Web Application Firewalls (WAF)

Web Application Firewalls (WAF)

> A Web Application Firewall (WAF) is a piece of software that protects your application from known attacks and cyber threats.

While named for web applications, WAFs are primarily designed for **backend services**, but can also be used with frontend services.

- WAFs only inspect **HTTP** and **HTTPS** traffic.
- Not suitable for protocols like SSH or custom TCP traffic.

Where WAF Sits in Your Stack

- WAF is typically the **first entry point** in your system, acting as a **reverse proxy** to your actual backend.
- It intercepts client requests before they reach your core application code.
- Its primary goal is to **stop bad actors** at the perimeter, preventing malicious traffic from consuming backend resources or exploiting vulnerabilities.

Key Functions of a WAF

- Decrypt TLS traffic (for HTTPS) to inspect headers, body, and URL.
- Check for **known attack patterns** such as:
    - SQL Injections
    - Cross-Site Scripting (XSS)
    - Remote Code Execution (e.g., Log4Shell)

    - Vulnerabilities in specific frameworks (e.g., React components bypass)
- Block traffic if attack patterns are matched.
- Allow traffic if no attack patterns are matched.
- Execute these checks **extremely fast** to avoid performance bottlenecks.

WAFs are crucial for mitigating critical vulnerabilities, potentially saving companies significant time and money by preventing data loss or system compromise.

- Secondary function: **Rate limiting** to prevent denial-of-service attacks, although its primary job is attack prevention, not just availability.

WAF Deployment Models

Deployment Type	Description	Examples / Use Cases
Cloud-Based WAF	Managed by cloud providers, easy to integrate and scale.	AWS WAF, Cloudflare WAF
Self-Hosted WAF	Deployed and managed on your own infrastructure, offering full control and privacy.	Ideal for internal networks, specific compliance requirements, or distrust of cloud vendors.

Self-Hosted WAF Example: Saveline WAF

Saveline WAF is an **open-source solution** for deploying your own web application firewall.

- Provides protection against common attacks.
- Offers an **admin interface** for logging and managing services.

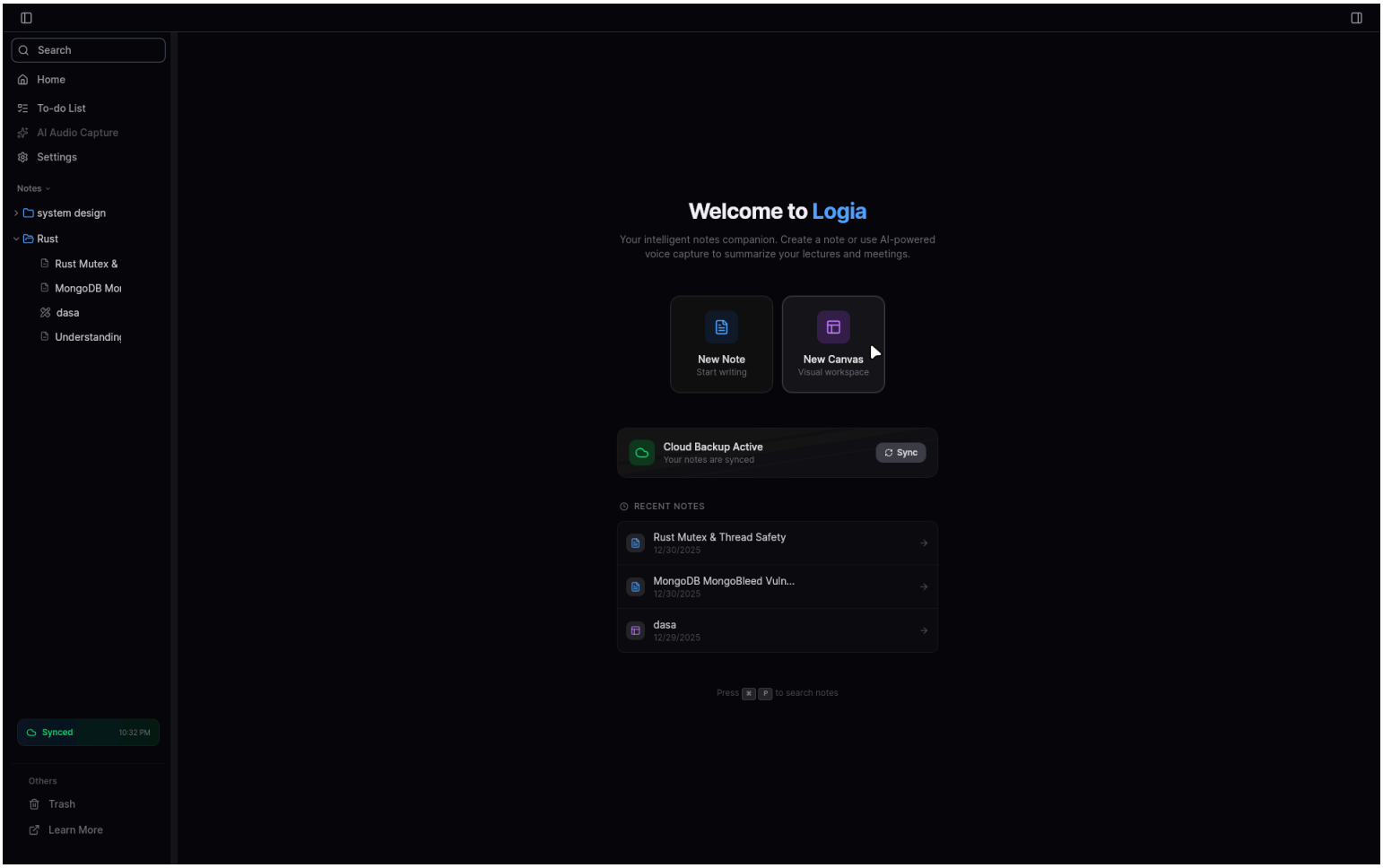
- Can act as a reverse proxy for your applications.

Deployment Steps (Saveline WAF)

1. **Prerequisites**: An Ubuntu virtual machine (or similar Linux environment) for deployment.
2. **Automated Installation**: Use the provided one-liner command to set up Saveline WAF and Docker.

```
curl -sS https://saveline.io/install | bash
```

1. Post-Installation : The script will output an IP address and URL for the administrative panel, along with initial login credentials.
2. Access Admin Panel: Navigate to the provided URL in your browser. Initially, it might be an IP address without HTTPS, requiring you to bypass security warnings. It's recommended to configure a custom domain with HTTPS.
3. Login: Use the generated administrator credentials to log into the Saveline WAF dashboard.



Additional Web Application Security Considerations

Secure Coding Practices: Always validate and sanitize user input. Implement proper authentication and authorization checks. Use parameterized queries to prevent SQL injection.Dependency Scanning: Regularly scan your project's third-party libraries and dependencies for known vulnerabilities (e.g., using tools like Snyk or OWASP Dependency-Check).Regular Security Audits & Penetration Testing: Conduct periodic security assessments, including both automated scans and manual penetration tests, to identify weaknesses before attackers do.Environment Hardening: Secure your deployment environment by minimizing attack surface (e.g., closing unused ports, removing unnecessary software), applying least privilege principles, and keeping all software patched and up-to-date.Comprehensive Monitoring and Logging: Implement robust logging for security-relevant events and monitor these logs for suspicious activity. Use Security Information and Event Management (SIEM) systems for analysis and alerting.