Analysis Report for: a.vbs

Decoded using latin-1...

Overall Functionality

This VBA macro is designed to process data within an Excel workbook, likely for report generation or data manipulation. It appears to interact with pivot tables, manipulate data ranges, format cells, and potentially generate an external text file. The macro has multiple functions, suggesting modular design. However, the use of `Environ`, `CreateTextFile`, and `CreateObject` raises significant security concerns, indicating possible malicious intent. The macro's primary function, `Formatear()`, performs data processing and updates pivot tables based on named ranges within the spreadsheet. The extensive error handling (`On Error GoTo Errores_existencia`) attempts to catch and manage potential runtime errors gracefully, but does not preclude malicious behaviour.

Function Summaries

* **`Formatear()`**: The main subroutine. It automates data processing, including updating pivot tables, formatting the worksheet, and potentially generating a text file. It uses other functions for sub-tasks.

* **`Marco(rango As String)`**: Adds borders to a specified range of cells. Takes a range name as input (string). No return value.

* **`FilaTexto(columna As String, Texto As String, Optional hoja As String, Optional iteracion As Integer) As Integer`**: Searches for a specific text within a given column and sheet. Returns the row number where the text is found. Handles multiple occurrences with the `iteracion` parameter.

* **`ConfigurarPagina(hoja As String, Optional ancho As Integer, Optional apaisado As Boolean, Optional filaTitulos As String)`**: Configures page setup for printing, including orientation, fit-to-pages, and print titles. No return value.

* **`ArrastrarFormula(celda As String, Optional hoja As String)`**: Autofills a formula down a column to a specified row (`filaFin`). Takes a cell address and optional sheet name as input. No return value.

* **`Desbloquear(hoja As String, area As String)`**: Unlocks cells within a specified range. Takes sheet and range names (strings) as input. No return value.

* **`ProtegerHoja(hoja As String)`**: Protects a worksheet without a password. Takes sheet name (string) as input. No return value.

* **`CrearHoja(nomHoja As String)`**: Creates a new worksheet. Takes the sheet name (string) as input. No return value.

* **`EliminarHoja(nomHoja As String)`**: Deletes a worksheet. Takes sheet name (string) as input. No return value.

* **`Rellenar(campo As String, longitud As Integer, relleno As String, izquierda As Boolean)`**: Pads a string with a specified character to a given length, either left or right. Returns the padded string.

* **`InsertarFila(fila As Integer)`**: Inserts a row at a specified position. Takes row number (integer) as input. No return value.

* **`EliminarFila(fila As Integer, Optional hoja As String)`**: Deletes a row at a specified position. Takes row number (integer) and optional sheet name as input. No return value.

* **`ColorFondo(rango As String, Color As String)`**: Sets the background color of a specified range of cells. Takes range name and color (strings) as input. No return value.

* **`CopiarCeldas(hojaOrigen, rangoOrigen, hojaDestino, celdaDestino)`**: Copies cells from one range to another. Takes sheet names and range addresses as input. No return value.

* **`FechaFinMes(fecha As Date)`**: Returns the last day of the month for a given date.

* **`MoverElementoTD(hojaTabla, nomTabla, campo, elemento, Optional final As Boolean)`**: Moves a pivot table item to the top or bottom of a field. Takes the sheet, pivot table, field, item names and a Boolean (final) as input. No return value.

* **`CreacionFichero()`**: Creates a text file. Prompts the user for a filename and path. This function raises serious security concerns due to the potential for writing data to an arbitrary location on the system.

* **`OcultarHoja(nomHoja As String)`**: Hides a worksheet. Takes the sheet name (string) as input. No return value.

* **`EliminarFilasVacias(filaFin As Long)`**: Deletes empty rows from a sheet. Takes the last row number as input.

Control Flow

The `Formatear()` subroutine is the main control flow. It calls many other functions sequentially and conditionally. It contains loops to iterate through columns, pivot tables and pivot items, performing various data manipulations. Conditional statements check for Excel versions and the presence of formulas, pivot tables, or specific environment variables. Error handling is implemented with a `GoTo` statement, jumping to an error handler.

The `CreacionFichero()` function contains a loop to process each row and write it to an external text file. The file's name and path are obtained from user input, creating a vulnerability if the filename is maliciously crafted.

Data Structures

The code primarily uses Excel worksheet data as its main data structure. It manipulates ranges of cells and named ranges. There are no explicitly defined custom data structures, but variables are used to store various data types like integers, strings, and ranges.

Malware Family Suggestion

Given the presence of functions like `Environ`, `CreateTextFile`, and `CreateObject`, along with the potential for writing arbitrary data to a user-specified file location (`CreacionFichero`), this VBA macro exhibits characteristics of a **file infector or dropper**. The macro could be part of a larger malware campaign, used to steal data, exfiltrate information, or install additional malicious components. The file creation and potential use of `CreateObject` to create OLE objects indicate it may also have features of a **macro virus**. The obfuscation attempts, including the use of `Chr` and potentially hex/base64 encoding (indicated by the oletools output), would classify it as a reasonably sophisticated piece of malware. The error handling isn't necessarily a characteristic of malware, however, it makes reverse engineering slightly more difficult. The absence of immediately obvious network communication does not rule out its malicious nature, as it might rely on subsequent actions or stages of the broader attack. In short, its functionality seems designed to conduct malicious operations rather than performing legitimate data processing.