Analysis Report for: b.vba

Decoded using latin-1...

Overall Functionality

This VBA code embedded within an Excel file (.xls) appears to be designed for managing and manipulating data related to a construction or engineering project. It includes functionalities for:

* **Data Management:** Saving the workbook with dynamically generated filenames based on cell contents, deleting data from specific ranges (header information and "C-Line"), restoring original cell formats.
* **Component Management:** Adding and deleting project components. This involves numerous subroutines that copy and move data between rows, suggesting a spreadsheet-based representation of components. This is coupled with user prompts using `MsgBox` for confirmation before deletion.
* **Reporting:** Setting print areas for different sections (Page 1 and both pages)
* **Database Interaction:** Transferring data from the main spreadsheet into a "Datenbank" (Database) sheet, and it includes formatting and data cleaning operations during the transfer.
* **Image Import:** Import images for visual representations into pre-defined areas of a spreadsheet.
* **External File Import:** Importing data from other Excel files ("ImportSK" and "ImportCL"), copying data from specific ranges into the current workbook.
* **Form Controls:** The presence of a `UserForm_Initialize` subroutine suggests the use of a user interface with a form containing several textboxes and checkboxes.

The code's structure is modular, using many subroutines to perform specific tasks. However, the sheer volume and repetitive nature of many subroutines (especially the `Zeile_XX_YY` series) raise concerns about potential obfuscation or unnecessary complexity. The password "ddookkuu1234" used in the `Tiefbauer` and related subroutines suggests an attempt to protect sensitive data or functionality.

Function Summaries

The code consists primarily of Subroutines (procedures that don't return a value) and one function:

* **`Speichern()`:** Saves the active workbook with a filename constructed from cell values in ranges "C1:E1", "W1", "C3:E3", and "C5:E5". It uses `Application.GetSaveAsFilename` to prompt the user for a filename. No return value.
* **`Lösch_K()`:** Prompts the user to confirm deleting contents from cells related to header data. No return value.
* **`Lösch_C()`:** Prompts the user to confirm deleting contents from a large range of cells (the "C-Line"). No return value.
* **`Seite1()`:** Sets the print area to "A1:Av60". No return value.
* **`Seite2()`:** Sets the print area to "A1:Av91". No return value.
* **`Orginalformat()`:** Copies and pastes a range, presumably restoring a default format. No return value.
* **`BT01_L()` to `BT16_L()`:** A series of subroutines for deleting a specific component (Bauteil). These subroutines use `Application.Run` to call other subroutines to perform the data manipulation. Each displays a confirmation `MsgBox` and then a success message. No return value.
* **`Zeile_17_12()` to `Zeile_87_82()`:** A series of subroutines that copy data from one block of cells to another. These are the actual data movers called by the `BT_XX_L` functions. No return value.
* **`BT01_P()` to `BT15_P()`:** A series of subroutines for inserting a component. They use `Application.Run` to call the `Zeile_XX_YY` subroutines in reverse order, suggesting data shifting to make space. Each displays a success `MsgBox`. No return value.
* **`Zeile_12_17()` to `Zeile_82_87()`:** A mirror series to `Zeile_17_12()` etc., that moves data between blocks of cells, similar to its counterpart series, but in the reverse direction. No return value.
* **`Datenbank()`:** Copies data from various cell ranges to a "Datenbank" sheet. Includes data cleaning (removing "Stk"). No return value.
* **`ABZW()`:** Copies the "Abzweiger" sheet, then clears contents from specific cell ranges. No return value.
* **`LAP()`:** Makes the "Lageplan (1)" sheet visible, copies it, and hides it again. No return value.
* **`NZP()`:** Makes the "Netzplan (1)" sheet visible, copies it, and hides it again. No return value.
* **`FOTO()`:** Copies the "FotoM" sheet, renames it to "Fotos X. BG" (incrementing X). No return value.
* **`AA_A()`:** Unprotects a sheet, changes a button's action, shows hidden columns, sets a version string, performs several replace operations within specified ranges, copies and pastes blocks of cells, hides columns again and protects the sheet. No return value.
* **`AA_DB()`:** Similar to `Datenbank()`, but appears to be called by a button action defined in `AA_A()`. No return value.
* **`A_UeP()`:** Unprotects a sheet, performs several `Replace` operations on formulas within cell ranges, copies and pastes data blocks, and protects the sheet. No return value.
* **`Messblatt()`:** Copies "Messwerte KUNDE" sheet, renames it, and clears contents from specified ranges. No return value.
* **`Tiefbauer()`:** Copies a range of cells (R6:V8) to several locations depending on whether the cell in X6, X10, X14, X18, X22 or X26 is empty, in which case it will copy the range to those cells. It inserts text "Tiefbauer n:" and then clears contents of other cells. No return value.
* **`Tiefbauer_Löschen()`:** Clears contents and formatting from a range of cells based on whether certain cells are not empty. It unprotects then protects the sheet. No return value.
* **`Pdf_drucken_LAP()`:** Shows a dialog (presumably for PDF printing). No return value.
* **`Komprimieren()`:** Executes a command to compress the workbook (likely using Excel's built-in functionality). No return value.
* **`Ausgabe()`:** A function that checks the value in cell "AB87". If it's not "Warren Robinett", it displays a message; otherwise, it displays a different message. Clears the contents of AB87. Returns nothing (implicitly void function).
* **`DDlöschen()`:** Unprotects the active sheet. No return value.
* **`ImportSK()`:** Imports data from another excel file ("Daten" Sheet) to this one ("Daten" Sheet). It handles errors and closes the source file. No return value.
* **`ImportCL()`:** Imports data from another excel file ("Berechnung" Sheet) to this one ("Berechnung" Sheet). It handles errors and closes the

source file. No return value.
* **`BTeinf()`:** Prompts the user for a component number (1-15), then runs the appropriate "BTXX_P" subroutine to insert that component. No return value.
* **`BTlös()`:** Prompts the user for a component number (1-16) and runs the appropriate "BTXX_L" subroutine to delete it. No return value.
* **`BT1lösch()` to `BT6lösch()`:** Subroutines for deleting a specific component on a different sheet (likely an "Abzweiger" sheet). It prompts the user for confirmation. No return value.
* **`Messwerte1()` to `Messwerte4()`:** These subroutines copy values from the "Messwerte KUNDE" sheet to the "Daten" sheet. They also handle merging and unmerging cells in the destination range. No return value.
* **`BilderImport()`:** Imports up to 32 images from a user-selected directory and places them within predefined ranges on a sheet. Uses `wscript.shell`, suggesting potential for more malicious behavior. No return value.
* **`RA_SUB()`:** Makes the "Rechnungsaufmaß SUB" sheet visible and activates it. No return value.
* **`RA_komplett()`:** Makes the "Rechnungsaufmaß KOMPLETT" sheet visible and activates it. No return value.

Control Flow

The control flow is generally straightforward, mostly linear within each subroutine. Key control structures include:

* **`If-Then-Else` statements:** Used extensively for conditional actions, particularly for user confirmation dialogs (`MsgBox`) and conditional data manipulation based on cell values.
* **`For Each` loops:** In `FOTO()`, `Messblatt()`, and `BilderImport()` to iterate over worksheets or selected items.
* **`Do While` loops:** In `BTeinf()` and `BTlös()` to repeatedly prompt the user until valid input is received. These loops can be seen as an attempt to obfuscate the direct calling of the BT functions.

Data Structures

The primary data structure is the Excel spreadsheet itself. Data is organized into cells and ranges within multiple sheets. The code heavily relies on the inherent structure of the spreadsheet to represent project components and their associated data. The `arrBereiche` array in `BilderImport()` is a simple one-dimensional array storing cell ranges where images will be inserted.

Malware Family Suggestion

While this code doesn't contain obvious malicious actions like direct execution of arbitrary code or network communication, its functionalities and structure exhibit characteristics of a potential macro virus or a more sophisticated piece of malware designed for data exfiltration or manipulation. The use of `MsgBox` for user interaction might be employed to make the actions appear less suspicious to the user while the extensive data copying and moving to different sections of the worksheet may indicate attempts to obfuscate its actions. The use of a password ("ddookkuu1234") to protect sheets, though not inherently malicious, is common in malware to hide its actions. The `BilderImport` function particularly raises red flags due to the use of `wscript.shell`, which has the potential to execute external commands beyond what the rest of the code does. It could be used for further malicious actions. The overall complexity and highly structured data management coupled with the ability to import data from external sources makes it potentially part of a larger attack framework.

The repetitive nature of numerous subroutines suggests potential for obfuscation, and the functions that copy and move data around the worksheet, especially the `Zeile_XX_YY` series, are a strong indicator of this.

Therefore, while not a clear-cut example of a specific known malware family, this code exhibits characteristics that could be part of a **macro virus** or a more advanced tool used in a **data exfiltration** or **data manipulation** campaign. Further analysis, especially dynamic analysis in a sandboxed environment, would be required to determine its true malicious capabilities. The presence of `wscript.shell` and a password protection mechanism should also be investigated further to better classify this code as potentially malicious.