CHAPTER 4

SYSTEM DESIGN
CHAPTER 4

SYSTEM DESIGN

System design sits at the technical kernel of the software engineering process and is applied regardless of the software process model that is used. System design is the first of the three technical activities – design, code generation and testing – that are required to build and verify the software. Software requirements, manifested by the data, functional, and behavioral models, feed the design process. The design process produces an architecture design, functionality design, data design and user interface design.

4.1 System Architecture Design

System architecture alludes to “the overall structure of the software and the ways in which that structure provides conceptual integrity for a system” [27]. In its simplest form, architecture is the hierarchical structure of program components, the manner in which these components interact, and the structure of the data that are used by these components. In a broader sense, however, “components” can be generalized to represent major system elements and their interactions. Figure 4.1 depicts the architecture design of IIS-SCAN.
Figure 4.1: IIS-SCAN System Architecture Design

Fundamentally, IIS-SCAN adopts a two-tier architecture design. The first tier refers to the database tier and the other tier refers to the user interface and logic of the IIS-SCAN program. Referring to Figure 4.1, the IIS program that is stored in the workstation will act as the client and typically sends and receives requests from the IIS server to be scanned, which is in this case, the server. Transmission of information can be restrained to an Intranet environment or can be scaled up to cover an Internet environment.

The underlying vulnerability database integrates closely with IIS-SCAN, serving as part of the scanning engine. For database access, the architecture of the accessing method will utilize Component Object Model (COM) components to encapsulate the database access via Open Database Connectivity (ODBC) drivers or Object Linking and Embedding Database (OLE DB) providers. In line with this architecture, any database can be plugged-in and out without requiring major changes to the program.
4.2 System Functionality Design

Functionality design deals with the purpose and collaboration of each module to achieve the overall system functionality specification.

4.2.1 Structure Chart

The main structure chart for IIS-SCAN is shown in Figure 4.2. Generally, IIS-SCAN can be divided into two major modules, Scan Module and Report Module.

![Structure Chart for IIS-SCAN program](image)

Figure 4.2: Structure Chart for IIS-SCAN program

The Scan Module can be further divided into sub-modules as shown in Figure 4.3.

![Structure Chart for Scan Module](image)

Figure 4.3: Structure Chart for Scan Module
4.2.1.1 Module Explanation

The following section will present a detailed explanation for each module in IIS-SCAN.

4.2.1.1.1 Scan Module

The scan module is the core of the IIS-SCAN system. Every function in this module revolves around the vulnerability scanning process. This module comprises of two sub modules, scan configuration and scan engine.

   i. Scan Configuration sub module
   The purpose of this sub module is to set up the initial or required parameters before the commencement of the scan engine’s operation.

   ii. Scan Engine sub module
   This sub module controls the scanning process and acts as the “vulnerability attacker” on an IIS system. Any vulnerability successfully detected by the scan engine will be reported to the user.

4.2.1.2 Report Module

This module solely focuses on the reporting of vulnerabilities found and the querying of vulnerabilities list in the database.

4.2.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical technique that depicts information flow and the transforms that are applied as data move from input to output. The data flow diagram may be used to represent a system or software at any level of abstraction [27].
Therefore, the DFD provides a mechanism for functional modeling as well as information flow modeling for the modules in IIS-SCAN. Figure 4.4 presents the context diagram or level 0 data flow diagram for IIS-SCAN. Figure 4.5 – Figure 4.7 depicts the data flow for sub modules 1.1, 1.2 and 2.0 respectively.

Figure 4.4: Context Diagram (Level 0 Data Flow Diagram)
Figure 4.5: Level 1 Data Flow Diagram for Process 1.1 (Scan Configuration)
Figure 4.6: Level 1 Data Flow Diagram for Process 1.2 (Scan Engine)
Figure 4.7: Level 1 Data Flow Diagram for Process 2.0 (Report Module)
4.2.3 Flow Chart

The flow chart provides excellent pictorial patterns that readily depict procedural detail. It helps to programmatically convey the algorithms of certain specific processes. Figure 4.8 shows the algorithm for process 1.2.1 (Internet Server Verification) from Figure 4.6.

![Flow Chart Diagram]

Figure 4.8: Flow Chart for Process 1.2.1 (Internet Server Verification)
Figure 4.9 provides an algorithmic view of process 1.2.2 (Vulnerability Scanning) from Figure 4.6.

![Flow Chart for Process 1.2.2 (Vulnerability Scanning)](image)

Figure 4.9: Flow Chart for Process 1.2.2 (Vulnerability Scanning)
Figure 4.10 depicts the programming logic of process 2.2 (Report Generation) from Figure 4.7.
4.3 Database Design

For IIS-SCAN database layer, Microsoft Access 2000 has been chosen for its database implementation. Since scalability is not a major issue for IIS-SCAN, Microsoft Access does have advantages in terms of portability between Windows platform and development compatibility with IIS-SCAN. Table 4.1 summarizes the attributes of the vulnerability database.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>VulDB.mdb</td>
</tr>
<tr>
<td>Database Type</td>
<td>Microsoft Access 2000</td>
</tr>
<tr>
<td>Usage</td>
<td>To store vulnerabilities information</td>
</tr>
<tr>
<td>Number of Tables</td>
<td>1</td>
</tr>
</tbody>
</table>

4.3.1 Data Dictionary

The table structure of the IIS-SCAN database is listed in the Table 4.2.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Numeric</td>
<td></td>
<td>Vulnerability ID</td>
</tr>
<tr>
<td>Abbv</td>
<td>Text</td>
<td>100</td>
<td>Vulnerability abbreviation</td>
</tr>
<tr>
<td>DoS</td>
<td>Text</td>
<td>1</td>
<td>Flag value to determine whether the specified vulnerability is a Denial of Service attack</td>
</tr>
<tr>
<td>Desc</td>
<td>Text</td>
<td>100</td>
<td>Vulnerability description</td>
</tr>
<tr>
<td>Discussion</td>
<td>Memo</td>
<td>-</td>
<td>Vulnerability discussion information</td>
</tr>
<tr>
<td>Exploit</td>
<td>Text</td>
<td>200</td>
<td>String or URL information on how the vulnerability can be exploited</td>
</tr>
<tr>
<td>Solution</td>
<td>Memo</td>
<td>-</td>
<td>Patches link or preventive measures to solve the vulnerability</td>
</tr>
<tr>
<td>Credit</td>
<td>Memo</td>
<td>-</td>
<td>Information on the person or organization responsible for finding the vulnerability</td>
</tr>
</tbody>
</table>

64
4.4 User Interface Design

IIS-SCAN interfaces are designed in a user-friendly yet efficient interface. The following design areas are taken into consideration: general interaction, information display and data entry. The subsequent guidelines are adopted in designing the interface for IIS-SCAN:

- **Be consistent.** Use a consistent format for menu selection, command input, data display and other myriad functions.

- **Ask for verification of any non-trivial destructive action.** If a user requests the deletion of a file, indicates that substantial information is to be overwritten, or asks for the termination of a program, an “Are you sure...?” message should appear.

- **Reduce the amount of information that must be memorized between actions.** The user should not be expected to remember a list of numbers or names so that he or she can reuse them in a subsequent function. Memory load should be minimized.

- **Seek efficiency in dialog, motion and thought.** Keystrokes should be minimized; the distance a mouse must travel between picks should be considered in designing screen layout.

- **Forgive mistakes.** The system should protect itself from errors that might cause it to fail.

- **Categorize activities by function and organize screen geography accordingly.** One of the key benefits of the pull-down menu is the ability to organize commands by type. In essence, the designer should strive for “cohesive” placement of commands and actions.

- **Maintain consistency between information display and data input.** The visual characteristics of the display (e.g. text size, color and placement) should be carried over to the input domain.

- **Deactivate commands that are inappropriate in the context of current actions.** This protects the user from attempting some action that could result in an error [27].
Figure 4.11 shows a sample screenshot of the IIS-SCAN main interface.

![Sample Screenshot of IIS-SCAN Main Display](image)

**Figure 4.11: Sample Screenshot of IIS-SCAN Main Display**

IIS-SCAN main display is designed with a three-pane information view. The scan display occupies the largest area, as it is the center of the scanning activities followed by the vulnerabilities found list that tracks all vulnerabilities found on the targeted server.

Menu items have been utilized in IIS-SCAN to provide a systematic interface for user to load up extra information where necessary and to avoid screen clustering. The configuration screen is located under Session → Settings menu item while Database → Vulnerability List menu item allows user to view all vulnerabilities information in the database.
Buttons location in IIS-SCAN is also closely located to avoid too much mouse movement. All controls can also be navigated through keyboard keys as an alternative and are equipped with shortcut keys.

IIS-SCAN has also taken into consideration for exceptional errors such as not specifying server’s name or IP address or printing reports without any vulnerability reported. Informative message boxes have been built in to cater for these cases and at the same time providing an error recovery mechanism for IIS-SCAN. Warning messages are also built in to inform users of selecting a Denial of Service vulnerability attack that may cause the targeted Internet server to crash. Besides that, confirmation message box is also added to ask for verification before exiting the application.