Risk ID	Technical Risk	Technical Risk Indicators	Impact Rating	Impact
95	Eval Injection	random code being implemented	M	arbitary PHP code executed on server
89	SQL Injection	excessive logins to protected accounts	M	arbitary SQL queries against database
80	XSS	content modified, lots of user-supplied input	H	populates HTTP response with user-supplied input, exploited cookies, modified presentation of content, compromise of information
259	hard coded password	excessive logins to protected accounts	H	account to tbe protected is logged in to
209	information exposure through error message	error messages appear on applicaiton	M	exposes information about application logic or names and versions of components, enables attacker to target known vulnerabilities.
565	reliance on cookies without validation and integrity checking	cookies changed	L	cookies can be modified , authentication can be bypassed, injecion attacks and cross-site scripting vulnerabilities
676	use of potentially dangerous functions	buffer overflows	H	bufer overflow

Mitigation	Validation Steps
Either properly validate user data before hand or use an alternative function	Set flag when validating user data if the data is suspicious
parameterized statements, escape characters with special meanings, pattern checking, limiting permissions on database logon	set flag when characters with special meanings used, set flag when wrong
Escaping from strings, work around filtered quotes, store passwords outside of code, use first time login that require unique strong password, use strong one-way hashes to passwords and store them in in configuration file with appropriate control	set flag when filter a destructive thing run strings command on firmware file
Manual analysis, exposed stack traces, fuzzing, obustness testing, fault injection	make error messages private
avoid using cookie data for security decisions, perform input validation on cookie data, att integrity checks	integrity checks,
avoid strcpy, avoid strcat, avoid sprintf	remove instances from code