Analysis Report for: a.txt

Overall Functionality

The VBA code within the provided Excel file (`5F33B77C81454AADAE60720149F4C648.xls`) exhibits behavior consistent with malicious activity. It performs a series of actions that manipulate the Excel sheet's contents, including deleting rows, pasting data, and selecting specific ranges. The repeated patterns and the inclusion of seemingly arbitrary strings like "■■■" (K■ Lìdá), "■■1" (Module 1), "■■" (Header), and others suggest obfuscation. Crucially, the code repeatedly modifies the `VBProject.VBComponents` property and `VBProject.Remove`, strongly indicating VBA stomping—a technique used to hide malicious code by altering the VBA project itself. The code appears designed to remove specific sheets ("PROFORMA INVOICE (2)", "■■■", "■■■2", "packing list") or parts of them after performing some operations, suggesting an attempt to hide its actions and exfiltrate information.

Function Summaries

The code contains three main subroutines: `msoTrue`, `id_029E`, and `id_02A0`. All three share a strikingly similar structure:

* **`msoTrue`**: This function initializes variables (`Range`, `B`, `Selection`, `Copy`) and performs actions based on the state of `Selection`. It manipulates ranges, pastes data (potentially images and other things based on string names), and deletes rows. It also exhibits VBA stomping behavior. The exact purpose is obfuscated but likely involves data manipulation and cleanup.

* **`id_029E`**: Similar to `msoTrue`, but operates on a different set of sheet names ("■■■2" instead of "PROFORMA INVOICE (2)"). Again, involves data manipulation, pasting, deleting and VBA stomping. It seems to process another sheet or section of the workbook.

* **`id_02A0`**: This function focuses on different sheet names ("packing list" and "PROFORMA INVOICE"). It also features similar data manipulation, pasting, and row deletion. The VBA stomping is also present. This could be manipulating a packing list type of sheet.

Control Flow

The control flow in each function is relatively straightforward, but the obfuscation makes the precise purpose difficult to determine:

1. **Initialization**: Variables are initialized (often with seemingly arbitrary string constants).
2. **Conditional Logic**: `If` statements often check the value of `Selection` against specific string values ("1521").
3. **Data Manipulation**: `ArgsMemCall` is heavily used, suggesting calls to Excel object methods to select, paste, and delete ranges of cells.
4. **Looping**: `ForEach` loops iterate over collections, likely sheets or shapes, to perform actions on each item. The "For Each" loops could be used to locate specific sheets to manipulate.
5. **VBA Stomping**: Repeatedly disables and enables the VB components before and after the manipulations are performed. The disabling is achieved using `LitVarSpecial(False)` and `MemSt VBComponents`. Enabling uses `LitVarSpecial(True)` and `MemSt VBComponents`. The removal of components is achieved using `LitVarSpecial(False)` and `MemSt Remove`. Enabling uses `LitVarSpecial(True)` and `MemSt Remove`.

Data Structures

The primary data structures are implicit in the Excel object model used by the VBA code. The code works with ranges (`Range`), shapes (`Shapes`), cells (`Cells`), rows (`Rows`), sheets (`ActiveSheet`), and collections of these objects. The code relies on the inherent structure of the Excel file and doesn't explicitly define custom data structures.

Malware Family Suggestion

Based on the analysis, this VBA macro shows characteristics of a **macro virus** or a component of a more sophisticated malware family. The VBA stomping strongly suggests an attempt to conceal its actions, making reverse engineering more difficult. The pattern of deleting rows and manipulating specific sheet names points to information stealing or data destruction as potential malicious goals. The code's complexity and obfuscation techniques also make it consistent with professional malware development efforts (possibly a dropper or installer). Further analysis, especially decoding the strings and examining the precise object method calls and DLL calls, could reveal more. The strings suggest it might target specific accounting software or template usage.