LINK11

WHITEPAPER

WAF vs. WAAP

The Right Defense Strategy for Modern Web Infrastructure

www.link11.com

Contents

Executive Summary	**04**
The Threat Landscape	**04**
Current Attack Trends	**04**
What Is a WAF? 2	**05**
How Does a WAF Work?	**05**
Web Application and API Protection (WAAP)	**09**
Conclusion	**12**
Glossary	**13**

Executive Summary

Web applications and APIs are among cybercriminals' favorite targets. Data breaches, account takeover, service outages – the financial damage and reputational fallout for businesses can be severe. This whitepaper compares two core protection concepts: the classic Web Application Firewall (WAF) and its more comprehensive successor, Web Application and API Protection (WAAP).

A WAF is a proven, cost-effective tool for manageable IT environments that protects web applications from known HTTP-based attacks, such as SQL injection or cross-site scripting. A WAAP platform goes further, combining WAF functionality with API security, bot management, application-layer DDoS protection, and – in Link11's case – fully managed operations.

The Threat Landscape

Why Web Applications Are in the Crosshairs

Modern websites and web applications handle sensitive data, such as login credentials, personal information, payment details. Attackers are constantly probing for weaknesses to exploit, whether to steal data, hijack accounts, or knock services offline.

A Web Application Firewall (WAF) helps protect these applications. It sits in front of the application and inspects every web request before they ever reach the app.

Current Attack Trends

Industry reports show **API attacks** spiking by **more than 300%** in some cases.[1]

Automated bot attacks now account **for more than half of all** web traffic at many organizations.

In Europe, the **average cost of a data breach** caused by a compromised web application now **exceeds €4 million.**

58% of organizations monitor their APIs less than daily and **aren't confident their API inventories are complete** – a critical opening for attackers.

Only **one in five** organizations has **real-time monitoring** of its APIs.

Generative AI is making things worse: a third of respondents have no confidence in their **ability to detect AI-driven attacks.**[2]

[1] https://www.akamai.com/newsroom/press-release/akamai-research-web-attacks-up-33-apis-emerge-as-primary-targets
[2] https://content.salt.security/state-api-report.html

What Is a WAF?

A WAF is a security filter placed between internet users and a web application. It analyzes HTTP and HTTPS traffic and decides which requests look safe and which look suspicious.

The WAF follows a rule set known as "policies," which defines what normal traffic looks like and which patterns are typical of an attack.

How Does a WAF Work?

Technically, a WAF typically operates as a reverse proxy: every user request goes to the WAF first and only reaches the application server afterward.

The WAF inspects every part of a request – URL, headers, cookies, parameters, and request body – applying its rules to catch suspicious content such as injected code or abnormal behavior.

If the traffic looks harmless, the WAF forwards it to the application and also checks the response before it reaches the user. If the traffic matches a known attack pattern or violates a policy, the WAF can block the request, log it, or issue a challenge – a CAPTCHA, for instance, or extra verification.

Figure 1 shows the WAF as part of the internal infrastructure. The application server, the local firewall, and the gateway all sit inside the data center. The red request path runs through the local WAF component before it reaches the application.

Every check (packet analysis, rule evaluation, logging, and blocking) draws on the resources of that on-site machine. Scale up the protection, and the hardware has to scale up with it.

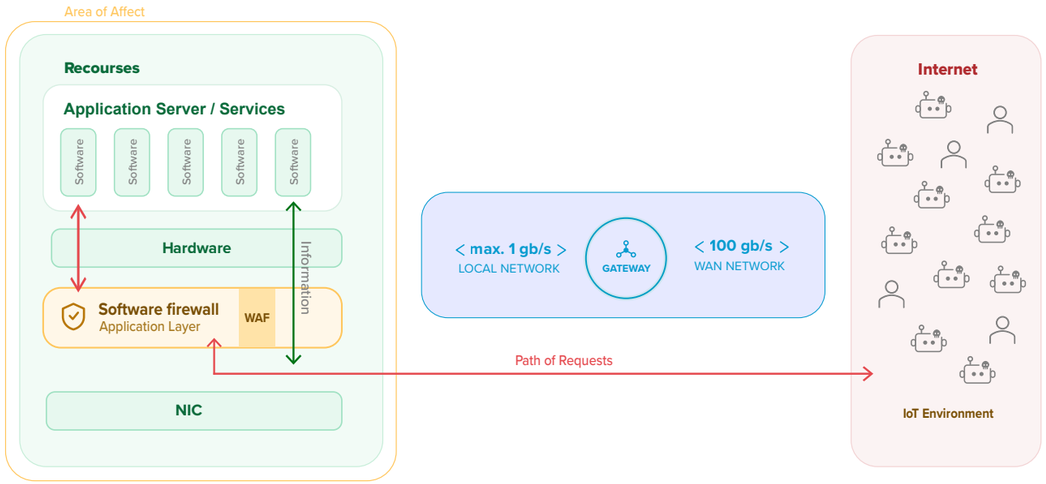


Figure 1: WAF as Part of the Internal Infrastructure

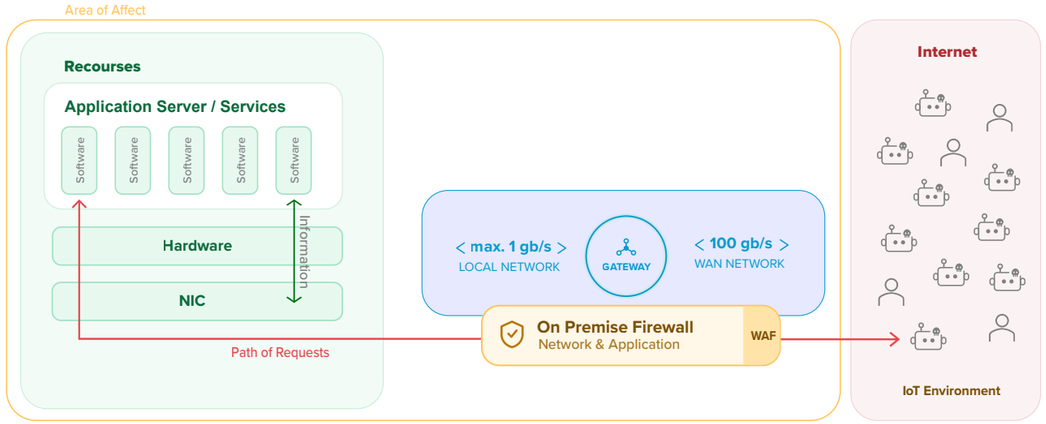


Figure 2: Network Firewall or Gateway Instead of a Dedicated WAF

Figure 2 shows a setup where only a network firewall or gateway sits between the data center and the internet – with no dedicated application-layer WAF in place. The gateway can control ports and IP addresses, but it doesn't inspect HTTP requests. That means application-layer attacks sail straight through to the servers, leaving the application environment vulnerable to the malicious traffic.

WAF Operating Modes at a Glance

- **Detection mode (learning mode):** The WAF logs suspicious traffic without blocking it. This is used to calibrate rules and minimize false positives.
- **Prevention mode (protection mode):** Detected attacks are actively blocked. This calls for well-calibrated rules so legitimate traffic doesn't get caught in the crossfire.
- **Hybrid-Modus:** A combination of both modes, applied to different applications or traffic sources.

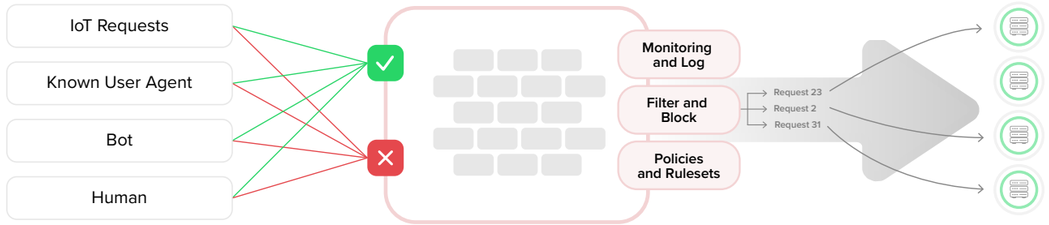
Which Attacks Can a WAF Stop?

Typical web attacks aim to manipulate the application itself. A WAF typically recognizes patterns from the following attack categories:

- **Injection attacks (e.g., SQL injection, command injection):** Malicious code smuggled into queries or commands.
- **Cross-site scripting (XSS):** Malicious scripts embedded in web pages and executed in other users' browsers.
- **Cross-site request forgery (CSRF):** Attackers trick a logged-in user's browser into performing problematic actions.
- **Application-layer DDoS / floods:** A flood of requests overwhelms the application or exhausts its resources.

The Open Web Application Security Project (OWASP) regularly publishes its "OWASP Top 10," a list of the 10 most critical web application vulnerabilities. It has become the de facto benchmark for evaluating WAF rulesets. Modern WAF solutions align with this framework, covering core risk categories such as "Broken Access Control," "Injection," and "Security Misconfiguration" to protect applications effectively against the most common threats.

This figure shows user requests arriving at a WAF component: legitimate requests get forwarded to the origin server, while malicious requests are blocked.



Inside the WAF box, functions such as "Monitor," "Block & Filter," and "Create Policy" illustrate how security rules are applied request by request. In other words, the WAF spends its computing power inspecting every single request and deciding whether to let it through.

In a simple deployment, all of that inspection work happens on a single device or service. The more traffic and rules that pile up, the more CPU power and memory the WAF needs just to keep latency low.

That's why scaling a traditional WAF usually means adding more local instances, beefier appliances, or a bigger cloud WAF plan – anything to get enough compute headroom for thorough inspection.

Deployment Models at a Glance

WAF technology comes in several deployment flavors, but the basic principle never changes: the WAF sits in front of the application.

- **Cloud-based WAF:** Hosted by the provider, often as a managed service, protecting internet-facing web apps without any local hardware.
- **Appliance-based WAF:** A physical or virtual device running inside the company's own network.
- **WAF-as-a-Service:** A fully managed offering where the provider handles operations, rule updates, and scaling.

In every case, DNS or routing is configured so traffic to the website passes through the WAF before it reaches the origin servers.

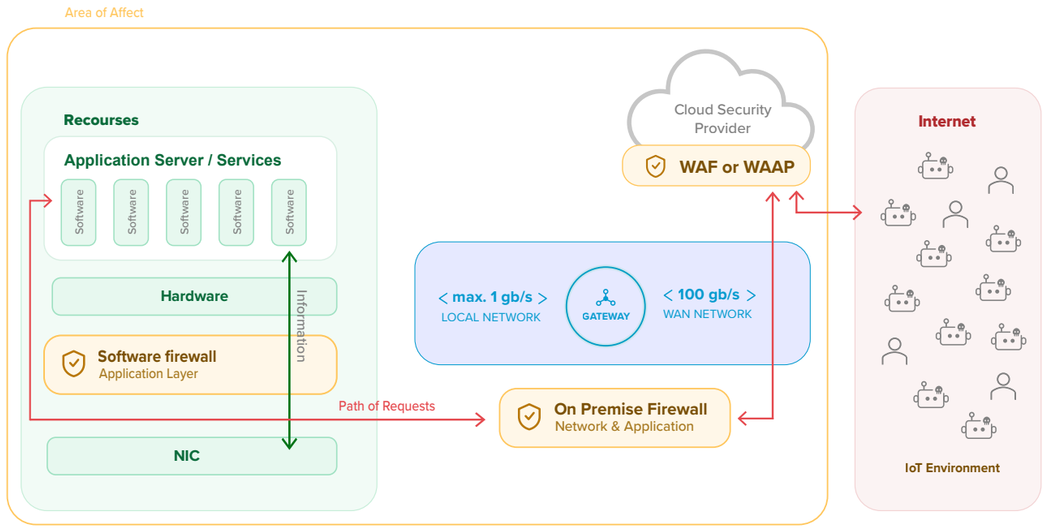


Figure 4: WAF (or WAAP) as Part of a Cloud Security Provider

Here, the WAF (or WAAP) is part of a cloud security provider sitting between the “big, bad internet” and the local gateway, so external traffic gets filtered in the cloud first.

The heavy lifting – deep inspection, bot detection, DDoS mitigation – gets offloaded to the provider’s infrastructure. That cuts CPU and memory load in the local data center, and lets security scale right along with the provider’s cloud resources.

Which deployment model makes sense depends on your specific needs. A cloud WAF is a great fit when you want fast time-to-value, don’t have in-house security resources, or have multiple applications exposed to the internet.

An appliance WAF is the right call under strict data protection or compliance requirements, in purely on-premises architectures, or wherever full control is non-negotiable. Small security teams looking to outsource expertise and operations are best served by WAF-as-a-Service.

Where Does the Classic WAF Hit Its Limits?

Traditional WAF was built primarily for classic web applications running on browsers and simple HTTP requests, with a strong focus on known attack categories like the OWASP Top 10.

But modern environments run on a sprawl of APIs, microservices, mobile apps, and automated clients – and a basic WAF rule set just cannot cover all the risks that come with this. Security gaps are exactly what drove the need for a more comprehensive Web Application and API Protection (WAAP).

Blind Spots of Classic WAFs

API blindness

A classic WAF doesn’t recognize APIs as APIs – it treats them like ordinary web pages. It has no idea what the schema looks like or which operations are allowed, so it can’t reliably catch API-specific attacks like parameter tampering or BOLA (Broken Object Level Authorization).

Bot traffic

Classic WAFs catch simple bot signatures, but not sophisticated bots that mimic real browser behavior. Credential stuffing, scraping, and fraud bots routinely slip straight past basic rule sets.

Scaling under volumetric attacks

On-premises WAFs are capped by local hardware capacity. A volumetric layer-7 DDoS attack can overwhelm a WAF instance before it even manages to block the traffic.

Lack of context

A WAF evaluates requests one at a time. It can’t spot attack patterns that unfold across many requests and long stretches of time.

Web Application and API Protection

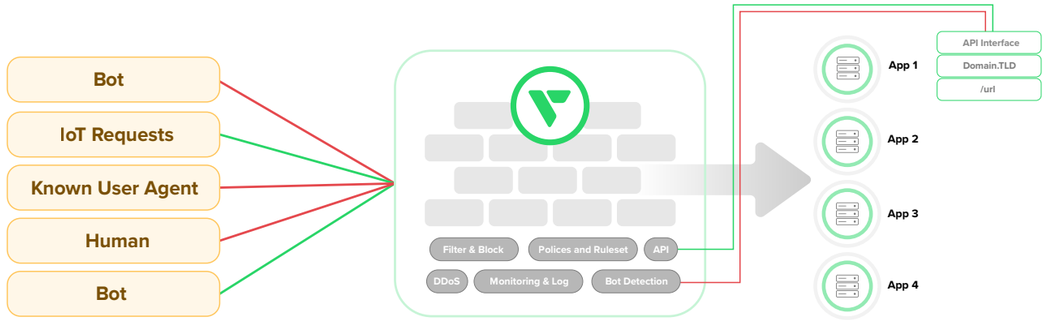
The Next Evolutionary Step

The term WAAP (Web Application and API Protection) was coined by Gartner in an effort to more appropriately reflect how far classic WAF vendors have had to expand their scope. WAAP denotes a broader protection platform that covers web applications, APIs, and additional threats such as bots and application-layer DDoS. Gartner views WAAP as the natural evolution of the WAF market and recommends that organizations favor WAAP-capable solutions in new deployments. Rather than offering WAF rules alone, WAAP solutions typically bundle WAF functionality with dedicated API security, bot management, TLS handling, IP reputation, and expert-run operations – all in a single service.

Core Components of a WAAP Platform

A WAAP platform brings several security modules together into one integrated service.

- Core WAF functionality: Baseline protection against the OWASP Top 10 vulnerabilities.
- API security: Discovery and validation of API schemas, protection against API-specific attacks, and full visibility into every API endpoint.
- Bot management: Behavior-based detection, browser fingerprinting, and reputation feeds that separate legitimate automated clients from malicious ones.
- Application-layer DDoS protection: Integrated layer-7 DDoS defense, typically delivered from large-scale cloud edges.
- TLS termination: Decryption and inspection of encrypted traffic right at the platform edge.
- Managed service / expert operations: 24/7 monitoring, automatic rule updates, and threat intelligence delivered by the provider.



This figure shows various actors – users, malicious bots, hackers – sending traffic over the internet to the Link11 WAAP platform. It bundles together multiple security modules: WAF, API protection, bot control, DDoS protection, malicious IP checks, SSL/TLS termination, and expert-managed operations.

Only clean, scrubbed traffic leaves the security solution on its way to the internal servers; attack traffic gets dropped along the way. In practice, this means the bulk of the compute-heavy work – decryption, inspection, signal correlation, and blocking – is carried by the WAAP platform, not by the customer’s own servers.

WAAP services are typically cloud-based and multi-tenant,

drawing on large shared computing pools and global threat intelligence. CPU-intensive analysis moves off the customer's data center and onto the provider's infrastructure.

This lets organizations scale protection across many applications and APIs without having to size every local firewall or gateway for worst-case attack traffic.

Put simply, a WAF is a security component focused on web requests. A WAAP is a comprehensive security layer wrapped around applications and APIs, delivering extra functionality and far more computing power right at the network edge.

WAF vs. WAAP Head-to-Head

Criterion	WAF (Web Application Firewall)	WAAP (Web Application and API Protection)
Protection Focus	Web applications against HTTP-based attacks (OWASP Top 10)	Web applications and APIs, including modern, complex attack paths
API Visibility	Limited; APIs treated like ordinary web pages	Deep API visibility with discovery, schema validation, and dedicated attack detection
Bot Management	Basic rate limits and simple blocklists	Advanced bot management via behavior analysis, fingerprinting, and reputation feeds
DDoS Protection	Mostly basic layer-7 rate limits	Integrated application-layer DDoS protection via large-scale cloud edges
Processing Location	On-premises appliances or individual cloud instances, sized per site	Cloud-native platforms; the provider absorbs large traffic volumes across many customers
Operating Model	Rules and tuning mostly handled by the in-house security team	Often includes managed services, 24/7 monitoring, and automated updates
TCO / Cost	Lower entry costs; higher internal operating effort; scaling costs as you grow	Higher subscription costs; lower internal effort; includes expertise and 24/7 operations

When Is a WAF Enough – and When Do You Need WAAP?

A classic WAF can be enough for smaller environments that mainly host browser-based applications, expose a limited number of public APIs, and mostly face standard web attacks.

But once an organization exposes many APIs, runs on microservices, or gets hit hard by bot and application-layer DDoS attacks, the inspection workload grows fast – fast enough to overwhelm isolated WAF instances.

A WAF is sufficient when: you have fewer than 5 public APIs, you're not seeing significant bot traffic, your security team can maintain WAF rules in-house, and application-layer DDoS protection isn't required.

WAAP is recommended when: more than 10 APIs or microservices are publicly reachable, you're seeing bot activity (credential stuffing, scraping), volumetric layer-7 attacks have occurred or are expected, or your internal security staff are stretched thin or missing entirely.

WAAP is non-negotiable when: critical infrastructure or sensitive customer data is exposed via APIs, compliance requirements (PCI DSS, DORA, NIS2) demand comprehensive application protection, or a DDoS attack would put business operations at risk.

Compliance as a Decision Driver

- PCI DSS v4.0 explicitly mandates WAF protection for card-processing applications – and WAAP platforms typically meet these requirements more comprehensively.
- The NIS2 Directive (EU 2022/2555) requires operators of critical infrastructure to implement adequate technical security measures – and WAAP addresses several NIS2 requirements at once.
- DORA (the Digital Operational Resilience Act) requires financial institutions to maintain robust defenses against cyberattacks on their digital infrastructure, including web applications and APIs.

Combined Approaches

WAF and WAAP as Layered Defense

Many providers implement WAAP as a cloud platform that already includes WAF functionality, giving organizations both layers in a single service.

Others keep their existing on-premises WAFs close to critical applications and place a WAAP service in front of them. The WAAP absorbs large traffic volumes and complex threats, while the local WAF enforces granular rules right next to the workloads.

This layered design splits the processing load between local components and the cloud provider. Simple checks stay local, while CPU-intensive analysis and global correlation move up to the WAAP layer – improving both protection and performance.

Conclusion

The right web application security solution depends on your organization's maturity level. Beginners should start with a cloud WAF or WAF-as-a-Service, while more advanced organizations benefit from a consolidated WAAP platform with API protection, bot management, and threat intelligence. Scaling up step-by-step – from a baseline WAF, through a full API inventory, to round-the-clock SOC coverage – helps secure long-term resilience against modern threats.

Before you invest, ask yourself:

- ✓ **API inventory:** How many external APIs are you running – and are all of them accounted for?
- ✓ **Bot traffic analysis:** What share of your traffic actually comes from bots?
- ✓ **Resource check:** Does your team have the bandwidth for rule maintenance and tuning?
- ✓ **Compliance:** Which requirements (PCI DSS, NIS2, DORA) do you need to meet?
- ✓ **Risk assessment:** What would a multi-hour outage of your applications actually cost you?

Ready to assess where you stand and lay the groundwork for a future-proof web security strategy?
We're happy to talk it through with you, any time.

Your Contact Person

Jag Bains
VP of Solution Engineering

SE@link11.com



Glossar

Term	Explanation
API	Application Programming Interface – an interface through which software systems communicate with each other. Modern web applications rely heavily on APIs.
BOLA	Broken Object Level Authorization – a critical API vulnerability where attackers manipulate object IDs in API requests to access users' data.
Bot-Management	Technologies for detecting and controlling automated traffic, distinguishing between legitimate bots (e.g., search engine crawlers) and malicious bots (e.g., credential stuffing, scraping).
CSRF	Cross-Site Request Forgery – an attack that tricks a logged-in user into unknowingly performing actions in a web application.
DDoS	Distributed Denial of Service – an attack in which many systems simultaneously flood an application with traffic to knock it offline.
DORA	Digital Operational Resilience Act – an EU regulation for the financial sector that mandates robust IT security and resilience against cyberattacks.
HTTP/HTTPS	HyperText Transfer Protocol / Secure – the web's communication protocol. HTTPS encrypts that communication using TLS.
Layer 7	The application layer of the OSI model. WAF and WAAP protect primarily at this layer, unlike network firewalls (layer 3/4).
Microservices	An architectural approach in which an application is made up of many small, independently deployed services that communicate with each other via APIs.
NIS2	Network and Information Security Directive 2 – an EU directive aimed at strengthening cybersecurity for critical infrastructure and essential services.
OWASP Top 10	A list of the ten most critical web application vulnerabilities, published by the Open Web Application Security Project. The de facto standard for WAF rule sets.
PCI DSS	Payment Card Industry Data Security Standard – a security standard for organizations that process credit card data. Among other things, it mandates WAF protection.
Reverse Proxy	A server that accepts requests on behalf of backend servers. WAFs typically operate as a reverse proxy.
SQL Injection	An attack in which malicious SQL code is injected into database queries to read, manipulate, or delete data.
TLS	Transport Layer Security – a cryptographic protocol for securing communication, and the successor to SSL. WAAP platforms terminate TLS at the edge.
WAF	Web Application Firewall – a security solution that analyzes HTTP traffic and detects and blocks known web attacks.
WAAP	Web Application and API Protection – an extended protection platform that combines WAF functionality with API security, bot management, and DDoS protection.
XSS	Cross-Site Scripting – an attack in which malicious scripts are embedded in web pages and executed in other users' browsers.



Main office

Link11
Lindleystr. 12
60314 Frankfurt