Analysis Report for: c.vbs

Decoded using latin-1...

Overall Functionality

The provided C code is actually VBA (Visual Basic for Applications) code embedded within an Excel file (.xls). The code implements a system for managing and manipulating data within an Excel spreadsheet, likely related to some kind of project or inventory tracking system. It features several modules:

* **Modul1:** Contains macros for saving the workbook with a dynamically generated filename, deleting header and specific row data, and setting print areas.
* **Modul2:** Contains macros for deleting components (Bauteil), each with a prompt before deletion. These macros are designed to delete data in specific ranges of the spreadsheet and they call other subroutines to move data around.
* **Modul3:** Contains macros for inserting components (Bauteil), with a similar structure to the deletion macros, shifting data around to make space for new entries.
* **Modul4:** A macro for copying data to a "Datenbank" sheet (database).
* **Modul5:** Macros for manipulating sheets named "Abzweiger," "Lageplan (1)," "Netzplan (1)," and "Fotos*." These seem to be related to creating copies of these sheets, possibly for creating new versions or reports.
* **Modul6:** Macros that manage the visibility of columns, manipulate cell values, and perform some data manipulation and copying actions. One is potentially an autoexec macro that is run on the file's opening.
* **Modul7:** A macro performing complex formula replacements within the spreadsheet, potentially altering calculations or data representation.
* **Modul8:** A macro that creates new sheets named "Messwerte BT*" (measurement values), deleting specific ranges within the newly created sheets.
* **Modul9:** An empty macro.
* **Modul10 & Modul11:** Macros for adding and deleting "Tiefbauer" entries, possibly related to subcontractor data. These macros use a password to protect the sheet.
* **Modul12:** A macro to print the sheet named "Lageplan" to a PDF.
* **Modul13:** A macro to compress the Excel workbook, along with an Easter Egg function.
* **Modul14:** A macro to unprotect a sheet, it seems to be incomplete.
* **Modul15 & Modul17:** Macros to import data from other Excel workbooks into the "Daten" (data) and "Berechnung" (calculation) sheets. These macros handle potential errors during the file opening and copying process.
* **Modul18:** Macros to add or delete components, prompting the user for the component number.
* **Modul19:** Macros to delete specific parts of "Bauteil" (component) 1 and 2, prompting the user for confirmation before deleting.
* **Modul20:** Macros to copy values from specific cells on a sheet named "Messwerte KUNDE" to the sheet named "Daten".
* **Modul21:** A macro to insert a picture file ("VrP Fotos") based on cell values from the active sheet. The file path suggests a network location.
* **Modul22:** A macro to import multiple images from a user-selected set of files from the desktop into a spreadsheet, positioning them at defined coordinates.
* **Modul23:** Macros to activate sheets named "Rechnungsaufmaß SUB" and "Rechnungsaufmaß KOMPLETT".
* **Tabelle111124:** This module contains an empty `CommandButton4_Click` subroutine, an empty `Worksheet_SelectionChange` event handler, and a `UserForm_Initialize` event handler that initializes the values of form controls.

Function Summaries

The code consists primarily of Subroutines (procedures that perform actions) rather than functions (procedures that return values). The only function is `Ausgabe()` in Modul13, which is an Easter egg.

* **`Ausgabe()`:** This function takes no parameters. It reads a value from cell `AB87`, displays a message box based on whether the value is "Warren Robinett," and then clears the contents of `AB87`. It serves as a trivial Easter egg.

Control Flow

The control flow is mostly straightforward, utilizing `If-Then-Else` statements for conditional logic and `For Each` loops for iterating through worksheets. Many subroutines follow a pattern of:

1. User confirmation via a `MsgBox`.
2. Conditional execution based on user input (`vbYes` or `vbNo`).
3. Data manipulation (copying, clearing, pasting, formula replacement) on specific cell ranges. The range specifications are quite extensive.
4. Optional message box indicating completion.

The `Tiefbauer` macros (Modul10, Modul11) and macros in Modul 18 and 19 show nested `If-Then-Else` structures to handle multiple conditions. `ImportSK()` and `ImportCL()` demonstrate error handling with `On Error GoTo` statements. `FOTO()` uses nested error handling in a somewhat unusual way.

Data Structures

The primary data structure is the Excel spreadsheet itself. The VBA code interacts with the spreadsheet's cells and ranges, treating them as data storage locations. The code also uses simple variables (integers and strings) to hold temporary values and process data from the spreadsheet cells. The `arrBereiche` array in `BilderImport` is a simple array of strings that are spreadsheet ranges.

Malware Family Suggestion

While the code itself isn't inherently malicious, its functionality raises concerns about potential malicious use:

* **File System Access:** The `Speichern` macro allows saving the workbook to a user-specified location, which could be exploited to overwrite or create files elsewhere on the system. The `BilderImport` macro reads images from the user's desktop. The `ImportSK` and `ImportCL` macros open files specified by the user and can read from them. These are all potential attack vectors for a more sophisticated malware program.
* **Macro-Based Actions:** The extensive use of macros makes it possible to perform a wide range of actions, including data manipulation, file system access, and potentially execution of external code if the macros were modified. The use of `Application.Run` makes the code more difficult to understand.
* **Password Protection:** The use of the password "ddookkuu1234" in the `Tiefbauer` and other macros indicates an attempt to hide or protect the functionality, which is a common obfuscation technique used in malware. However, this password is easily bypassed.
* **Data Exfiltration:** The `Datenbank` macro copies a large amount of data to a sheet named "Datenbank," making it suitable for exfiltrating potentially sensitive information.
* **Obfuscation:** The extensive use of German comments obscures the code's functionality to non-German speakers but does not make it particularly more secure. The repetitive nature of several routines is likely to allow for potential optimization which is unusual for a program designed for security.

Based on these features, the VBA code, if modified maliciously, could be part of a **macro virus** or a more complex **file infector**, potentially with data exfiltration capabilities. Its functionality shows some signs that this VBA code is not just for data management of a project or inventory. The use of `Application.Run`, and potential file system access, along with the password protected routines suggest some attempt to obscure the code. The presence of the Easter Egg is not necessarily an indication of malware but does suggest a possible lack of professionalism in the creation of this code. Without further analysis of the broader context (e.g., the actual data contained in the spreadsheet), it's difficult to definitively classify it as malicious. However, it's certainly suspicious and warrants further investigation.