API PROTECTION: THE NEW FRONTIER

ebook
An SC Media publication

Sponsored by THREATX

From WAF to WAAP

Attackers increasingly target web APIs to gain a foothold into companies to deny access, install bots, escalate privilege, and find sensitive data. Traditional WAFs can't handle the onslaught. David F. Carr explains how the best defense is an integrative approach called WAAP.

Every time computing takes a step forward, security programs struggle to keep up. The proliferation of web-based APIs is no different.

Using HTTP or HTTPS to exchange application requests and responses – often with payloads in JavaScript Object Notation (JSON) or XML format – has become a de facto standard for distributed computing. Securing those transactions requires a rethinking of application architecture and a retooling of application firewalls.

APIs are useful to companies wanting to achieve feature parity between web and mobile versions of the same app, but they can open vulnerabilities in the process. Tom Pohl, pentesting team manager and senior cybersecurity consultant for LGM Security, saw that play out recently when testing a modern web application.

"I found most of the issues with the application were in the APIs," he said. "Before, applications were typically form-based – you submit a form, and you get some data back. Now, you have an endpoint and it submits data using JSON, or some other format that the application is expecting, and that opens up all sorts of attack angles because it's a little bit newer." In several cases, he has found he can exploit an application by specifically targeting how JavaScript Object Notation (JSON) is parsed by the server app.

Just as with input submitted by a web form, API inputs need to be carefully sanitized and validated, Pohl says.

Too often, application builders don't fully understand the security implications of the API approach. While traditional forms-based web applications were certainly notorious for exposing security flaws, at least they were discrete applications. With public web-based APIs, the organization is exposing the building blocks of applications, which sometimes can be recombined or accessed in unexpected ways to create a security breach.

An example of this was the April 2021 discovery of a flaw in Experian's credit lookup APIs. As reported in Krebs on Security, Bill Demirkapi, a sophomore at the Rochester Institute of Technology, was shopping for student loan deals when he became curious about the credit score information he gained access to through a loan aggregator's website. Finding evidence that the data was provided by an Experian API on the back end, he discovered that the API could be manipulated to provide a credit report for almost anyone in the service bureau's database. All he had to supply as input was name, address, and a string of zeros for date of birth.

"No one should be able to perform an Experian credit check with only publicly available information," Demirkapi said at the time.

OUR EXPERTS: API Security: The New Frontier

Tom Pohl: Pentesting team manager and senior cybersecurity consultant, LGM Security

Abib Khan: Offensive security lead, United Bank Limited

Michael Bissell: API evangelist, Silicon Valley Bank

Anshu Gupta: Investor, Silicon Valley CISO Investments

Neil Weitzel: SOC manager, ThreatX

Goodbye WAF, hello WAAP

Just as common flaws in web applications gave rise to a market for Web Application Firewall (WAF) systems, issues specific to APIs have rendered WAFs obsolete. To better confront the threat, security teams are turning to the next generation of web app security: Web Application & API Protection (WAAP), which is more focused on API protection, bot management and expanded protection against distributed denial-of-service attacks.

Many of the most common vulnerabilities are related to authentication and authorization flaws, according to Neil Weitzel, manager of the Security Operations Center at ThreatX, which provides a WAAP solution. “Authorization checks are sometimes configured loosely with APIs because assumptions are made about the client endpoints interacting with it. This leaves application functions or API calls available to any attacker who is able to steal the API keys – or, worse yet, any attacker who is able to discover an API for which no keys are required.”

Neil Weitzel, SOC manager, ThreatX

For example, suppose an airline has configured its applications to offer special deals only to selected partners. If the partner ID submitted with a request is not checked to confirm a match with the client’s credentials, the application can be manipulated to give those special deals to anyone.

“A WAAP helps ensure strict schemes are enforced on API requests and only trusted clients are permitted access, while analyzing interactions with the API for nefarious behavior,” Weitzel said. “ThreatX backs up automated detection with a security operations center that can intervene to block novel patterns of attack. Equally important, the SOC professionals help keep false positives out of the list of sites to be blocked.”

Like any good security plan, an API security program should include multiple layers of defense, but a WAAP should be one of them, Weitzel says.

Adib Khan, a pentesting specialist who focuses on API security at United Bank Limited in Pakistan, agreed. “You should not just rely on the firewall, you should configure apps in a secure manner,” he said.

Khan contributed to the OWASP API Top 10, an API-specific list of common vulnerabilities the Open Web Application Security Project Foundation published as an addendum to its list of common web application vulnerabilities. Many of the basic issues are the same, but the methods of exploitation are different, he said.

> “A WAAP helps ensure strict schemes are enforced on API requests and only trusted clients are permitted access, while analyzing interactions with the API for nefarious behavior.”
>
> *– Neil Weitzel, manager of the Security Operations Center at ThreatX*

An application firewall can provide some protections, particularly by providing rate limiting and spotting common exploits like SQL injection, he said. But he’s skeptical of the ability of a firewall to protect against flaws buried deeper in the application logic. For example, if an application allows any authenticated user to access any credit report – if that looks like normal activity to the firewall – the request won’t necessarily be blocked.

It's better for security to be baked into an application and its APIs up front, Pohl said. Application firewalls "are just band aids over what really should happen," he said. "Your application should be architected well enough that you don't need a web application firewall."

Tom Pohl, pentesting team manager and senior cybersecurity consultant, LGM Security

That said, we don't live an ideal world, he acknowledged.

The most useful features a WAAP can provide include rate limiting (detecting an excessive volume of requests to an API, which might be associated with scanning for or exploiting vulnerabilities) and correlating access to multiple APIs. Some vulnerabilities arise from APIs having been designed in isolation, not appreciating how an attacker might chain them together to create an exploit, so a system capable of spotting those patterns would be useful.

Ideally, if you feed the WAAP enough "known good" traffic, it ought to be able to use machine learning techniques to build statistical models that will spot the possibly harmful outliers, Pohl said.

THE OWASP API TOP 10

The Open Web Application Security Project (OWASP) Foundation drafted an API-specific version of its warnings against common security vulnerabilities, published in 2019. While many of the issues also apply to web user interface applications, here's what they judged to be the biggest threats to APIs:

1. Broken Object Level Authorization
2. Broken User Authentication
3. Excessive Data Exposure (exposing object properties, regardless of sensitivity; relying on client-side filtering)
4. Lack of Resources & Rate Limiting
5. Broken Function Level Authorization
6. Mass Assignment (improper filtering leads to clients being able to change properties they shouldn't have access to).
7. Security Misconfiguration
8. Injection (SQL injection and other forms of command injection)
9. Improper Assets Management
10. Insufficient Logging & Monitoring

The Log4j vulnerability – a serious flaw in an open source Java library for event logging that's embedded in many different applications – is the sort of thing a WAAP can help protect against, said Michael Bissell, API evangelist at Silicon Valley Bank.

"That particular exploit you're probably not going to catch on day zero because you won't have firewall rules against it," Bissell said. But organizations with well-organized security plans were well positioned to block Log4j – buying them time to track down and patch all the places where the software was present.

A closer look at how APIs are exploited

Web APIs can be deployed (and exploited) in many ways. Some are employed in server-to-server connections, meaning they are a step removed from any web or mobile client. Log4j is an example of what was intended to be an internal server-to-server API but often was deployed in a way that made it publicly accessible.

But perhaps the most common use – and where problems often crop up – are APIs used as the back end to mobile apps or JavaScript-powered web user interfaces.

If a web application invokes an API via JavaScript, a hacker can learn a lot from "view source" and monitoring of the network requests made by the script. "You can do that simply with the browser – you don't need any further tools," said United Bank Limited's Khan. That is, the same web development utilities built into the popular browsers for debugging can be the starting point for hacking.

Once API calls are identified, a hacker (or penetration tester) can experiment with submitting invalid data to them until something breaks. Part of the danger comes from the way an attacker can begin cataloging system objects, often by their JSON representation, and gaining access to objects that should be disallowed, Khan said.

Anshu Gupta, investor, Silicon Valley CISO Investments

APIs do have some security advantages over web applications, Pohl added. For example, APIs are commonly protected with long cryptographic keys, rather than a password that a human user is supposed to be able to remember. And the organization granting access can generate those keys randomly, rather than relying on unimaginative users to provide passwords that, too often, they've also used on other sites.

"I think you get a couple of really good things out of that," Pohl says. "They're not giving the user the ability to determine what that password, or that key, is going to be." However, what still can easily happen is that API clients are given access to more data and more application functions than they ought to be entitled to. "The fundamentals haven't changed – that's just the least privilege principle – only provide the access needed for the work that needs to be done," he said.

Often, the issue is not so much with the initial authentication as with what an authenticated client can do after gaining access. An example that is extremely prevalent, which you would think wouldn't be after all these years, is extended data exfiltration, in which any application can be accessed via an account number or other identifier that the client can change, independent of their login ID.

Attackers often discover they can iterate through all possible account numbers until they find one that works using enumeration or randomly generated identifiers until they find one that works. A properly designed application would detect that an authenticated user was attempting to access data they shouldn't have access to.

Checking authentication up front without monitoring subsequent accesses is like securing the front door of the bank without securing the bank vault or the safety deposit boxes – giving anyone allowed in the door an opportunity to steal from any other customer. A good WAAP can play the role of the alert security guard who spots problems at the vault door.

How to choose and deploy a WAAP Platform

Anshu Gupta, a past CISO and member of the investment syndicate Silicon Valley CISO Investments, suggests that anyone shopping for a WAAP look for one with advanced botnet protection and machine learning capabilities. "If a bot mimics human behavior, it can be very hard to catch that," he said. With a business-to-business API, the machine learning capabilities should be smart enough to detect that a partner that normally requests megabytes of information is suddenly requesting gigabytes of data – and flag that behavior as suspicious.

> "If a bot mimics human behavior, it can be very hard to catch that."
>
> – Anshu Gupta, investor, Silicon Valley CISO Investments

The trick is to analyze traffic well enough to separate legitimate from illegitimate traffic without adding so much overhead that performance is ruined for the legitimate users, Gupta said.

Beyond looking at the reputation of a WAAP provider and how long they have been in business, Pohl said he would choose one based on how well it responds to any issues that may arise with the system. Hopefully, a WAAP system wouldn't have its own security flaws – although Pohl has seen issues with access to the management consoles in security products – but they can have performance and reliability issues like any other software.

"I'm looking for an open, honest dialog," Pohl said. "I don't care about the features in your product as much as I care about how you respond when issues arise."

Once having selected a product, be sure to allow adequate time to tune it for your specific applications, Pohl said. "If anyone tells you that you don't have to tune their product, it just works – well, either it's going to miss a lot, or it's going to make everyone really frustrated."

Like many other security products, WAAPs can suffer from false positives, meaning they can block things that shouldn't be blocked. Pohl said one of his favorites is where a firewall will block inputs that contain the pipe character and "nc," mistaking it for a command injection attack targeting the netcat command line utility often targeted by hackers – when really the application is using the abbreviation for North Carolina.

Most application firewall products can be configured to start out in some sort of "learning mode" where they initially generate alerts without blocking traffic, allowing the organization to identify the false positives that may occur in normal application traffic, Pohl said. "You look at illegitimate versus legitimate traffic, and then you start tuning the system. Tuning is where you'll spend most of your time with these products."

In other words, WAAP systems need to be managed properly like any other element of your technology infrastructure. Once configured, however, they provide an essential layer of protection for any application with web APIs – which is most applications, these days.

ThreatX's Weitzel said a good API protection platform is supported 24/7 by trained expert security professionals that use attacker behavior analytics, application profile and traffic, and known attacks and patterns to analyze all HTTP traffic to APIs and applications.

"This allows the SOC to apply risk scores to potentially harmful traffic. As would-be attackers attempt to progress down the kill chain, they are blocked and reported for further investigation." ■

For more information about ebooks from SC Media, *please contact Bill Brenner, VP, content strategy, at bill.brenner@cyberriskalliance.com.*

If your company is interested in sponsoring an ebook, please contact Dave Kaye, chief revenue officer, at (917) 613-8460, or via email at dave.kaye@cyberriskalliance.com.

Sponsor

THREATX

The ThreatX 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. ThreatX's attacker-centric behavioral analytics capability secures APIs from advanced security threats across cloud, on-prem and hybrid environments. Our multi-layered detection capabilities accurately identify malicious actors and dynamically initiate defensive action to protect known, rogue and zombie APIs. In addition, our Managed Services combine threat hunting with access to experts 24×7, significantly reducing the direct operational costs and maintenance for enterprises. With our advanced API protection platform, managed services and threat research, ThreatX effectively and efficiently protects APIs for companies in every industry across the globe.

More information can be found at www.threatx.com

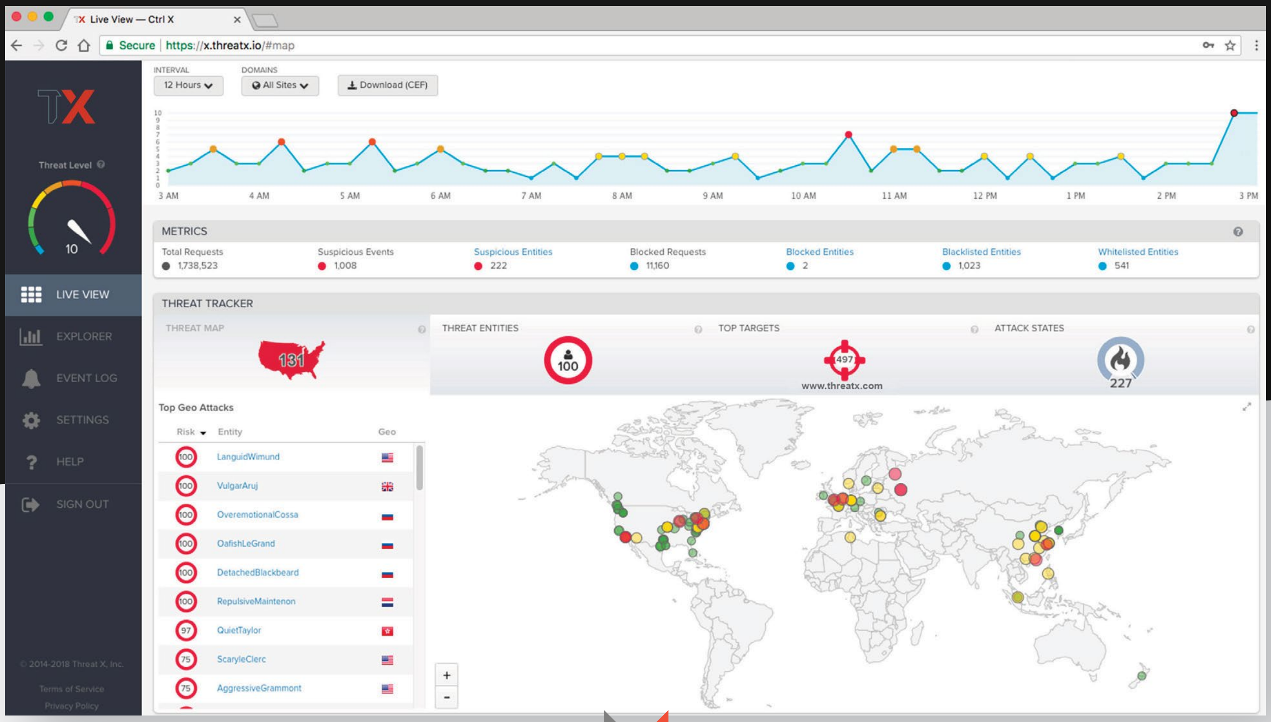
Masthead

EDITORIAL
VP, CONTENT STRATEGY
Bill Brenner
bill.brenner@cyberriskalliance.com
PROJECTS MANAGER
Victor Thomas
victor.thomas@cyberriskalliance.com

SALES
CHIEF REVENUE OFFICER
Dave Kaye
(917) 613-8460 *dave.kaye@cyberriskalliance.com*
VP, SALES
Matthew Allington
(707) 651-9367 *matthew.allington@cyberriskalliance.com*

All your apps. All your APIs. All the threats.

ONE SOLUTION.



METRICS

Total Requests	Suspicious Events	Suspicious Entities	Blocked Requests	Blocked Entities	Blacklisted Entities	Whitelisted Entities
● 1,738,523	● 1,008	● 222	● 11,160	● 2	● 1,023	● 541

Top Geo Attacks

Risk	Entity	Geo
100	LanguidWimund	🇺🇸
100	VulgarAruj	🇬🇧
100	OveremotionalCossa	🇷🇺
100	OafishLeGrand	🇷🇺
100	DetachedBlackbeard	🇷🇺
100	RepulsiveMaintenon	🇳🇱
97	QuietTaylor	🇭🇰
75	ScaryleClerc	🇺🇸
75	AggressiveGrammont	🇺🇸

ThreatX's single Web Application and API Protection (WAAP) platform addresses the full spectrum of API protection and application security. From traditional OWASP attacks, bots and malicious automation, to DDoS mitigation, and API-specific threats, we've got you covered.

- **Reduce false positives by watching attacker behavior, not attack signatures**
- **Let AppSec experts in our managed services SOC ease your team's burdens**
- **Speed time-to-blocking across all attack types, not just a few**

Find out how ThreatX delivers comprehensive coverage across the entire threat landscape.

learn more at **threatx.com**

THREATX

Protecting the web apps & APIs that run the world.