Malicious Documents Analysis Lab Guide

soundblaster

Lab Solutions

What type of file is this? How do you know?

The first two bytes of the file are `50 4B` ("PK") which indicates it is a ZIP archive. The file's unzipped contents contain a word directory, which indicates the file is an OOXML Word document.

What is the document extension type? How do you know?

The ContentType for Word document part `/word/document.xml` is `application/vnd.ms-word.document.macroEnabled.main+xml`, which indicates that the document is a macro-enabled document (`.docm`).

Are there any macros? How do you know? If any macros are present, what do they do?

Yes, the document contains a `vbaProject.bin` part with `ContentType` `application/vnd.ms-office.vbaProject`, indicating the presence of VBA macros. The macros contain multiple auto execution entry points. Only the `AutoOpen` entrypoint is valid. The VBA macros use the SAPI API to vocalize "FLARE RULES"

Lab Walkthrough

What type of file is this? How do you know?

Open the file in *010 Editor* and examine the file signature. The first two bytes of the file are `50 4B` ("PK") which indicates it is a ZIP archive.

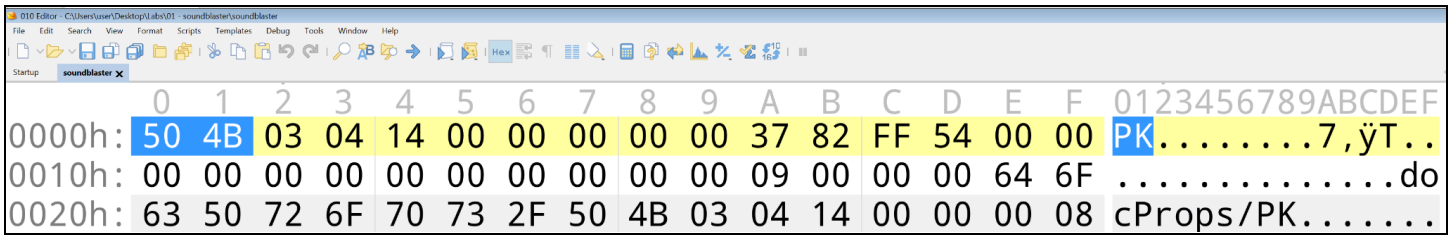

Figure 1: Hex dump of beginning of binary includes file signature

Use *7-Zip* to extract the archive contents. Right-click on the file and select *7-Zip - Extract Here*. The decompressed contents are now accessible.

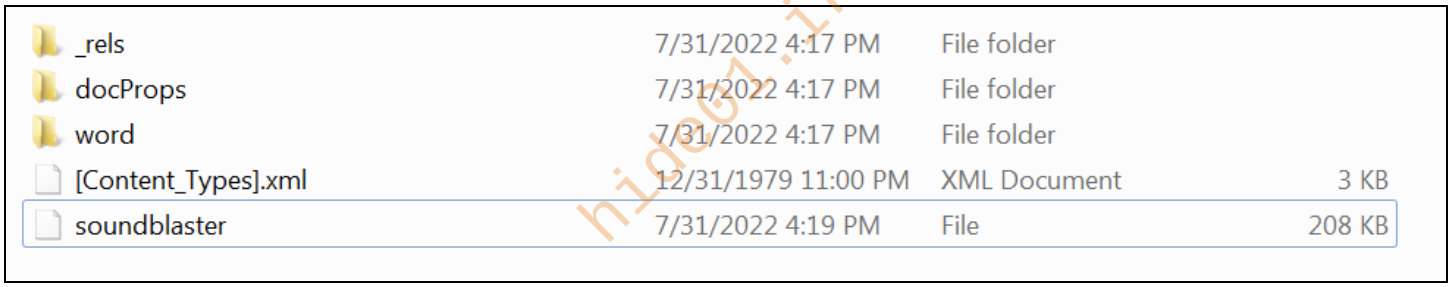

FIgure 2: Directory contents now include extracted content

The file's unzipped contents contain a word directory, which indicates the file is an OOXML Word document.

What is the document extension type? How do you know?

Right-click on the file `[Content_Types].xml` and select *Open with Code*. Now open *CyberChef* and paste the file contents into the input panel. Activate the operation *XML Beautify*. Copy the contents of the output window and paste them back into the original file in *VS Code*. This modifies the XML to include spacing to make it more human readable. *VS Code* applies syntax highlighting to make interpretation easier.

[Content_Types].xml

C: > Users > user > Desktop > Labs > 01 - soundblaster > [Content_Types].xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Types
    xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
    <Default Extension="bin" ContentType="application/vnd.ms-office.vbaProject"/>
    <Default Extension="png" ContentType="image/png"/>
    <Default Extension="rels" ContentType="application/vnd.openxmlformats-package.relationships+xml"/>
    <Default Extension="xml" ContentType="application/xml"/>
    <Override PartName="/word/document.xml" ContentType="application/vnd.ms-word.document.macroEnabled.main+xml"/>
    <Override PartName="/word/vbaData.xml" ContentType="application/vnd.ms-word.vbaData+xml"/>
    <Override PartName="/word/styles.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.styles+xml"/>
    <Override PartName="/word/settings.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.settings+xml"/>
    <Override PartName="/word/webSettings.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.webSettings+xml"/>
    <Override PartName="/word/footnotes.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.footnotes+xml"/>
    <Override PartName="/word/endnotes.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.endnotes+xml"/>
    <Override PartName="/word/header1.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.header+xml"/>
    <Override PartName="/word/header2.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.header+xml"/>
    <Override PartName="/word/footer1.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.footer+xml"/>
    <Override PartName="/word/footer2.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.footer+xml"/>
    <Override PartName="/word/header3.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.header+xml"/>
    <Override PartName="/word/footer3.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.footer+xml"/>
    <Override PartName="/word/fontTable.xml" ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.fontTable+xml"/>
    <Override PartName="/word/theme/theme1.xml" ContentType="application/vnd.openxmlformats-officedocument.theme+xml"/>
    <Override PartName="/docProps/core.xml" ContentType="application/vnd.openxmlformats-package.core-properties+xml"/>
    <Override PartName="/docProps/app.xml" ContentType="application/vnd.openxmlformats-officedocument.extended-properties+xml"/>
</Types>
```

Figure 3: Beautified XML viewed in VS Code with syntax highlighting

Observe line 8. It describes the part `/word/document.xml` as having `ContentType` `application/vnd.ms-word.document.macroEnabled.main+xml`. This indicates that the document is a macro-enabled document (`.docm`).

Are there any macros? How do you know? If any macros are present, what do they do?

Open the file in Word. Do not click the Enable Content button, which would allow the macros to execute. Instead, navigate to View - Macros and select Edit.

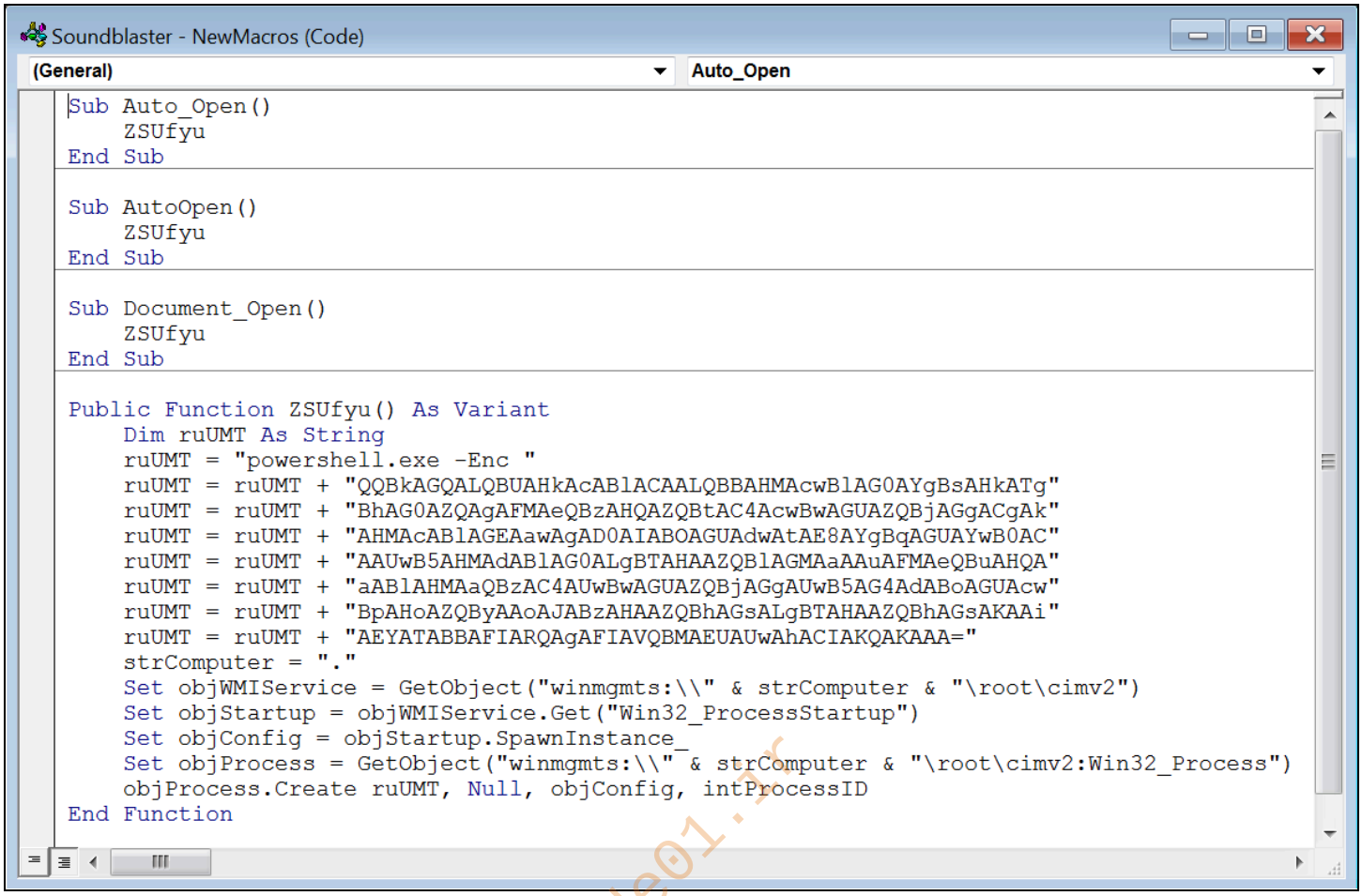

Figure 4: Macros viewed in Word

The subroutines `Auto_Open`, `AutoOpen`, and `Document_Open` each invoke the function `ZSUfyu`. Among these entrypoints, only `AutoOpen` will execute – `Auto_Open` is only valid for Excel documents and event handlers like `Document_Open` must be defined within the `ThisDocument` class to be valid.

The function `ZSUfyu` builds a string `ruUMT` that includes Base64 text. `objProcess` is a Windows Management Instrumentation (WMI) object that represents a process. It is used to launch a new process. The process is a cmd.exe command - "`powershell.exe -Enc <Base64_string>`". This launches a PowerShell process that decodes `<Base64_string>` and runs it.

You can decode the Base64 manually or use the debugger to decode dynamically. To decode manually, copy the relevant lines and paste them into CyberChef. Use the operations "*Find / Replace*", "*Remove whitespace*", "*From Base64*", and "*Decode text*". Remember to use the `\` character to escape special characters.

Example CyberChef URL (paste into FLARE VM web browser URL input):

```
file:///C:/ProgramData/chocolatey/lib/cyberchef.flare/tools/cyberchef.html#recipe=Find_/_Replace(%7B'option':'Regex','string':'ruUMT%20%3D%20ruUMT%20%5C%5C%2B%20'%7D,'',true,false,true,false)Find_/_Replace(%7B'option':'Regex','string':'%5C%5C%22'%7D,'',true,false,true,false)Remove_whitespace(true,true,true,true,true,false)From_Base64('A-Za-z0-9%2B/%3D',true,false)Decode_text('UTF-16LE%20(1200)')
```

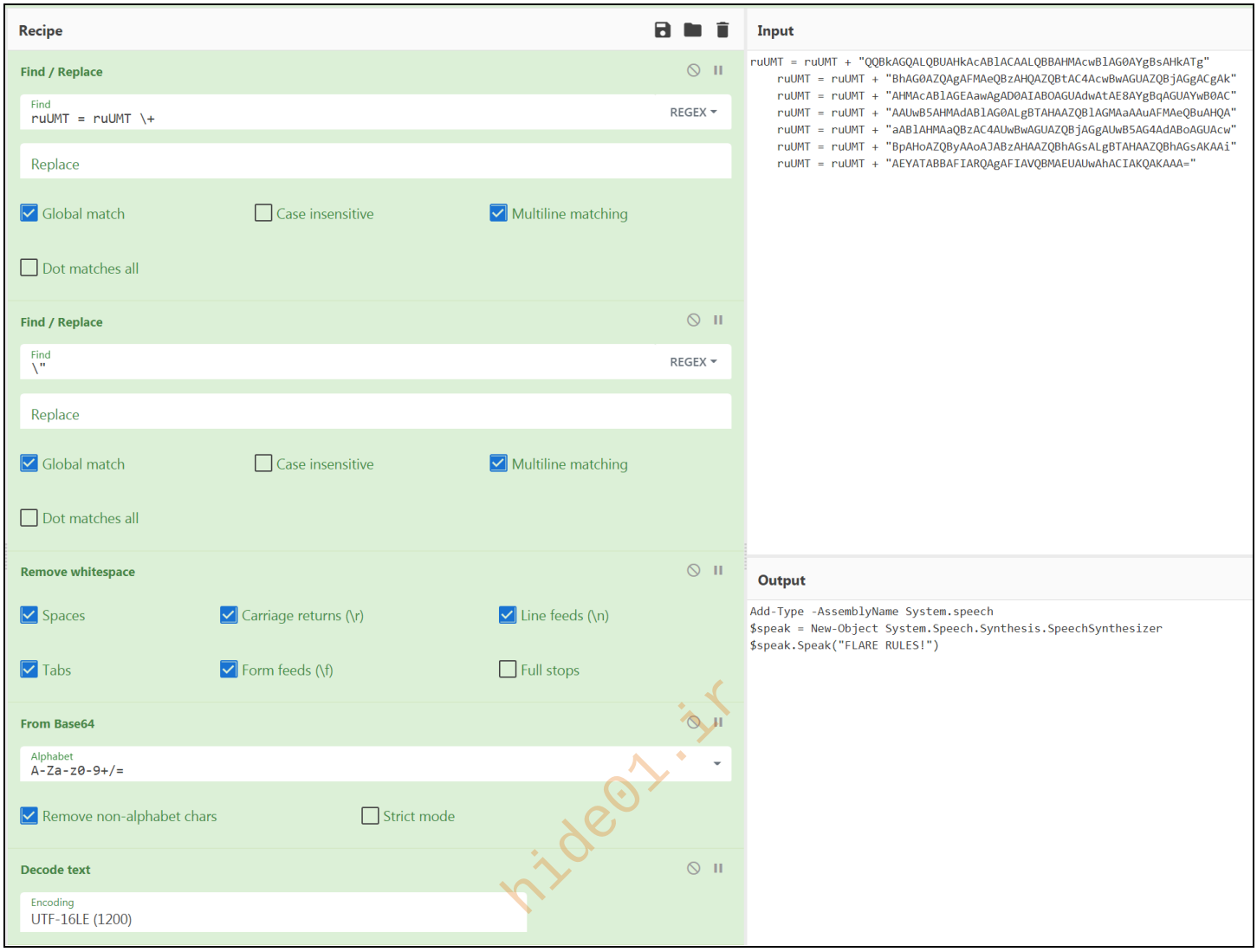

Figure 5: Cyberchef recipe to decode Base64 sequence

In this example we use "*Find / Replace*" to remove unwanted text, "*Remove whitespace*" to remove spaces and newlines, "*From Base64*" to decode the Base64, and "*Decode text*" to interpret the resulting text as UTF-16LE (Unicode little-endian).

The decoded PowerShell:

```
Add-Type -AssemblyName System.speech
$speak = New-Object System.Speech.Synthesis.SpeechSynthesizer
$speak.Speak("FLARE RULES!")
```

The VBA macros create a .NET object which is used to vocalize "FLARE RULES".