Analysis Report for: a.vbs

Overall Functionality

The provided code is VBA macro code embedded within an Excel spreadsheet (`.xls`). The primary function, `text()`, manipulates the column widths and inserts new columns in a systematic pattern within a spreadsheet. It then iterates through merged cells, merging additional cells based on specific conditions related to adjacent cell merging and border styles. The code's actions suggest an attempt to reformat the spreadsheet, possibly to conceal or obfuscate data. The presence of hex-encoded strings (`' 0085 62 BOUNDSHEET : Sheet Information - worksheet or dialog sheet, visible - b'T\x00e\x00l\x00e\x00x\x00 \x00R\x00e\x00l\x00e\x00a\x00s\x00e\x00 \x00A\x00p\x00p\x00l\x00i\x00c\x00a\x00t\x00i\x00o\x00n\x00 \x001\x00'`) also hints at possible obfuscation techniques.

Function Summaries

* **`text()`**: This is the main subroutine. It takes no parameters and has no explicit return value. Its purpose is to modify the appearance and structure of an Excel worksheet by adjusting column widths, inserting columns, and merging cells based on specific criteria.

Control Flow

The `text()` subroutine follows a linear flow, broken down into several distinct sections:

1. **Column Width and Insertion:** This section uses repetitive blocks of code to set the column widths of various columns (A, B, C...etc.) and insert new columns using `Columns("X:X").Insert shift:=xlToRight`. The pattern suggests a regular formatting scheme.

2. **Merged Cell Processing (Loop 1):** A `For Each` loop iterates through the `UsedRange` of the active sheet. It identifies merged cells and stores them in the `UR` range variable. The logic checks if a cell is part of a merged area, and if it's not already included in `UR`, it adds it along with its merged area.

3. **Merged Cell Processing (Loop 2):** Another `For Each` loop iterates through the `UR` range (containing the merged cells). This loop checks if cells to the right of a merged cell are also not merged. If they aren't merged, it merges a range including the original merged area and the adjacent two cells to the right.

4. **Border Processing (Loop 3):** The final `For Each` loop again iterates through `UR`. It checks if the left border of each merged cell has a non-null style. If so, it sets the border style to solid, thin, and automatic color.

Data Structures

The primary data structure used is the Excel worksheet itself, manipulated through the VBA object model. Variables like `XR`, `UR`, and `MR` are range objects used to represent selections of cells within the worksheet. `UR` accumulates merged cell areas, and `MR` holds the current merged area being processed in the loop.

Malware Family Suggestion

While the code itself isn't inherently malicious, its function—reformatting a spreadsheet in an unusual and repetitive way—raises suspicion. The code's behavior is consistent with techniques used by macro-based malware to obfuscate its actions or modify the spreadsheet to make it less easily analyzed. The hex-encoded string further supports this suspicion. It's not directly damaging, but it could be part of a larger malicious script designed for data exfiltration, data manipulation, or even to serve as a staging area for more harmful actions triggered later. This type of macro could be classified as a **potential dropper or obfuscator**, part of a larger malicious payload. Further investigation of the surrounding Excel file and its other components is crucial to determine its true nature. Without further context, labeling it definitively as a specific malware family is not possible.