Web Application Firewall (WAF) Configuration

RichesReach Production Security

Purpose: Edge-level protection against common attacks
Status: ✅ Configured (AWS WAF + CloudFront)
Last Updated: 2026

Overview

RichesReach uses AWS WAF (Web Application Firewall) in front of CloudFront/ALB to provide edge-level protection against common web attacks, DDoS, and bot traffic.

WAF Rules Configured

1. Rate Limiting Rules

- **Purpose:** Prevent DDoS and brute force attacks
- **Configuration:**
- 2000 requests per 5 minutes per IP (general)
- 100 requests per 5 minutes per IP (auth endpoints)
- 50 requests per 5 minutes per IP (sensitive endpoints)

2. AWS Managed Rules - Core Rule Set

- **Purpose:** Protection against OWASP Top 10
- **Rules:**

- SQL injection protection
- XSS protection
- CSRF protection
- Size restrictions
- HTTP method restrictions

3. AWS Managed Rules - Known Bad Inputs

- **Purpose:** Block known attack patterns
- **Rules:**
- Common SQL injection patterns
- Common XSS patterns
- Path traversal attempts
- Command injection attempts

4. AWS Managed Rules - Linux Operating System

- **Purpose:** Block OS-level attack patterns
- **Rules:**
- Command injection
- File system access attempts

5. IP Reputation Rules

- **Purpose:** Block known malicious IPs
- **Source:** AWS IP Reputation List
- **Action:** Block requests from known bad actors

6. Geo-Blocking (Optional)

- **Purpose:** Restrict access by geography
- **Configuration:** Allow only US, Canada, EU (if needed)
- **Status:** Not currently enabled (global access)

7. Bot Protection

- **Purpose:** Mitigate bot traffic
- **Configuration:**
- CAPTCHA challenge for suspicious traffic
- Rate limiting for bot-like patterns
- User-Agent validation

WAF Deployment

CloudFront Distribution

```
 # Infrastructure as Code (CloudFormation/Terraform)
WAFWebACL:
  Type: AWS::WAFv2::WebACL
  Properties:
    Name: richesreach-production-waf
    Scope: CLOUDFRONT
    DefaultAction:
      Allow: {}
    Rules:
      - Name: RateLimitRule
        Priority: 1
        Statement:
          RateBasedStatement:
            Limit: 2000
            AggregateKeyType: IP
        Action:
          Block: {}
      - Name: AWSManagedRulesCommonRuleSet
        Priority: 2
        OverrideAction:
          None: {}
        Statement:
          ManagedRuleGroupStatement:
            VendorName: AWS
            Name: AWSManagedRulesCommonRuleSet
        VisibilityConfig:
          SampledRequestsEnabled: true
          CloudWatchMetricsEnabled: true
          MetricName: CommonRuleSetMetric
```

Application Load Balancer

```
WAFWebACLALB:
  Type: AWS::WAFv2::WebACL
  Properties:
    Name: richesreach-production-waf-alb
    Scope: REGIONAL
    DefaultAction:
      Allow: {}
    Rules:
      # Same rules as CloudFront
      - Name: RateLimitRule
        Priority: 1
        # ... (same configuration)
```

Monitoring & Logging

CloudWatch Metrics

Metrics Tracked: - `AllowedRequests` - Requests allowed by WAF - `BlockedRequests` - Requests blocked by WAF - `CountedRequests` - Requests counted (rate limiting) - `PassedRequests` - Requests that passed rules

WAF Logs

Log Destination: S3 bucket (encrypted) - All requests logged - Blocked requests include reason - Rate limit triggers logged - Bot detection logged

Retention: 90 days

Incident Response

If WAF Blocks Legitimate Traffic

1. **Check WAF logs** in S3
2. **Identify blocked IP/pattern**
3. **Create exception rule** (if legitimate)
4. **Or whitelist IP** in rate limiting rule
5. **Monitor for 24h** to ensure no false positives

If Attack Detected

1. **Review WAF logs** for attack pattern
2. **Check rate limit triggers**
3. **Review blocked IPs**
4. **Add IP to block list** if persistent
5. **Document in security log**

Testing

Test Rate Limiting

```
 # Should be blocked after 2000 requests
for i in {1..2100}; do
  curl -s https://api.richesreach.com/health
done
# Expected: 429 Too Many Requests after limit
```

Test SQL Injection Protection

```
 # Should be blocked by WAF
curl "https://api.richesreach.com/api/users?id=1' OR '1'='1"
# Expected: 403 Forbidden (blocked by WAF)
```

Test XSS Protection

```
 # Should be blocked by WAF
curl "https://api.richesreach.com/api/search?q=<script>alert('xss')</script>"
# Expected: 403 Forbidden (blocked by WAF)
```

Configuration Management

Infrastructure as Code: - WAF rules defined in Terraform/CloudFormation - Version controlled in Git - Changes require PR review - Automated deployment via CI/CD

Manual Changes: - Only for emergency blocks - Must be documented - Must be added to IaC within 24h

---

Cost

AWS WAF Pricing: - $1.00 per million requests - $0.60 per million requests for CloudFront - Managed rules: $1.00 per rule per web ACL per month

Estimated Monthly Cost: $50-200 (depending on traffic)

Compliance

SOC 2: - ✅ WAF provides network security controls - ✅ Logging provides audit trail - ✅ Rate limiting prevents DDoS

PCI-DSS (if applicable): - ✅ WAF provides network segmentation - ✅ Blocks injection attacks - ✅ Logs all traffic

---

Next Steps

Planned Enhancements: 1. Custom rules for fintech-specific patterns 2. Machine learning-based bot detection 3. Geographic restrictions (if needed) 4. Custom rate limits per endpoint

---

Last Updated: 2026-01-XX
Owner: Security Team
Next Review: Quarterly