	A	B	C	D	E	F	G	H
1	Risk ID	Technical Risk	Technical Risk Indicators	Related CWE or CVE IDs	Imapct Rating	Impact	Mitigation	Validation Steps
2	1	User authentication to the WordPress blog can be brute-forced.	Number of incorrect logins for accounts seen in logs; performance of login server has been degrading.	CWE-521: https://cwe.mitre.org/data/definitions/521.html	H	Increased load on login server; slower performance; possible denial of service	Lock out user account on 5 incorrect password tries by setting account	Account lockout flag set for user account on 5 incorrect password
3	2	Eval() Injection	Id query evaluated on Evil-Homer Page, system commands evaluated	CWE-95: http://cwe.mitre.org/data/definitions/95.html	H	Enables read/write access to all file stored on server; enables execution of arbitrary code	Avoid the use of Eval in general; sanitize user input	Code injeted in id query isn't executed
4	3	SQL Injection	Id query used to get data from the database evaluates SQL commands on board.php; username field on login page on board also vulnerable to SQL injections; SQL errors shown	CWE-89: http://cwe.mitre.org/data/definitions/89.html	H	Enables execution of SQL commands, which allows attackers to dump the DB	Sanitize user input to make sure it only accepts the proper format; avoid dynamically constructed SQL queries	Malicious SQL command isn't executed
5	4	Cross-site Scripting	Redirection when accessing board.php; a million alert windows; and all sorts of crazy images that shouldn't be there are all classic syptoms of XSS	CWE-80: http://cwe.mitre.org/data/definitions/80.html	H	Allows attackers to deface the website, execute malicious Javascript code client side, hijack session cookes and more	Sanitize user input by properly escaping script tags etc.	Website no longer executes arbitrary Javascript scripts
6	5	Hard-coded password	hardcoded password for PHP database in board.php	CWE-259: http://cwe.mitre.org/data/definitions/259.html	L	Exposes the database credentials to anyone who has the source php files, which can be obtained in other ways	Store the credentials in local config variables	Password is no longer hard-coded in plain text in the php files
7	6	Weak password for registered WP user	User bobo has weak password "scorpion", which can be easily cracked and allows full access to the account	CWE-521: https://cwe.mitre.org/data/definitions/521.html	H	Allows access to user account, which gives attackers the ability to make changes to the WP blog	Use stronger password with a combination of uppercase letters, lowercase letters, and numbers	New password cannot be cracked in a reasonable amount of time
8	7	Reliance on cookie without integraity checking	Key admin present in cookie on login page	CWE-565: https://cwe.mitre.org/data/definitions/565.html	H	Attackers can modify the boolean value of admin in cookie on login page to bypass authentication	Deploy tamper-detection to make sure cookies aren't altered; or avoid storing critical data via cookies	The value of admin can no longer b altered or is removed
9	8	Publicly accessible WP Readme	Readme page can be access without authentication	Unclear	L	Attackers can gain sensitive information regarding the architecture of the WP site and the version No., which can be used to find other vulnerabilities.	Delete readme.html	readme.html is no longer visible
10	9	Certain URLs that should be protected can be accessed directly	/FLAG route accessible without authentication	CWE-425: https://cwe.mitre.org/data/definitions/425.html	L	Unauthorized access to content	Implement correct access control for all routes	/FLAG can no longer be accessed directly
11	10	Content encoded in base64	A flag was encoded in base64	CWE-261: https://cwe.mitre.org/data/definitions/261.html	M	Information is not protected properly with weak encoding	Use more secure hashing algorithms to generate better keys	base64 is replaced with other algorithms