Rachel Hogue
Intro to Security
Assignment 4

Risk ID	Technical Risk	Technical Risk Indicators	Related CVE, CWE, or OSVDB IDs	Impact Rating	Impact	Mitigation	Validation Steps
1	This database query contains a SQL injection flaw, www/board.php, www/scoreboard/index.php	The function call constructs a dynamic SQL query using a variable derived from user-supplied input. An attacker could exploit this flaw to execute arbitrary SQL queries against the database.	CWE 89	H	User can view and modify sensitive and confidential data. The user can gain more privileges over the database and could potentially delete all of the data.	Sanitize inputs and remove special characters from inputs. "Avoid dynamically constructing SQL queries. Instead, use parameterized prepared statements to prevent the database from interpreting the contents of bind variables as part of the query. Always validate user-supplied input to ensure that it conforms to the expected format, using centralized data validation routines when possible." –CWE website	Attempt malicious SQL queries, such as: *1' and '1'='1*
2	Use of hard-coded, plaintext password: www/board.php, www/scoreboard/index.php	The software contains a hard-coded password, which it uses for database authentication	CWE 259	H	"The hard-coded password is the same for each installation of the product, and it usually cannot be changed or disabled by system administrators without manually modifying the program, or otherwise patching the software. If the password is ever discovered or published (a common occurrence on the Internet), then anybody with knowledge of this password can access the product. Finally, since all installations of the software will have the same password, even across different organizations, this enables massive attacks such as	Do not store passwords in the application code.	Run Veracode to ensure that there are no hard-coded passwords.

					worms to take place." - CWE website		
3	Cross-Site Scripting: www/board.php (challenge 1), www/scoreboard/index.php	An attacker can send malicious code that's shown to other users. The web application uses untrusted data in the output it generates without validating or encoding it.	CWE-80	M	XSS can be used to modify web page content and can potentially compromise confidential information	Sanitize user input before using it as web application content.	Try to submit javascript as user input and ensure that the javascript isn't executed.
4	Directory Traversal	User can view files in 67.23.79.113/ctf/wp-content/uploads/	CWE-73	M	An attacker may be able to access or modify protected system resources that would normally be inaccessible to end users	Give minimal permissions to users to access directories and files.	Ensure that the user can't visit 67.23.79.113/ctf/wp-content/uploads/ and see the directory listing.
5	Information Leakage: www/board.php, www/scoreboard/index.php, www/logout.php, www/wp-content/uploads/2015/10/uncleherbert2.jpg, www/wp-content/uploads/2015/10/readme.txt, www/flag.txt (challenges 8, 9, +10)	User can view sensitive information on these pages and in these images (keys).	CWE-200	L	An attacker has access to information that they are not explicitly authorized to have access to.	Don't expose sensitive data.	Grep the page source for each page and run strings on each image to ensure that no sensitive data is out in the open.
6	Weak password: www/wp-admin.php bobo / supermodel (challenge 12)	The product does not require that users should have strong passwords, which makes it easier for attackers to compromise user accounts.	CWE 521	H	Lack of password complexity significantly reduces the search space when trying to guess user's passwords, making brute-force attacks easier.	Require users to have strong passwords.	Attempt to create a user with a weak password, and verify that you cannot create the user, and that you receive a message that you need a stronger password.
7	Unnecessary Service (ftp, challenge 2)	Many FTP servers support a default account with the user ID "anonymous" and password "ftp@", which can allow a malicious user access to files on the server.	CVE - 1999-0497	M	User could gain access to sensitive or confidential information.	Turn off ftp service, or remove the anonymous user.	Attempt to connect using ftp and anonymous user. Verify that you cannot gain access.
8	Response Discrepancy Information Disclosure www/wp-admin/login.php	A difference in failed-login messages could allow an attacker to determine if the username is valid or not.	CWE 204	M	This difference enables a potential attacker to discover a valid username by trying different values until the incorrect password message is returned.	Return the same message upon failed login, regardless of whether or not the username is valid.	Attempt to login with a valid username with an invalid password, and attempt to login with an invalid username. Verify that the message displayed is the

							same for both cases.
9	User Interface (UI) Misrepresentation of Critical Information, www/runme.exe (challenge 3)	Seeing an incorrect file extension, the user may try to run or open a file that they should not.	CWE-451	L	Erroneous data could trick the user into performing the wrong action.	Use correct filename extensions.	Check that each file extension is correct for files that users have access to.
10	Reliance on Cookies without Validation and Integrity Checking: www/main.php	Contains cookie that can be set to true. (lg)	CWE-565	M	Will display confidential information if manipulated correctly.	Avoid using cookie data for a security-related decision or add integrity checks.	Try to manipulate the cookie (if you decide to add integrity checks) and verify that you can't access anything the user shouldn't be able to access.
11	Using root user for database, www/wp-config.php, www/scoreboard/index.php	The application uses the root database user when it should use a user with minimal permissions.	CWE-250	H	The user may be able to gain access to sensitive data or perform malicious acts to the database, such as dropping tables.	Create a user with minimal access instead of using root.	Try to perform administrative actions while logged in as the db user, and ensure that you are not allowed to.
12	Information Exposure Through an Error Message: www/board.php, www/scoreboard/index.php	Error message reveals that a connection to the database failed, and the reason why: , "I cannot connect to db because:"	CWE-209	L	A user may be able to use information revealed through the error message (such as environment or software running) in order to launch an attack.	Use generic messages that don't reveal any sensitive information to the user, such as, "An error has occurred. Please contact the site administrator."	Do something that causes an error, such as using the wrong password for the db user, and make sure that you see a generic, non-revealing error message.
13	User authentication to system can be brute forced (challenge 12)	Number of incorrect logins for accounts seen in logs; performance of login server can degrade.	OSVDB-902, CVE-1999-1074	H	Increased load on login server; slower performance; possible denial of service	Lock out user account on 5 incorrect password tries by setting account lockout flag to true.	Account lockout flag set for user account on 5 incorrect password tries.