Analysis Report for: b.txt

Overall Functionality

The provided code is VBA (Visual Basic for Applications) macro code embedded within an Excel file (indicated by the file extension and OLE stream names like "ThisWorkbook," "Sheet1," "Sheet2"). The primary functionality is to restrict user interaction with specific cells in "Sheet1." The macros are triggered by the `Worksheet_SelectionChange` event, meaning they are executed whenever a user selects a cell. If a user selects a cell in column 1 (rows 2 and above 4) or column 8 (rows above 4), a beep sounds, the selection moves to the adjacent cell to the right, and a message box appears indicating that the selected cell is read-only. "Sheet2" and "ThisWorkbook" contain empty macros. The `xlm_macro.txt` section suggests the presence of additional data, possibly metadata related to the Excel workbook, but doesn't contain executable code in itself.

Function Summaries

The code only contains one significant subroutine:

* **`Worksheet_SelectionChange(ByVal Target As Range)`**: This is an event handler that executes automatically whenever the user's selection changes within "Sheet1". It takes a `Range` object (`Target`) as input, representing the newly selected cell(s). It has no explicit return value.

Control Flow

The `Worksheet_SelectionChange` subroutine's control flow is primarily based on nested `If-ElseIf-End If` statements:

1. **Check Column 1:** It first checks if the selected cell (`Target`) is in column 1 (`Target.Column = 1`).
2. **Row Checks within Column 1:** If in column 1, it further checks the row number:
- **Row 2:** If the row is 2, it executes the actions (beep, move selection, display message box).
- **Row > 4:** If the row is greater than 4, it also executes the same actions.
3. **Check Column 8:** If the selected cell is not in column 1, it checks if it's in column 8 (`Target.Column = 8`).
4. **Row Checks within Column 8:** If in column 8 and the row is greater than 4, it executes the same actions (beep, move selection, display message box).
5. **No Action:** If none of the above conditions are met, no action is taken.

The actions within each conditional block are always the same: `Beep`, `Cells(Target.Row, Target.Column).Offset(0, 1).Select` (moves selection one cell to the right), and `MsgBox` (displays a read-only message).

Data Structures

The primary data structure used is the `Range` object provided by VBA, representing a cell or a selection of cells within the Excel sheet. The code manipulates this object using its properties (like `.Column`, `.Row`, `.Address`) and methods (like `.Offset`, `.Select`). No other explicit data structures are defined.

Malware Family Suggestion

The VBA macro itself is not inherently malicious; it simply restricts editing of specific cells in an Excel sheet. However, the presence of hex strings in the `xlm_macro.txt` section raises suspicion. These hex strings could be used to obfuscate malicious commands or data. Given the limited information, it's not possible to definitively classify this code as belonging to a specific malware family. However, the overall behavior could be used as part of a larger malware scheme to hide malicious actions or data within an seemingly benign Excel file. Further investigation of the hex strings and the wider context of where this file was found would be necessary. It could be a component of a more complex attack, possibly social engineering or document-based malware, aiming to control specific information in a spreadsheet or avoid detection. Consider this potentially suspicious and deserving of further analysis with more advanced tools.