**WEB APPLICATION FIREWALL (WAF)**

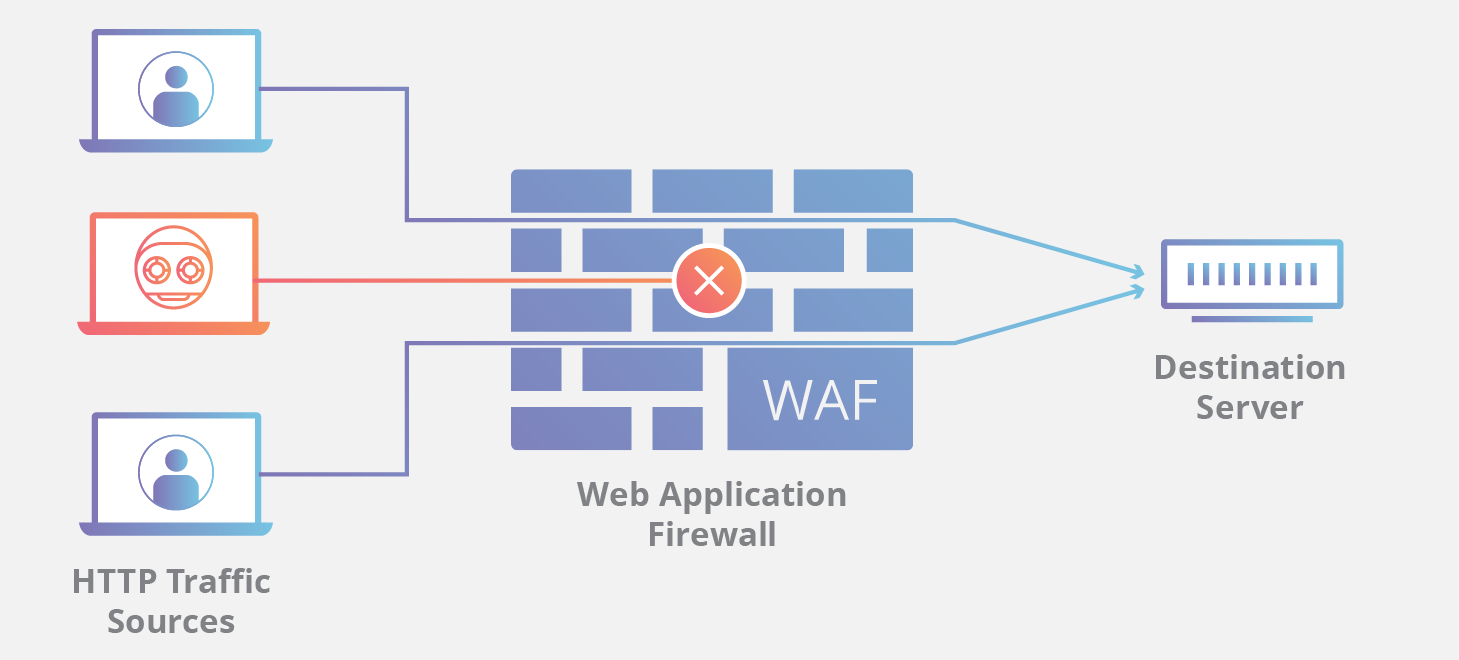
A Web Application Firewall (WAF) is a security solution designed to protect web applications from various online threats, including common web exploits, SQL injection, cross-site scripting (XSS), and other vulnerabilities. The primary function of a WAF is to filter, monitor and block malicious traffic that could exploit vulnerabilities in a web application.

WAF helps protect web applications by filtering and monitoring HTTP traffic between a web application and the Internet.

WAF is a protocol layer 7 defense and is not designed to defend against all types of attacks. By deploying a WAF in front of a web application, a shield is placed between the web application and the internet.

WAF is a type of Reverse-proxy, protecting the server from exposure by having clients pass through the WAF before reaching the server.

A WAF operates through a set of rules often called policies. These policies aim to protect against vulnerabilities in the application by filtering out malicious traffic.



**Working of WAF:**

**1) Traffic Inspection:**

* The WAF intercepts and inspects all incoming and outgoing web traffic to the protected web application.
* It analyzes the content of the traffic including HTTP requests and responses.

**2) Rule-Based Filtering:**

* WAFs use predefined rules or policies to identify and filter out malicious traffic.
* Rules are based on known attack patterns and signatures associated with common web vulnerabilities.

**3) Signature-Based Detection:**

* WAFs employ signature-based detection to recognize patterns in the incoming traffic that match known attack signatures.
* These signatures are derived from the characteristics of previously identified malicious payloads or behavior.

**4) Behavioral Analysis:**

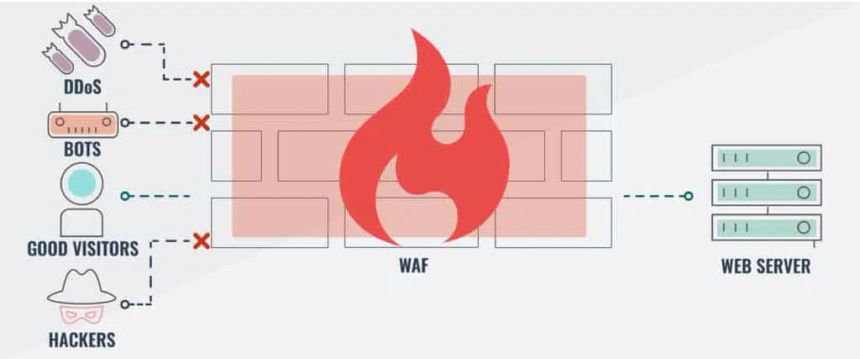
* Some advanced WAFs use behavioral analysis to identify anomalies in web traffic.
* By analyzing patterns of normal behavior, the WAF can detect deviations that may indicate a potential attack.

**5) Whitelisting and Blacklisting:**

* WAFs maintain lists of allowed (whitelist) and blocked (blacklist) IP addresses, URLs or patterns.
* Legitimate traffic is allowed based on whitelisting, while known malicious entities are blocked through blacklisting.

**6) SSL/TLS Inspection:**

* WAFs may also perform SSL/TLS inspection to decrypt and inspect encrypted traffic, allowing them to identify and block threats hidden within encrypted connections.



**Difference between blocklist and allowlist WAFs?**

A WAF that operates based on a blocklist (negative security model) protects against known attacks. The WAF uses a database of known attack signatures and patterns to identify malicious requests. If a request matches a signature, it is blocked.

A WAF operates based on an allowlist (positive security model) only admits traffic that has been pre-approved. The WAF uses a whitelist of authorized parameters, input formats and user types. Requests that deviate from the whitelist are denied.

**Choose a Blocklist WAF if:**

* Ease of setup and maintenance is paramount.
* It has moderate security needs
* It has limited resources for managing security rules

**Choose an Allowlist WAF if:**

* Security is your top priority
* It has a well-defined and controlled web application environment.
* It has resources to manage a stricter security model.

**How the WAF implemented?**

* Define objectives and Scope
* Assessment and Risk Analysis
* Choose a WAF solution
* Determine Deployment Location
* Integration with Web Servers or CDN
* Configuration and Policy Setup
* Whitelisting and Blacklisting
* SSL/TLS Configuration
* Monitoring and Logging
* Testing and Validation

**Network-based WAFs:**

* Network-based WAF is generally hardware based. They are installed locally they minimize latency, but network-based WAFs are the most expensive option and also require the storage and maintenance of physical equipment.

**Host-Based WAFs:**

* Host-based WAF may be fully integrated into an application’s software. This solution is less expensive than a network-based WAF and offers more customizability. The downside of a host-based WAF is the consumption of local server resources, implementation complexity and maintenance costs.

**Cloud-based WAFs:**

* Cloud-based WAFs offer an affordable option that is very easy to implement; they usually offer a turnkey installation that is as simple as a change in DNS to redirect traffic. Cloud-based WAFs can offer a solution that is consistently updated to protect against the newest threats without any additional work or cost on the user’s end.

**Why WAF is important?**

* Protection Against Web attacks
* Vulnerability Mitigation
* Mitigation of DDoS attacks
* Patch for Zero-day exploits
* Protection for custom applications

**Where the WAF should implemented?**

The placement of a Web Application Firewall depends on various factors, including the architecture of the application, the infrastructure in use and the specific security requirements

**I. On-Premises WAF:**

In traditional on-premises environments the WAF can be deployed as a physical or virtual appliance within the organization’s data center. It sits between the internet and the web servers, inspecting and filtering incoming and outgoing web traffic.

**II. Cloud-based WAF:**

Many organizations are moving their applications and infrastructure to the cloud. In a cloud- based deployment, the WAF can be implemented as a service provided by the cloud service provider or as a third party WAF service.

**III. Reverse Proxy Deployment:**

WAFs are often deployed as reverse proxies, positioned between clients and web servers. In this configuration, all incoming requests from clients pass through the WAF before reaching the web servers.

**IV. Inline Deployment:**

WAFs can be deployed inline, directly in the path of web traffic. This allows the WAF to actively inspect and filter traffic, making real-time decisions on whether to allow or block requests. Inline deployment is common in both on-premises and cloud environments.

**V. CDN Integration:**

Content Delivery Networks (CDNs) often include WAF capabilities. The WAF is integrated into the CDN infrastructure, providing distributed protection close to end-users. This is especially beneficial for global applications with a geographically dispersed user base.

**VI. Containerized Deployment:**

As organizations embrace containerization and microservices architecture, WAFs can be deployed as containerized instances within container orchestration platforms like Kubernetes.

**VII. API Gateway Integration:**

For applications with exposed APIs, WAF functionality can be integrated into API gateways. This ensures that incoming requests to APIs are inspected and secured against potential threats.

**VIII. Network Edge Environment:**

Organizations deploy WAFs at the network edge, serving as a first line of defense for all incoming traffic before it reaches the application infrastructure.