🔒 SSL/TLS Troubleshooting Guide - Advies N Consultancy BV

🚨 Current Issue: ERR_SSL_VERSION_OR_CIPHER_MISMATCH

Issue Summary

- **Domain**: www.adviesnconsultancy.nl
- **Error**: ERR_SSL_VERSION_OR_CIPHER_MISMATCH
- **Root Cause**: TLS handshake failure during SSL negotiation
- **Impact**: Website inaccessible via HTTPS externally

Diagnostic Results

```
$ curl -I -k -v https://www.adviesnconsultancy.nl
* TLSv1.3 (OUT), TLS handshake, Client hello (1)
* TLSv1.0 (IN), TLS header, Unknown (21)
* TLSv1.3 (IN), TLS alert, handshake failure (552)
* error:0A000410:SSL routines::sslv3 alert handshake failure
```

🔍 Root Cause Analysis

1. TLS Handshake Failure Causes

- **Cipher Suite Mismatch**: Client and server cannot agree on cipher
- **TLS Protocol Incompatibility**: Version mismatch between client/server
- **Certificate Chain Issues**: Incomplete or invalid certificate chain
- **Server Configuration**: Restrictive SSL/TLS settings

2. Application-Level Conflicts

- **Security Headers**: Strict HSTS or CSP policies
- **Middleware Interference**: Custom security headers conflicting
- **Proxy Configuration**: Load balancer or CDN SSL termination issues

🛠️ Immediate Solutions

Solution 1: Middleware Security Headers Fix

Current Issue in `middleware.ts`:

```
// Line 42-45: Potential HSTS conflict
if (request.url.startsWith('https://')) {
  response.headers.set('Strict-Transport-Security', 'max-age=3600')
}
```

Temporary Fix - Disable HSTS:

```
// TEMPORARY: Comment out HSTS to isolate SSL issue
// if (request.url.startsWith('https://')) {
//   response.headers.set('Strict-Transport-Security', 'max-age=3600')
// }
```

Apply this fix:

```
cd /home/ubuntu/advies-n-consultancy/app
# Edit middleware.ts and comment out HSTS lines
# Then test the website
```

Solution 2: Relaxed Content Security Policy

Current CSP may be too restrictive:

```
// Replace current CSP with more permissive version
const cspPolicy = [
  "default-src 'self' https: data:",
  "script-src 'self' 'unsafe-inline' 'unsafe-eval' https: data:",
  "style-src 'self' 'unsafe-inline' https: data:",
  "font-src 'self' https: data:",
  "img-src 'self' data: https: blob:",
  "connect-src 'self' https: wss: ws:",
  "frame-src 'self' https:",
  "object-src 'none'",
  "base-uri 'self'",
  "form-action 'self' https:",
  // Remove upgrade-insecure-requests directive
]
```

Solution 3: Next.js Configuration Update

Add SSL debugging to `next.config.js`:

```
const nextConfig = {
  // Existing config...

  // Add custom headers for SSL debugging
  async headers() {
    return [
      {
        source: '/(.*)',
        headers: [
          {
            key: 'X-SSL-Debug',
            value: 'enabled'
          },
          {
            key: 'X-TLS-Version',
            value: 'TLSv1.2,TLSv1.3'
          }
        ]
      }
    ]
  },

  // Ensure HTTPS redirect is properly configured
  async redirects() {
    if (process.env.NODE_ENV === 'production') {
      return [
        {
          source: '/:path*',
          has: [
            {
              type: 'header',
              key: 'x-forwarded-proto',
              value: 'http',
            },
          ],
          destination: 'https://adviesnconsultancy.nl/:path*',
          permanent: true,
        },
      ]
    }
    return []
  }
}
```

🔧 Platform-Specific Solutions

Cloudflare Configuration (If Using Cloudflare)

1. **SSL/TLS Settings:**
   `SSL/TLS → Overview → Encryption Mode: "Full (strict)"`
   `SSL/TLS → Edge Certificates → Minimum TLS Version: 1.2`
   `SSL/TLS → Edge Certificates → TLS 1.3: Enabled`

2. **Cipher Suite Configuration:**
   `SSL/TLS → Edge Certificates → Cipher Suites: Modern`
   `- Disable legacy cipher suites`
   `- Enable ECDSA certificates`
   `- Enable ChaCha20-Poly1305`

3. **HSTS Configuration:**
   `SSL/TLS → Edge Certificates → HSTS:`
   `  - Enable: Yes`
   `  - Max Age: 6 months`
   `  - Include subdomains: Yes`
   `  - Preload: No (disable to avoid conflicts)`

Hosting Provider Configuration

Required Settings for Hosting Provider:

```
TLS Configuration Requirements:
- TLS 1.2 minimum, TLS 1.3 preferred
- Modern cipher suites only
- Complete certificate chain including intermediates
- ECDSA + RSA certificate support
- ALPN protocol negotiation support
- No SSL 3.0 or TLS 1.0/1.1 support
```

Contact hosting provider with these requirements:

```
Subject: SSL/TLS Configuration Issue - Certificate Handshake Failure

Domain: www.adviesnconsultancy.nl
Issue: TLS handshake failure (error:0A000410:SSL routines::sslv3 alert handshake fail-
ure)

Required Actions:
1. Verify complete certificate chain installation
2. Enable modern cipher suites
3. Ensure TLS 1.2/1.3 support
4. Check for cipher suite compatibility
5. Verify ALPN protocol support
```

🧪 Testing & Validation

Test Commands

```
# Test different TLS versions
openssl s_client -connect www.adviesnconsultancy.nl:443 -tls1_2 -servername www.adviesn
consultancy.nl
openssl s_client -connect www.adviesnconsultancy.nl:443 -tls1_3 -servername www.adviesn
consultancy.nl

# Check available cipher suites
nmap --script ssl-enum-ciphers -p 443 www.adviesnconsultancy.nl

# Verify certificate chain
openssl s_client -connect www.adviesnconsultancy.nl:443 -showcerts -servername www.advi
esnconsultancy.nl

# Test with different user agents
curl -H "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36" ht-
tps://www.adviesnconsultancy.nl
curl -H "User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 14_0 like Mac OS X)" https://
www.adviesnconsultancy.nl

# SSL Labs test (external)
# Visit: https://www.ssllabs.com/ssltest/analyze.html?d=www.adviesnconsultancy.nl
```

Browser Testing

```
# Test in different browsers
# Chrome: Check Developer Tools → Security tab
# Firefox: Check Certificate viewer
# Safari: Check Certificate details
# Edge: Check Connection security
```

Validation Checklist

- [ ] TLS 1.2 handshake successful
- [ ] TLS 1.3 handshake successful
- [ ] Certificate chain complete
- [ ] Modern cipher suites working
- [ ] No mixed content warnings
- [ ] HSTS header present (after re-enabling)
- [ ] CSP not blocking resources

📋 Step-by-Step Resolution Process

Phase 1: Application-Level Fixes (Immediate)

1. **Disable HSTS temporarily** in `middleware.ts`
2. **Relax CSP policy** to eliminate header conflicts
3. **Add SSL debugging headers** in `next.config.js`
4. **Test website accessibility**

Phase 2: Server-Level Investigation (Within 24h)

1. **Contact hosting provider** with SSL configuration requirements
2. **Request SSL certificate chain verification**
3. **Verify TLS version and cipher suite support**
4. **Check for proxy/CDN SSL termination issues**

Phase 3: Validation & Re-enabling (After fixes)

1. **Test SSL handshake with various clients**
2. **Verify certificate chain completeness**
3. **Re-enable HSTS with gradual max-age increase**
4. **Monitor for any regression issues**

🔄 Quick Implementation Guide

Immediate Action (5 minutes)

```
cd /home/ubuntu/advies-n-consultancy/app

# Backup current middleware
cp middleware.ts middleware.ts.backup

# Edit middleware.ts - comment out HSTS lines (42-45)
# Replace:
#   if (request.url.startsWith('https://')) {
#     response.headers.set('Strict-Transport-Security', 'max-age=3600')
#   }
# With:
#   // TEMPORARY FIX: Disabled HSTS for SSL troubleshooting
#   // if (request.url.startsWith('https://')) {
#   //   response.headers.set('Strict-Transport-Security', 'max-age=3600')
#   // }

# Test the change
yarn build
yarn start
```

Test the Fix

```
# Test external access
curl -I https://www.adviesnconsultancy.nl

# If successful, website should be accessible
# If still failing, proceed to Phase 2 (hosting provider)
```

📞 Emergency Contacts & Escalation

Internal Escalation

1. **Developer Team**: Immediate notification of fix attempts
2. **DevOps Team**: Hosting provider communication
3. **Business Team**: Client communication about temporary issues

External Contacts

1. **Hosting Provider Support**: Priority SSL/TLS issue ticket
2. **SSL Certificate Provider**: Certificate validation if needed
3. **CDN Provider**: SSL termination configuration if applicable

Escalation Timeline

- **0-2 hours**: Application-level fixes
- **2-24 hours**: Hosting provider engagement
- **24-48 hours**: Alternative hosting/CDN evaluation
- **48+ hours**: Emergency hosting migration if needed

📈 Success Metrics

Resolution Confirmation

- [ ] Website accessible via HTTPS
- [ ] No SSL/TLS errors in browser
- [ ] SSL Labs grade A or A+
- [ ] All pages loading correctly
- [ ] Forms and functionality working
- [ ] Performance metrics maintained

Monitoring Setup

- [ ] SSL certificate expiry monitoring
- [ ] HTTPS redirect verification
- [ ] TLS handshake monitoring
- [ ] Certificate chain monitoring
- [ ] Cipher suite compatibility monitoring

Status: 🚨 Active Issue - Requires immediate action
Priority: Critical - Business impact
Estimated Resolution: 2-24 hours with hosting provider cooperation
Last Updated: July 12, 2025