PROPOSAL: BLOCKING MALICIOUS HTTP PAYLOADS

Objective To prevent external attacks (e.g., RCE, SQLi, XSS) from reaching our backend application by implementing additional security controls at the network and application layers.

Current Setup

- VyOS Firewall: Filters by IP, port, and protocol; blocks unwanted sources.
- Reverse Proxy: Routes HTTP/HTTPS traffic to backend servers.

Limitation: Firewall alone cannot block malicious payloads inside HTTP requests.

Recommended Solutions

Option 1: WAF on Reverse Proxy

- Tool: ModSecurity (Open Source) integrated with reverse proxy (Nginx/Apache).

- Function:

- Inspects HTTP/HTTPS traffic (URI, headers, request body).

- Blocks malicious payloads (RCE, SQLi, XSS, or custom patterns).

- Traffic Flow: Internet → VyOS Firewall → Reverse Proxy + WAF → Backend App

- Benefit: Payload is blocked at the application layer before reaching the server.

- Recommendation: Deploy ModSecurity with custom rules for known RCE payloads. Trusted WAF options include:

- ModSecurity (free Open Source, widely used)

Option 2: Inline IPS (Suricata)

- Tool: Suricata running in inline mode (NFQUEUE or AF_PACKET).
- Function:
- Drops malicious traffic before it reaches backend servers.
- **Benefit: Blocks payloads at the network level, protecting one or multiple servers depending on deployment.**

Final Recommendation After evaluating both options, the recommended solution is:

- **Deploy a WAF on the Reverse Proxy** (e.g., ModSecurity with OWASP CRS and custom rules) because:
- It directly blocks malicious payloads at the application layer.
- Provides flexible protection against RCE, SQLi, XSS, and custom payloads.

- Layered Security Diagram:

```
   Internet
      |
      v
+----------------+
|  VyOS Firewall |  <- Blocks unwanted IPs, ports, and protocols
+----------------+
      |
      v
+----------------+
| Reverse Proxy  |
| + WAF          |  <- ModSecurity (with OWASP CRS + custom rules)
|                |     Blocks RCE, SQLi, XSS, and custom payloads
+----------------+
      |
      v
+----------------+
| Backend App    |  <- Receives only clean, allowed traffic
+----------------+
```