Corvit System Multan

ASSIGNMENT NO. 06

SUBMITTED BY:

Muhammad Adnan

SUBMITTED TO:

Supervisor Muhammad Bilal

DATE:

31-08-2024

TOPIC:

DDL IJECTION
API HOOKING

COURCE TITLE: CERTIFIED ETHICAL HACKER

TABLE OF CONTENT:

DDL INJECTION:

DDL INJECTION:

DEFINATION AND OVERVIEW:

'**DDL (Data Definition Language) Injection** is a type of security vulnerability that occurs when an attacker injects malicious DDL commands into an application's database queries. DDL commands, which include '`CREATE`', '`ALTER`', and '`DROP`', are used to define or alter the structure of a database. When exploited, DDL Injection allows attackers to modify the database schema, potentially leading to unauthorized data access, data loss, or corruption.'

COMMON TECHNIQUES:

Schema Modification:

Injecting commands to create, alter, or drop tables or other schema objects.

Privilege Escalation:

Exploiting injected DDL commands to escalate database privileges or gain higher access levels.

Data Corruption:

Using DDL injections to corrupt or destroy data by altering schema structures.

EXAMPLES OF VULNERABILITIES:

Unvalidated User Input:

Applications that do not properly validate or sanitize user inputs before constructing DDL queries.

Improper Error Handling:

Systems that expose detailed error messages or database structure information through flawed error handling.

PREVENTAION AND MITIGATION:

Input Validation:

Ensure that all user inputs are properly validated and sanitized before being used in DDL statements.

Parameterized Queries:

Use parameterized queries and prepared statements to prevent direct injection of DDL commands.

Least Privilege Principle:

Restrict database user privileges to minimize the impact of potential DDL Injection attacks.

Regular Audits:

Conduct regular security audits and code reviews to identify and fix potential vulnerabilities related to DDL Injection.

API HOOKING:

DEFINATION AND OVERVIEW:

'API Hooking is a technique used to intercept and modify calls made to an API (Application Programming Interface). By inserting custom code (hooks) into API functions, developers or attackers can monitor, alter, or extend the functionality of API calls made by an application. This technique is commonly used for debugging, monitoring, or enhancing software behaviour.'

COMMON TECHNIQUES:

Function Hooking:

Replacing or modifying the original API function with a custom implementation.

Code Injection:

Injecting additional code into an API function to alter its behavior.

DLL Injection:

Using dynamic-link libraries (DLLs) to hook API calls in applications running on Windows.

USE CASES AND APPLICATION:

Debugging:

Developers use API hooking to analyze and debug the behavior of software by intercepting API calls

.

Security Monitoring:

Security professionals hook APIs to monitor and detect malicious activities or unauthorized access.

Feature Enhancement:

Adding new features or modifying existing functionality in applications by intercepting API calls.

RISK AND SECURITY CONSIDERATION:

Security Vulnerabilities:

Improper use of API hooking can introduce security risks, such as unauthorized data access or manipulation.

Performance Overhead:

Hooking can impact application performance due to the added overhead of intercepting and processing API calls.

Compatibility Issues:

API hooking may cause compatibility issues with other software or future updates of the API.

PREVENTATION AND MITIGATION:

Code Reviews:

Regularly review and audit code to ensure that API hooking is used appropriately and securely.

Secure Coding Practices:

Follow best practices in secure coding to minimize the risk of introducing vulnerabilities through API hooking.

Monitoring and Alerts: Implement monitoring systems to detect and respond to suspicious or unauthorized API hooking activities.