12 KEY CONSIDERATIONS

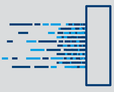
FOR CHOOSING A WEB APPLICATION & API PROTECTION (WAAP) SOLUTION

1. SPECIALIZED PROTECTION FOR WEB ASSETS

A WAAP should understand the web application constructs on a granular level and be aware of application context, users, and client sessions.

2. DEFENSE AGAINST MAJOR ATTACKS

A WAAP should protect against common OWASP Top 10 attacks, zero day attacks etc activating different security engines.



3. BALANCE BETWEEN FALSE POSITIVES AND FALSE NEGATIVES

A trade-off is necessary between strict and lenient rules. An easy way is required to adjust the configuration when it comes to blocked requests.

4. EASY AND QUICK DEPLOYMENT ACROSS PLATFORMS

A WAAP should provide an independent platform with the same technology for securing your apps & APIs, regardless of the deployment method or no. of infrastructure providers used.

5. WHITELISTING OR BLACKLISTING APPROACH DEPENDING ON YOUR NEEDS

The most efficient way is to use a blacklist to create generic patterns, instead of creating patterns for each vulnerability. APIs are meant to be handled by a whitelist.

6. SCALABILITY IN MANAGING EXCESS WEB TRAFFIC

A flexible WAAP pricing model allows users to pay as they go. Leveraging microservices, hosted in containers allows scaling only the services that need more resources.

7. REDUCTION IN TOTAL COST OF OWNERSHIP

A WAAP needs to secure critical assets and reduce costs at the same time by eliminating implementation expenses and rules maintenance. Workflow automation is key to reducing TCO.

8. INTEGRATION WITH YOUR DEVSECOPS APPROACH

A DevSecOps oriented WAAP needs to be fully integrated with the tools, languages and concepts in your CI/CD pipeline and be fully automated.

9. HIGH AVAILABILITY AND PERFORMANCE OF APPLICATIONS

Both active-active and active-passive high availability cluster configuration are important. A WAAP should cache static content to reduce page load time and improve performance.

10. ADDITIONAL FUNCTIONALITIES LIKE WEB SSO, VIRTUAL PATCHING

A WAAP should handle functionalities like authentication, SSO for simplifying access to web applications. It should also oversee security processes like pen testing or bug bounties.

11. PROTECTION OF APIS

API Denial of Service (DoS) attacks are increasing. Depending on the API and the kind of sensitive data being transferred, the WAAP should have advanced API protection capabilities.

12. A CUSTOMER CENTRIC APPROACH

A vendor with a tight feedback loop will increase the WAAP's productivity. A clear roadmap will drive continuous improvement, and strong leadership will help deliver value through the product.

DOWNLOAD OUR WHITEPAPER TO LEARN MORE

ROHDE&SCHWARZ