Risk ID	Technical Risk	Technical Risk Indicators	Impact Rating	Impact	Mitigation	Validation Steps	Files
1, 5	Eval Code Injection	Data loss/manipulation without proper explanation. Unexplained behavior is noticed on target system.	H	Arbitrary code can be executed on the system. The injected code could access restricted data/files. This leads to loss of data integrity.	Refactor code so it does not use eval(). Sanitize input data.	Verify eval() is not used anywhere in codebase	plupload.silverlight.xap silverlightmediaelement.xap

126	**Code Injection: 'PHP Remote File Inclusion'**	Data loss/manipulation without proper explanation. Unexplained behavior is noticed on target system.	H	The PHP application receives user-supplied input but does not properly restrict the input before using it in require(), include(), or similar functions. This can allow an attacker to specify a URL to a remote location from which the application will retrieve code and execute it.	Validate all user-supplied input to ensure that it conforms to the expected format, using centralized data validation routines when possible. Use white lists to specify known safe values rather than relying on black lists to detect malicious input.	Verify 'PHP Remote File Inclusion' attacks no longer work after remediating code.	www/wp-admin/update.php 90

15 47 82 130 168 172 211	SQL Injection	Partially formed SQL statements appear in system/server logs as inputs to field values or URL paramaters.	H	An attacker can manipulate database queries in order to access, modify, or delete arbitrary data.	Always validate user input & use parameterized prepared statements instead of dynamically constructed SQL queries.	Verify that all previous vulnerabilities are no longer susceptible to SQL injection attacks.	www/…/SimplePie/Cache/MySQL.php 344 www/wp-includes/wp-db.php 797 www/board.php 30 www/includes/dblib.php 23 www/scoreboard/index.php
55, 89, 97, 116, 120, 124, 134, 140, 147	Use of Hard-coded Password	Account is compromised.	H	Account can be hacked into and become compromised putting all users of the system at risk.	Store passwords out-of-band from the application code. (e.g in database)	Perform code scan to verify that all hard-coded passwords have been removed.	www/board.php www/includes/dblib.php www/scoreboard/index.php www/…/network/site-new.php

9-12, 14, 18-20, 22, 29, 31, 34, 41-43, 46, 48, 52, 53, 56, 62, 68, 69, 70, 72, 73, 80, 81, 83, 88, 96, 98-100, 104, 113, 115, 117-119, 121, 123, 128, 129, 131, 133, 135, 136, 141, 143, 144, 146, 150, 154, 157, 159, 161, 162, 164, 166, 169, 170, 174, 175, 177, 184, 187, 190, 192, 193, 199, 201, 203, 210, 212	Cross-Site Scripting (XSS)	External content is displayed on this system or the presentation of content to the customer is not what it should be.	H	Presentation of content can be manipulated leading to phishing attacks etc. This can lead to sensitive information being compromised on the user's end. In addition, cookies can be stolen or manipulated.	Validate user-supplied input (e.g no <script> tag). Don't permit users to include HTML content in posts, notes, or other data.	Verify that all input fields have proper validation in place.	board.php, index.php www/wp-admin/import.php, install.php, media.php my-sites.php, plugins.php, press.this.php, themes.php, upgrade.php, upload.php, user-edit.php, user-new.php, widgets.php www/…/class-ftp.php, class-ftp-pure.php, class-ftp-sockets.php, class-phpmailer.php, class-wp-editor.php, includes/ajax-actions.php, includes/media.php, includes/nav-menu.php, media-template.php, network/themes.php, tinymce/wp-tinymce.php www/…/wp-admin/async-upload.php, edit-comments.php, includes/file.php, includes/post.php, link-manager.php, load-

2	Cleartext Storage of Sensitive Information in Memory (i.e password)	Unexplained logins to the system that were not initiated by the owner of the user account.	M	An attacker with access to the system running the application may be able to obtain access to the sensitive data by examining core dumps and swap files. Or by attaching to the running process a debugger and searching mapped memory pages.	Always clear sensitive data after use by explicitly zeroing out the memory.	Verify that code is in place to zero out the memory used to store sensitive data after use.	plupload.silverlight.xap

4	Insufficient Entropy	Attackers are gaining unauthorized access to a system that uses random numbers for authentication & authorization	M	An attacker can brute force the output of pseudorandom number generators gaining access to a session key or session identifier	Use a trusted cryptographic random number generator instead. CryptoAPI (Windows) or OpenSSL (open source)	Verify that code was remediated to use a trusted cryptographic random number generator.	plupload.silverlight.xap
127 149 176	Missing Encryption of Sensitive Data	Sensitive Data is compromised	M	Private data such as cryptographic keys or sensitive information may be erroneously exposed	Encrypt all sensitive data that is passed into functions	Verify that all sensitive data that is passed into functions are encrypted	www/…/class-ftp-sockets.php 138 …/class-wp-filesystem-ftptext.php 68 …/class-wp-filesystem-ftptext.php 70

6, 8, 16, 17, 21, 23, 24, 26, 27, 28, 30, 32, 33, 36, 37, 38, 40, 44, 45, 49, 50, 51, 54, 57, 58, 59, 61, 63, 64, 65, 66, 71, 74, 75, 76, 77, 78, 79, 84, 85, 86, 87, 90, 92, 93, 94, 101, 102, 103, 105, 106, 107, 108, 109, 111, 112, 114, 125, 132, 137, 139, 145, 148, 151, 153, 156, 158, 160, 163, 167, 171, 173, 178, 179, 181, 182, 183, 185, 186, 188, 189, 191	Broken or Risky Cryptographic Algorithm	Sensitive information is exposed	M	An attacker can expose the broken/risky cryptographic algorithm to gain access to the sensitive information	Replace broken/risky cryptographic algorithm with more robust cryptographic algorithm that are less risky	Verify that the old cryptographic algorithms have been replaced by the safer ones	Author.php, bookmark.php, Caption.php, Category.php, class-pcizip.php, class-phpass.php, class-phpmailer.php, class-pop3.php, class-simplepie.php, class-smtp.php, class-snoopy.php, class-wp-embed.php, class-wp-ms-themes-list-table.php, class-wp-plugins-list-table.php, class-wp-theme.php, class-wp-upgrader.php, class-wp.php, comment.php, Copyright.php, Credit.php, cron.php, dashboard.php, default-constants.php, Enclosure.php, file.php, general-template.php, getid3.php, gzdecode.php, image.php, Item.php, link-template.php, Memcache.php, module.tag.apetag.php, module.tag.id3v2.php, ms-blogs.php, ms-

3 67 122 152	External Control of File Name or Path	Files that are nomally inaccessible to end users start to inexplicably become modified	M	An attacker can access or modify otherwise protected system resources that would normally be inaccessible to end users	Validate all user-supplied input to ensure that it conforms to the expected format, using centralized data validation routines when possible. When using black lists, be sure that the sanitizing routine performs a sufficient number of iterations to remove all instances of disallowed characters	Verify that all input-data is properly sanitized	class-wp-upgrader.php shell.php plupload.silverlight.xap

13 60 91 95 142 165 180	Information exposure through an error message	An error message that provides to much information is displayed to the user	L	Sensitive information about its environment, users, or associated data are displayed to the user during an error message.	Generalize these error message so that they do not reveal any additional details other than what they absolutely need to know.	Verify that the previous error messages no longer provide additional details about the environment, users, or other associated data	board.php dblib.php index.php plugins.php themes.php
7 25 35 39 110 138 155 208	External Initialization of Trusted Variables or Data Stores	You start to notice arbitrary code being executed	L	If optarg is used in an unbounded string copy, an attacker can specify overly long command line arguments and overflow the destination buffer, potentially resulting in execution of arbitrary code	Limit the size of data copied from the optarg variable	Verify that all data copied from optarg variable has been limited in size	class-phpmailer.php getid3.lib.php getid3.php shell.php