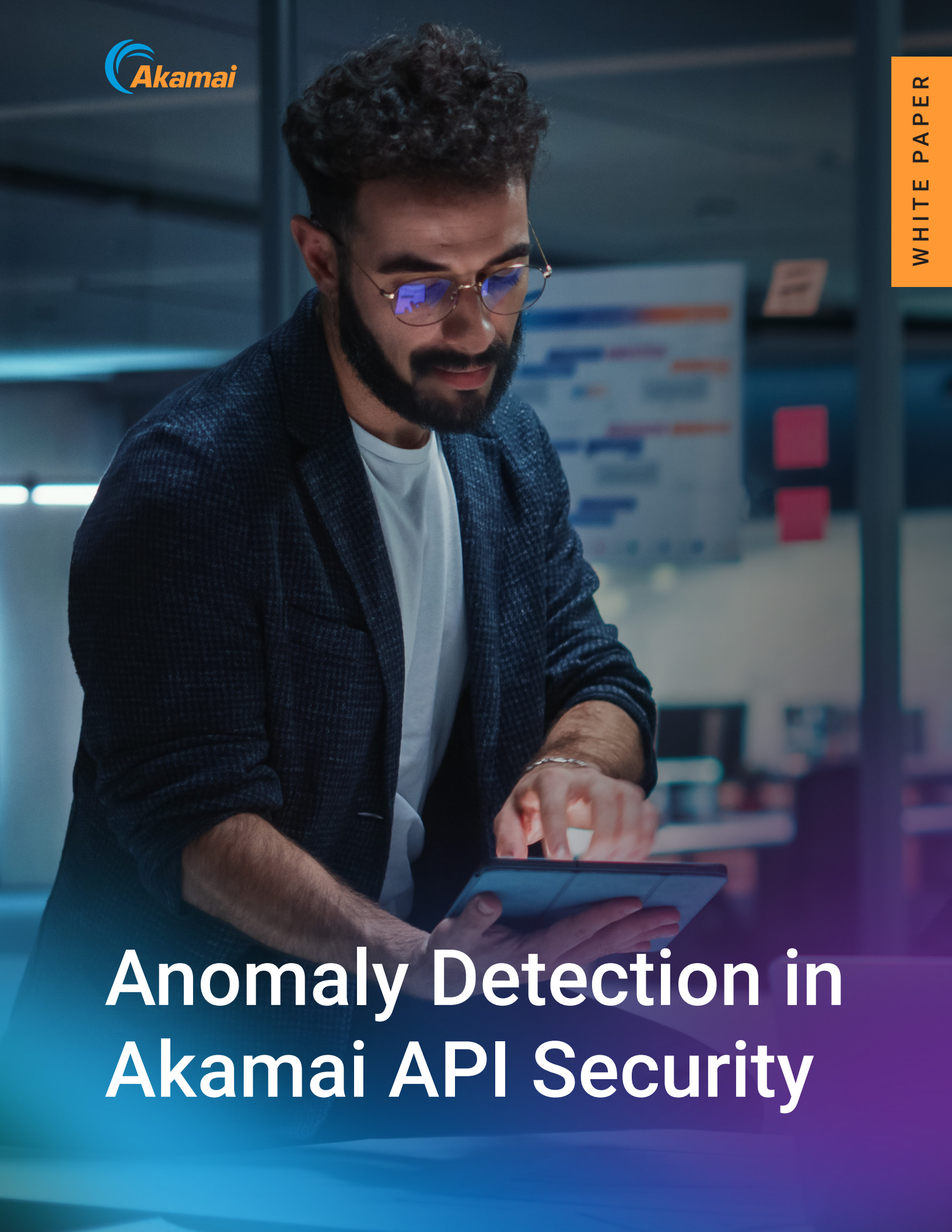


WHITE PAPER

Anomaly Detection in Akamai API Security

APIs are a key component of your organization's ability to serve customers, generate revenue, and operate efficiently. However, their continuous growth, proximity to sensitive data, and lack of security controls make APIs an appealing target for today's attackers. Gaining real-time insight into user behavior is key for proactively identifying the signs of potential API abuse or an attack.

The goal of the Akamai API Security solution's anomaly detection capabilities is to identify anomalous user behavior that indicates potentially malicious attempts to exploit the organization's APIs. By establishing a baseline of normal traffic, Akamai's anomaly detection capabilities can compare incoming requests to the baseline and determine if it is likely to be conducted by an attacker.

Our anomaly detection algorithm identifies anomalous behavior, such as:

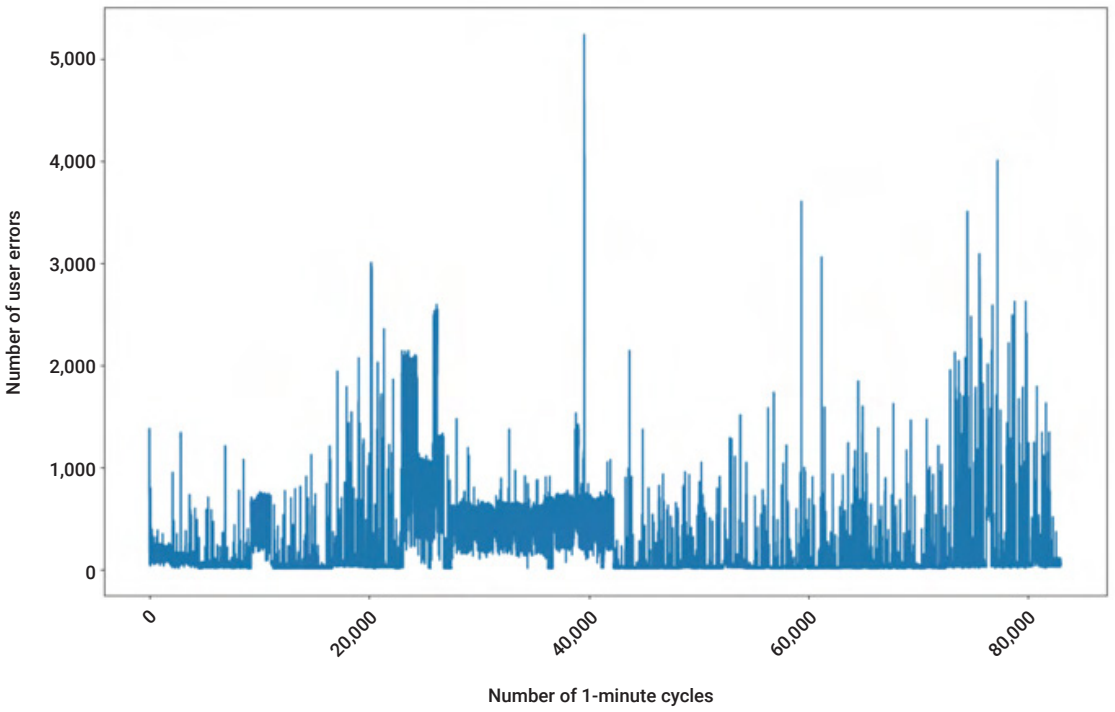
- Using an unexpected field in the API request
- Pulling more data from the server than the regular user
- Trying to use other user/admin resources
- Calling the APIs in an unexpected order

The algorithm is based on an unsupervised online learning artificial intelligence and machine learning (AI/ML) model that learns the multiple features of the statistical behavior of the traffic and detects abnormal incidents after a fixed learning period. Our model is adaptive to changes in the traffic over time and to anomalies labeled as false positives by users.

During the learning phase, our system parses the customer's data and identifies the different APIs, authentication methods, users, data types, etc. As for each API, the model develops a list of features of the regular user traffic, including the number of API hits, the number of errors generated, the percentage of authenticated requests, the amount of data retrieved from the server, and more. Our algorithm detects user abnormalities by comparing the user's and API's characteristics with the results expected by the statistical model that our algorithm has learned.

How Akamai API Security anomaly detection works

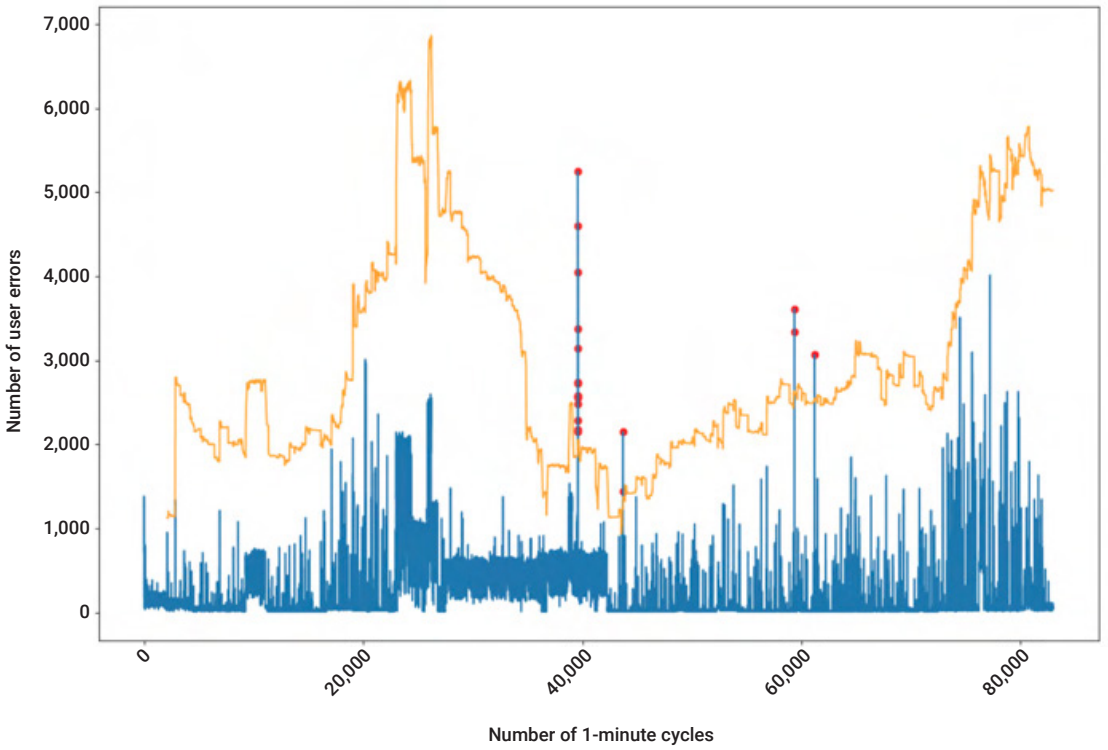
Akamai API Security's anomaly detection capabilities identify users that create excessively more errors than other users. This allows us to identify attacks such as brute force, path scanning, and scraping. The following plot shows the maximum amount of user errors generated by a user every one-minute cycle in an environment.



There are multiple challenges in identifying anomalies in this scenario:

1. The model needs to account for data drift when calculating the threshold.
2. We will want to avoid learning abnormalities during the learning period of the model.
3. The learning is conducted in stream, meaning the model never sees the entire data and needs to adjust every time step.
4. Alerts must be in real time, therefore our algorithm cannot rely on future data to predict an anomaly.
5. To avoid spamming the user, our model needs to learn a statistically guaranteed threshold on the data.

In the plot below, we can see how our model satisfies those requirements by adjusting thresholds according to incoming data.



The orange line shows the threshold function calculated by the model, and the red dots show the anomalies it detected based on that function.

Frequently Asked Questions

What is the necessary learning period for Akamai's anomaly detection algorithm?

Most of our algorithms require a learning period of two to seven days. In addition, the algorithm's learning period is also affected by how many different user behaviors were observed during the learning period.

When an anomalous behavior is detected, how long does it take for the alert to be generated?

Our algorithm will create a relevant alert for the client within 30 to 60 seconds, in most cases, from the moment it receives the anomalous traffic.

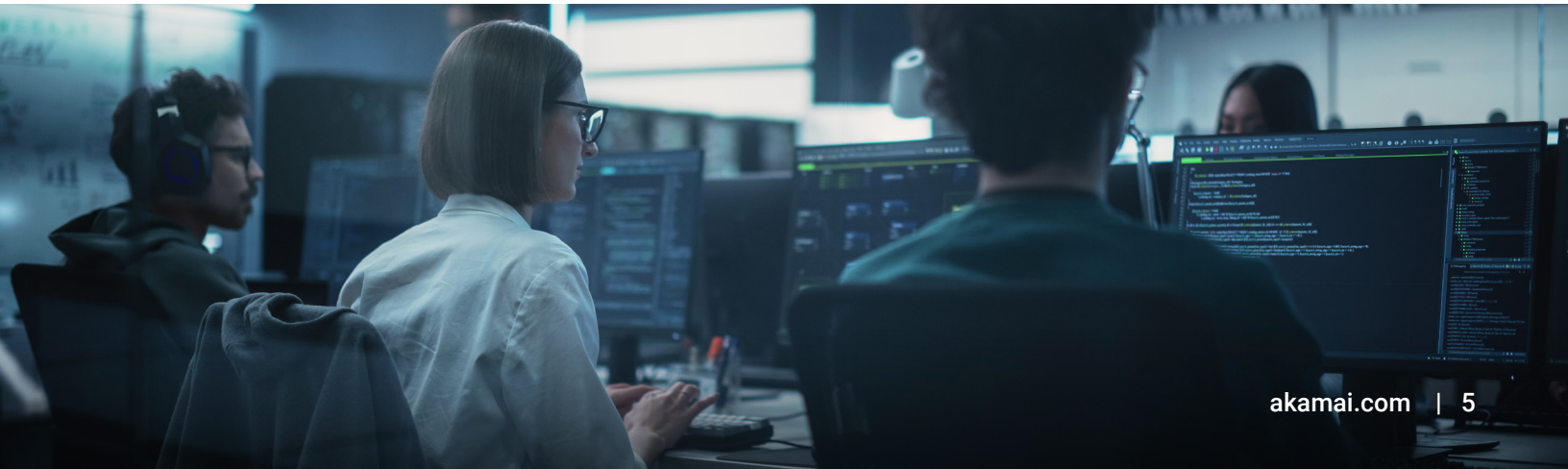
Does the algorithm use a supervised or unsupervised model?

Our algorithm is based on an unsupervised model, which allows it to adapt to each customer's environment without having previous knowledge on its characteristics. In addition, our algorithm uses online learning to adjust to changes in the environment over time.

What are the different types of anomalies detected by Akamai API Security?

Akamai API Security detects two types of anomalies:

- Pattern-based — anomalies that are based on identifying malicious patterns in the traffic such as web exploitation techniques and known malicious user agents such as command injection, path traversal, and suspicious user agent.
- Behavioral-based — anomalies that are based on the learning behavior of users, and identifying abnormal users, such as excessive API usage, range violation, and Broken Object Level Authorization.



Which parameters does Akamai API Security take into account when triggering an anomaly?

Our algorithms are based on multiple features engineered by conducting a statistical analysis of the traffic, such as the:

- Number of different users that use an API
- Authentication status of the API
- Server's response code
- Amount of data that is pulled by the user
- User's IP geolocation
- User's user agent, etc.

Can the user control the sensitivity of the algorithm?

Yes, the user can control the sensitivity of each anomaly by modifying the relevant policy sensitivity. The policy sensitivity is a number between 1 (low) and 5 (high); the highest value makes the system the most sensitive that can be configured for each anomaly policy in Akamai API Security. Our algorithm takes this parameter into account as part of the model.

Can the user mark an issue alerted by Akamai as a false positive, and how will it affect the algorithm?

Yes, for improving our anomaly detection, our users can mark relevant issues as a "false positive." When an issue is marked as a false positive, our algorithm takes this into account and adjusts the model according to the input from the user.

How does Akamai avoid "spamming" the client with a user that keeps sending the same attack scenario?

Our algorithm will identify similar issues that keep being triggered on the same user and API. In this case, our algorithm will ignore similar issues for a constant period.

How does Akamai handle drift/seasonality in the data?

Akamai API Security uses several different algorithms to detect anomalies in the data. Depending on the underlying data preprocessing and algorithm complexity, we may relax the threshold adjustment or enforce adjustments every cycle where we need guaranteed statistical thresholds for anomaly detection. In conjunction with the spam control, we allow for a hassle-free interface even when a specific algorithm requires additional cycles to adjust the thresholds.

How does Akamai handle data poisoning?

Being an online learning algorithm, Akamai API Security needs to address a variety of challenges, such as:

- New APIs
- New field(s) in existing API(s)
- Change of value type/range in a field
- Server availability issues
- Bugs in APIs that may cause errors (404, 500, etc.) and other challenges to decide which are to be learned and which are not (Akamai takes precautions to not learn these abnormalities by requiring a combination of minimal user count, time period, and persistence in order to trigger learning)

Learn how we can help you by scheduling a customized Akamai API Security demo.

Akamai Security protects the applications that drive your business at every point of interaction, without compromising performance or customer experience. By leveraging the scale of our global platform and its visibility to threats, we partner with you to prevent, detect, and mitigate threats, so you can build brand trust and deliver on your vision. Learn more about Akamai's cloud computing, security, and content delivery solutions at akamai.com and akamai.com/blog, or follow Akamai Technologies on X, formerly known as Twitter, and LinkedIn. Published 12/24.