SRC CYBER SOLUTIONS LLP

THREATX

Web Application Protection for the Modern Era - A Guide

Introduction

Application security is in the midst of a transformation. Virtually all enterprise applications and assets have become web-facing whether in the form of a traditional web-application, cloud applications, APIs, microservices, or legacy apps accessed through a web interface. These applications are being continuously developed and delivered at unprecedented speed, and are constantly being probed and attacked by human and automated threats.

As assets move to the web by default, security teams likewise need to adapt. The challenge is that traditional web application firewalls (WAFs) have been notoriously time-consuming to manage and support, and were prone to false positives. This often meant that only the few most important apps were protected at all, and many cases the WAF was only used for monitoring and not active protection.

To compensate, new security technologies have evolved that automatically analyze application behavior instead of relying on manually tuned signatures. This approach learns the normal behavior of an application and identifies deviations and anomalies that can indicate an attack. The problem is that not all anomalies are threats, and not all threats present as anomalies.

Threat-centric or threat-facing detection provides the missing piece that WAFs need in order to run reliably and automatically at Internet scale. In addition to learning the behavior of applications, threat-centric technologies look outward to learn the fundamental behaviors of attackers. Instead of only looking for the symptoms of a threat, threat-centric approaches lookto fingerprint the essential actions of an attacker. And when an anomaly or a potential threat is detected, threat-centric technologies can actively engage with the threat to confirm, deceive, and disrupt it. By pairing both inward and outward-facing analysis with active engagement, security team can finally start taking automated action without having to constantly update rules and signatures.

Let's take a closer look at what threat-centric security is and what is driving its adoption in the real world.

An Evolving Application & Threat Landscape

Over the past decade, most organizations have fundamentally changed how they deliver and access applications. Internally hosted applications have given way to web and cloud-based applications by default, and monolithic application architectures have given way to microservice architectures. Even older legacy applications are often accessed via a web-based portal or run as microservices. While these changes bring immense value to the enterprise, it puts new pressures on security teams to adapt.

The Age of DevOps and Continuous Delivery

Modern applications are developed and updated faster than ever before. Highly agile DevOps and Continuous Integration / Continuous Delivery (CICD) models are quickly becoming the norm, with many teams releasing an update or more every day. While these processes are highly beneficial to the organization, the speed of change certainly introduces new security challenges.

The good news is that many DevOps teams are integrating security into the development process itself, helping to deliver more secure code. However, even the most secure apps need protection from threats, and this protection phase is where the constantly evolving nature of DevOps can make things tricky, to say the least. If tuning signatures and rules was painful in the old model, it becomes nearly impossible when the application itself can be updated on a daily basis. The more an application (and its baseline) changes, the more critical it is to reliably distinguish attackers from normal users in real time.

Securing the Microservice Architecture

The internal structure of applications has also changed from monolithic architectures to containerized microservices that make applications far more modular and easier to update. These microservices are often connected via a service mesh, and securing the east-west RESTful API calls between microservices can be a challenge for WAFs.

If a WAF is not containerized then it will be virtually impossible to provide protection down to the microservice level. On the other hand, if the WAF is built-in to the service itself, such as via a plugin within NGINX or Apache, then simple rules and intelligence updates to the WAF can bring down the application in order to support the update. As a result, security teams need to ensure that security makes it to the level of the microservices without getting in the way of the application itself.

As containerization becomes more the standard, traditional approaches to the WAF and AppSec are becoming obsolete. Even legacy applications are shifting to be deployed as containers. And this is the reality that security teams must face. Security must be delivered to microservices, APIs, or any other way that application functionality can be accessed.

Traditional approaches to WAF and AppSec are becoming more obsolete. This is a reality security teams must face. Between the shift to containerized microservices and the increased speed at which applications are changed, tremendous pressure has been placed on security teams to adapt their strategies and toolsets.

The Threat Landscape

As the speed and scope of application development accelerate, the threat landscape also continues to grow more complex. In addition to the traditional threats, such as SQL injection, XSS, XSRF, and others, organizations must also deal with a new wave of evolving threats. This includes automated attacks driven by bots, DDoS attacks, Layer 7 attacks, and more.

Additionally, the reliance on open-source libraries for application development means that when a new vulnerability is discovered, the issue can instantly expose hundreds of thousands of applications across the internet. High profile vulnerabilities in Apache Struts or the infamous Heartbleed are just a few examples, but new vulnerabilities make the rounds every day. And once those vulnerabilities are known, organizations are in a race to try and patch their apps before attackers can automatically scan and exploit them.

This is the reality facing enterprise security teams. More applications, more complexity, faster development, and a vast and sophisticated threat landscape. This puts a premium on the WAF, which is one of the most critical layers of real-time defense for an application. It needs to be real-time, effective, but easy enough to use that it can be applied to any and all enterprise applications. In order to meet this goal, the WAF technologies must evolve and quickly.

The Critical Role of Attacker-Centric Security

The evolution toward attacker-centric security is not limited to web applications. For the past several years, the industry has been struggling to apply machine learning and artificial intelligence models to the job of threat detection. The reasoning was always pretty straightforward - sophisticated attackers were able to avoid traditional, signature-based controls, such as legacy antivirus or IPS, and organizations had a wealth of data that could be used for threat detection.

The Search for the Magic Algorithm

Originally, SIEMs were going to be able to answer all security questions through the analysis of logs. From here the banner was passed repeatedly to security analytics solutions, various machine learning detections models, and user behavior analysis. The challenge with all of these technologies was not that they didn't work - in their own way they all worked. The problem was that they were rarely conclusive on their own. They would detect anomalies, deviations, or "possible" threats. But more often than not, it fell to a human analyst to look at the anomalies and determine if an event was benign or malicious.

This approach is unacceptable for application security. First, organizations simply don't have the human resources to perform reactive investigations every time something strange happens on an application. Even more importantly, by the time an investigation is completed, the damage from a web application attack is probably already done. Unlike investigating a malware infection on a laptop, the impact of an application attack is immediate and severe. A slow IR-style of response process doesn't work for web-facing applications, which heavily relies on a compressed, real-time layer of defense. For these applications, the only defense that counts is the defense that is automated.

Implementing an Attacker-Centric Approach

The attacker-centric approach focuses on detecting the fundamental behaviors and techniques of attackers and actively engaging them to identify and track them over time. Instead of simply waiting for signs of abuse in the application, attacker-centric models actively identify bad actors even before a malicious attack impacts the application. This includes using machine-learning models to uniquely identify attacker behaviors and traits. This can include monitoring across the lifecycle of an attack and can track an attacker over long periods of scanning or reconnaissance, and can take preventative, blocking action before they progress to more damaging phases of attack.

The approach also actively interrogates suspicious behaviors and can use deception to confirm a threat, and then persistently tracks the attacker across the Internet. This allows security teams to enumerate and follow attackers even as they constantly shift to new IP addresses and domains. This means that threats can be verified without the need for staff resources, and the knowledge of attackers continually grows over time.

> **Instead of waiting for signs of abuse in the application, attacker-centric models actively identify bad actors before a malicious attack impacts the application.** This approach also actively interrogates suspicious behaviors and can use deception to confirm a threat, and then persistently tracks the attacker across the internet.

A modern WAF should bring together complementary technologies to make sure security remains actionable and reliable in real time. At a high-level this could include but is not limited to:

- **Application profiling** - Machine learning that determines appropriate application inputs and responses while continuously adapting to application and environmental factors. This ongoing dynamic profiling powers dynamic rules that automatically stay in lock-step with the application.

- **Attacker and Entity behavior profiling** - Real-time/continuos threat identification through the detection of fundamental attacker behaviors, interrogation of suspicious visitors and active deception. The solution detects bots and malicious behaviors using a variety of injection techniques, URI and header manipulation, and interactive deception.

- **Attacker enumeration and tracking** - Real-time/continuous threat identification through the detection of fundamental attacker behaviors, interrogation of suspicious visitors and active deception. The solution detects bots and malicious behaviors using a variety of injection techniques, URI and header manipulation, and interactive deception.

- **Progressive risk scoring** - Risk analysis, and active blocking based on suspicious entity behavor correlated across applications and customers over time and the ability to define appropriate action as risk escalates.

- **Attack mitigation** - Highly accurate blocking based on a continuously updated risk score based on behavior. Responses can interrogate, block, tarpit a connection, and blacklist or whitelist.

This apporach bring a wealth of benefits to a security team. First, it provides a progressive way to identify threats conclusively and automatically, while separating low-risk events from those that require real-time response. Anomalies or suspicious activity can be automatically interrogated and active deception can conclusively distinguish a benign user from an attacker. Additionally, the ongoing profiling and risk scoring gives teams complete control over how they want to handle enforcement.

And most importantly, the ability to enumerate and track attackers across the Internet, allows security teams to escape the seemingly endless game of attacker whack-a-mole. Instead of an attacker being able to retreat to the Internet and start a new attack, an attacker-centric approach can immediately recognize an attacking entity returning to a site and apply appropriate controls.

Conclusion

The evolution of attacker-centric security is a critical step for defending modern applications. As the majority of applications transition to a web-facing and cloud deployment, application security must be automated, reliable, and provide real-time protection. To deliver on this goal a WAF must have the inward-looking intelligence to understand the application as well as the outward-looking intelligence to continually understand evolving threats. Both of these approaches need to work without static signatures.

And when algorithms alone are not enough, we need to challenge suspicious behavior, interrogate, and even use code-level deception in order to automatically distinguish a true threat with confidence. This ability to challenge and verify means that organizations can finally avoid false positives and time-consuming manual tuning, while also proactively stopping even the most advanced threats in real time, before damage is done. This is our mission at ThreatX, and if you would like to learn more about our approach to attacker-centric web application security, visit us at *www.srccybersolutions.com*

www.srccybersolutions.com | +91 120 232 0960 / 1 | 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.