AppSentinels.ai
Application Security Re-invented

Why Web Application Firewalls (WAFs) are inadequate against API Attacks

Why Web Application Firewalls (WAFs) are inadequate against API Attacks

During our various customer interactions, we often discuss how AppSentinels solution is different compared to a Web Application Firewall (WAF) in protecting against API's attacks:

The core difference is that AppSentinels API Security Platform knows the context of what it is protecting while unfortunately WAFs don't.

Let me explain why I am saying this and why this is important:

WAFs were built 2 decades ago to protect web applications. There is no standard way to describe what a web application does and how to interact with it. With that challenge, WAFs start with a negative security model, i.e, a denylist. Such an approach leverages a library of threats or known attacks (which by definition puts it behind time) in the form of regular expressions, describing patterns to look for in the traffic. To execute this denylist, WAFs match regex in a single network session and it does not have context to understand either the application or the user behavior. **In summary, WAFs are a general-purpose security solutions, protecting any web application in the same manner, regardless of the application's functionality and purpose**.

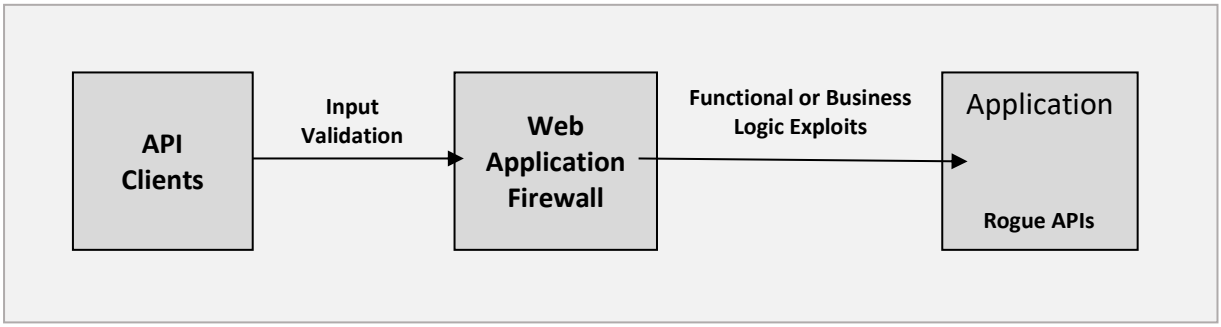
This negative model creates multiple issues for an organization:

1. **False Positive Management**: As we discussed to match a deny-list, WAF doesn't maintain context of application. It generates alerts based on regex matching a network session. This passes the buck of analyzing alert for a valid issue or false-positive to the security teams, adding to the alert fatigue security teams are already facing.
2. **Continuous Tuning**: To ensure WAF doesn't end up blocking critical transactions, WAF administrators create complex and specific rules to bypass certain traffic. As software keeps evolving, Security teams are in a continuous iteration to maintain delicate balance in security - to block attacks and ensuring critical transactions are not impacted. Again they are always a version or two behind.

In recent years, WAF vendors have started adding capability to inspect APIs using OpenAPI/Swagger schema. Relying solely on this approach for API security has serious limitations as explained below -

1) This approach is limited to performing syntax checks on API structure. It doesn't build context of the application and hence cannot detect functional or business-logic exploits.
2) These features depend on organizations to provide precise API schema. Most organizations are struggling with basic visibility of APIs and their API documentation is lagging. The result is outdated or incomplete schemas, renders such features irrelevant and noisy.
3) Furthermore, as WAFs require constant tuning and maintenance, the variety and pace of APIs development makes them practically impossible to stay on top of. Too often, security teams are unaware of changes being made to APIs - or even aware of all the APIs the organization has to begin with.

Let's see this graphically:



API attacks are different

Unlike web URL's, APIs have defined interfaces and are meant to give control to callers - means it's the client using API that has control of the flow (or business logic) than the servers. This opens multiple hidden paths in the application that were not exposed earlier or considered during development.

Let's see with few examples how API attacks are different than traditional attacks and why organizations using API need to take a different approach than only relying on WAF to protect their applications.

To attack APIs, hackers use discovery and inspection techniques to understand the structure and logic of APIs. The tools they use are very similar to what organizations will use to build their APIs. Attackers may make minor changes to a current and a valid API calls to see how the API responds. This process takes time and requires multiple attempts. Few attempts may succeed but most attempts fail. Attackers may try stuff like –

1. Enumerate an object and see if API returns information associated with the new object. The new object may or may not be associated with current user session.
2. Replace parameters of one user session with another's and see if API returns error or the other users information.
3. Try changing the method of the API and see if API is successful.
4. Try API's not associated with the current user role like admin APIs and see if APIs are successful.
5. Try checking out security mechanisms implemented in APIs like authentication, authorization enforcements, rate-limits etc.

Once attackers have mapped out API logic and found the API's vulnerabilities, they will try to chain multiple API's. These sequences of APIs will be different compared to the ones seen during normal workflows. Further an attacker may repeat this process with different set of APIs, till they get information or is able to break application business logic.

Looking at these activities in silos, one transaction at a time, doesn't reveal any risk, but putting the pieces together can provide context and insight that someone is probing and breaking logic of APIs and is up to no good.

Right architecture is foundational to do AppSec right

An attacker's reconnaissance & discovery attempts provides a clear signal that someone is probing APIs. As discussed earlier, to detect this activity, solutions need context that cannot be learned looking at each transaction in silos. Solutions must analyse and stitch as much data as possible to understand normal behaviour, identify outliers, and put together the pieces to form a bigger picture. Traditional tools lack big data, AI and ML and don't have ability to gather and stitch these events to come up with complete picture.

By analyzing complete API workflows, AppSentinels creates baseline of normal behaviour, models complex relationship within the application to discover workflows and identifies anomalous users not displaying normal behavior. With such a context, it's able to differentiates between software changes or simple mistakes vs. malicious activity. This allows AppSentinels to stop attackers in their reconnaissance or chain attempts to prevent successful attacks. With it's deep understanding of the behavior, AppSentinels is able to stop attacks in advanced stages as well.

Note on OWASP API Security Top 10

Aside from its widely recognized list of the top 10 Web Application Security Risks, in 2019 OWASP published a separate list dedicated to API Security. This list provides a well-researched, detailed review of common vulnerabilities exploited to abuse APIs, listed in descending order of risk based on ease of exploitation, prevalence, ease of detection for attackers, and potential impact.

It's important to remember that while OWASP covers the common denominator for API vulnerabilities, each API is different and has its own unique vulnerabilities associated with its business logic and processes, and user behavior become more contextual than others.

Below table provides OWASP API Top-10 coverage of AppSentinels vs WAF, API-Management and API-Gateway solution:

OWASP API Protection	AppSentinels	WAFs	API Management	API Gateway
OWASP API-1 Broken Object Level Authorization	**Complete**	None	None	None
OWASP API-2 Broken User Authentication	**Complete**	Partial	Complete	Partial
OWASP API-3 Excessive Data Exposure	**Complete**	Partial	Partial	None
OWASP API-4 Lack of resources and rate limiting	**Complete**	None	None	None
OWASP API-5 Broken Function Level Authorization	**Complete**	None	None	None
OWASP API-6 Mass Assignment	**Compete**	None	Partial	None
OWASP API-7 Security Mis-configuration	**Complete**	Partial	None	None
OWASP API-8 Injection	**Complete**	Complet e	None	Partial
OWASP API-9 Improper Assets Management	**Complete**	Partial	None	None
OWASP API-10 Insufficient Logging & Monitoring	**Partial**	Partial	None	None

About AppSentinels.ai

AppSentinels is next generation API security company that is redefining the way Application Security is done in organizations.

- With it's continuous discovery, AppSentinels helps bring complete visibility of API's along with sensitive data exposure even when applications are evolving.
- With multi-layer defence shield – it protects organizations against all known and unknown attacks in production environment.
- With remediation, it helps developers address them with pin-pointed accuracy.
- It's intelligent DAST uncovers yet to be exploited API business logic issues and helps transform organization from reactive to pro-active security posture.

