Macgill Davis

Comp 116 Ming

Assignment 4

Risk ID	Technical Risk	Technical Risk Indicators	Impact Rating	Impact	Mitigation	Validation Steps
1	Eval Injection	The software allows user-controlled input to be fed directly into a function (e.g. "eval") that dynamically evaluates and executes the input as code, usually in the same interpreted language that the product uses	H	Attacker can input code that can be evaluated to essentially do anything. An attacker can takeover website and exract any information they want.	Validate all user-supplied input	Stop using eval() and validate all user input so that input will not be executed as code
2	PHP Remote File Inclusion	The PHP application receives user-supplied input but does not properly restrict the input before using it in require(), include(), or similar functions.	H	This can allow an attacker to specify a URL to a remote location from which the application will retrieve code and execute it. Similar to eval, this gives an attacker a tremendous amount of power to impact the website and extract data.	Validate all user-supplied input	With validation injection cannot occur
3	SQL Injection	SQL injection vulnerabilities occur when data enters an application from an untrusted source and is used to dynamically construct a SQL query.	H	This allows an attacker to manipulate database queries in order to access, modify, or delete arbitrary data.	Validate all user-supplied input. Use prepared statements for DB queries. Normalize all user-supplie data.	Input of SQL queries either yields an error or fails to impact database.
4	User authentication to system can be brute-forced	Number of incorrect logins for accounts seen in logs. Ability to run brute-force scripts such as 'http-wordpress-brute' wit nmap and no reaction from website.	H	Attackers can guess every password and username combination using brute-force techniques until gaining entrance to website.	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
5	Trust of client-side data (cookies)	Sections of site rely on client-side cookies to determine authentication.	H	Client-side data is inherently not trustworthy and can be easily modified by attackers. In this specific case modifying cookies led to sign-in ability.	Never trust client-side data for any reason. Use other techniques like unique authentication tokens that cannot be easily recreated.	Altering of cookies fails to yield any authentication on the site.

	6	Encryption through encoding	Valuable information stored in 'Not Global Thermonuclear War' game only encoded through Base64.	M	Any information encoded through an encoding scheme can be decoded using that scheme. This is not secure as an attacker need only know the encoding scheme to find out the information.	Use a hash algorithm to hash information or store protected in database.	Attackers will not be able to decode the original data without the hash. If information is simply encoded, attackers can just decode using that scheme to reveal information.
	7	User of Hardcoded Password	The use of a hard-coded password significantly increases the possibility that the account being protected will be compromised. Moreover, the password cannot be changed without patching the software	M	Attacker can easily find password and gain access to website	Store passwords out-of-band from the application code	Restrain from use of a hard-coded password
	8	Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)	This call contains a cross-site scripting (XSS) flaw. The application populates the HTTP response with user-supplied input, allowing an attacker to embed malicious content, such as Javascript code, which will be executed in the context of the victim's browse	M	XSS vulnerabilities are commonly exploited to steal or manipulate cookies, modify presentation of content, and compromise confidential information	Validate all user input. Be sure to filter out code and escape all untrusted data.	If user input is validated and XSS prevented attackers will be unable to input code to be executed
	9	Cleartext Storage of Sensitive Information in Memory	The application reads and/or stores sensitive information (such as passwords) unencrypted in memory, leaving it susceptible to compromise or erroneous exposure	M	An attacker with access to the system running the application may be able to obtain access to this sensitive data by examining core dumps and swap files	Encrypt all sensitive data.	With data encrypted, even if attackers cause core dumps they still will be unable to decrypt sensitive data
	10	Insufficient Entropy	Standard random number generators do not provide a sufficient amount of entropy when used for security purposes	M	Attackers can brute force the output of pseudorandom number generators such as rand().	Use a trusted cryptographic random number generator instead	There are trusted random number generators like CryptoAPI and OpenSSL
	11	Missing Encryption of Sensitive Data	The application exposes potentially sensitive data by passing it into a function unencrypted	M	This could allow private data such as cryptographic keys or other sensitive information to be erroneously exposed	Encrypt sensitive data even when passing between functions.	Prevents attacker from catching sensitive data decrypted.

	12	Use of a Broken or Risky Cryptographic Algorithm	The use of a broken or risky cryptographic algorithm is an unnecessary risk that may result in the disclosure of sensitive information	M	Attacker can utilize known flaws in algorithm to decrypt sensitive data	Use a trusted cryptographic algorithm! There are a bunch out there.	Using a cryptographic algorithm will prevent the attacker from using known flaws to decrypt data
	13	Directory Traversal	Allowing user input to control paths used in filesystem operations may enable an attacker to access or modify otherwise protected system resources that would normally be inaccessible to end users	M	Attacker can use brute force techniques to access or modify files just by traversing through website. Never hide information this way because it is not truly hidden and will be found.	Validate all user input. Be sure not to leave anything important available through simple traversing through files.	
	14	Inclusion of .git file	Including a git file could potentially reveal private or sensitive information including the build of the website which could in turn show flaws and vulnerabilities.	M	Programmers often post .git files by accident or without thinking. If an attacker grabs the .git file it gives him/her access to all the files of the website.	Do not include .git file on hosted website.	With no .git file the attacker cannot use the .git file.
	15	External Control of File Name or Path	This call contains a path manipulation flaw. The argument to the function is a filename constructed using user-supplied input.	M	If an attacker is allowed to specify all or part of the filename, it may be possible to gain unauthorized access to files on the server, including those outside the webroot, that would be normally be inaccessible to end users	Validate all user input	