



WHITEPAPER

 SRC CYBER SOLUTIONS LLP

**THREATX**

# Designing AppSec in the Age of APIs and Microservices

# Executive Summary



Change is a constant part of security. Security teams face new threats, vulnerabilities, and intelligence on a daily basis. However, change is occurring on a much larger scale. Fundamental changes to the way that applications are developed, architected, and delivered are challenging some of the basic assumptions that Web Application Firewalls (WAFs) have relied on for decades.

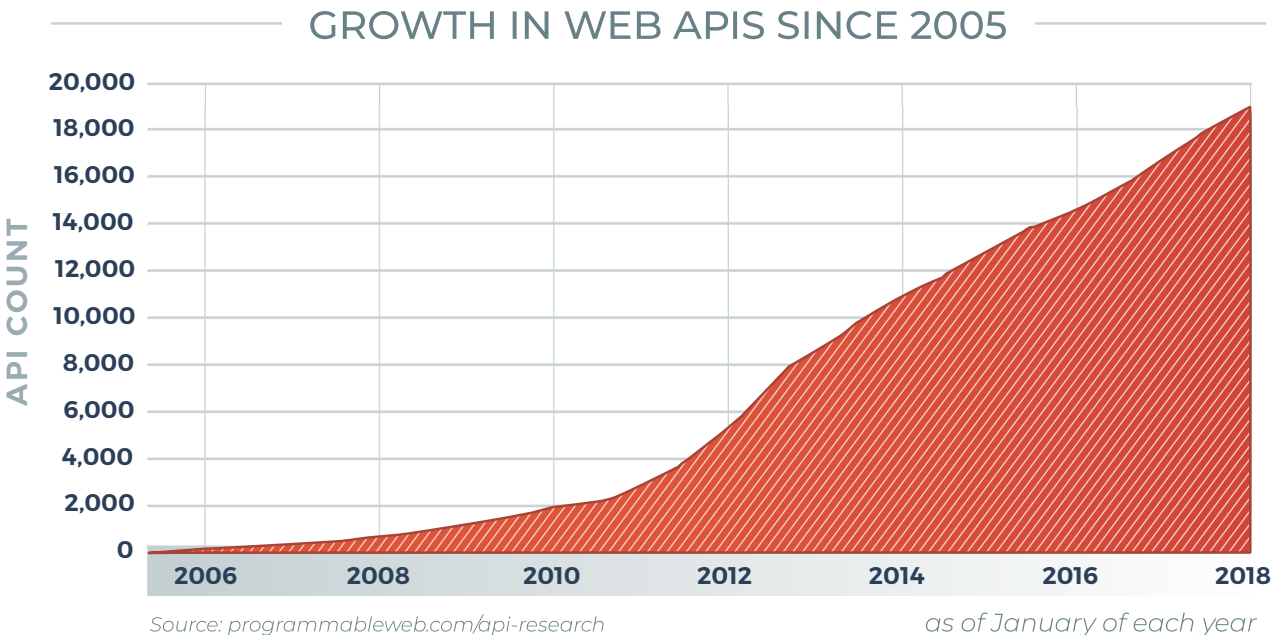
Organizations no longer have a few applications with a few paths to access. Instead, they have many applications, each often dependent on and delivered through a myriad of APIs. Microservice architectures, mini-apps themselves, have made applications more modular and easier to develop, but rely heavily on internal communication that traditional security appliances can't see. Lastly, applications are being developed in new ways. DevOps and CI/CD pipelines are driving fast, continuous development and security needs to be able to keep pace without slowing down delivery or losing efficacy.

This paper analyzes each of these major shifts and how they impact modern AppSec strategies. We will analyze some of the drivers behind the trends, the challenges they pose to traditional security, and finally, provide examples of how security can move forward.

[Read on for more info...](#)

# APIs and the New Application Attack Surface

APIs have steadily transformed the face of modern applications. Instead of every capability being directly coded into a site, modern applications use an array of HTTP-based APIs to build and deliver functionality. For example, services like GoogleMaps or PayPal can be added to an application via an API without developers having to reinvent the wheel. And while these are well-known examples, there are well over 20,000 public APIs in the API economy according to ProgrammableWeb.



APIs are also at the heart of how many applications are delivered to end users. Mobile apps and cloud services, for example, rely heavily on APIs to deliver an application experience that remains consistent over a wide range of end-user devices. APIs also provide much of the all-important communication between modules and services of containerized applications. Whether Internet-facing, mobile, or container-based, APIs have become crucial to modern applications.

## Challenge: A New Wave of Attacks Against Old Defenses

The scale and diversity of APIs make the modern application attack surface much more complicated. In the past, all user experiences were predictably funneled through a single path of a WWW website. If we think of this traditional web application as water flowing through a funnel, an API-based application is more like a colander with many different paths for users to connect and for traffic to flow. With many more paths to the applications, security teams suddenly have many more paths to defend. And with the speed of modern development, security teams are often not aware of all the APIs that exist in their application environment.

Each of these APIs is a potential target and attackers have shifted to this unguarded attack surface in droves. In fact, Gartner Research predicts that “by 2022, API abuses will be the most-frequent attack vector resulting in data breaches for enterprise web applications.” This is a fundamental shift that immediately requires security to adapt.

“

**By 2022, API abuses will be the most-frequent attack vector resulting in data breaches for enterprise web applications.**

- Gartner Research

Legacy WAFs have simply proven too cumbersome and ultimately ineffective at defending against API-based threats. This is because...

- » Legacy WAFs must be deployed at centralized choke points where all traffic can be seen.
- » APIs can provide a variety of paths and backend integrations that can naturally circumvent this chokepoint and thus are invisible to security.
- » APIs can experience different types of attacks and abuses versus a web app or webpage.
- » Traditional WAFs are designed to protect the web front-end of an application and may not have detection capabilities to recognize abuses against APIs.
- » Even if signatures are available, the work required to tune signatures and rules for each API can quickly become overwhelming.

Many organizations have also deployed API gateways to provide additional protection for their APIs. And while these gateways can provide important services such as authentication, load balancing, and rate limiting, they do not perform the detailed traffic analysis and inspection to detect threats. This leaves APIs exposed to injection attacks, reconnaissance, fuzzing, Layer 7 DoS attacks and much more.



This ultimately means that, for most organizations, the fastest growing part of their application attack surface (APIs) lacks the appropriate levels of detection and prevention.

## How the Next-Generation WAF Secures APIs

ThreatX takes a new approach to the WAF, which is built for the unique challenges of modern applications and their many APIs. This includes ensuring that staff can quickly and easily find and secure all of their APIs, while also providing new detection strategies to mitigate the threats common to APIs.



**Ensuring Full API Coverage** - Unlike traditional WAFs, ThreatX is designed to protect both the front-end of an application as well as the back-end APIs. At its heart, ThreatX protects the application environment itself and does not rely on a centralized choke point in the way that a physical or virtual appliance would. For example, ThreatX Sensors can be deployed as part of a pod alongside any containerized services that need to be protected. This ensures that regardless of how the application is accessed, all paths lead to security.



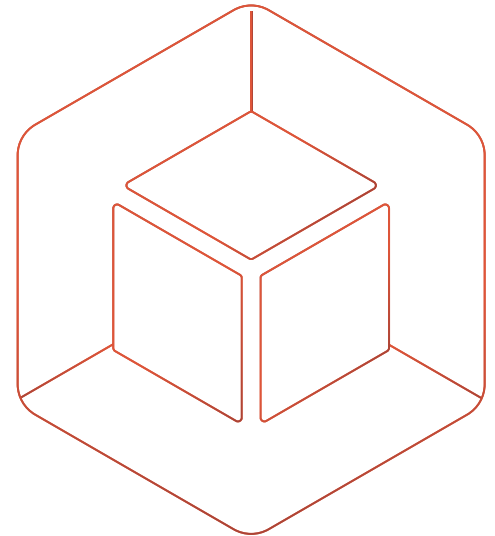
**Intelligence to Detect API Threats** - APIs can be attacked in a variety of ways, including login and credential attacks, data theft, DDoS attacks as well as traditional injection attacks. Instead of using signatures and rules, ThreatX detection uses a combination of application-centric and attacker-centric, behavior-based methodologies.

Application-centric models can reveal unusual application behavior as well as high intensity of use that indicate automated attacks. This view is complemented by attacker-centric analysis that learns the unique behaviors and indicators of attackers and tracks them over time and across the attacker kill-chain. The system then actively interrogates, fingerprints, and even deceives attackers automatically without the need for signatures and rules. **This provides a highly automated way of detecting and mitigating the wide range of attacks that can target APIs, without requiring security teams to maintain complex rulesets.**

# Building Security for Microservices & Kubernetes

While APIs have had a major role in reshaping modern applications, it is only one aspect of the sea-change in how that applications are being architected and delivered. The containerization of applications has allowed developers to abstract their applications away from the underlying hardware that the applications will run on. This is incredibly beneficial as it means that organizations can build applications that can run in a variety of environments (local data center, cloud, hybrid, etc) and on a variety of underlying hardware and operating systems.

Additionally, containers have made it easy to break an application into underlying logical services. Instead of a single massive monolithic codebase, these smaller services can be coded and maintained individually, making the codebase far more manageable for the developer. Likewise, testing becomes easier because each service simply needs to deliver inputs and outputs to other services in the application, and there are fewer chances for code in one part of the application to break something in another.



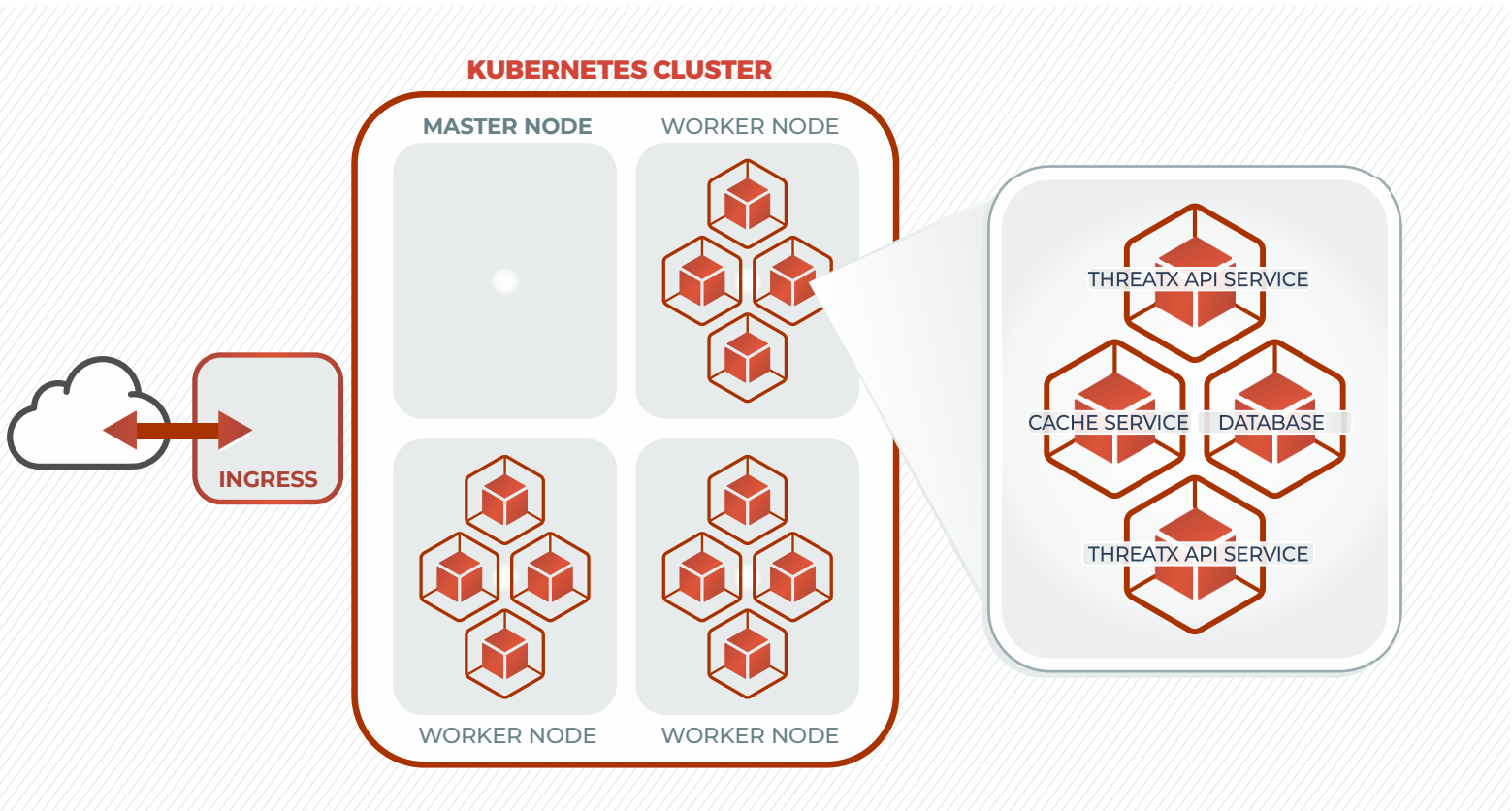
As overall development becomes faster and more reliable, orchestrating the deployment and scaling of containerized applications becomes critical, Kubernetes brings all of this together allowing developers to easily pick and choose the components they need, connect them, and then scale up on the fly.

## Challenge: External Defenses Miss the Inside Story

The containerization of applications however, creates a big challenge for legacy WAFs, which are typically based on physical or virtual appliance models. In a Kubernetes environment, there is no longer a single monolithic stream of traffic where an appliance can sit with visibility to all that traffic. Instead, the application will have many pods that can spin up or down and microservices that may need to communicate to one another internally. Once again, developers rely heavily on APIs for the east-west communication between services. But just as importantly, this means that a considerable amount of application behavior can occur between services within the application itself. An appliance-based WAF simply would not be able to see this internal traffic and behavior.

## How a Next-Gen WAF Secures Microservices

ThreatX allows security to be containerized right alongside any services that need to be protected. In a Kubernetes context, a containerized ThreatX Sensor can be deployed into a pod that houses a containerized service. This makes the WAF sensor a standard part of a new pod when it is created and ensures that security can automatically scale and adapt as the application itself changes. It also means that, once again, all paths lead to security regardless of how the application is architected.



Additionally, the analysis of internal communication between microservices opens up new opportunities for threat detection. ThreatX constantly profiles and learns the behavior of the application and its underlying services, and then finds abnormalities that can indicate an attack. As services become smaller and more specialized, their behavior becomes far more predictable. This makes it easier to train the behavioral learning models and spot signs of an attack. **So not only does containerized security ensure visibility, it actually empowers new styles of analysis.**

# Security Alignment and Independence in the Age of DevOps

In the same way that APIs and microservices are changing the building blocks of applications, the rise of DevOps and CI/CD pipelines are transforming the operational side of how they are built.



The slow, linear waterfall development strategies of the past, are being replaced by fast, iterative development cycles that can adapt to the changing needs of customers and the market.

These changes not only reflect a change in development philosophy but in many cases, involve organizational changes within teams as well. In the past, Development teams and Operations teams were separate entities. Ops teams often didn't even begin testing until long after code was committed. DevOps recognizes the interdependence of these two teams and aims to bring development and testing into a unified process. Instead of massive releases that are released occasionally, organizations have shifted to rapid, highly-focused updates that can be delivered on daily basis or even multiple times a day.

## Challenge: Run Security at DevOps Speed While Remaining Independent

While DevOps has benefits for the business, it creates a new series of challenges for old-school security practices. Maintaining complex signatures and rulesets has always been a challenge for traditional WAFs. However, this challenge gets decidedly more complex when the application itself is constantly changing. In addition to keeping up with new signatures and vulnerabilities, the application, its services, and functions can change on a daily basis.

Additionally, with the success of DevOps, many organizations are interested in further integrating security functions into the DevOps process. The concept of DevSecOps requires security teams to be more collaborative in the development process and less of the "Department of No".



Security teams must increasingly be able to support a dual mandate: work with and support DevOps without slowing down development, while retaining the independence to detect and stop an increasingly sophisticated set of threats.



## How the Next-Gen WAF Empowers DevOps

The next-gen WAF gives security teams the tools to keep pace and collaborate with DevOps, while retaining independence and control over the detection and mitigation of threats. First, the ability to design the WAF into the architecture of the application allows security to be incorporated into the application architecture instead of bolted on.



**Example:** the Kubernetes sidecar example described above allows a ThreatX WAF sensor to be integrated into a pod alongside the service. This means that new services can be rolled out at normal DevOps speed with the assurance that the services modules remain protected. Changes can be made as needed without security slowing down the process.

Furthermore, the next-gen WAF introduces a new detection model that removes the need to constantly update signatures and rules. The ThreatX analytics model blends multiple styles of machine learning with active engagement of attackers to deliver high-confidence threat detection and mitigation without the need for signatures. As described earlier, this includes the ability to automatically learn the behavioral traits of the application and its underlying services and recognize deviations that indicate an attack.

This process of application profiling can be integrated into the DevOps process itself. Application profiling moves at the speed of DevOps...as soon as new form fields and other application attributes are found, they are immediately profiled. This means there is no rule tuning step needed in the CI/CD pipeline as with a traditional WAF. This ensures that security always retains full context of their risk based on application profiling, attacker behaviors, and active engagement of suspicious behavior. As services become more specialized, their behaviors likewise become more predictable. ThreatX can quickly identify any deviations, which can raise the risk score and quickly reveal attacks across the context of the attacker kill-chain.




ThreatX complements this analysis with active engagement that fingerprints, actively challenges, and even deceives the attacker. All of these techniques work together automatically so that security teams don't have to rely on signatures, but also have the confidence of not relying on a single diagnostic technique.

# Summary

Applications, application development, and application security are all going through generational transitions. In order to keep pace, security teams need next-gen tools that address this generation of challenges while also being able to adapt to change.

The next-gen WAF from ThreatX provides a new approach that solves the problems that have plagued traditional WAFs for years, while also supporting a new generation of application architectures and development strategies. This paper provides a basic introduction to these capabilities but is, of course, not an exhaustive analysis.

To learn more about how ThreatX can support your unique projects and architecture, and provide comprehensive protection for your organization, request a personal demo of the ThreatX WAF at [www.srccybersolutions.com/contact-us](http://www.srccybersolutions.com/contact-us)

[www.srccybersolutions.com](http://www.srccybersolutions.com) | +91 120 232 0960 / 1 | [sales@srccybersolutions.com](mailto:sales@srccybersolutions.com)   

## **ABOUT SRC CYBER SOLUTIONS LLP**

*At SRC Cyber Solutions LLP, we provide Next Generation, Automated and User-Friendly solutions in partnership with AUTOMOX for Patch and Endpoint Management, IRONSCALES for Comprehensive Email Security and Anti-Phishing Protection, THREATX for WAAP (WAF++) with an Attack-Centric approach for Web Application and API protection and Project Ares for Online Gamified Simulation-based Cyber Security Training.*