Document File Analysis 2 (Deobfuscation)

By: Ryan Stewart

Reviewing the olevba output:

[olevba filename.doc > filename.vba]

*Use Visual Studio Code (VS Code) a source-code editor developed by Microsoft.

The next command we'll use is:

```
olevba --deobf --reveal [filename.vba] > filename_deobf.vba
```

This command utilizes the `olevba` tool with the `--deobf` and `--reveal` options to **deobfuscate** and reveal the content of a VBA file named `filename.vba`. The output is then redirected to a new file named `filename_deobf.vba`.

We will examine the deobfuscated file for Indicators of Compromise (IOCs), including but not limited to PowerShell commands, names of executable files, ping activities, URLs, and specific strings.

Why should we deobfuscate?

Deobfuscation is essential for several reasons:

1. **Understanding Code Intent:** Obfuscated code is intentionally made complex and challenging to comprehend. Deobfuscation helps reveal the original intent and functionality of the code, allowing for a clearer understanding of its operations.
2. **Security Analysis:** In cybersecurity, deobfuscation is crucial for analyzing potentially malicious code. It enables security analysts to uncover hidden threats, identify vulnerabilities, and develop effective countermeasures.

3. **Incident Response:** During incident response activities, deobfuscation is necessary to unravel obscured details within code or scripts. This aids in the investigation of security incidents and helps organizations respond effectively to cyber threats.
4. **Forensic Analysis:** Deobfuscation is valuable in digital forensics when examining artifacts containing obfuscated code. It assists forensic analysts in reconstructing the original code and understanding the actions taken by attackers.
5. **Malware Research:** Researchers often encounter obfuscated code in the study of malware. Deobfuscating such code is vital for uncovering the malware's functionality, identifying its propagation methods, and developing effective countermeasures.

Why would adversaries want to obfuscate their malware?

Adversaries employ obfuscation techniques in malware for several strategic reasons:

Obfuscation is a technique used in software development and cybersecurity to deliberately make code, data, or communication unclear or difficult to understand. The primary purpose of obfuscation is not to secure the system but to deter reverse engineering, tampering, or unauthorized access. This process involves making the code intentionally convoluted or complex without changing its functionality, making it challenging for someone to comprehend or analyze the original source.

Obfuscation is commonly employed in various domains, such as:

1. **Software Protection:** Developers may obfuscate their code to protect intellectual property and deter unauthorized copying or reverse engineering.
2. **Malware Defense:** Malicious actors often use obfuscation to hide the true intent of their code, making it harder for security analysts and antivirus programs to detect and analyze malicious activities.
3. **Code Compression:** Obfuscation techniques can be used to compress code, reducing its size and potentially improving performance.
4. **Code Minification:** In web development, obfuscation is often applied to reduce the size of JavaScript files for faster loading times in web applications.

Common obfuscation techniques include renaming variables and functions to nonsensical names, inserting unnecessary code or control structures, encrypting or encoding strings, and using various code transformations to make the original code less human-readable.

Let's execute the command: `vmonkey --iocs [filename.vba]`

"Finally, we'll utilize 'Vmonkey,' a tool purposefully crafted for the in-depth analysis of Visual Basic macros. It is tailored to meet the requirements of security researchers and analysts aiming to understand and evaluate macro behavior without actual execution. Macro analysis is a fundamental practice in cybersecurity, playing a critical role in identifying potential threats. This significance becomes particularly pronounced when examining malicious documents or emails that may conceal harmful code.