THREATX

Why an Attacker-Centric Approach Is **Key to API Protection**

INTRODUCTION

APIs are a cyberattacker's dream. By their very nature, APIs expose sensitive data, they're everywhere, and their proliferation has outpaced security oversight and control.

Legacy WAF and many API protection solutions don't stop API-based attacks at the source. By focusing on individual attacks rather than the source of the malicious activity, these solutions allow attackers to forge ahead with their objectives — to steal your data, commit fraud, or disrupt services. In the meantime, security teams must deal with false positives and high operational overhead, all for minimal — if any — protection against API-based attacks.

In this white paper we:

- → Dive under the hood of an API attack
- → Explore why existing solutions fail to provide API protection
- → Introduce attacker-centric behavioral analytics as an effective approach to stopping attacks at the source

Anatomy of an API Attack

The best way to understand the complexity of an API attack is to walk through one from start to finish.

We'll do that, but it's important to call out that no two API attacks are alike. Attackers have a wide arsenal of attack and evasion techniques that they mix and match to avoid detection. We'll walk through a specific example using broken object level authorization (BOLA) as our exploit.

As the No. 1 API security vulnerability on the OWASP API Security Top 10, BOLA is both common and dangerous. According to OWASP, "This issue is extremely common in API-based applications because the server component usually does not fully track the client's state, and instead, relies more on parameters like object IDs, that are sent from the client to decide which objects to access."

OWASP API SECURITY TOP 10

1 Broken Object Level Authorization	2 Broken User Authentication	3 Excessive Data Exposure	4 Lack of Resources & Rate Limiting	5 Broken Function Level Authorization
6 Mass Assignment	7 Security Misconfiguration	8 Injection	9 Improper Assets Management	10 Insufficient Logging & Monitoring

Source: owasp.org/www-project-api-security

BOLA exploits are possible in any endpoint that uses user-supplied input. The most common attacks involve enumerating IDs, either user IDs or object IDs, to see if the API will respond with data. BOLA attacks can have serious consequences further down the kill chain; data leakage, privilege escalation, and account takeover (ATO) can follow.

In addition to being common and dangerous, a BOLA attack is one of the more difficult to protect against. Attackers attempt to access objects (generally targeting personally identifiable information) by manipulating identities in URLs, HTTP headers, bodies, etc. If the endpoint doesn't check permissions and/or parameters with every sequential call, the attack can easily succeed, and data can be accessed and exfiltrated through methods as trivial as GUID enumeration. Meanwhile, the application shows no sign of error, making it difficult to detect if an attacker has exploited an instance of this vulnerability.

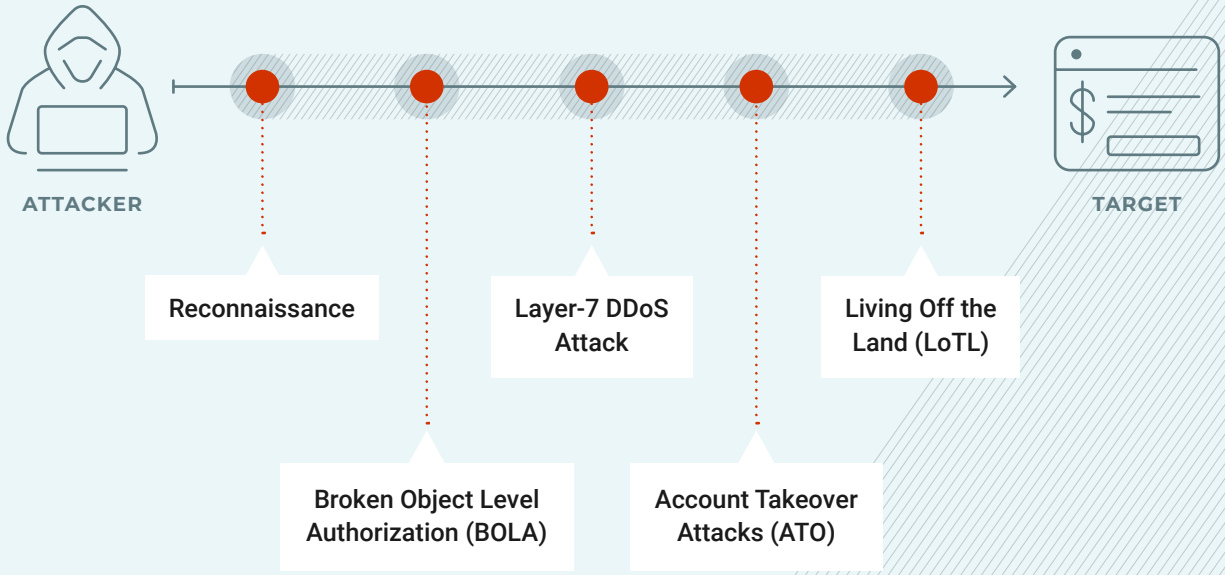
Given the significant threat BOLA attacks pose to organizations and their customers, we chose it for our attack example. But keep in mind that our example is just one of many variations of an API-based attack.

HOW TO PREVENT BOLA ATTACKS

Enforce a standardized authorization strategy within all endpoints of your API, taking into account the user authorization level and the privacy requirements of all referenced resources.

An Attack in Action

Let's imagine an attacker is targeting a financial services website.



Reconnaissance

The first step of an API-based attack is much like any other. The attacker does reconnaissance. Using open-source intelligence, or discovered methods and parameters, the attacker can begin hacking in earnest. Using cheap and ubiquitous botnet nodes available for rent by the hour, the attacker cycles user agents and IP addresses repeatedly, probing endpoints with valid or invalid account numbers, always staying just below the threshold of detection of legacy web application firewall (WAF) tripwires. Eventually, the attacker gets a hit. They find an endpoint that returns some end user PII without appropriate authorization and credentials. Then they put all that reconnaissance together into a map, and they do a precision attack.

BOLA

The attacker attempts to exploit a case of BOLA by manually testing fields until they find one that allows a transfer of funds without re-checking credentials.

Layer-7 DDoS Attack

When they find this exploit, they may launch a layer-7 DDoS attack simultaneously, using that same botnet to overwhelm an authorization endpoint. While buried in the noise of the DDoS attack, the attacker empties the account.

ATO

But the attacker doesn't stop there. They can use the information acquired during the reconnaissance phase to also carry out account takeover attacks using stolen credentials and leveraging reused passwords, to gain direct access into an account on the application server.

LoTL

From there, they may dwell, living off the land — looking for legitimate functionality to exploit — or they may probe the application even deeper, looking for new ways to exfiltrate data or steal financial assets.

Treating the Symptoms, Not the Disease

As you can see, API-based attacks are multi-faceted.

Attackers do not use a singular attack method. They string together a series of tactics. They're clever and careful. They map not only the API endpoints, but also the perimeter protection and security practices protecting those endpoints. Attackers know that their attempts will be thwarted, and they plan accordingly. Security teams are often unprepared for this reality. The tools they have in place lack the sophistication to stop attacks at the source, at best treating the symptoms and not the disease.

THE PROBLEM WITH SIGNATURES

Most organizations today have a legacy WAF deployed in their tech stack. These tools rely on signatures to detect attacks — but API-based attacks have no clear signature. Attackers rarely, if ever, follow a linear path in their efforts to breach a target. Attacks are multi-pronged. They change over time and in real time. Furthermore, attacks may begin slowly, then vary in intensity and approach, ebbing and flowing over months or longer. Effective API security needs to understand this attacker behavior and adapt to it.

A GAME OF WHACK-A-MOLE

Some API security solutions consider this multi-faceted attack mode and avoid reliance on signatures to recognize malicious activity. However, the activity these solutions do see is out of context because they only look at one API call, from a single IP address, at a single point in time. They can't see the bigger picture to recognize the attack chain and tie it to an attacker. As a result, they fail to shut down the attack at the source. The attacker simply moves to another IP address. At best, these solutions thwart *individual* API calls, maybe burning one host/IP address in a botnet comprised of tens of thousands of nodes. When this happens, the attacker simply cycles through botnet nodes and continues their pursuit. At their worst, these solutions block legitimate user behavior.

PLAYING BY THE RULES

Some solutions are better at seeing the big picture — but they see it too late and introduce significant overhead. These solutions correlate behavior across IPs *offline* so that a rule can be written to detect and block the attack. But before the attack can be blocked, someone (sometimes a tool, usually an actual human) must conclusively identify a pattern in the attack that can be described in a static rule. This is difficult to do. The baseline pattern may be similar to legitimate user behavior and can therefore result in many false positives. For example, the behavior of a user who forgets their password looks a lot like an attacker cycling through usernames and passwords.

Once a rule is written, it must be fed into another system, which may then need to interpret the rule before it can block the attack. Because new attacks are discovered all the time, security teams must constantly create and apply new rules that are null and void by the time they're done. This methodology, reliant on static signatures, recreates the failed architecture of legacy WAFs. It fails to scale, and places a massive operational burden on security operational staff.

As you can see, the rules-based approach has several shortcomings. However, the fundamental problem with these solutions is the fact that they work offline. As soon as you move any capability offline, you eliminate the opportunity to react in real time. These solutions do not monitor behavior in real time, nor can they block attacks in real time. Even if a security organization has the resources to create and apply effective rules in a timely manner, it could still be too late. An attacker could have gotten what they were after — or they could have abandoned the effort for the time being. When the attacker does come back, they're highly unlikely to follow the same attack pattern that will be detected by the rule.

AN OUTDATED APPROACH

All of these solutions apply an outdated approach to detecting an advanced attack. They lack the nuanced analytic capabilities needed to discern malicious behavior from benign, recognize bot-level attacks, and see the entire attack chain. As a result, these solutions may shut down attack after attack, but the target organization remains at risk until malicious activity is shut down at the source.

Attacker-Centric Behavioral Analytics:

Getting to the Intent of an Attack

To effectively defend against API-based attacks, organizations must expand their defense beyond discrete attacks and stop the attackers themselves.

Instead of simply blocking one malicious API call, it's important to be able to block an entire distributed botnet attack comprised of tens of thousands of nodes and millions of API calls. In this context, understanding the motivations and intent of the threat is key to defending APIs. Security teams must be able to identify the traits and behaviors of an attacker. In other words, they require attacker-centric behavioral analytics to identify, track, and defend against sophisticated API security threats.

Attacker-centric behavioral analytics is an evolved approach to identifying key elements of an attack and responding appropriately before damage is done. This approach continuously monitors all users as they interact with an application or API, looking for key indicators of suspicious behavior and tracking risk over time and across multiple endpoints. By combining attacker fingerprinting, multiple markers that identify unique users, along with numerous detection techniques, attacker-centric behavioral analytics identifies suspicious patterns of behavior. This combination allows the solution to track suspicious and malicious users across multiple IP addresses as they use various evasion techniques and modify their attack parameters over time.

The key to this approach is the shift from "pattern matching" against a single request to "user profiling" — monitoring every request from every user to characterize their behavior and map their intent. All inbound traffic, including all elements of a request (for example, header information and parameters), is scanned in real time. This deep, real-time monitoring enables advanced threat engagement techniques such as IP fingerprinting, interrogation, and tarpitting. Proactively challenging the attacker in this way allows the solution to identify and stop the most complex attacks, including large-scale bots and DDoS-level threats to API endpoints, in real time.

WATCHING AND BLOCKING BOLA WITH ATTACKER-CENTRIC BEHAVIORAL ANALYTICS

It's critical to detect probing/reconnaissance activity targeting BOLA vulnerabilities in APIs. Attacker-centric behavioral analytics can identify and flag resource ID enumeration, which indicates an attempt to exploit a BOLA vulnerability. This approach can also identify other anomalous activities, such as repeated malformed API requests, that indicate malicious intent. If the behaviors collectively reach a predefined risk threshold — whether taken alone or, as is usually the case, when observed in conjunction with other suspicious behavior — the solution can block the attacker and record the events for later review.

ThreatX:

The Pioneer in Attacker-Centric Behavioral Analytics

As the pioneer in attacker-centric behavioral analytics, ThreatX enables security teams to stay ahead of an ever-changing threat landscape and defend against attackers who are intent on exploiting vulnerable APIs and web applications.

The ThreatX Platform protects APIs from all threats, including DDoS attempts, bot attacks, API abuse, exploitations of known vulnerabilities, and even zero-day attacks.

REAL-TIME BLOCKING

ThreatX scans all inbound API traffic in real time, identifying and blocking attacks. As risk rises, ThreatX blocks attacks — stopping the threat in its tracks in real time. For example, ThreatX recognizes attacker behavior indicative of a BOLA attack, then flags and watches that user. This real-time monitoring enables ThreatX to execute advanced threat engagement techniques such as IP fingerprinting, interrogation, and tarpitting. When a series of user interactions indicate a certain risk threshold has been met, ThreatX blocks the user in real time.

MULTI-STEP ATTACKS

Rather than requiring a single, significantly risky event or identifying a known signature, ThreatX identifies and blocks more threats more accurately by analyzing behavior from multiple vantage points. In this way, ThreatX can correlate several behaviors back to one attacker and identify behavior that is suspicious but wouldn't be flagged by other security solutions, such as resource ID enumeration, to exploit a BOLA vulnerability.

FEWER FALSE POSITIVES

ThreatX's blocking modes are designed to block malicious requests and stop suspicious entities from attacking APIs while allowing benign traffic and real users through. ThreatX can tell the difference between a legitimate user who forgot their password and an attacker cycling through usernames and passwords as part of a concerted ATO attack.

CONCLUSION

Attackers use every tool and technique at their disposal to evade detection while carrying out complex, multi-step API-based attacks, and they've been largely successful — up until now.

ThreatX's attacker-centric behavioral analytics engine brings a new level of sophistication to web application and API security, enabling organizations to get to the root of an attack and stop it at the source.

Learn more about ThreatX's API attack protection capabilities. →

Or request a demo. →

THREATX

www.threatx.com
info@threatx.com

ABOUT THREATX

ThreatX's API protection platform makes the world safer by protecting APIs from all threats, including DDoS attempts, BOT attacks, API abuse, exploitations of known vulnerabilities, and zero-day attacks. Its multi-layered detection capabilities accurately identify malicious actors and dynamically initiate appropriate action. ThreatX effectively and efficiently protects APIs for companies in every industry across the globe. For more information, visit: www.threatx.com.