Mitigation	**Rotate All Credentials** Assume credentials may have been compromised. Reset: • WordPress administrator passwords • Hosting control panel credentials • SFTP/FTP passwords • SSH keys • Database credentials • CDN credentials • Cloudflare credentials • API keys and integration secrets Enable MFA wherever available. **Identify the Malicious Injection** The visible prompt is typically generated by injected JavaScript. **Inspect:** **Theme Files** Review: • wp-content/themes/<theme>/ • Pay particular attention to: • header.php • footer.php • functions.php **Plugin Files** Review: • wp-content/plugins/ Look for: • Recently modified files • Unrecognized plugins • Abandoned or unsupported plugins • Unexpected JavaScript files **Database Content** Examine: • wp_options • wp_posts • wp_postmeta Search for: • <script

	• eval(• fromCharCode • unescape(• atob(Large obfuscated JavaScript blocks are a common indicator. **Search for Persistence Mechanisms** Many organisations remove the visible code only to have it reappear days later. Check for: **Rogue Administrator Accounts** Review all WordPress users: Users → All Users Remove: • Unknown administrators • Dormant accounts • Recently created privileged accounts **Malicious Scheduled Tasks** Check for: • WP-Cron jobs • Server cron jobs • Hosting control panel scheduled jobs. **Backdoors** Search the webroot for: • eval(• base64_decode(• gzinflate(• str_rot13(• create_function(While not all occurrences are malicious, unexpected use in plugin or theme files warrants investigation. **Webshells** Look for suspicious files in: • wp-content/uploads/ • wp-includes/ • wp-admin/ PHP files should generally not exist in upload directories.

	Determine the Initial Compromise Vector Removing malware without addressing the entry point often results in reinfection. Common entry points include: • Vulnerable WordPress plugins • Vulnerable themes • Stolen administrator credentials • Supply-chain compromise of abandoned plugins • WordPress-specific vulnerabilities exploited at scale. Review: • Web server logs • Authentication logs • WordPress activity logs • Hosting provider audit logs Look for: • Unexpected admin logins • New plugin installations • File modifications • Requests associated with known vulnerable components **Replace Compromised Core Components** Where practical: • Reinstall WordPress core from a trusted source. • Replace themes with clean copies. • Replace plugins with freshly downloaded versions. • Remove unused themes and plugins. A file-by-file cleanup is often less reliable than replacement from known-good sources. **Check for Additional Infrastructure Compromise** While ClickFix is frequently used to deliver credential theft and remote access malware to visitors, threat actors may have broader objectives in leveraging compromised infrastructure. Look for unauthorised modifications to: • DNS records • Cloudflare settings

	• Redirect rules • Email configuration • Analytics scripts • Payment integrations • Third-party JavaScript inclusions **Perform Post-Cleanup Validation** Validate: • Homepage • Key landing pages • Blog posts • Login pages • Mobile versions Confirm: • No clipboard-copy behaviour • No unexpected JavaScript • No redirects • No fake verification prompts • No suspicious outbound connections Perform testing from multiple networks and user agents to account for conditional malware delivery.