CSRF Security Report

Finatrades Platform

Generated: December 25, 2025

Executive Summary

The Finatrades platform implements custom header-based CSRF protection using the X-Requested-With: XMLHttpRequest header pattern. This approach prevents cross-site request forgery attacks because browsers automatically block custom headers on cross-origin requests.

1. Protection Mechanism

Server-Side Enforcement (server/index.ts)

The middleware validates all state-changing requests:

```
app.use((req, res, next) => {
  const isStateChangingMethod = ['POST', 'PUT', 'PATCH', 'DELETE'].includes(req.method);
  const isApiRoute = req.path.startsWith('/api/');

  if (isStateChangingMethod && isApiRoute && !isExempt) {
    const csrfHeader = req.headers['x-requested-with'];
    if (csrfHeader !== 'XMLHttpRequest') {
      return res.status(403).json({
        message: 'CSRF validation failed.'
      });
    }
  }
  next();
});
```

Protection Level: All POST, PUT, PATCH, DELETE requests to /api/* endpoints are protected.

2. Exempt Endpoints

The following endpoints are intentionally exempt from CSRF protection:

Authentication (Pre-login flows)

/api/auth/login, /api/auth/register, /api/auth/forgot-password, /api/auth/reset-password, /api/auth/send-verification, /api/auth/verify-email, /api/admin/login

MFA Verification

/api/mfa/verify - Stateless verification with challenge token

Public Endpoints

/api/contact, /api/gold-price, /api/geo-restriction/check, /api/platform-config/public, /api/cms/pages, /api/branding, /api/fees, /api/verify-certificate

External Webhooks

/api/webhooks/*, /api/binancepay/webhook, /api/ngenius/webhook, /api/stripe/webhook

3. Client-Side Implementation

Centralized Helper (client/src/lib/queryClient.ts)

```
export async function apiRequest(method: string, url: string, data?: unknown) {
  const headers = {
    'X-Requested-With': 'XMLHttpRequest',  // CSRF header
    'Content-Type': 'application/json'
  };

  const res = await fetch(url, {
    method,
    headers,
    body: data ? JSON.stringify(data) : undefined,
    credentials: "include",  // Required for session cookies
  });
  return res;
}
```

4. Coverage Summary

All state-changing API requests across the codebase have proper CSRF headers:

Area	Files	Status
Contexts	AuthContext, FinaPayContext, NotificationContext	Complete
User Pages	KYC, Security, Settings, HelpCenter, Notifications	Complete
Admin Pages	15+ files (UserDetails, Transactions, etc.)	Complete
Payment Modals	DepositModal, BuyGoldModal, BuyGoldWingoldModal	Complete
Payment Components	HybridCardPayment, EmbeddedCardForm	Complete
Communication	FloatingAgentChat, AdminChat, NotificationCenter	Complete
Other Components	BiometricSettings, TradeCertificate, etc.	Complete

Total Files with CSRF Headers: 37 files across the codebase

5. Security Features

' Custom Header Validation (X-Requested-With: XMLHttpRequest) - Active
' Credentials Include (credentials: include for session cookies) - Active
' Same-Site Cookies (sameSite: lax on session cookies) - Active
' HTTP-Only Cookies (httpOnly: true on session cookies) - Active
' Secure Cookies (secure: true in production) - Active
' User-Friendly Error Messages - Active

6. Error Handling

When CSRF validation fails, users see:

Server Response

403 Forbidden with message "CSRF validation failed. Please refresh the page and try again."

Client Display

"Your session may have expired. Please refresh the page and try again."

7. Why This Approach Works

• Browser Security: Browsers enforce CORS restrictions on custom headers for cross-origin requests
• No Token Management: Unlike token-based CSRF, no need to manage/rotate tokens
• Simple Implementation: Single header check vs. complex token

synchronization
• Works with SPAs: Ideal for React applications with JSON API calls

8. Recommendations

High Priority: All state-changing requests have CSRF headers - DONE

Medium Priority: Add automated test to verify CSRF rejection for missing headers - Suggested

Low Priority: Consider rate limiting on failed CSRF attempts - Suggested

Conclusion

The Finatrades platform has comprehensive CSRF protection across all 37+ files that make state-changing API requests. The custom header approach combined with proper cookie configuration (SameSite, HttpOnly, Secure) provides strong protection against cross-site request forgery attacks.