Remediation Guide

Plain-English action plan · prepared for Riverside Family Law, PLLC

PREPARED FOR	**Riverside Family Law, PLLC**
TARGET	**https://www.riversidefamilylaw.com**
SCAN TYPE	**Full Scan**
SCAN STARTED	**Apr 15, 2026, 9:42:18 AM PDT**
SCAN DURATION	**22 minutes, 18 seconds**
GENERATED	**Apr 15, 2026**
GENERATED BY	**RiskMeter Cybersecurity**

SAMPLE · ILLUSTRATIVE

How to use this guide

This guide is the plain-English companion to the RiskMeter Technical Report. It describes every issue we found in language a business owner can act on — without learning security vocabulary first.

Each finding has the same five sections:

What this means. A description of the issue in everyday language.

Why it matters. The real-world consequence if it goes unaddressed.

What we found. How we observed the issue during the scan.

How to fix it. Numbered steps, written for a non-technical reader.

Estimated effort. How long the fix typically takes, and who does it.

How to read severity

Critical	Stop and address today. Active risk, low effort to exploit.
High	Address this week. A real attacker tool will find this.
Medium	Address this month. Real impact, but not the easiest path in.
Low	Address next quarter. Defense in depth — small individually, real together.
Informational	Address when convenient. Good hygiene, no immediate risk.

Working with your IT vendor

Most of these fixes call for an IT vendor — someone who can edit your site's configuration, install plugins, and update software. A few are things you can do yourself in five minutes (we mark those clearly). When you forward this guide to a vendor, the numbered steps under “How to fix it” are written so they can be acted on with no additional context from you.

SAMPLE · ILLUSTRATIVE

Overall threat level

High

We found at least one high-severity issue. The pages that follow describe every issue in plain English, with an effort estimate and step-by-step fix for each. Address the High-severity findings first.

Scan detail

Target	https://www.riversidefamilylaw.com
Scan type	Full Scan
Start time	Apr 15, 2026, 9:42:18 AM PDT
Scan duration	22 minutes, 18 seconds
Authentication profile	—

SAMPLE · ILLUSTRATIVE

Findings at a glance

Severity	Vulnerabilities	Instances
Critical	1	1
High	3	3
Medium	5	5
Low	4	4
Informational	4	4
Total	**17**	**17**

All findings, ordered by severity

Vulnerability group	Severity	Count
Backup file exposed (wp-config.php.bak)	Critical	1
.git repository exposed on web root	High	1
WordPress core is outdated (5.8.x — multiple known CVEs)	High	1
WordPress login page lacks rate limiting	High	1
Outdated jQuery 1.7.2 — multiple known XSS vulnerabilities	Medium	1
Directory listing enabled on /wp-content/uploads/	Medium	1
TLS/SSL weak cipher suites enabled	Medium	1
Contact form lacks CSRF protection	Medium	1
SSL certificate expires within 30 days	Medium	1

SAMPLE · ILLUSTRATIVE

Vulnerability group	Severity	Count
Cookies not marked as Secure	Low	1
Cookies not marked as HttpOnly	Low	1
HSTS header missing	Low	1
X-Frame-Options header missing (clickjacking)	Low	1
Server version disclosed in HTTP header	Informational	1
WordPress version disclosed in generator meta tag	Informational	1
Email address disclosure in page source	Informational	1
Permissions-Policy header not implemented	Informational	1

SAMPLE · ILLUSTRATIVE

How to fix what we found

Each finding below describes a single issue, why it matters, and what to do about it. Items are grouped by severity, with the most urgent first.

Critical-severity findings

A backup of your site's master settings is publicly downloadable

Critical

Estimated effort: 30 min · IT vendor

WHAT THIS MEANS

Someone made a backup of your site's main settings file (the file that stores your database password) and left it in a place where anyone on the internet can download it.

WHY IT MATTERS

If anyone downloads this file, they get the keys to your database — every contact-form submission, every login record, every piece of data your site stores.

WHAT WE FOUND

We typed a URL into a browser and downloaded the file *wp-config.php.bak* directly from your site.

HOW TO FIX IT

1. Ask your IT vendor to delete *wp-config.php.bak* and any sibling files ending in *.bak*, *.old*, or *.orig* from your website.

2. Change your database password and rotate WordPress's secret keys (your vendor will know how — there's a free tool at api.wordpress.org/secret-key/1.1/salt/).

3. Have your vendor configure the web server to block public downloads of any file with a backup extension.

High-severity findings

Your website's source-code history is publicly accessible

High

Estimated effort: 1 hour · IT vendor

WHAT THIS MEANS

When your site was deployed, the developer accidentally included the full version-control history. Anyone visiting a specific URL can download every file your site has ever had — including older versions of files that may have been removed.

WHY IT MATTERS

Source-code history often contains things that were deleted because they were sensitive — old passwords, internal email addresses, or private notes. Anything that's ever been there is recoverable.

WHAT WE FOUND

We accessed a special file at */.git/HEAD* that confirms the version-control history is published.

HOW TO FIX IT

1. Have your IT vendor delete the *.git* folder from your live site.

2. Have them configure the web server to block any URL starting with a period (this prevents similar future leaks).

3. Have them review what's in the leaked history and rotate any credentials they find.

Your WordPress version is several years old

High

Estimated effort: Half day · IT vendor (with testing)

WHAT THIS MEANS

Your site is running WordPress 5.8.6, which came out in mid-2022. Since then, the WordPress team has fixed many security issues that affect older versions.

WHY IT MATTERS

Outdated software is the single most common way websites get attacked. Automated tools constantly scan the internet for old WordPress versions and try the known exploits. The fix is straightforward but needs to be done carefully so plugins don't break.

WHAT WE FOUND

Your site's HTML source includes a tag identifying the WordPress version as 5.8.6.

HOW TO FIX IT

1. Have your IT vendor make a complete backup of the site first.

2. Apply the WordPress update in a staging environment to verify your plugins and theme still work.

3. Roll the update to production.

4. Turn on automatic minor updates so this doesn't happen again.

Anyone can try unlimited password guesses against your login page

High

Estimated effort: 30 min · IT vendor

WHAT THIS MEANS

When someone tries to log in to your WordPress admin and gets the password wrong, your site doesn't slow them down or lock them out. They can try thousands of passwords per minute.

WHY IT MATTERS

This is how most WordPress sites get broken into. Attackers run automated tools that try millions of common passwords until one works. Once they're in, they can publish anything, redirect your contact form, or hold your site for ransom.

WHAT WE FOUND

We attempted 250 logins in 90 seconds with no rate-limit response from your site.

HOW TO FIX IT

1. Have your IT vendor install a brute-force protection plugin like Limit Login Attempts Reloaded or Wordfence.

2. Require multi-factor authentication for any account that can publish or modify content.

3. Have them disable or rate-limit *xmlrpc.php* — it's a back door that bypasses login protection.

Medium-severity findings

Your site uses an outdated JavaScript library with known security holes

Medium

Estimated effort: 1–2 hours · IT vendor

WHAT THIS MEANS

Your site uses jQuery 1.7.2, a JavaScript library released in 2012. Multiple security issues have been fixed in newer versions over the years.

WHY IT MATTERS

Old JavaScript can sometimes be tricked into running attacker-supplied code in a visitor's browser. For a law firm, this could be used to display fake login prompts targeting your clients.

WHAT WE FOUND

Our scanner downloaded *jquery.js* from your site and found the version comment *jQuery v1.7.2*.

HOW TO FIX IT

1. Updating WordPress core (see the related finding) usually pulls in a current jQuery automatically.

2. Have your IT vendor check whether any plugin or theme is forcing the old version, and update or replace it if so.

3. Test the site after the update to confirm interactive elements still work.

Anyone can see a list of every file uploaded to your site

Medium

Estimated effort: 15 min · IT vendor

WHAT THIS MEANS

When someone visits the URL of your uploads folder directly, your web server shows them an index of every file in there — going back years.

WHY IT MATTERS

Files uploaded for a single client (case PDFs, photos, signed documents) become discoverable by anyone willing to browse the folder. This is a real problem for any practice that handles confidential client materials.

WHAT WE FOUND

We visited */wp-content/uploads/* and your site responded with a directory listing showing folders for 2024, 2025, and 2026.

HOW TO FIX IT

1. Have your IT vendor turn off directory listing on your web server.

2. Have them place an empty *index.html* file in the uploads folder as a backup defense.

3. Move any sensitive client materials out of */wp-content/uploads/* and into a folder with proper access controls.

SAMPLE · ILLUSTRATIVE

Your site supports outdated and weak encryption settings

Medium

Estimated effort: 30 min · IT vendor

WHAT THIS MEANS

When a visitor's browser connects to your site over HTTPS, the two sides agree on what kind of encryption to use. Your site still allows old, weakened encryption methods that modern browsers refuse.

WHY IT MATTERS

Modern browsers won't actually use the weak settings, but every cyber-insurance application and client security questionnaire checks for them. A 'B' grade or worse is something you'll have to explain repeatedly.

WHAT WE FOUND

Our scanner negotiated outdated TLS versions and weak cipher suites with your server.

HOW TO FIX IT

1. Have your IT vendor follow the Mozilla 'Intermediate' TLS configuration guide.

2. Specifically: turn off TLS 1.0 and TLS 1.1, and remove cipher suites containing RC4 or 3DES.

3. After the change, run your site through SSL Labs (https://www.ssllabs.com/ssltest/) and confirm an A or A+ grade.

Your contact form can be submitted from other websites

Medium

Estimated effort: 30 min · IT vendor

WHAT THIS MEANS

Your contact form accepts submissions from any source on the internet, not just from your own site. A malicious page hosted elsewhere can silently submit the form when someone visits it.

WHY IT MATTERS

Attackers can pollute your intake records with fake submissions, which costs your team time chasing leads that don't exist. It's also an embarrassing thing to come up in a client questionnaire.

WHAT WE FOUND

We submitted your contact form from an unrelated website and your site accepted the submission with no challenge.

HOW TO FIX IT

1. If you're using a contact-form plugin (Contact Form 7, WPForms, Gravity Forms), have your IT vendor turn on its built-in CSRF protection.

2. Add a honeypot field as a defense-in-depth measure (most modern form plugins include this).

3. Add basic rate limiting to your form so the same source can't spam dozens of submissions.

Your security certificate expires in less than a month

Medium

Estimated effort: 5 min · you (or 15 min · IT vendor)

WHAT THIS MEANS

Your site uses a security certificate (the lock icon in browsers) that expires in 26 days. If it expires before it's renewed, every visitor will see a security warning instead of your site.

WHY IT MATTERS

An expired certificate immediately breaks trust with visitors and clients — many will not click past the browser warning. Automated systems that connect to your site (CRM integrations, contact-form callbacks) will refuse to work.

WHAT WE FOUND

Your TLS certificate is dated to expire on May 11, 2026.

HOW TO FIX IT

1. If you're using Let's Encrypt (free certificates), it should auto-renew. Have your IT vendor confirm the renewal job is running.

2. If you're using a paid certificate, place the renewal order with your provider this week.

3. Set up expiry monitoring — most hosting providers offer free alerts at 30, 14, and 7 days before expiry.

Low-severity findings

Your login cookies aren't marked as 'HTTPS-only'

Low

Estimated effort: 20 min · IT vendor

WHAT THIS MEANS

Cookies are tiny pieces of data your site stores in a visitor's browser. There's a flag called 'Secure' that tells the browser to only ever send the cookie over an encrypted connection. Several of your cookies don't have that flag set.

WHY IT MATTERS

If anyone can ever get a connection to your site to fall back to non-encrypted HTTP, the cookies — including your admin login session — would travel in the clear. Combined with the missing HSTS header (see below), this becomes a real attack on someone using public Wi-Fi.

WHAT WE FOUND

Several cookies set by WordPress on your site are missing the Secure flag.

HOW TO FIX IT

1. Have your IT vendor add *define('FORCE_SSL_ADMIN', true);* to your *wp-config.php* file.

2. Confirm the WordPress site URL setting starts with *https://* (not just *http://*).

3. Combine with HSTS (see related finding) so browsers refuse plain HTTP altogether.

Some cookies can be read by JavaScript

Low

Estimated effort: 20 min · IT vendor

WHAT THIS MEANS

There's a flag called 'HttpOnly' that tells browsers to keep certain cookies invisible to JavaScript code on the page. Some of your cookies don't have it.

WHY IT MATTERS

If your site ever gets a small JavaScript injection vulnerability somewhere — a real possibility for any WordPress site with plugins — those exposed cookies become readable to whatever script the attacker injects.

WHAT WE FOUND

Several cookies in your site's responses are missing the HttpOnly flag.

HOW TO FIX IT

1. Have your IT vendor audit your site's cookies and add HttpOnly to every cookie that doesn't specifically need to be readable by JavaScript.

2. For most small-business WordPress sites, this means HttpOnly should be on for every cookie.

Your site doesn't tell browsers to always use HTTPS

Low

Estimated effort: 15 min · IT vendor

WHAT THIS MEANS

There's a header your site can send that tells browsers 'always use HTTPS for this domain, even if someone types *http://*.' Without it, the very first visit to your site can be intercepted on an insecure network like coffee-shop Wi-Fi.

WHY IT MATTERS

Modern security questionnaires expect this header to be set. It's also one of the simplest things to fix — a single header value.

WHAT WE FOUND

We checked the response headers from your site and the *Strict-Transport-Security* header is not present.

HOW TO FIX IT

1. Have your IT vendor add the response header *Strict-Transport-Security: max-age=63072000; includeSubDomains* on every page.

2. Test the site after the change — if anything breaks, dial back the *max-age* and roll forward gradually.

3. Once stable, consider HSTS preloading for stronger protection.

Other websites can hide your pages inside their own

Low

Estimated effort: 15 min · IT vendor

WHAT THIS MEANS

Your site doesn't have a header that tells browsers 'don't let other websites embed me in a hidden frame.' That means another site could load your login page invisibly and trick a logged-in user into clicking buttons they didn't intend to.

WHY IT MATTERS

This is a real attack pattern called 'clickjacking.' For sensitive pages — login, intake forms — it's a question of when, not if.

WHAT WE FOUND

Your responses don't include the *Content-Security-Policy* or *X-Frame-Options* headers.

HOW TO FIX IT

1. Have your IT vendor add the header *Content-Security-Policy: frame-ancestors 'self'* on every page.

2. For especially sensitive pages (login, payment), use *'none'* instead.

3. This pairs naturally with the HSTS and Permissions-Policy headers — they're typically configured in the same place.

Informational-severity findings

Your site reveals which web-server software and version it uses

Informational

Estimated effort: 10 min · IT vendor

WHAT THIS MEANS

Every page on your site sends a header telling the visitor's browser exactly what web-server software and which patch version is running. This is information attackers can match against the public list of known security holes.

WHY IT MATTERS

It doesn't directly let anyone in, but it makes attacker reconnaissance easier. Hiding it is a low-effort, high-courtesy improvement.

WHAT WE FOUND

Your site sends the response header *Server: Apache/2.4.41 (Ubuntu)* on every page.

HOW TO FIX IT

1. On Apache: have your IT vendor set *ServerTokens Prod* and *ServerSignature Off*.

2. On Nginx: have them set *server_tokens off*.

3. Also disable PHP's version exposure with *expose_php = Off* if applicable.

Your WordPress version is visible in your site's HTML

Informational

Estimated effort: 5 min · IT vendor (one-line code change)

WHAT THIS MEANS

WordPress automatically adds a small tag to every page that says exactly which version is running. Anyone can right-click and View Source to read it.

WHY IT MATTERS

Same reasoning as the server-version finding — it's reconnaissance fuel. Combined with updating WordPress core (see the High-severity finding), the version is both hidden and current.

WHAT WE FOUND

Every page on your site includes *<meta name="generator" content="WordPress 5.8.6" />*.

HOW TO FIX IT

1. Have your IT vendor add a small piece of code to your theme's *functions.php* that removes the generator tag.

2. Specifically: *remove_action('wp_head', 'wp_generator');*

3. Combine with updating WordPress core so the version is both hidden and not actually old.

Your team's email addresses are scrapable from your website

Informational

Estimated effort: 30 min · IT vendor or you

WHAT THIS MEANS

Several email addresses on your site are written as plain *mailto:* links in the page source. Automated programs that crawl the web specifically look for these and add them to spam and phishing lists.

WHY IT MATTERS

Law-firm partners are a known target for wire-fraud and case-document phishing. Reducing the visibility of email addresses cuts down on the number of targeted attempts.

WHAT WE FOUND

We found *<a href="mailto:partner@riversidefamilylaw.com">* and other staff addresses in the HTML source of your team page.

HOW TO FIX IT

1. The strongest fix: replace direct email links with a contact form that routes to the right person internally.

2. If direct emails must remain, have your IT vendor render them via JavaScript so they aren't visible in the static HTML.

3. Train staff on wire-fraud and case-document phishing patterns regardless — these reach you no matter what.

Your site doesn't restrict access to powerful browser features

Informational

Estimated effort: 15 min · IT vendor

WHAT THIS MEANS

Modern browsers can let websites access powerful features — camera, microphone, location. There's a header that lets you say 'my site doesn't need any of these, refuse all requests.' Your site doesn't set that header.

WHY IT MATTERS

If you ever embed a third-party widget that gets compromised, it could request sensitive permissions in your site's name. Setting this header up front is preventive.

WHAT WE FOUND

We checked the response headers from your site and the *Permissions-Policy* header is not present.

HOW TO FIX IT

1. Have your IT vendor add *Permissions-Policy: camera=(), microphone=(), geolocation=(), payment=()* to your site's response headers.

2. If your site ever needs one of these features (e.g., a video-conferencing tool), allow only the specific origin that needs it.