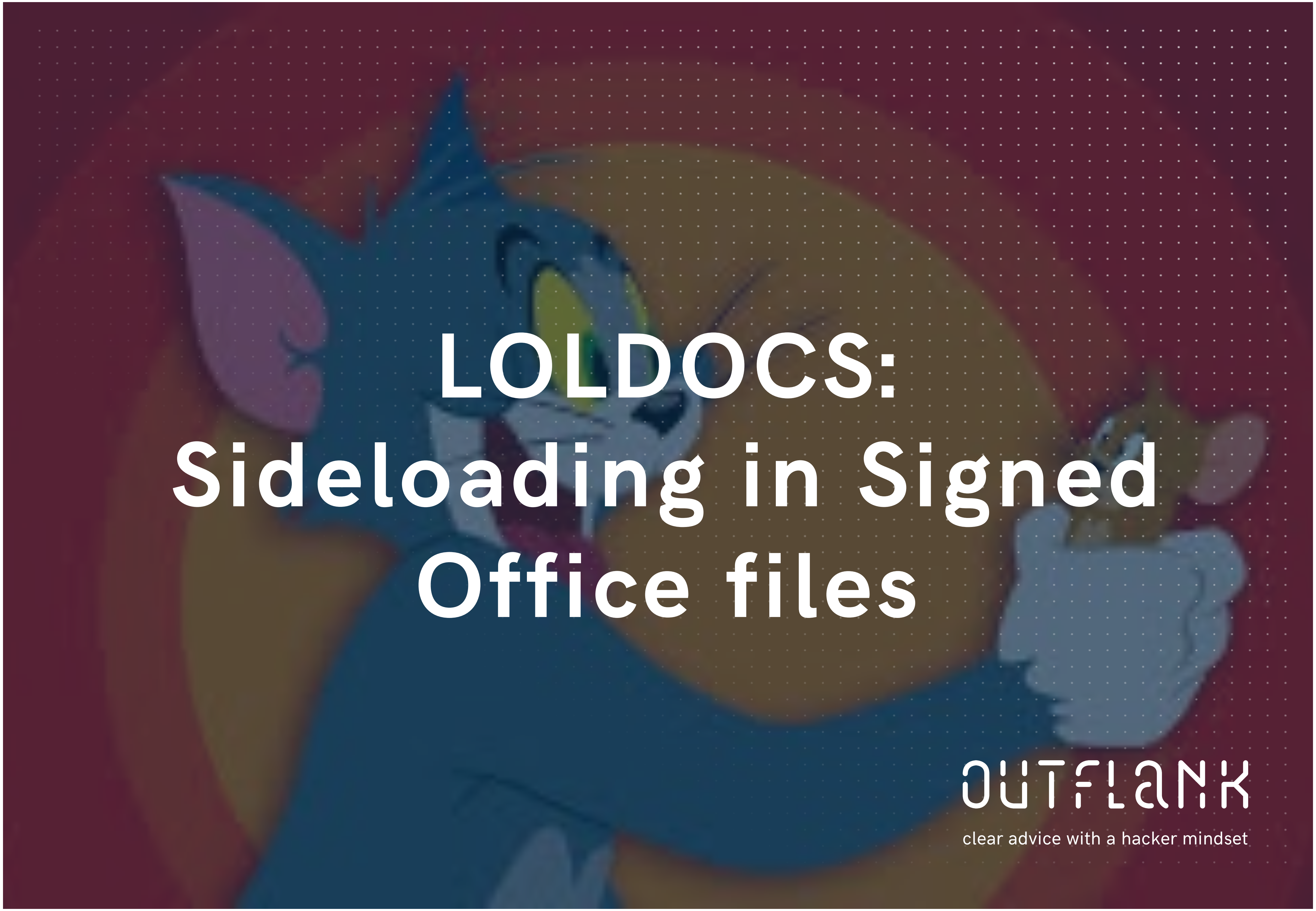
LOLDOCS: Sideloading in Signed Office files



OUTFLANK

clear advice with a hacker mindset

About Dima

Dima van de Wouw @DaWouw

Red Teamer and Offensive Developer

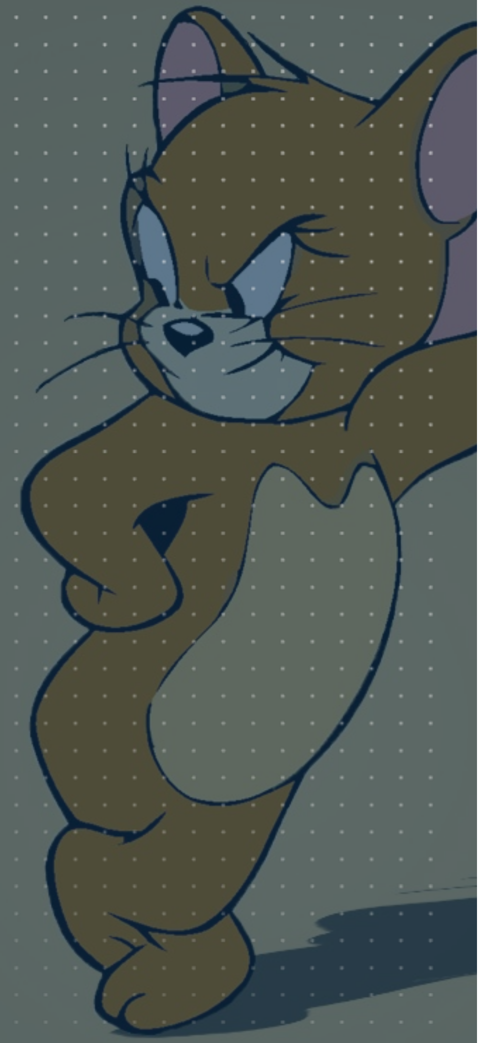
Enjoys building things that break things and using them in operations

About Pieter

Pieter Ceelen @ptrpieter

Red Teamer & Offensive R&D

I am an MS Office power-user and like clicking all buttons



INTRO

- Companies moving to 'signed macro's'
- We located a vulerability in a Microsoft signed document
- What other vulnerabilities are there?

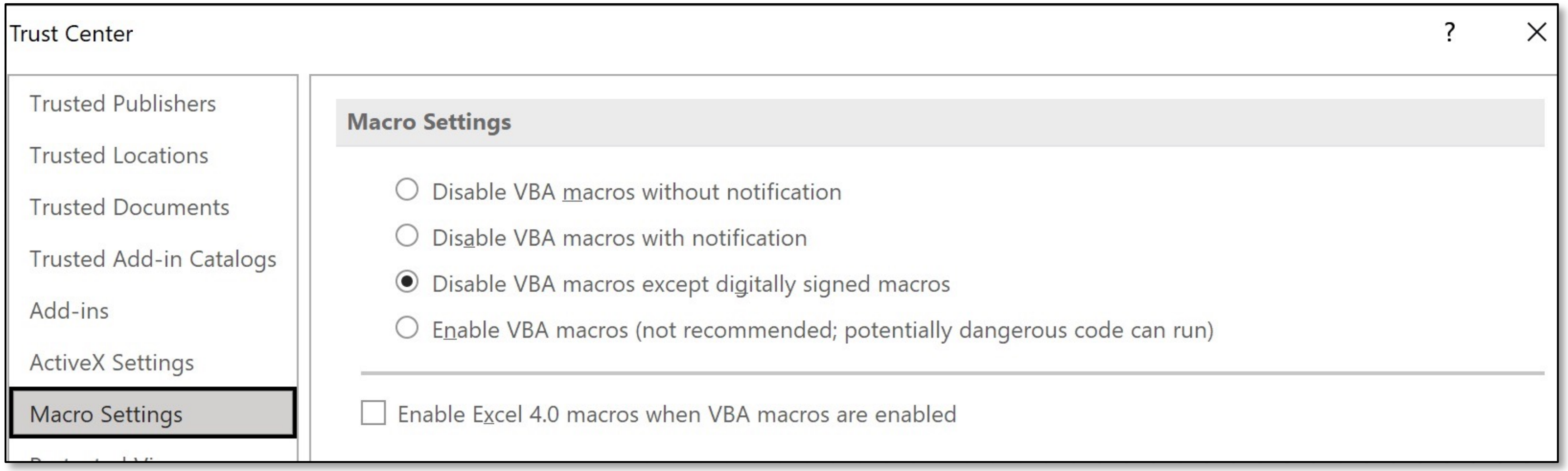
- This is about Office, not only as a phishing vector
- Not everything is patched, in process of disclosing

Hardening Against Macro’s

OUTFLANK

clear advice with a hacker mindset

SETTINGS: ONLY DIGITALLY SIGNED MACROS

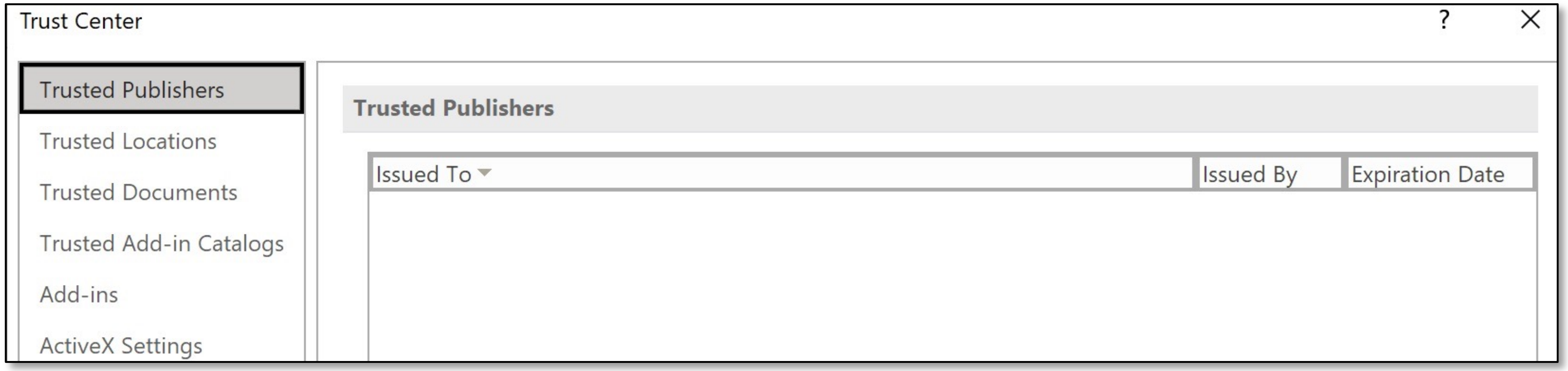

Trust Center

Trusted Publishers
Trusted Locations
Trusted Documents
Trusted Add-in Catalogs
Add-ins
ActiveX Settings
Macro Settings

Macro Settings

- ○ Disable VBA macros without notification
- ○ Disable VBA macros with notification
- ● Disable VBA macros except digitally signed macros
- ○ Enable VBA macros (not recommended; potentially dangerous code can run)

☐ Enable Excel 4.0 macros when VBA macros are enabled


Trust Center

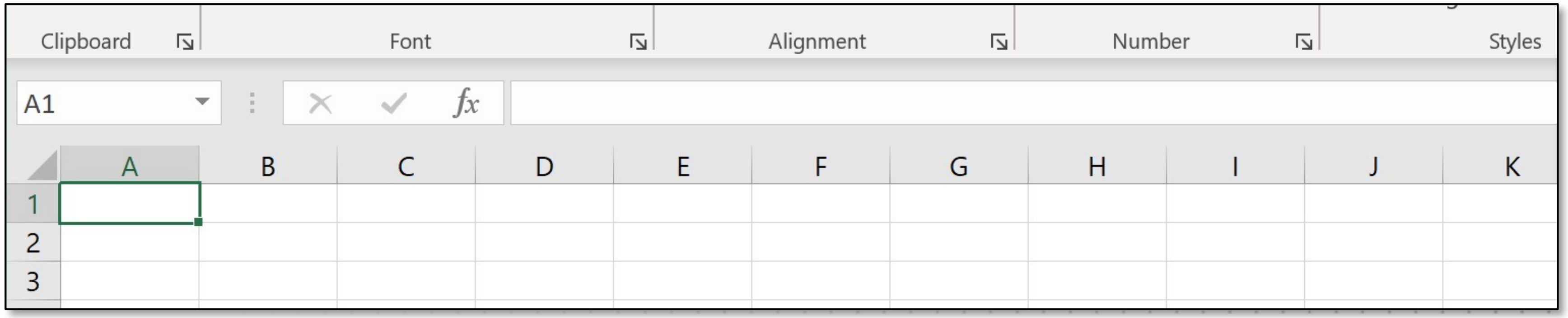
Trusted Publishers
Trusted Locations
Trusted Documents
Trusted Add-in Catalogs
Add-ins
ActiveX Settings

Trusted Publishers

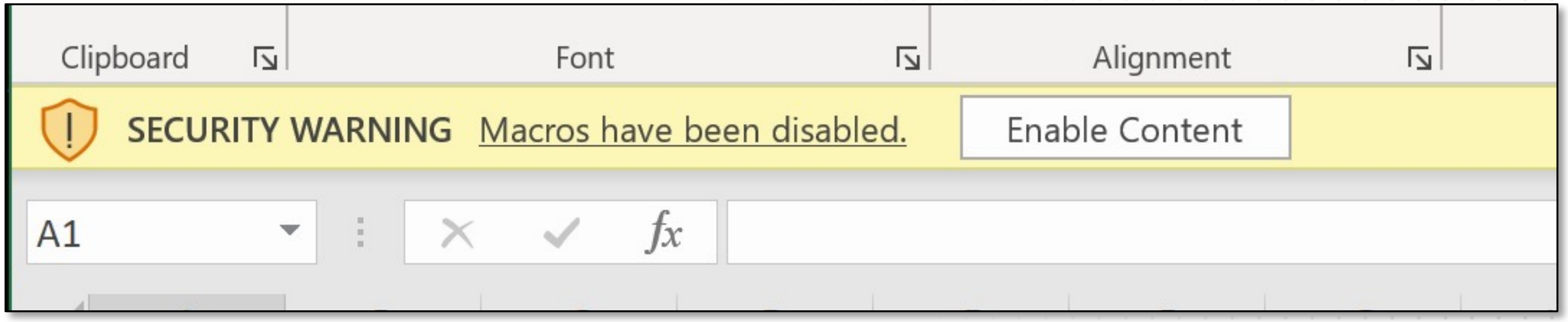
Issued To	Issued By	Expiration Date

OFFICE GUI BEHAVIOUR

Unsigned document - No warning – No execution

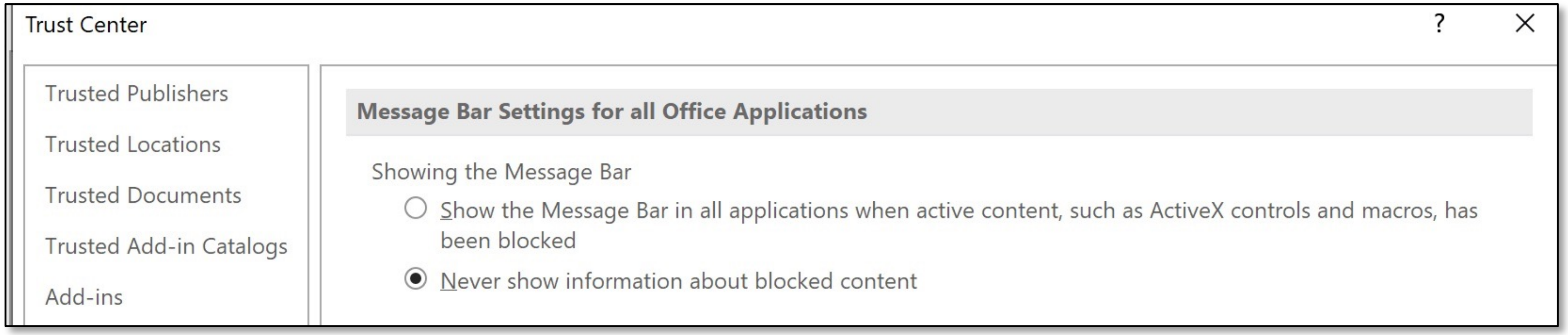


Self-signed document – Yellow bar – User executed



SECURITY WARNING Macros have been disabled. Enable Content

SETTINGS: REMOVE THE MESSAGE BAR


Trust Center

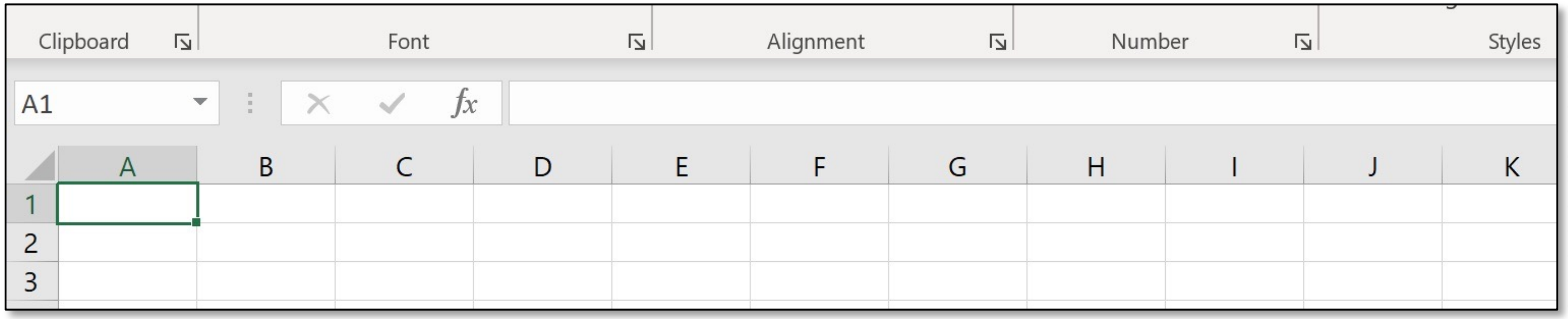
- Trusted Publishers
- Trusted Locations
- Trusted Documents
- Trusted Add-in Catalogs
- Add-ins

Message Bar Settings for all Office Applications

Showing the Message Bar

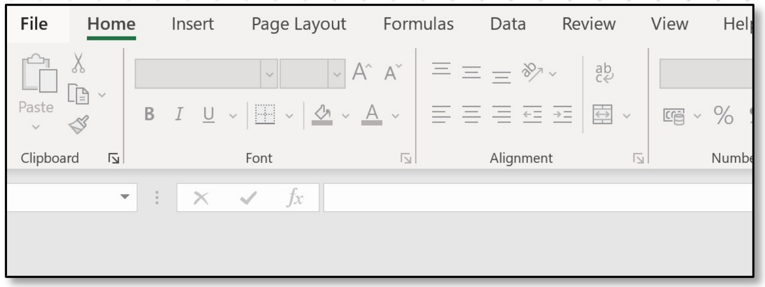
- ○ Show the Message Bar in all applications when active content, such as ActiveX controls and macros, has been blocked
- ● Never show information about blocked content

Self-signed document – No warning - No execution



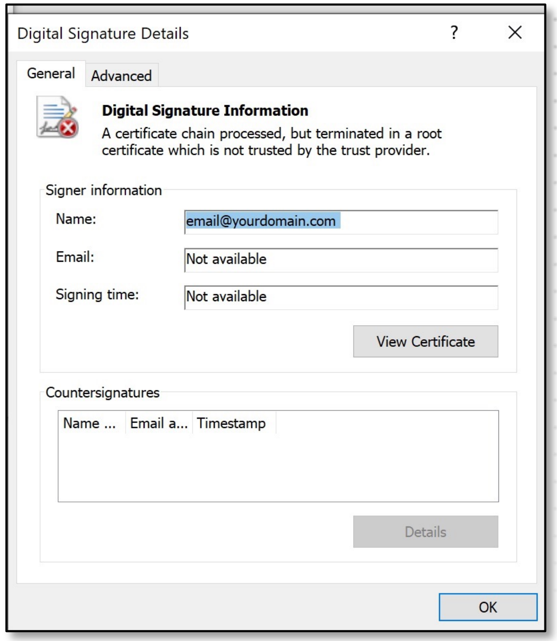
XLA/XLAM – EXCEL ADD-INS

Unsigned xlam – No warning - No execution



Self-Signed xlam – No bar but a dialogue – User executed

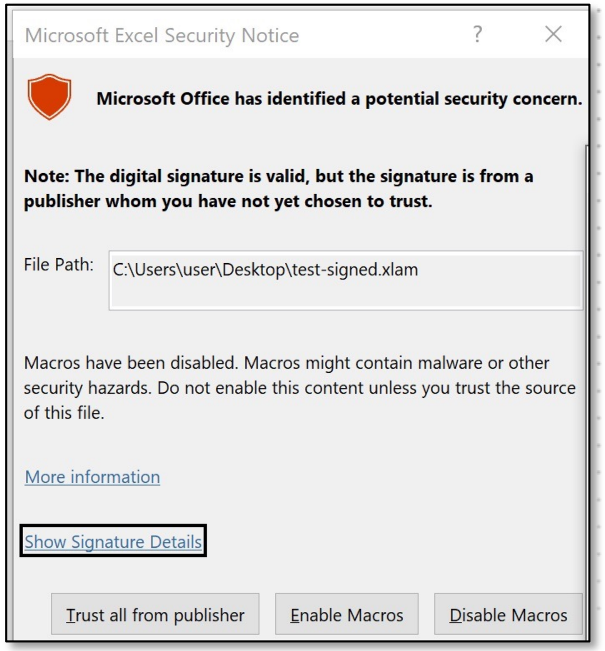
Transpose XLSM->XLAM



Digital Signature Information
A certificate chain processed, but terminated in a root certificate which is not trusted by the trust provider.

Signer information

Name:	email@yourdomain.com
Email:	Not available
Signing time:	Not available



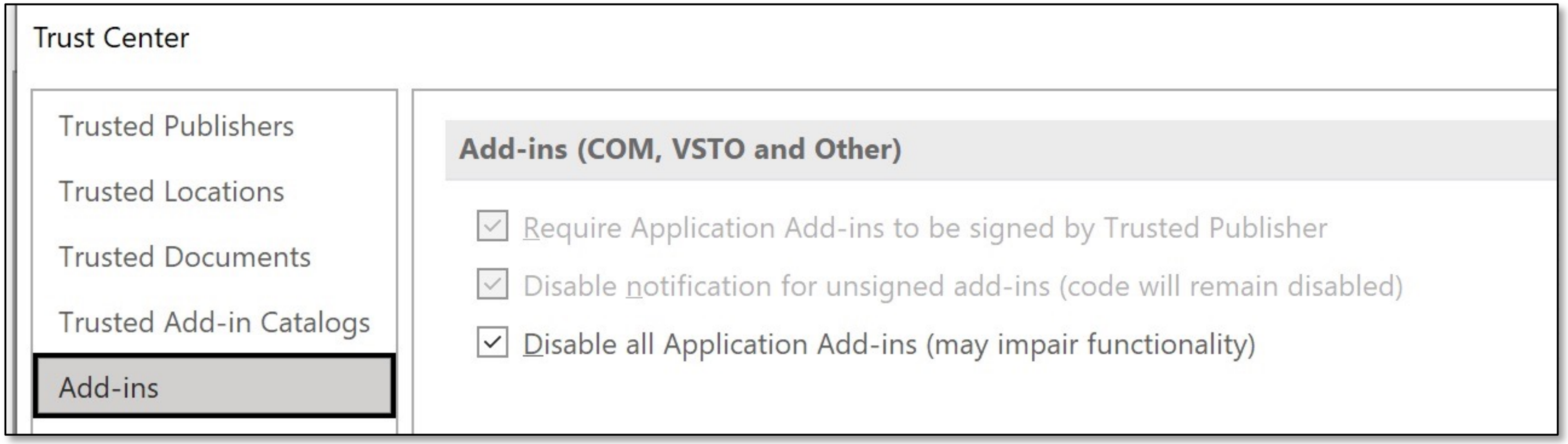
Microsoft Office has identified a potential security concern.

Note: The digital signature is valid, but the signature is from a publisher whom you have not yet chosen to trust.

File Path: C:\Users\user\Desktop\test-signed.xlam

Macros have been disabled. Macros might contain malware or other security hazards. Do not enable this content unless you trust the source of this file.

SETTINGS: ADD-IN CONFIGURATIONS



Trust Center

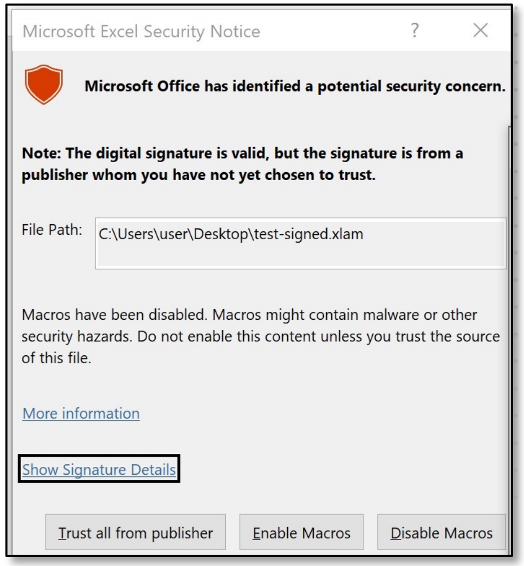
- Trusted Publishers
- Trusted Locations
- Trusted Documents
- Trusted Add-in Catalogs
- Add-ins

Add-ins (COM, VSTO and Other)

- [✓] Require Application Add-ins to be signed by Trusted Publisher
- [✓] Disable notification for unsigned add-ins (code will remain disabled)
- [✓] Disable all Application Add-ins (may impair functionality)

Now opening a Self-Signed XLA/XLAM...

Still a dialogue!!



Microsoft Excel Security Notice

Microsoft Office has identified a potential security concern.

Note: The digital signature is valid, but the signature is from a publisher whom you have not yet chosen to trust.

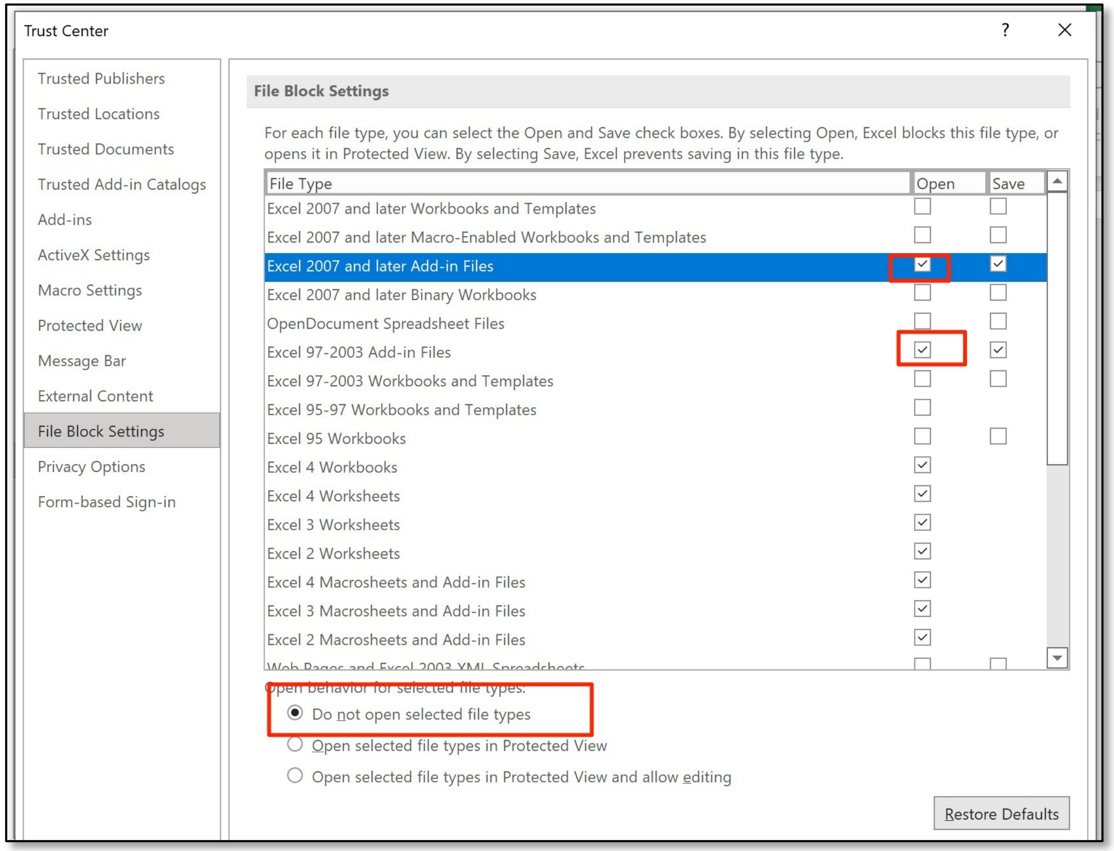
File Path: C:\Users\user\Desktop\test-signed.xlam

Macros have been disabled. Macros might contain malware or other security hazards. Do not enable this content unless you trust the source of this file.

More information

Show Signature Details

BLOCKING XLA/XLAM FILES – FOR GOOD!



Trust Center

Trusted Publishers
Trusted Locations
Trusted Documents
Trusted Add-in Catalogs
Add-ins
ActiveX Settings
Macro Settings
Protected View
Message Bar
External Content
File Block Settings
Privacy Options
Form-based Sign-in

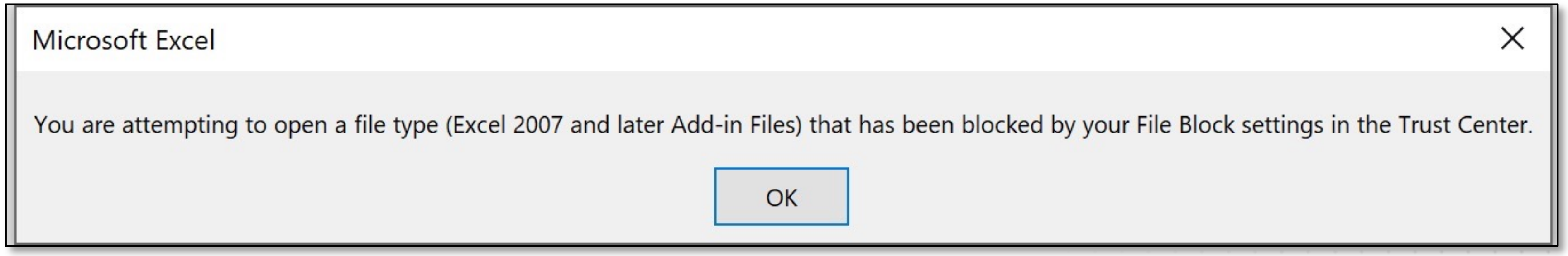
File Block Settings

For each file type, you can select the Open and Save check boxes. By selecting Open, Excel blocks this file type, or opens it in Protected View. By selecting Save, Excel prevents saving in this file type.

File Type	Open	Save
Excel 2007 and later Workbooks and Templates	☐	☐
Excel 2007 and later Macro-Enabled Workbooks and Templates	☐	☐
Excel 2007 and later Add-in Files	✓	✓
Excel 2007 and later Binary Workbooks	☐	☐
OpenDocument Spreadsheet Files	☐	☐
Excel 97-2003 Add-in Files	✓	✓
Excel 97-2003 Workbooks and Templates	☐	☐
Excel 95-97 Workbooks and Templates	☐	
Excel 95 Workbooks	☐	☐
Excel 4 Workbooks	✓	
Excel 4 Worksheets	✓	
Excel 3 Worksheets	✓	
Excel 2 Worksheets	✓	
Excel 4 Macrosheets and Add-in Files	✓	
Excel 3 Macrosheets and Add-in Files	✓	
Excel 2 Macrosheets and Add-in Files	✓	
Web Pages and Excel 2003 XML Spreadsheets	☐	☐

Open behavior for selected file types:

[✓ Do not open selected file types]
Open selected file types in Protected View
Open selected file types in Protected View and allow editing



Microsoft Excel

You are attempting to open a file type (Excel 2007 and later Add-in Files) that has been blocked by your File Block settings in the Trust Center.

RELEVANCE & RECAP

Takeaways

- XLAM’s are cool
- Settings are complex, companies will fail

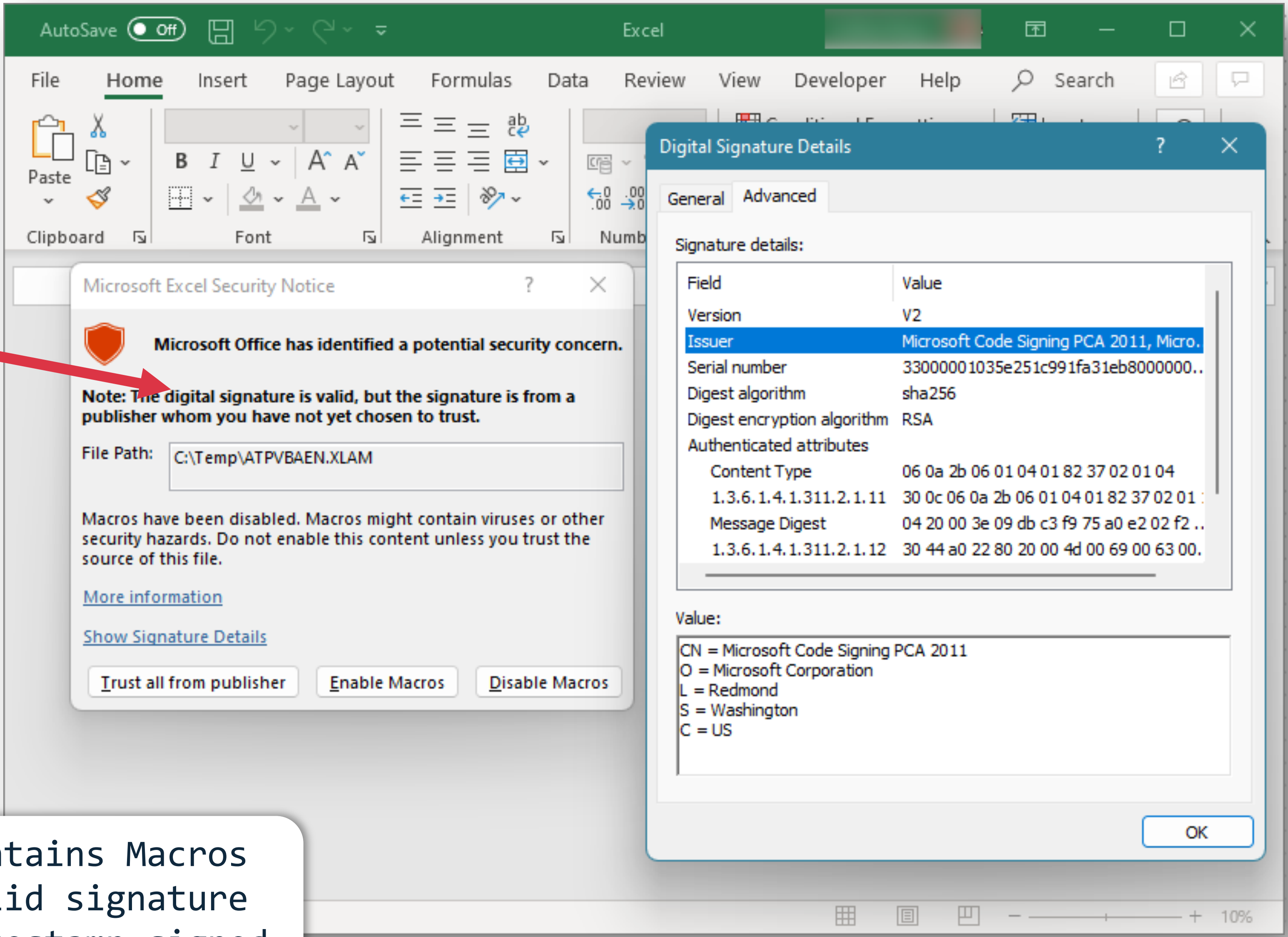
Pattern 1:

Read from cells - MS signed XLAMs

OUTFLANK

clear advice with a hacker mindset

SIGNED BY MICROSOFT



Microsoft Excel Security Notice

Microsoft Office has identified a potential security concern.

Note: The digital signature is valid, but the signature is from a publisher whom you have not yet chosen to trust.

File Path: C:\Temp\ATPVBAEN.XLAM

Macros have been disabled. Macros might contain viruses or other security hazards. Do not enable this content unless you trust the source of this file.

More information

Show Signature Details

Trust all from publisher | Enable Macros | Disable Macros

Digital Signature Details

General | Advanced

Signature details:

Field	Value
Version	V2
Issuer	Microsoft Code Signing PCA 2011, Micro...
Serial number	33000001035e251c991fa31eb8000000...
Digest algorithm	sha256
Digest encryption algorithm	RSA
Authenticated attributes	
Content Type	06 0a 2b 06 01 04 01 82 37 02 01 04
1.3.6.1.4.1.311.2.1.11	30 0c 06 0a 2b 06 01 04 01 82 37 02 01
Message Digest	04 20 00 3e 09 db c3 f9 75 a0 e2 02 f2 ..
1.3.6.1.4.1.311.2.1.12	30 44 a0 22 80 20 00 4d 00 69 00 63 00.

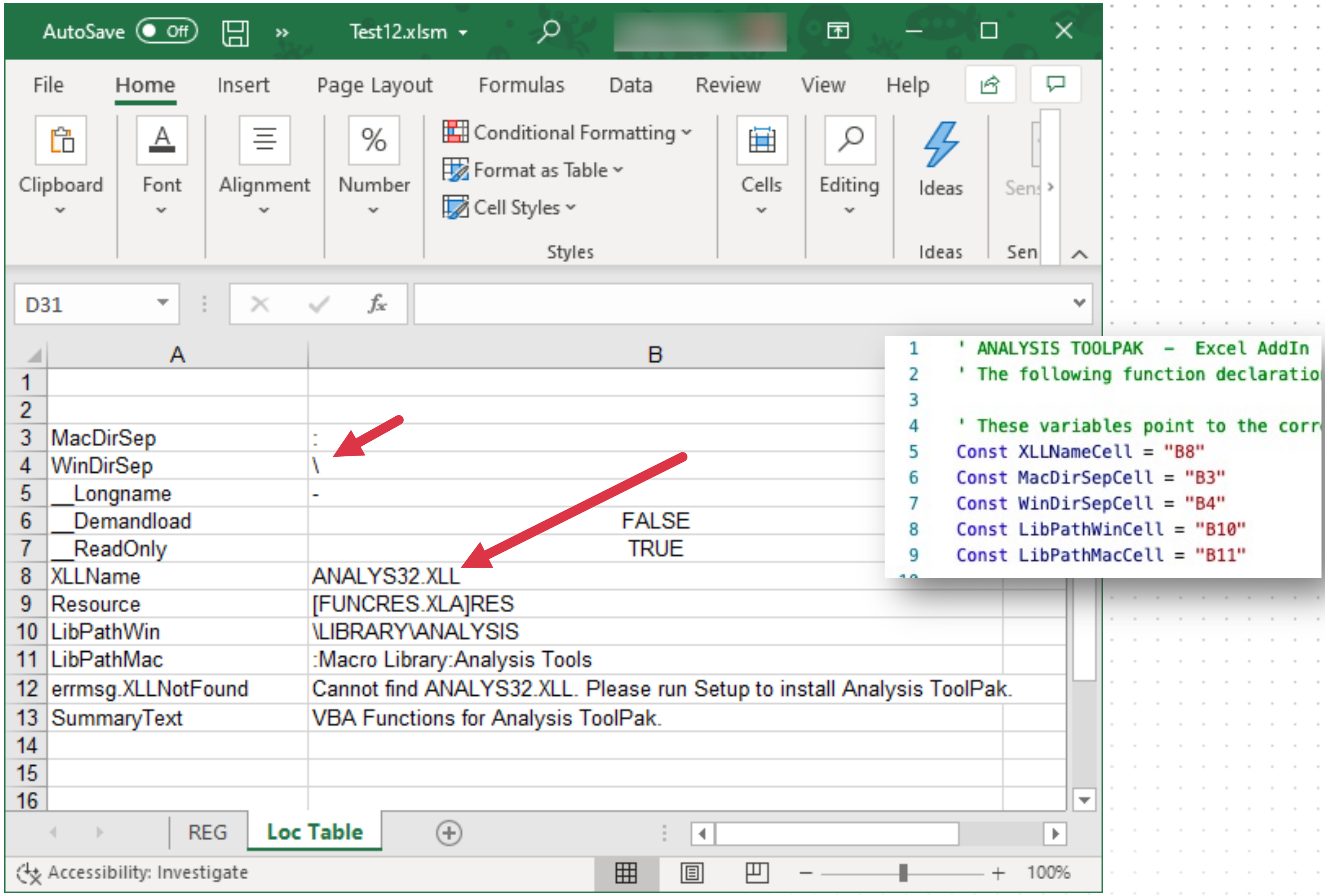
Value:

```
CN = Microsoft Code Signing PCA 2011
O = Microsoft Corporation
L = Redmond
S = Washington
C = US
```

OK

```
Contains Macros
Valid signature
Timestamp signed
```

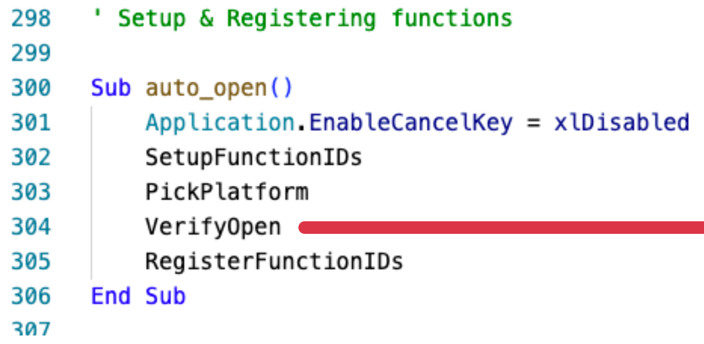
VARIABLE VALUES IN CELLS


AutoSave Off
Test12.xlsm
File Home Insert Page Layout Formulas Data Review View Help
Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles Styles Cells Editing Ideas
D31
REG Loc Table
Accessibility: Investigate
100%

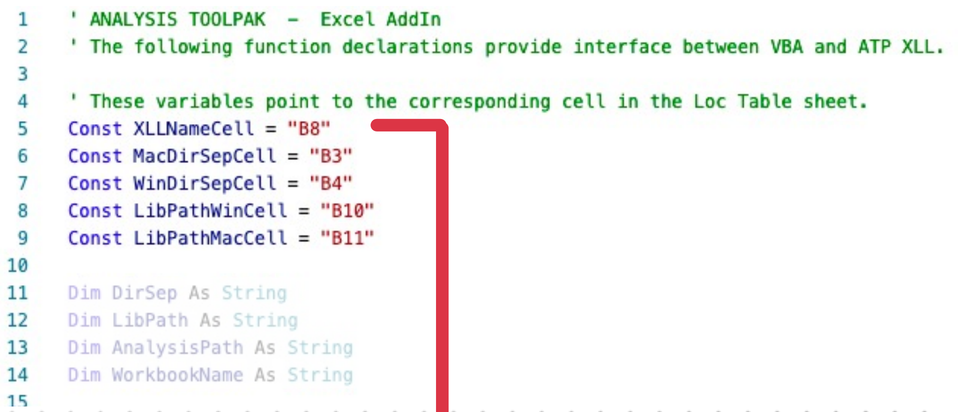
	A	B
1		
2		
3	MacDirSep	:
4	WinDirSep	\
5	__Longname	-
6	__Demandload	FALSE
7	__ReadOnly	TRUE
8	XLLName	ANALYS32.XLL
9	Resource	[FUNCRES.XLA]RES
10	LibPathWin	\LIBRARY\ANALYSIS
11	LibPathMac	:Macro Library:Analysis Tools
12	errmsg.XLLNotFound	Cannot find ANALYS32.XLL. Please run Setup to install Analysis ToolPak.
13	SummaryText	VBA Functions for Analysis ToolPak.
14		
15		
16		

```
' ANALYSIS TOOLPAK  -  Excel AddIn
' The following function declaratio

' These variables point to the corr
Const XLLNameCell = "B8"
Const MacDirSepCell = "B3"
Const WinDirSepCell = "B4"
Const LibPathWinCell = "B10"
Const LibPathMacCell = "B11"
```

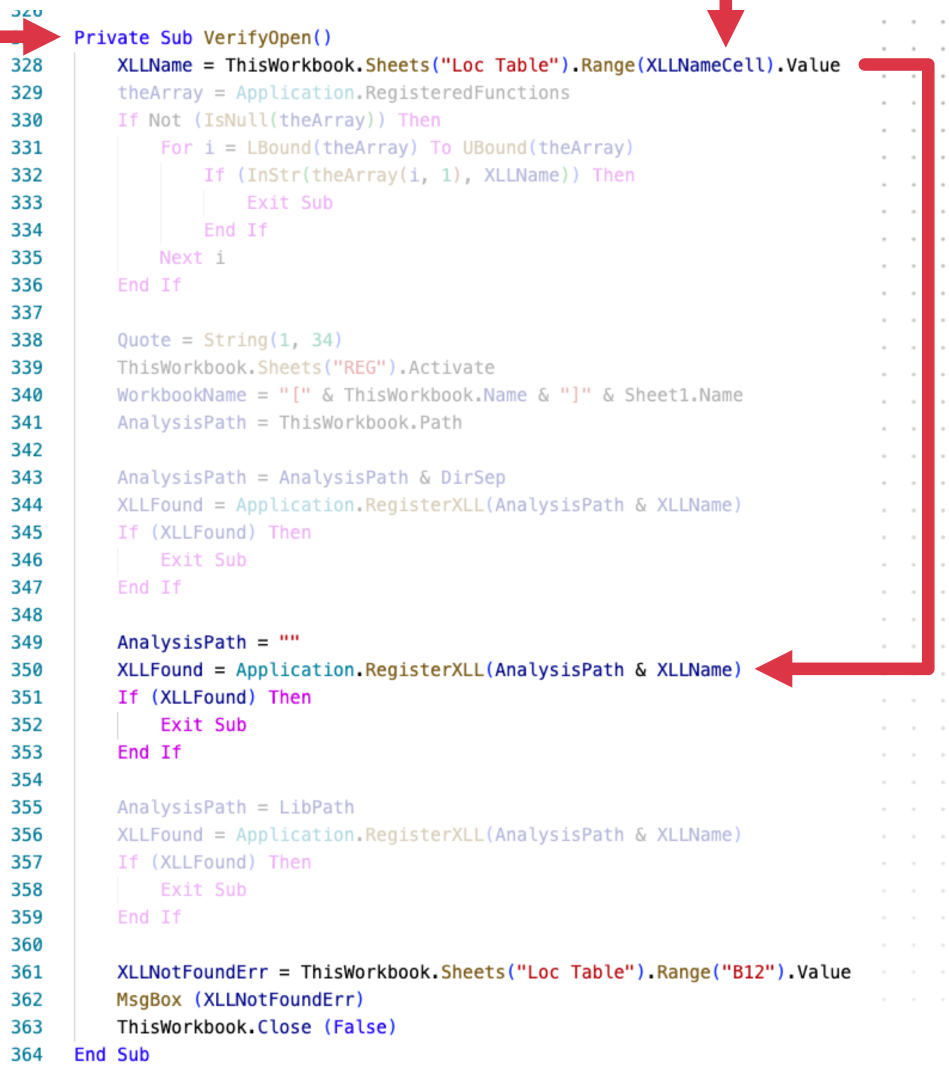
FLOW TO LOAD AN XLL



```
' Setup & Registering functions

Sub auto_open()
    Application.EnableCancelKey = xlDisabled
    SetupFunctionIDs
    PickPlatform
    VerifyOpen
    RegisterFunctionIDs
End Sub
```



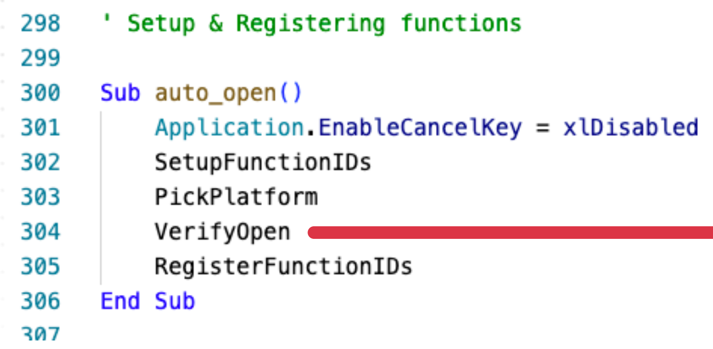
```
' ANALYSIS TOOLPAK  -  Excel AddIn
' The following function declarations provide interface between VBA and ATP XLL.

' These variables point to the corresponding cell in the Loc Table sheet.
Const XLLNameCell = "B8"
Const MacDirSepCell = "B3"
Const WinDirSepCell = "B4"
Const LibPathWinCell = "B10"
Const LibPathMacCell = "B11"

Dim DirSep As String
Dim LibPath As String
Dim AnalysisPath As String
Dim WorkbookName As String
```



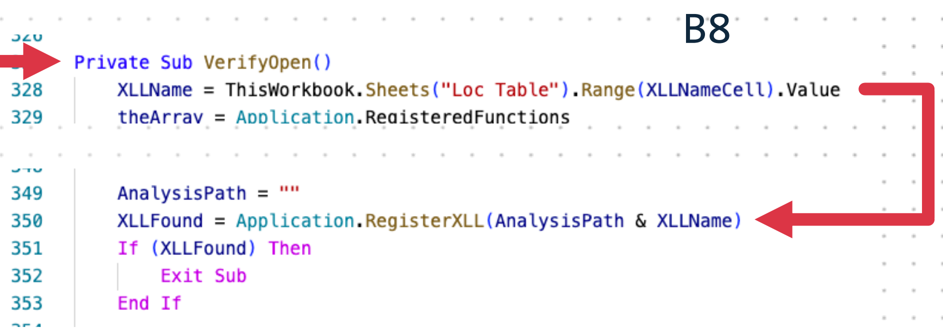
```
Private Sub VerifyOpen()
    XLLName = ThisWorkbook.Sheets("Loc Table").Range(XLLNameCell).Value
    theArray = Application.RegisteredFunctions
    If Not (IsNull(theArray)) Then
        For i = LBound(theArray) To UBound(theArray)
            If (InStr(theArray(i, 1), XLLName)) Then
                Exit Sub
            End If
        Next i
    End If

    Quote = String(1, 34)
    ThisWorkbook.Sheets("REG").Activate
    WorkbookName = "[" & ThisWorkbook.Name & "]" & Sheet1.Name
    AnalysisPath = ThisWorkbook.Path

    AnalysisPath = AnalysisPath & DirSep
    XLLFound = Application.RegisterXLL(AnalysisPath & XLLName)
    If (XLLFound) Then
        Exit Sub
    End If

    AnalysisPath = ""
    XLLFound = Application.RegisterXLL(AnalysisPath & XLLName)
    If (XLLFound) Then
        Exit Sub
    End If

    AnalysisPath = LibPath
    XLLFound = Application.RegisterXLL(AnalysisPath & XLLName)
    If (XLLFound) Then
        Exit Sub
    End If

    XLLNotFoundErr = ThisWorkbook.Sheets("Loc Table").Range("B12").Value
    MsgBox (XLLNotFoundErr)
    ThisWorkbook.Close (False)
End Sub
```

- Code runs on open
- XLLName is obtained from cell B8
- Code loads XLL using RegisterXLL function with XLLName as input

PRACTICAL ABUSE



```
' Setup & Registering functions

Sub auto_open()
    Application.EnableCancelKey = xlDisabled
    SetupFunctionIDs
    PickPlatform
    VerifyOpen
    RegisterFunctionIDs
End Sub
```



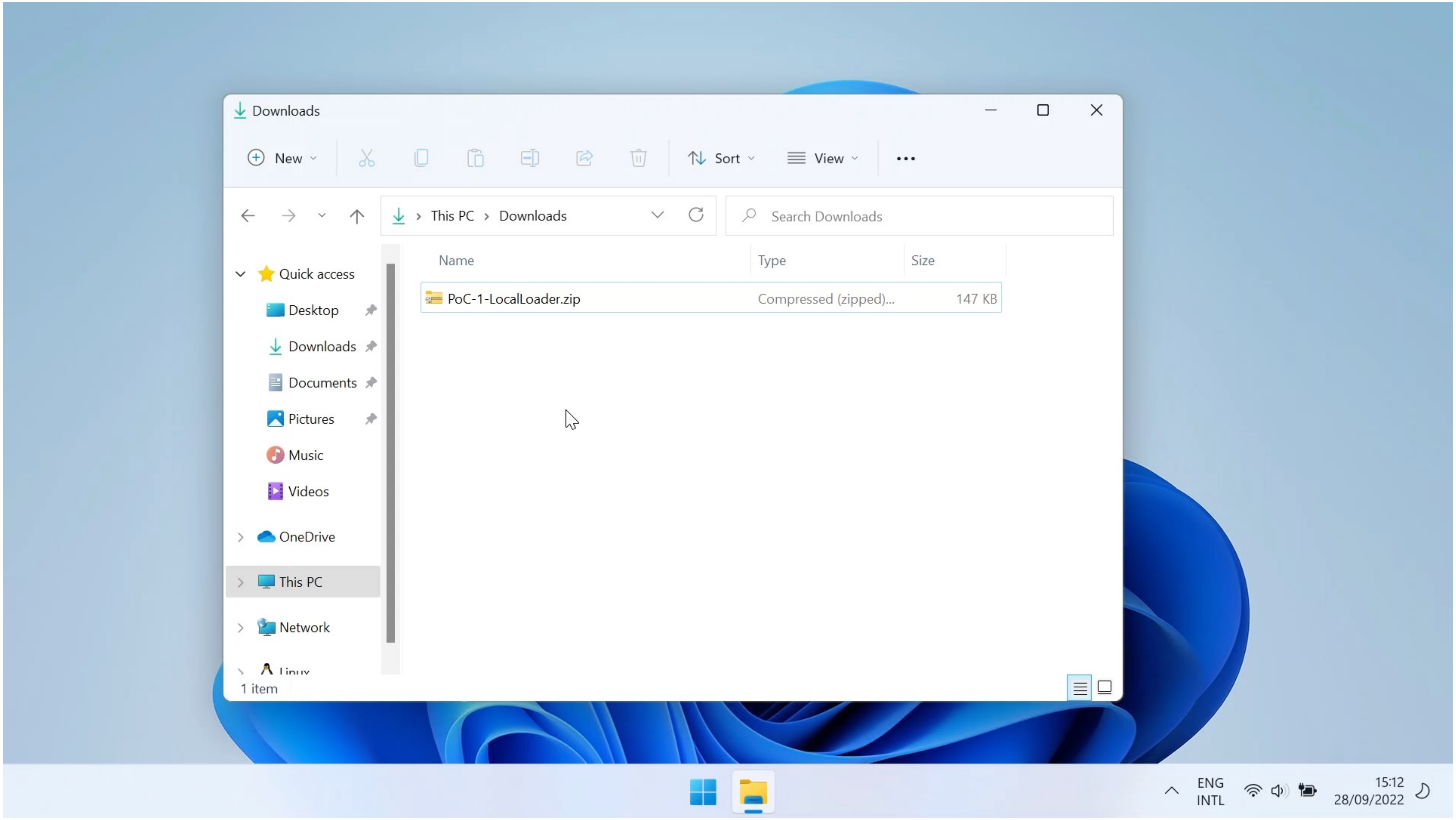
```
Private Sub VerifyOpen()
    XLLName = ThisWorkbook.Sheets("Loc Table").Range(XLLNameCell).Value
    theArray = Application.RegisteredFunctions

    AnalysisPath = ""
    XLLFound = Application.RegisterXLL(AnalysisPath & XLLName)
    If (XLLFound) Then
        Exit Sub
    End If
```

Original B8=
ANALYS32.XLL

PoC 1 - B8=
demo64.dat
Success: Different extension, XLL loads!

PoC 2 - B8=
= "demo" & IF(ISERROR(SEARCH("64";INFO("OSVERSION"))); "32"; "64") & ".dat"
Success: Formulas work!
demo64.dat for x64, demo32.dat for x86 process!

DEMO



Name	Type	Size
PoC-1-LocalLoader.zip	Compressed (zipped)...	147 KB

RECAP

- Microsoft signed XLAM depended on cell contents
  - RegisterXLL (Load XLL) & ExecuteExcel4Macro (Excel4 abuse)

- Using signed files 'out of context'
- Owned if you had MS as 'trusted publisher'

- Patch for CVE-2021-28449
  - Fix input validation
  - No signing downgrade: Unsigned XLL's are not loaded from signed VBA
  - Complex: Files are timestamp signed, no easy option to revoke
  - Microsoft also patched other files...

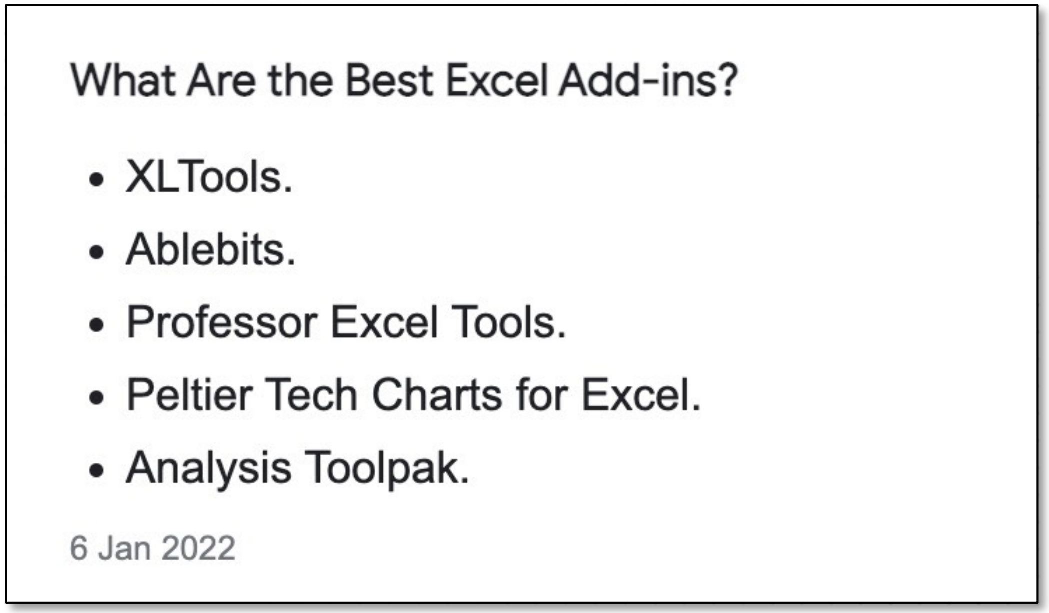
Open question: What other vulnerable signed files are there!

More
Signed Files

OUTFLANK

clear advice with a hacker mindset

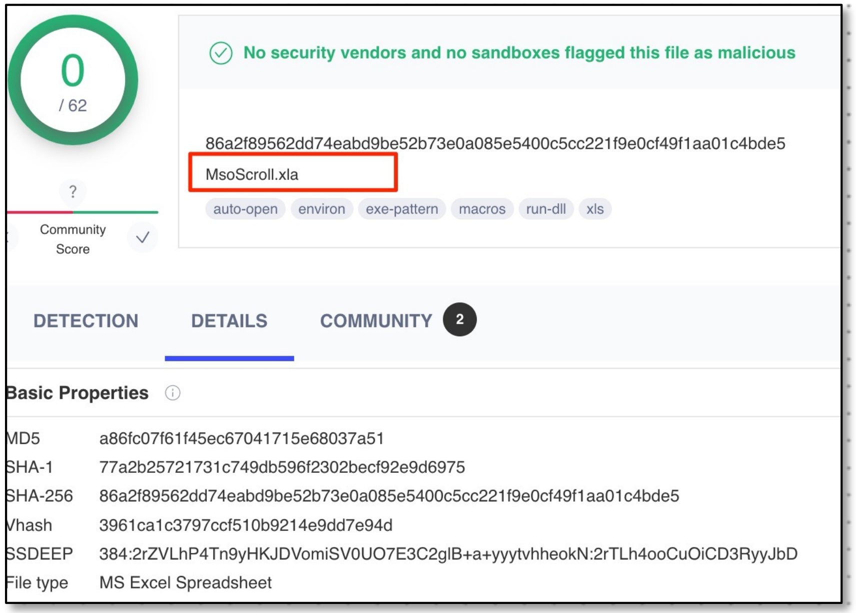
LOCATING SIGNED CONTENT



What Are the Best Excel Add-ins?

- XLTools.
- Ablebits.
- Professor Excel Tools.
- Peltier Tech Charts for Excel.
- Analysis Toolpak.

6 Jan 2022



0 / 62

No security vendors and no sandboxes flagged this file as malicious

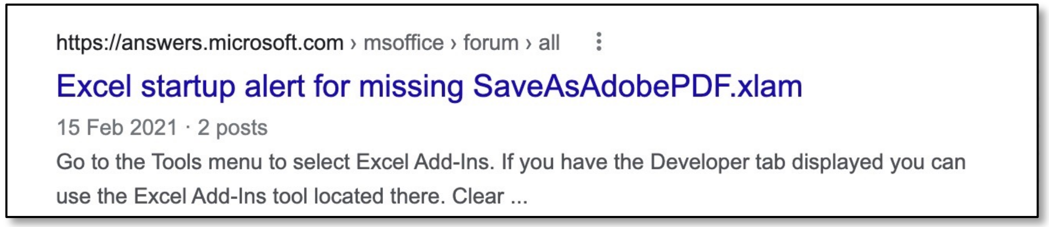
86a2f89562dd74eabd9be52b73e0a085e5400c5cc221f9e0cf49f1aa01c4bde5

MsoScroll.xla

auto-open environ exe-pattern macros run-dll xls

Basic Properties

MD5	a86fc07f61f45ec67041715e68037a51
SHA-1	77a2b25721731c749db596f2302becf92e9d6975
SHA-256	86a2f89562dd74eabd9be52b73e0a085e5400c5cc221f9e0cf49f1aa01c4bde5
Vhash	3961ca1c3797ccf510b9214e9dd7e94d
SSDEEP	384:2rZVLhP4Tn9yHKJDVomiSV0UO7E3C2glB+a+yyytvhheokN:2rTLh4ooCuOiCD3RyyJbD
File type	MS Excel Spreadsheet

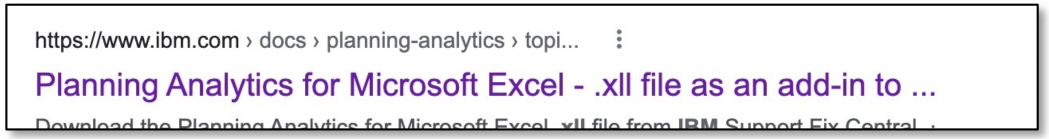


https://answers.microsoft.com › msoffice › forum › all

Excel startup alert for missing SaveAsAdobePDF.xlam

15 Feb 2021 · 2 posts

Go to the Tools menu to select Excel Add-Ins. If you have the Developer tab displayed you can use the Excel Add-Ins tool located there. Clear ...

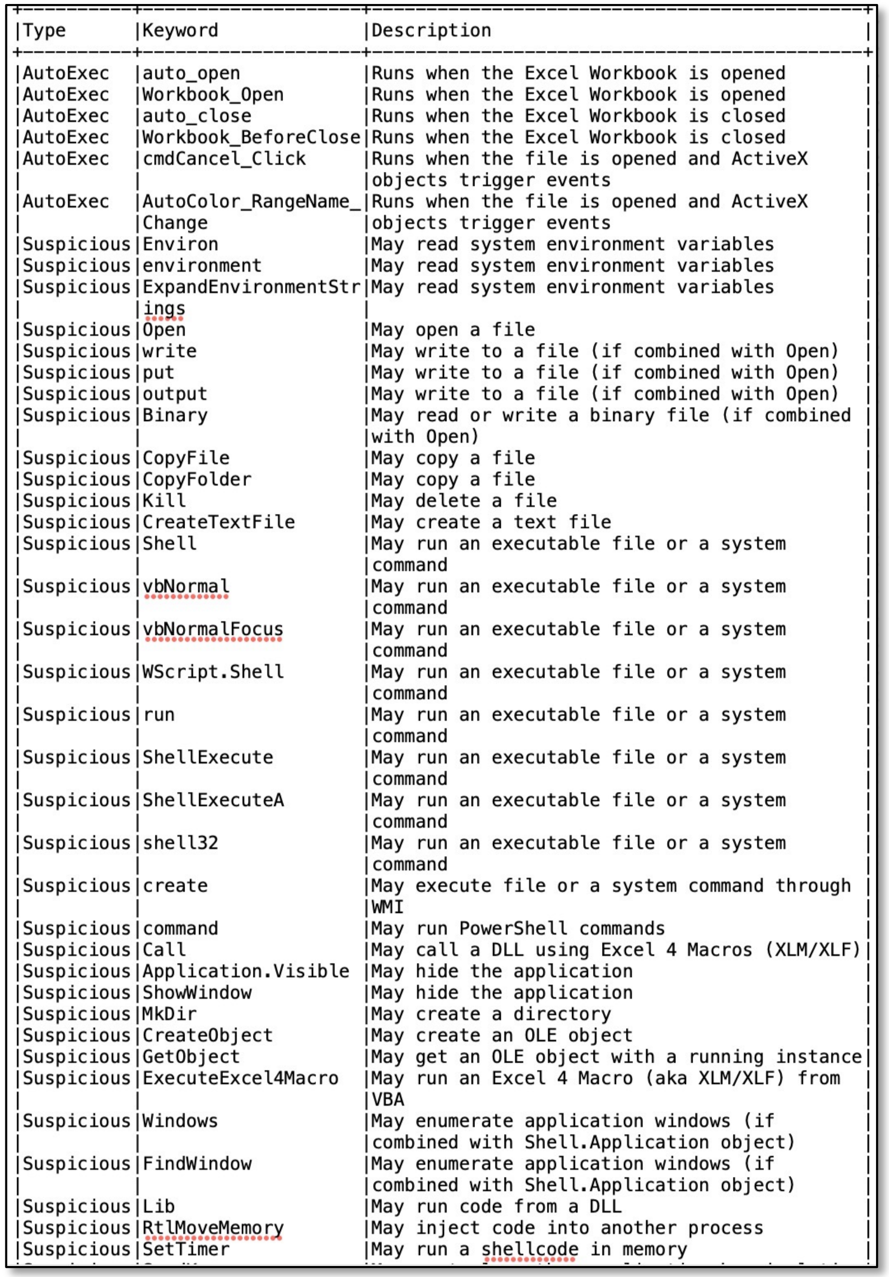


https://www.ibm.com › docs › planning-analytics › topi...

Planning Analytics for Microsoft Excel - .xll file as an add-in to ...

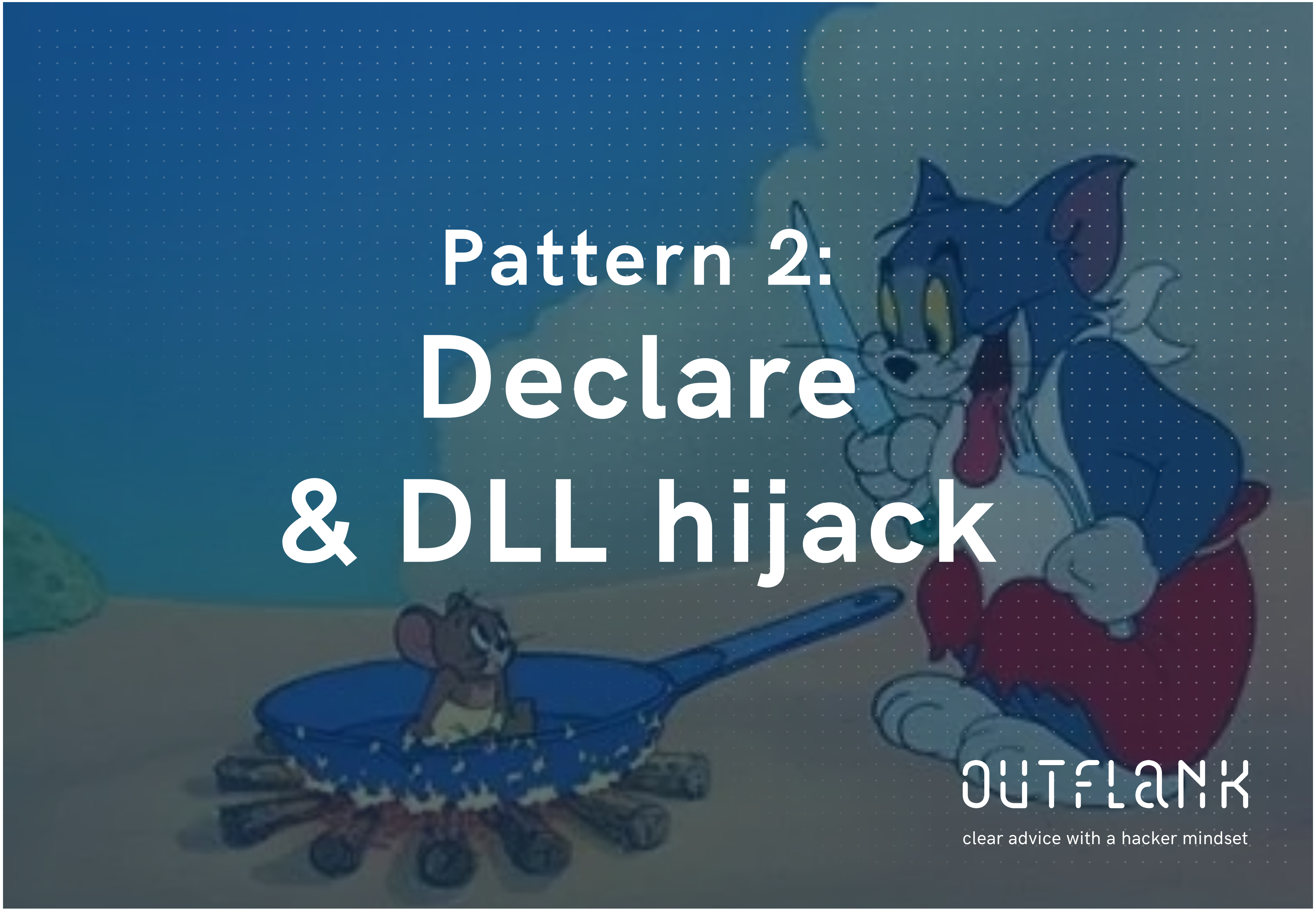
Internal recon

SOME INTERESTING FILES?



```
+----------+--------------------+---------------------------------------------+
|Type      |Keyword             |Description                                  |
+----------+--------------------+---------------------------------------------+
|AutoExec  |auto_open           |Runs when the Excel Workbook is opened       |
|AutoExec  |Workbook_Open       |Runs when the Excel Workbook is opened       |
|AutoExec  |auto_close          |Runs when the Excel Workbook is closed       |
|AutoExec  |Workbook_BeforeClose|Runs when the Excel Workbook is closed       |
|AutoExec  |cmdCancel_Click     |Runs when the file is opened and ActiveX     |
|          |                    |objects trigger events                       |
|AutoExec  |AutoColor_RangeName_|Runs when the file is opened and ActiveX     |
|          |Change              |objects trigger events                       |
|Suspicious|Environ             |May read system environment variables        |
|Suspicious|environment         |May read system environment variables        |
|Suspicious|ExpandEnvironmentStr|May read system environment variables        |
|          |ings                |                                             |
|Suspicious|Open                |May open a file                              |
|Suspicious|write               |May write to a file (if combined with Open)  |
|Suspicious|put                 |May write to a file (if combined with Open)  |
|Suspicious|output              |May write to a file (if combined with Open)  |
|Suspicious|Binary              |May read or write a binary file (if combined |
|          |                    |with Open)                                   |
|Suspicious|CopyFile            |May copy a file                              |
|Suspicious|CopyFolder          |May copy a file                              |
|Suspicious|Kill                |May delete a file                            |
|Suspicious|CreateTextFile      |May create a text file                       |
|Suspicious|Shell               |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|vbNormal            |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|vbNormalFocus       |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|WScript.Shell       |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|run                 |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|ShellExecute        |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|ShellExecuteA       |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|shell32             |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|create              |May execute file or a system command through |
|          |                    |WMI                                          |
|Suspicious|command             |May run PowerShell commands                  |
|Suspicious|Call                |May call a DLL using Excel 4 Macros (XLM/XLF)|
|Suspicious|Application.Visible |May hide the application                     |
|Suspicious|ShowWindow          |May hide the application                     |
|Suspicious|MkDir               |May create a directory                       |
|Suspicious|CreateObject        |May create an OLE object                     |
|Suspicious|GetObject           |May get an OLE object with a running instance|
|Suspicious|ExecuteExcel4Macro  |May run an Excel 4 Macro (aka XLM/XLF) from  |
|          |                    |VBA                                          |
|Suspicious|Windows             |May enumerate application windows (if        |
|          |                    |combined with Shell.Application object)      |
|Suspicious|FindWindow          |May enumerate application windows (if        |
|          |                    |combined with Shell.Application object)      |
|Suspicious|Lib                 |May run code from a DLL                      |
|Suspicious|RtlMoveMemory       |May inject code into another process         |
|Suspicious|SetTimer            |May run a shellcode in memory                |
```

~ 60K lines of VBA…
Hits all OLEVBA warnings
is legit…



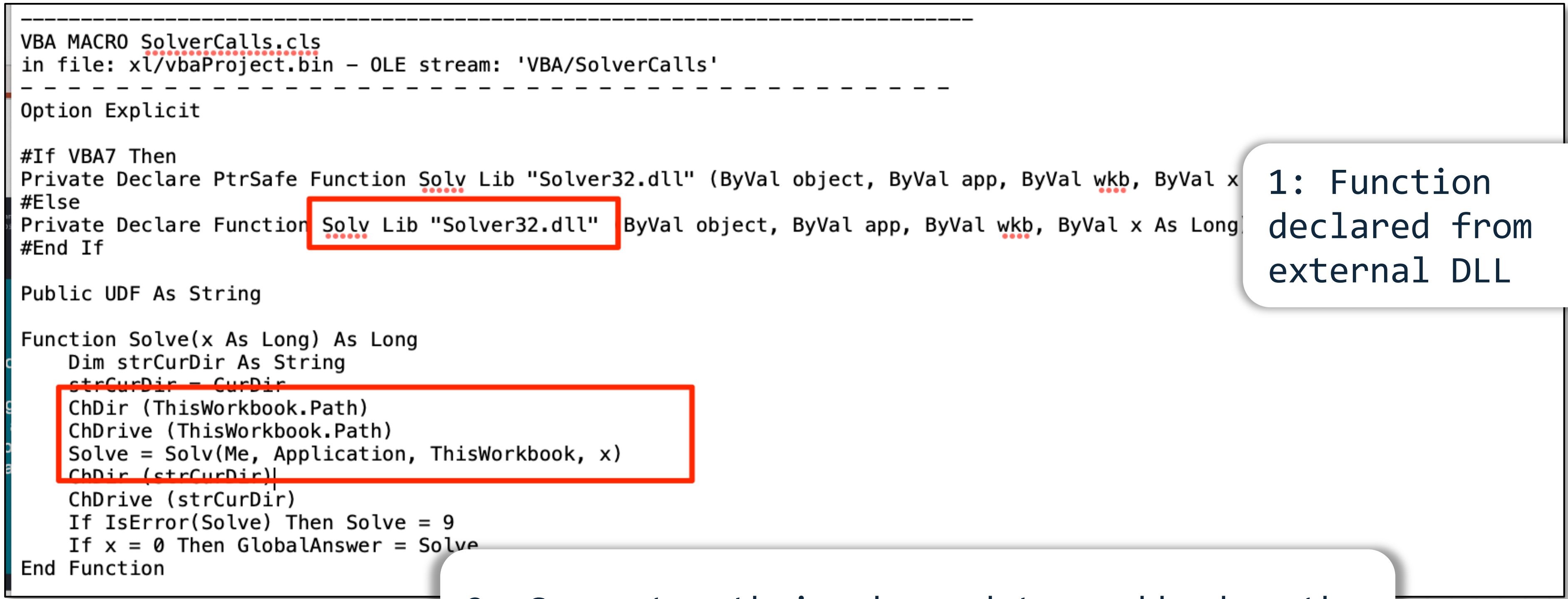
Pattern 2:

Declare & DLL hijack

OUTFLANK

clear advice with a hacker mindset

ABUSING DECLARES - PATCHED IN CVE-2021-28449

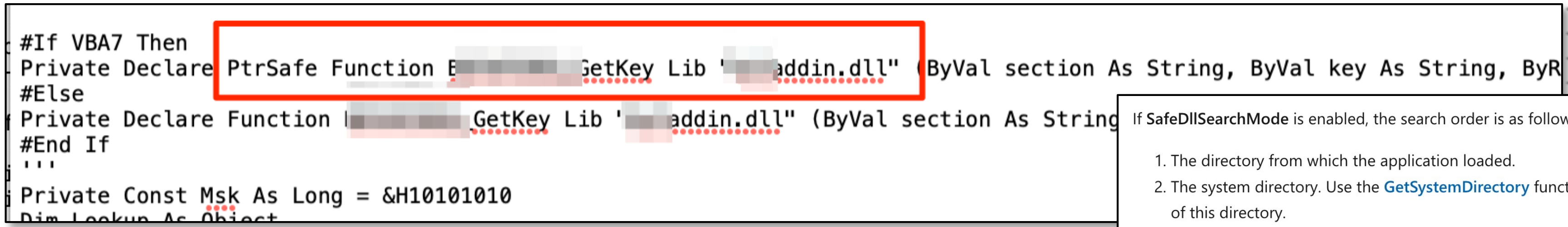


```
-------------------------------------------------------------------------------
VBA MACRO SolverCalls.cls
in file: xl/vbaProject.bin - OLE stream: 'VBA/SolverCalls'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Option Explicit

#If VBA7 Then
Private Declare PtrSafe Function Solv Lib "Solver32.dll" (ByVal object, ByVal app, ByVal wkb, ByVal x
#Else
Private Declare Function Solv Lib "Solver32.dll" (ByVal object, ByVal app, ByVal wkb, ByVal x As Long)
#End If

Public UDF As String

Function Solve(x As Long) As Long
    Dim strCurDir As String
    strCurDir = CurDir
    ChDir (ThisWorkbook.Path)
    ChDrive (ThisWorkbook.Path)
    Solve = Solv(Me, Application, ThisWorkbook, x)
    ChDir (strCurDir)
    ChDrive (strCurDir)
    If IsError(Solve) Then Solve = 9
    If x = 0 Then GlobalAnswer = Solve
End Function
```

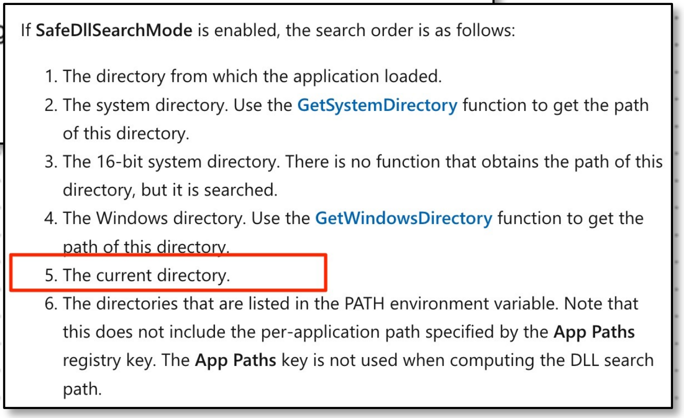
1: Function declared from external DLL

2: Current path is changed to workbook path
3: Declared function executed (=DLL load)

ABUSING DECLARES & SEARCH ORDER



```
#If VBA7 Then
Private Declare PtrSafe Function [redacted]GetKey Lib "[redacted]addin.dll" (ByVal section As String, ByVal key As String, ByR
#Else
Private Declare Function [redacted]GetKey Lib "[redacted]addin.dll" (ByVal section As String
#End If
'''
Private Const Msk As Long = &H10101010
Dim Lookup As Object
```

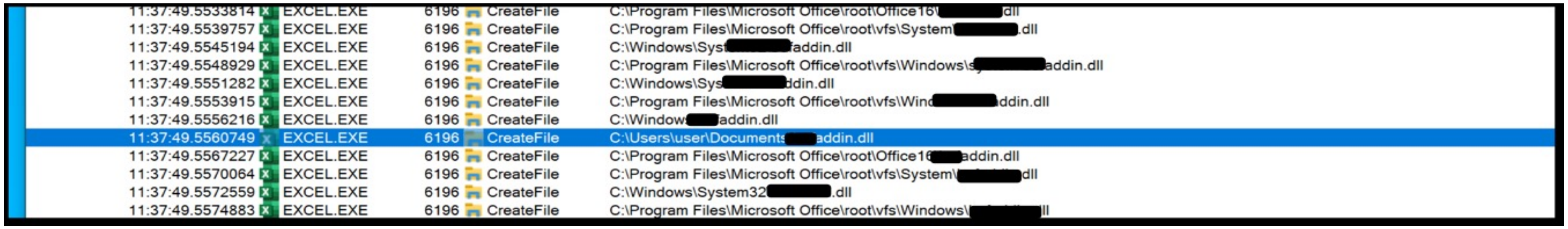


If **SafeDllSearchMode** is enabled, the search order is as follows:

1. The directory from which the application loaded.
2. The system directory. Use the GetSystemDirectory function to get the path of this directory.
3. The 16-bit system directory. There is no function that obtains the path of this directory, but it is searched.
4. The Windows directory. Use the GetWindowsDirectory function to get the path of this directory.
5. The current directory.
6. The directories that are listed in the PATH environment variable. Note that this does not include the per-application path specified by the **App Paths** registry key. The **App Paths** key is not used when computing the DLL search path.

Without the VBA directory changes

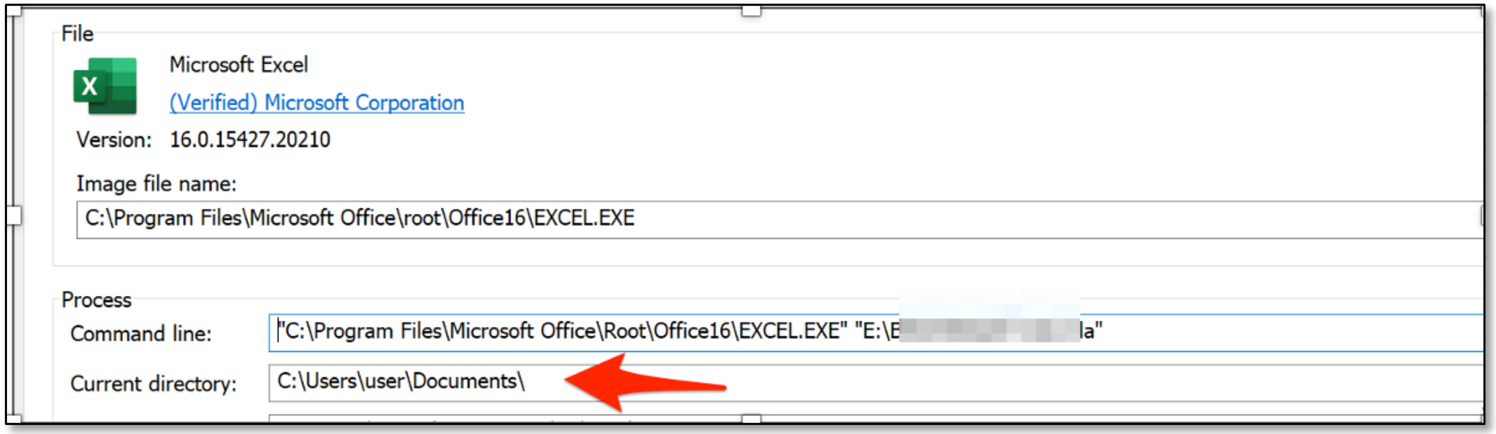
tried to abuse similar declares, but initially failed..



```
11:37:49.5533814  EXCEL.EXE  6196  CreateFile  C:\Program Files\Microsoft Office\root\Office16\[redacted]dll
11:37:49.5539757  EXCEL.EXE  6196  CreateFile  C:\Program Files\Microsoft Office\root\vfs\System[redacted].dll
11:37:49.5545194  EXCEL.EXE  6196  CreateFile  C:\Windows\Sys[redacted]addin.dll
11:37:49.5548929  EXCEL.EXE  6196  CreateFile  C:\Program Files\Microsoft Office\root\vfs\Windows\[redacted]addin.dll
11:37:49.5551282  EXCEL.EXE  6196  CreateFile  C:\Windows\Sys[redacted]ddin.dll
11:37:49.5553915  EXCEL.EXE  6196  CreateFile  C:\Program Files\Microsoft Office\root\vfs\Wind[redacted]ddin.dll
11:37:49.5556216  EXCEL.EXE  6196  CreateFile  C:\Window[redacted]addin.dll
11:37:49.5560749  EXCEL.EXE  6196  CreateFile  C:\Users\user\Documents[redacted]addin.dll
11:37:49.5567227  EXCEL.EXE  6196  CreateFile  C:\Program Files\Microsoft Office\root\Office16[redacted]addin.dll
11:37:49.5570064  EXCEL.EXE  6196  CreateFile  C:\Program Files\Microsoft Office\root\vfs\System\[redacted]dll
11:37:49.5572559  EXCEL.EXE  6196  CreateFile  C:\Windows\System32[redacted].dll
11:37:49.5574883  EXCEL.EXE  6196  CreateFile  C:\Program Files\Microsoft Office\root\vfs\Windows\[redacted]ll
```

Document is on desktop, searches in "documents" folder. Why?

SEARCH ORDER IN OFFICE



File

Microsoft Excel
(Verified) Microsoft Corporation

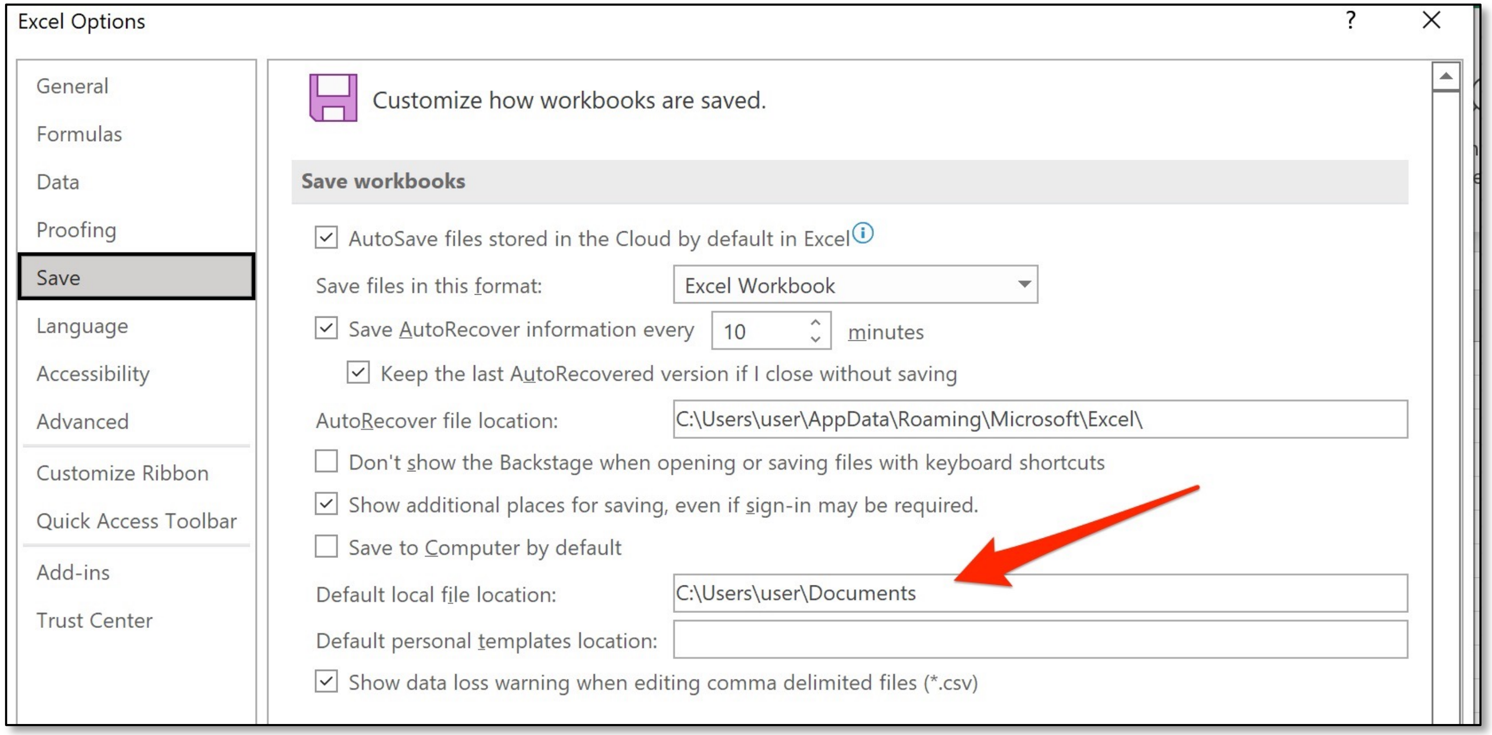
Version: 16.0.15427.20210

Image file name:
C:\Program Files\Microsoft Office\root\Office16\EXCEL.EXE

Process

Command line: "C:\Program Files\Microsoft Office\Root\Office16\EXCEL.EXE" "E:\B[redacted]la"

Current directory: C:\Users\user\Documents\



Customize how workbooks are saved.

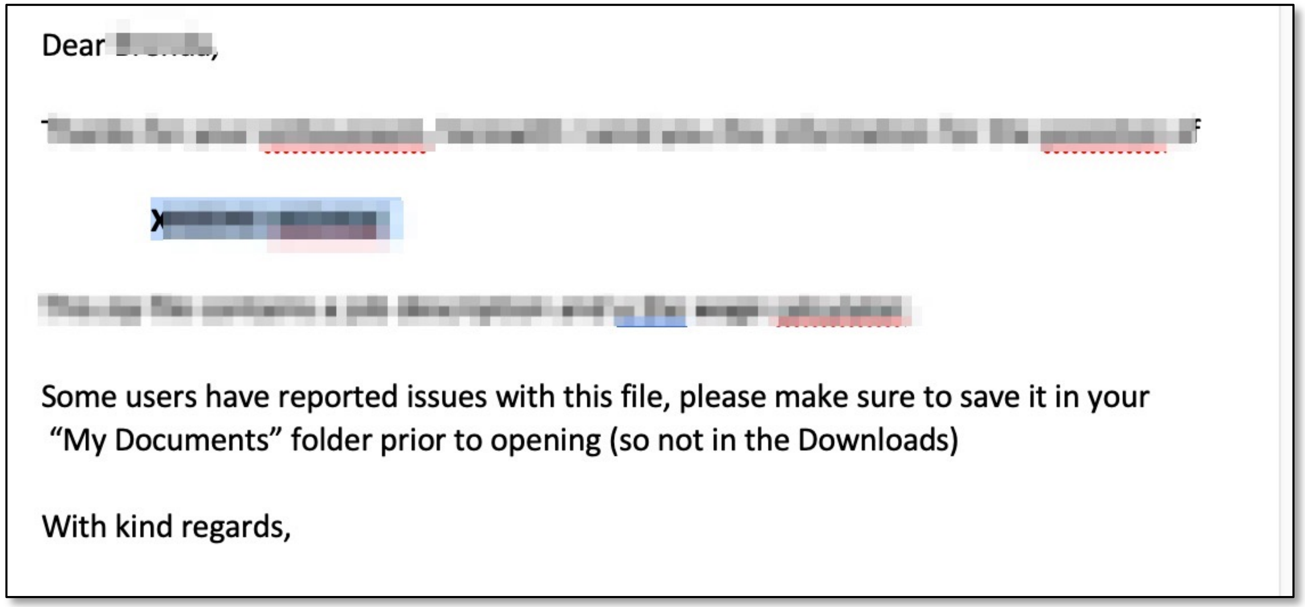
Save workbooks

- ✓ AutoSave files stored in the Cloud by default in Excel
- Save files in this format: Excel Workbook
- ✓ Save AutoRecover information every 10 minutes
  - ✓ Keep the last AutoRecovered version if I close without saving
- AutoRecover file location: C:\Users\user\AppData\Roaming\Microsoft\Excel\
- ☐ Don't show the Backstage when opening or saving files with keyboard shortcuts
- ✓ Show additional places for saving, even if sign-in may be required.
- ☐ Save to Computer by default
- Default local file location: C:\Users\user\Documents
- Default personal templates location:
- ✓ Show data loss warning when editing comma delimited files (*.csv)

MANIPULATING THE DOCUMENTS DIRECTORY

Social engineering

Make sure that the user places the provided files in the ‘documents’ folder



Dear [redacted],

[redacted]

[redacted]

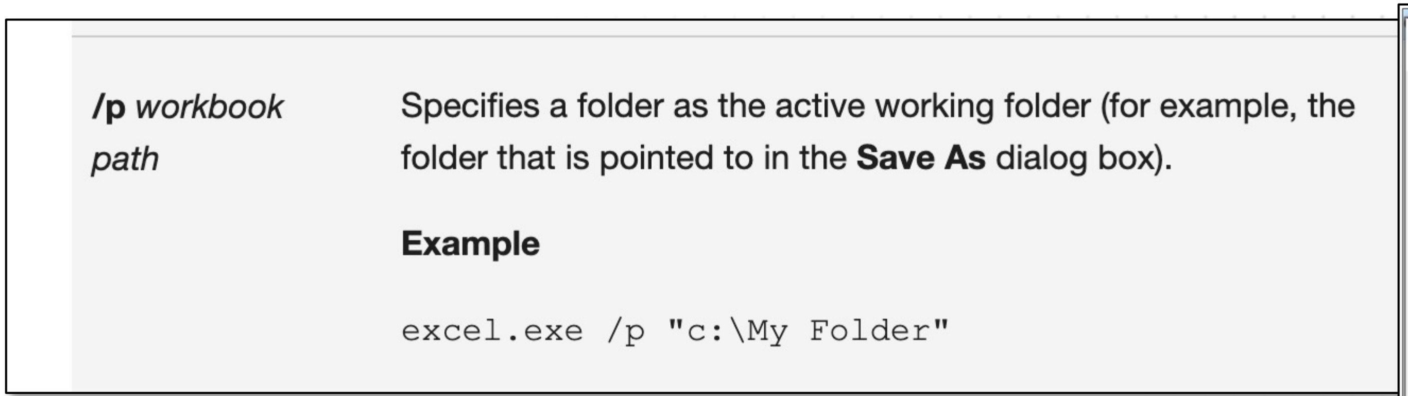
[redacted]

Some users have reported issues with this file, please make sure to save it in your "My Documents" folder prior to opening (so not in the Downloads)

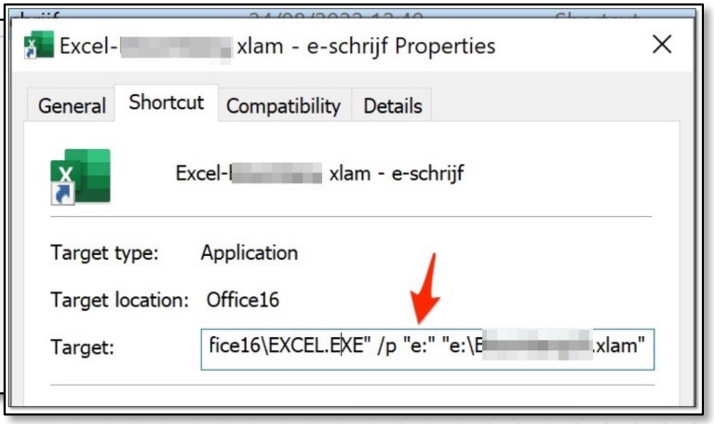
With kind regards,

Technical engineering

Startup flags can manipulate the searchpath (e.g. send .lnk)



/p workbook path	Specifies a folder as the active working folder (for example, the folder that is pointed to in the Save As dialog box). **Example** `excel.exe /p "c:\My Folder"`



Excel-[redacted] xlam - e-schrijf Properties

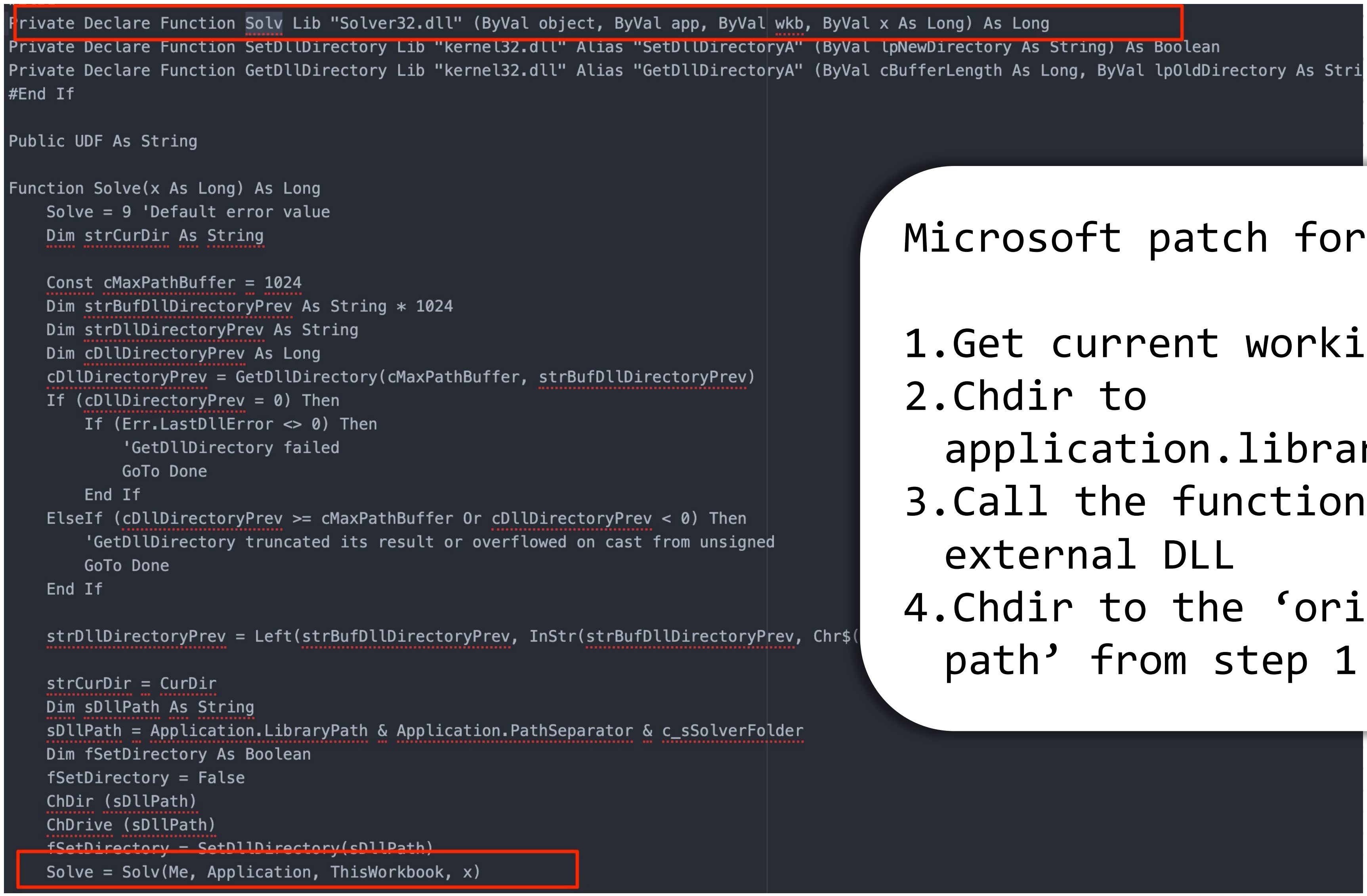
Excel-[redacted] xlam - e-schrijf

Target type: Application

Target location: Office16

Target: fice16\EXCEL.EXE" /p "e:" "e:\B[redacted].xlam"

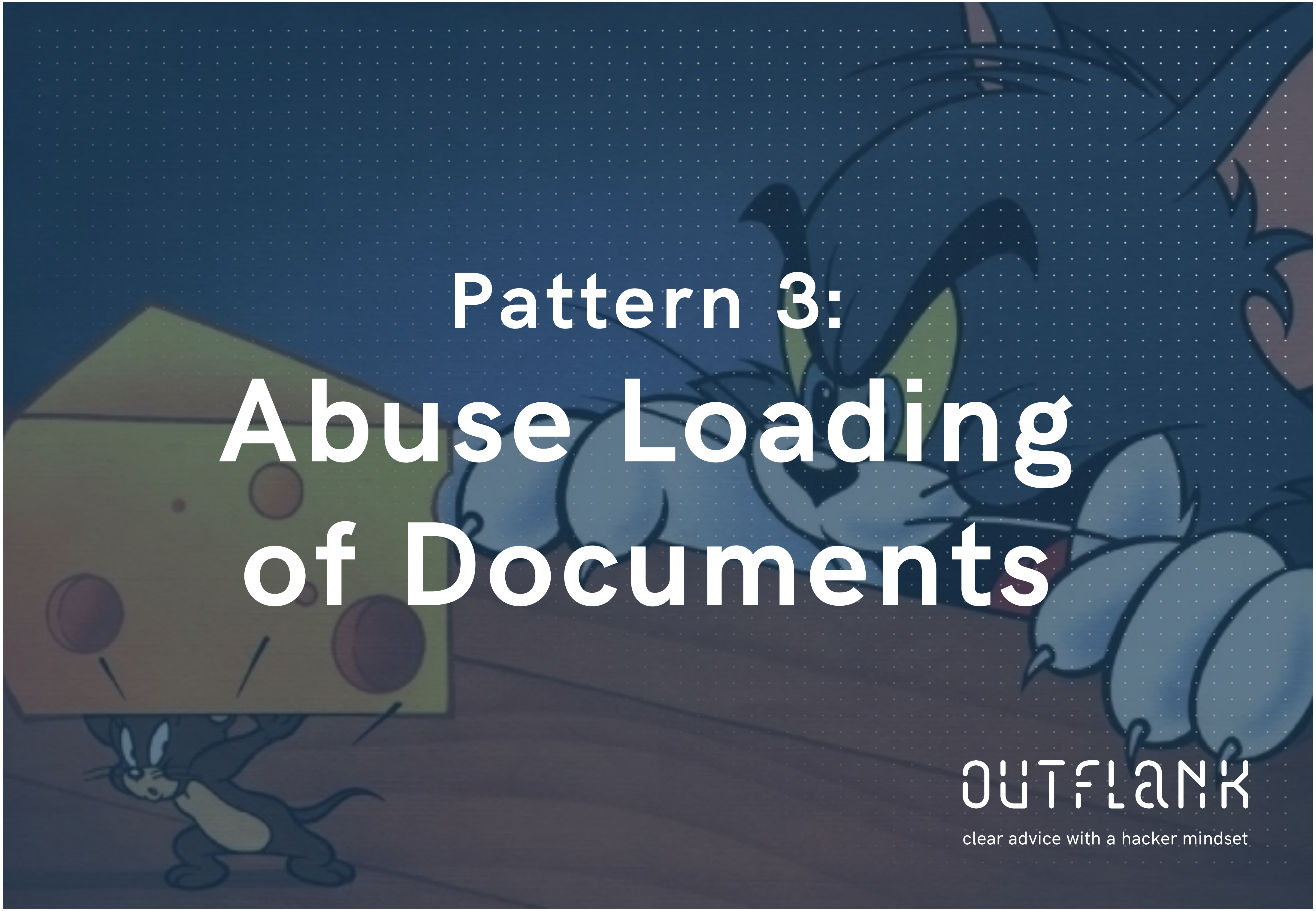
MITIGATIONS ON DECLARES



```
Private Declare Function Solv Lib "Solver32.dll" (ByVal object, ByVal app, ByVal wkb, ByVal x As Long) As Long
Private Declare Function SetDllDirectory Lib "kernel32.dll" Alias "SetDllDirectoryA" (ByVal lpNewDirectory As String) As Boolean
Private Declare Function GetDllDirectory Lib "kernel32.dll" Alias "GetDllDirectoryA" (ByVal cBufferLength As Long, ByVal lpOldDirectory As Stri
#End If

Public UDF As String

Function Solve(x As Long) As Long
    Solve = 9 'Default error value
    Dim strCurDir As String

    Const cMaxPathBuffer = 1024
    Dim strBufDllDirectoryPrev As String * 1024
    Dim strDllDirectoryPrev As String
    Dim cDllDirectoryPrev As Long
    cDllDirectoryPrev = GetDllDirectory(cMaxPathBuffer, strBufDllDirectoryPrev)
    If (cDllDirectoryPrev = 0) Then
        If (Err.LastDllError <> 0) Then
            'GetDllDirectory failed
            GoTo Done
        End If
    ElseIf (cDllDirectoryPrev >= cMaxPathBuffer Or cDllDirectoryPrev < 0) Then
        'GetDllDirectory truncated its result or overflowed on cast from unsigned
        GoTo Done
    End If

    strDllDirectoryPrev = Left(strBufDllDirectoryPrev, InStr(strBufDllDirectoryPrev, Chr$(

    strCurDir = CurDir
    Dim sDllPath As String
    sDllPath = Application.LibraryPath & Application.PathSeparator & c_sSolverFolder
    Dim fSetDirectory As Boolean
    fSetDirectory = False
    ChDir (sDllPath)
    ChDrive (sDllPath)
    fSetDirectory = SetDllDirectory(sDllPath)
    Solve = Solv(Me, Application, ThisWorkbook, x)
```

Microsoft patch for Solver:

1. Get current working dir
2. Chdir to application.librarypath
3. Call the function from external DLL
4. Chdir to the ‘original path’ from step 1



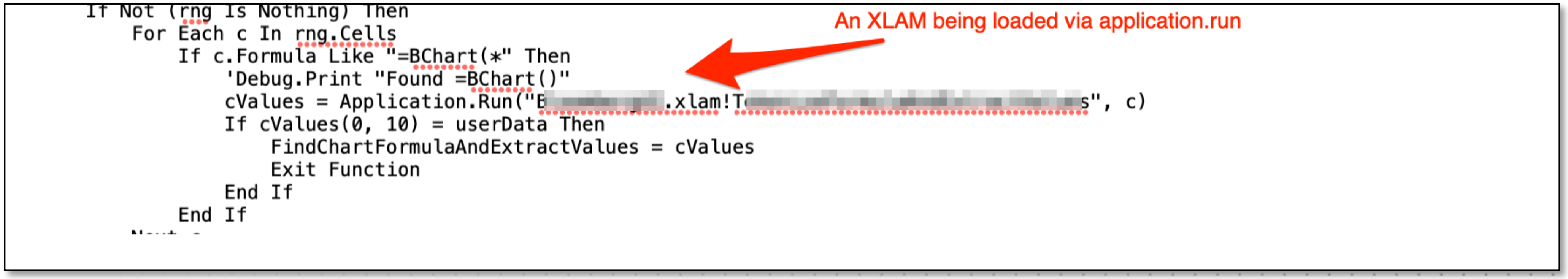
Pattern 3:

Abuse Loading of Documents

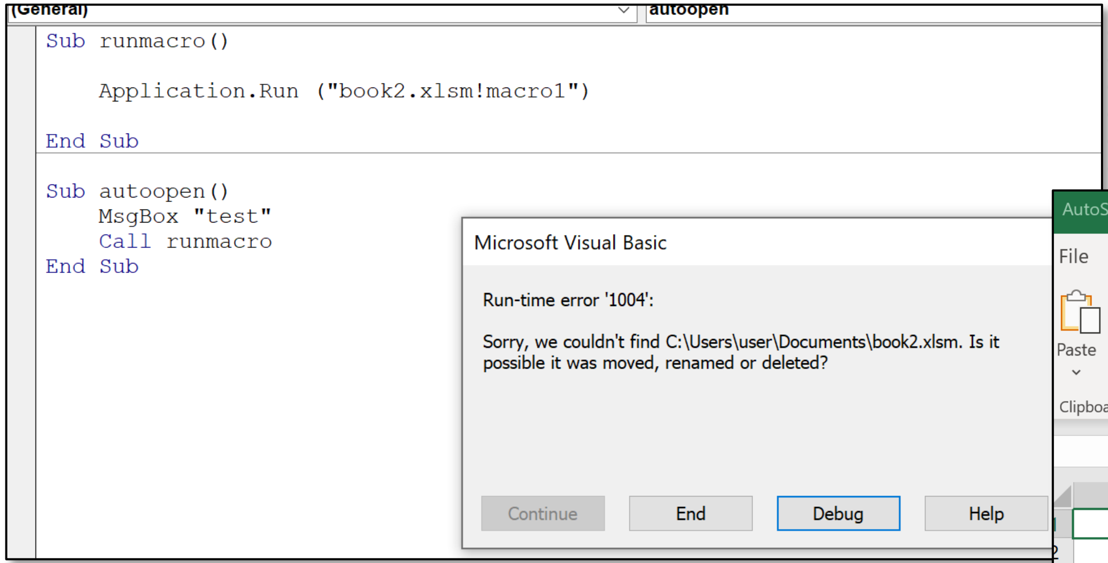
OUTFLANK

clear advice with a hacker mindset

ABUSE LOADING OF DOCUMENTS – APPLICATION.RUN



```
    If Not (rng Is Nothing) Then
        For Each c In rng.Cells
            If c.Formula Like "=BChart(*" Then
                'Debug.Print "Found =BChart()"
                cValues = Application.Run("B[redacted].xlam!T[redacted]s", c)
                If cValues(0, 10) = userData Then
                    FindChartFormulaAndExtractValues = cValues
                    Exit Function
                End If
            End If
        Next c
```

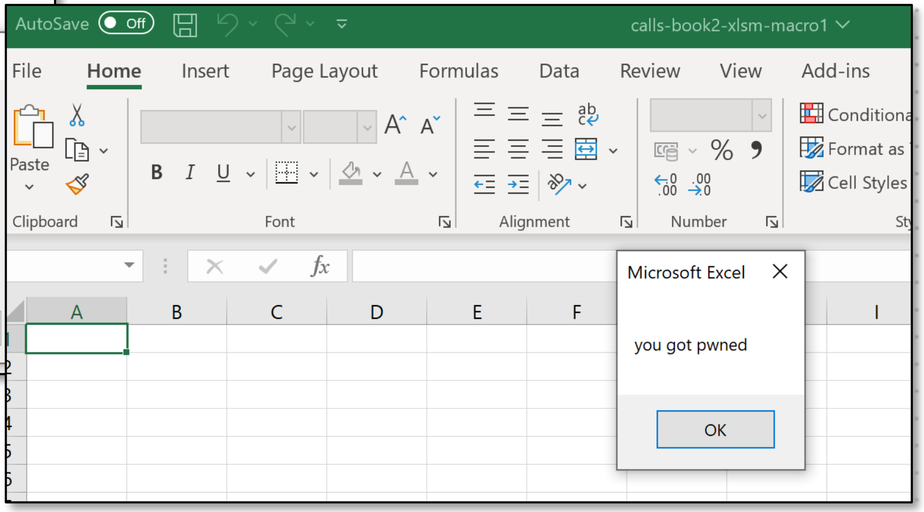


```
Sub runmacro()

    Application.Run ("book2.xlsm!macro1")

End Sub

Sub autoopen()
    MsgBox "test"
    Call runmacro
End Sub
```

Microsoft Visual Basic

Run-time error '1004':

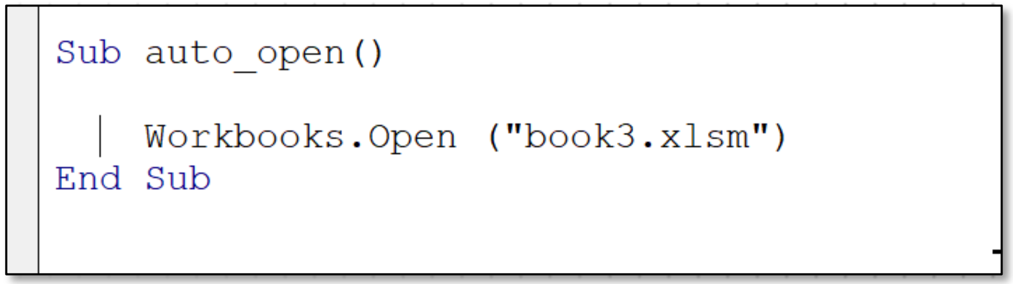
Sorry, we couldn't find C:\Users\user\Documents\book2.xlsm. Is it possible it was moved, renamed or deleted?



Microsoft Excel

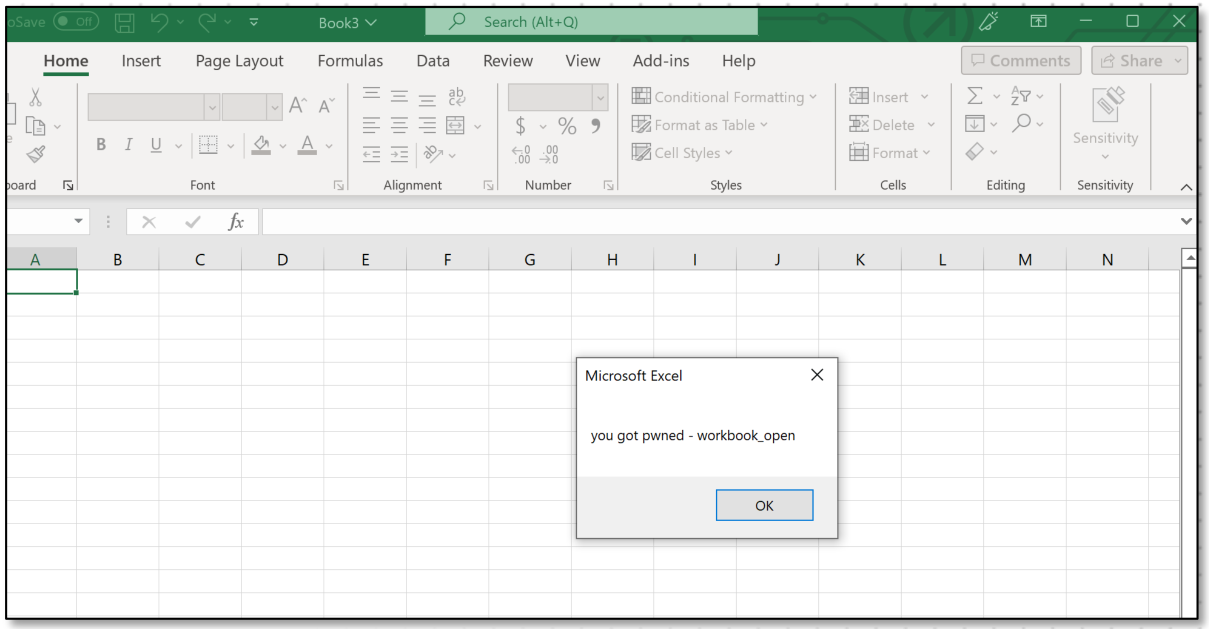
you got pwned

ABUSE LOADING OF DOCUMENTS – OPEN DOC



```
Sub auto_open()

    Workbooks.Open ("book3.xlsm")
End Sub
```

Vulnerable or not?



Microsoft Excel

you got pwned - workbook_open

OK

CROSS OFFICE APP FILE OPENING

StackOverflow: Excel instantiates Word and opens file

```
Sub RamsOpen2()

Dim Doc
Dim DocPath
Dim DocObj
Dim VarResult

DocPath = "C:\Users\mariuszk\Desktop\cf10\RAMS.docx"
Set DocObj = CreateObject("word.Application")
Doc = DocObj.Documents.Open(DocPath)
DocObj.Visible = True

With Doc.ActiveDocument
    Set myRange = .Content
    With myRange.Find
        .Execute FindText:="FindText", ReplaceWith:="Repla
    End With
End With

VarResult = Doc.GetSaveAsFilename( _
FileFilter:="DP Document (*.doc), *.doc, DP Document (*.dc
initialvalue:="InitialDocument")

End Sub
```

Question

What methods to ‘open a file’ via VBA and what impact on ‘macros running or not’

CROSS OFFICE APP FILE OPENING

Vulnerable

```
appWord.Run (ThisWorkbook.Path & "\doc1.docm!macro")

appWord.Documents.Open (ThisWorkbook.Path & "\doc1.docm")
```

Vulnerable when called with doc ,docm,..., not vulnerable when called with docx (note: a renamed docm->docx fails)

Not vulnerable?

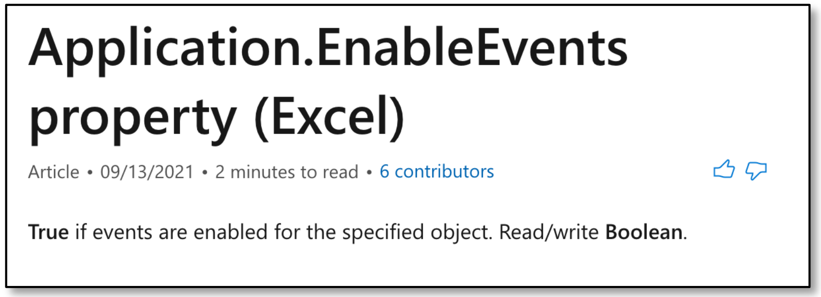
```
docu = appWord.Documents.Add(ThisWorkbook.Path & "\doc1.doc")
```

OTHER MITIGATION DIRECTIONS

Can we disable macros while running macros?

One option seems to ”Disable events”

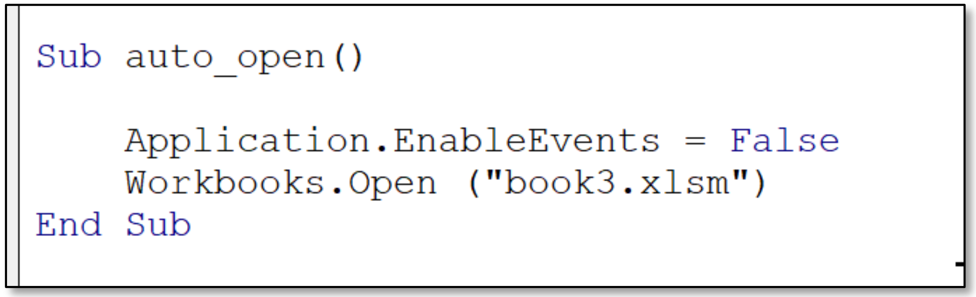
avoids autoopen, document_open events from firing



Application.EnableEvents property (Excel)

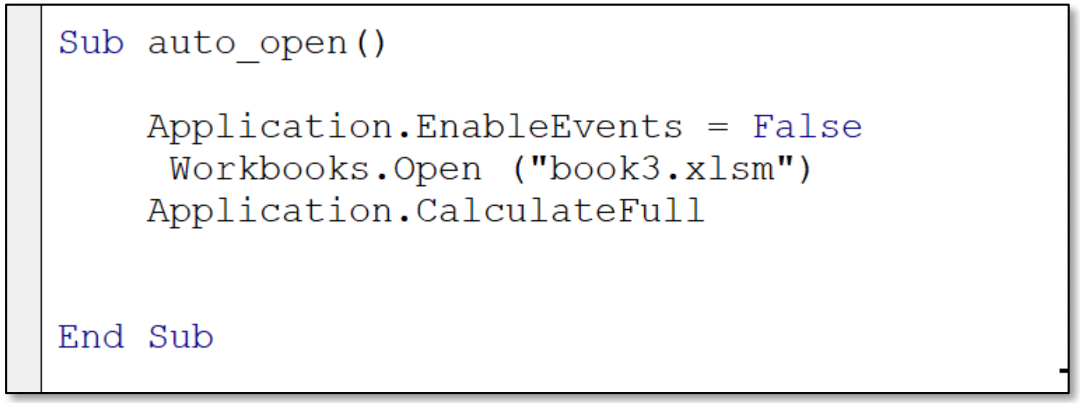
Article • 09/13/2021 • 2 minutes to read • 6 contributors

True if events are enabled for the specified object. Read/write **Boolean**.



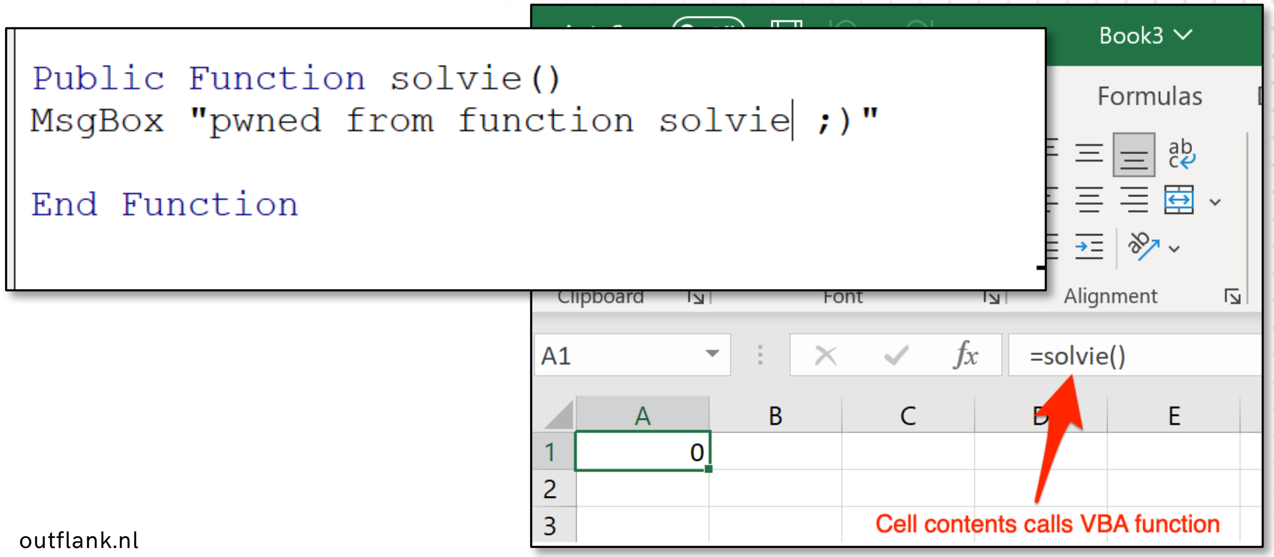
```
Sub auto_open()

    Application.EnableEvents = False
    Workbooks.Open ("book3.xlsm")
End Sub
```

THIS IS SUBTLE

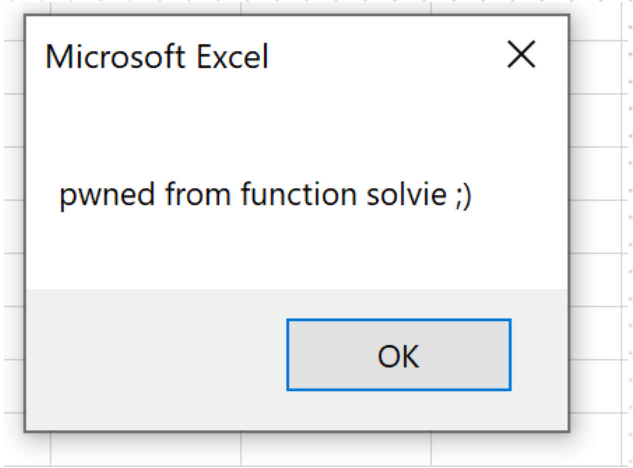


```
Sub auto_open()

    Application.EnableEvents = False
     Workbooks.Open ("book3.xlsm")
    Application.CalculateFull


End Sub
```

Vulnerable or not?



```
Public Function solvie()
MsgBox "pwned from function solvie ;)"

End Function
```



pwned from function solvie ;)

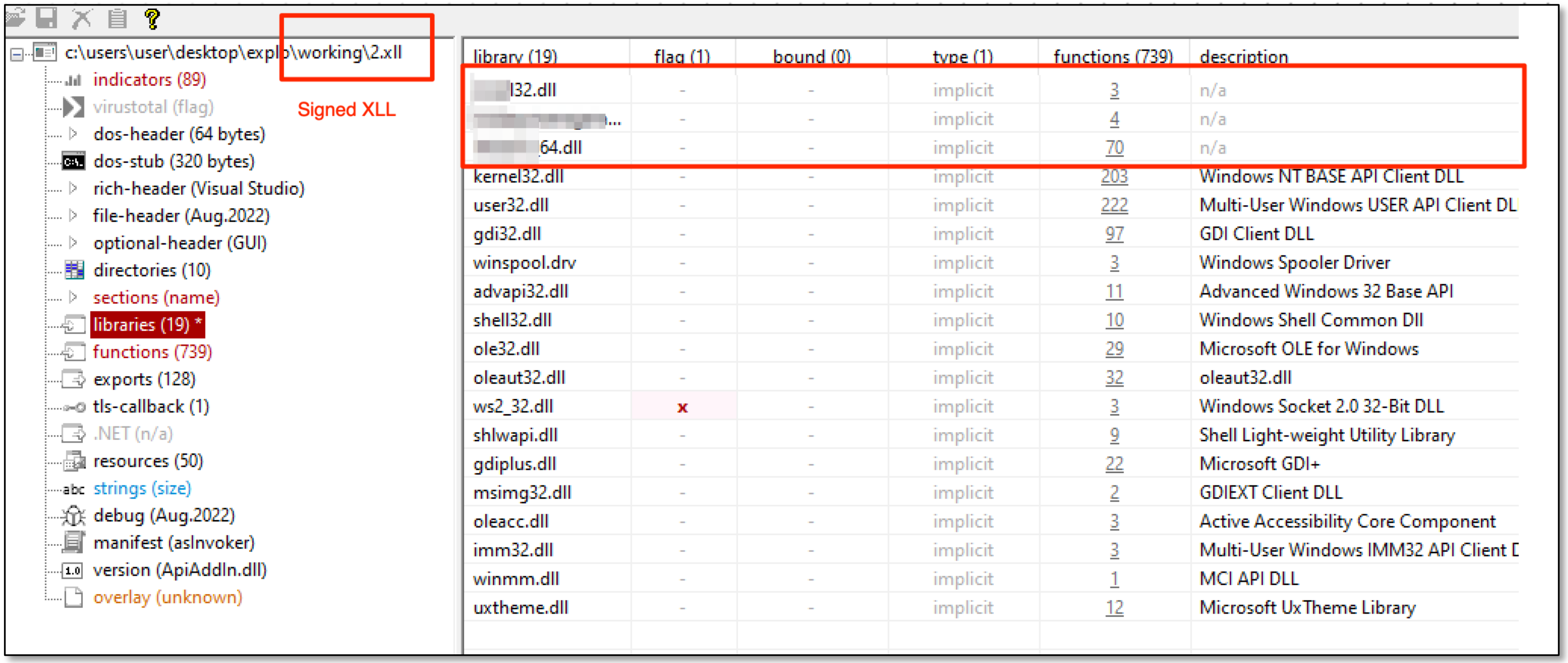
Pattern 4:

XLL: Ghost DLL hijack

OUTFLANK

clear advice with a hacker mindset

SIGNED XLLS

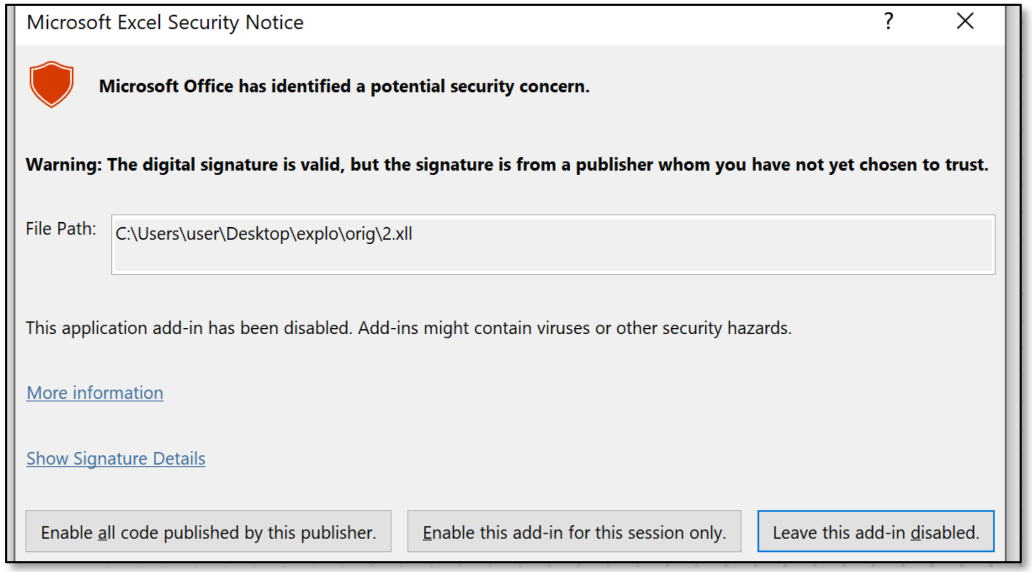


c:\users\user\desktop\expl\working\2.xll

- indicators (89)
- virustotal (flag)
- dos-header (64 bytes)
- dos-stub (320 bytes)
- rich-header (Visual Studio)
- file-header (Aug.2022)
- optional-header (GUI)
- directories (10)
- sections (name)
- libraries (19) *
- functions (739)
- exports (128)
- tls-callback (1)
- .NET (n/a)
- resources (50)
- strings (size)
- debug (Aug.2022)
- manifest (asInvoker)
- version (ApiAddIn.dll)
- overlay (unknown)

library (19)	flag (1)	bound (0)	type (1)	functions (739)	description
[redacted]l32.dll	-	-	implicit	3	n/a
[redacted]...	-	-	implicit	4	n/a
[redacted]64.dll	-	-	implicit	70	n/a
kernel32.dll	-	-	implicit	203	Windows NT BASE API Client DLL
user32.dll	-	-	implicit	222	Multi-User Windows USER API Client DL
gdi32.dll	-	-	implicit	97	GDI Client DLL
winspool.drv	-	-	implicit	3	Windows Spooler Driver
advapi32.dll	-	-	implicit	11	Advanced Windows 32 Base API
shell32.dll	-	-	implicit	10	Windows Shell Common Dll
ole32.dll	-	-	implicit	29	Microsoft OLE for Windows
oleaut32.dll	-	-	implicit	32	oleaut32.dll
ws2_32.dll	x	-	implicit	3	Windows Socket 2.0 32-Bit DLL
shlwapi.dll	-	-	implicit	9	Shell Light-weight Utility Library
gdiplus.dll	-	-	implicit	22	Microsoft GDI+
msimg32.dll	-	-	implicit	2	GDIEXT Client DLL
oleacc.dll	-	-	implicit	3	Active Accessibility Core Component
imm32.dll	-	-	implicit	3	Multi-User Windows IMM32 API Client D
winmm.dll	-	-	implicit	1	MCI API DLL
uxtheme.dll	-	-	implicit	12	Microsoft UxTheme Library

XLL & SEARCH ORDER



Microsoft Excel Security Notice

Microsoft Office has identified a potential security concern.

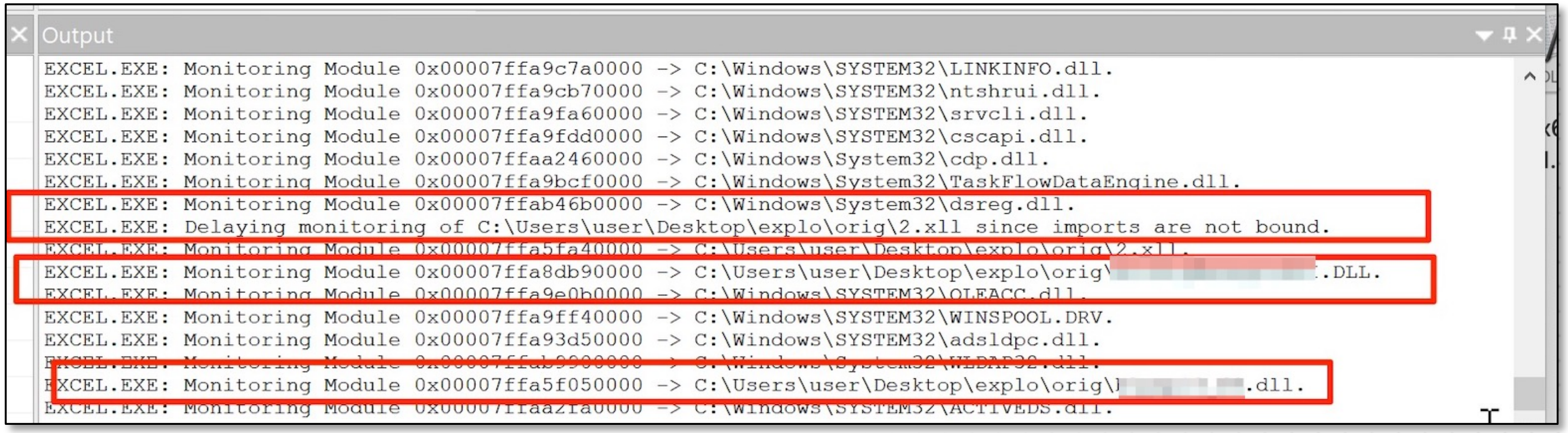
Warning: The digital signature is valid, but the signature is from a publisher whom you have not yet chosen to trust.

File Path: C:\Users\user\Desktop\explo\orig\2.xll

This application add-in has been disabled. Add-ins might contain viruses or other security hazards.

More information
Show Signature Details
Enable all code published by this publisher.
Enable this add-in for this session only.
Leave this add-in disabled.

Excel loads dlls from current dir, not in "documents"

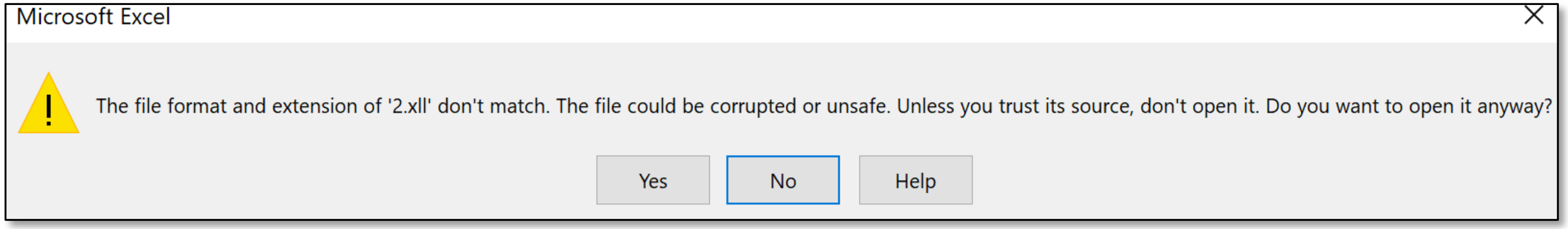


Output

```
EXCEL.EXE: Monitoring Module 0x00007ffa9c7a0000 -> C:\Windows\SYSTEM32\LINKINFO.dll.
EXCEL.EXE: Monitoring Module 0x00007ffa9cb70000 -> C:\Windows\SYSTEM32\ntshrui.dll.
EXCEL.EXE: Monitoring Module 0x00007ffa9fa60000 -> C:\Windows\SYSTEM32\srvcli.dll.
EXCEL.EXE: Monitoring Module 0x00007ffa9fdd0000 -> C:\Windows\SYSTEM32\cscapi.dll.
EXCEL.EXE: Monitoring Module 0x00007ffaa2460000 -> C:\Windows\System32\cdp.dll.
EXCEL.EXE: Monitoring Module 0x00007ffa9bcf0000 -> C:\Windows\System32\TaskFlowDataEngine.dll.
EXCEL.EXE: Monitoring Module 0x00007ffab46b0000 -> C:\Windows\System32\dsreg.dll.
EXCEL.EXE: Delaying monitoring of C:\Users\user\Desktop\explo\orig\2.xll since imports are not bound.
EXCEL.EXE: Monitoring Module 0x00007ffa5fa40000 -> C:\Users\user\Desktop\explo\orig\2.xll.
EXCEL.EXE: Monitoring Module 0x00007ffa8db90000 -> C:\Users\user\Desktop\explo\orig\[redacted].DLL.
EXCEL.EXE: Monitoring Module 0x00007ffa9e0b0000 -> C:\Windows\SYSTEM32\OLEACC.dll.
EXCEL.EXE: Monitoring Module 0x00007ffa9ff40000 -> C:\Windows\SYSTEM32\WINSPOOL.DRV.
EXCEL.EXE: Monitoring Module 0x00007ffa93d50000 -> C:\Windows\SYSTEM32\adsldpc.dll.
EXCEL.EXE: Monitoring Module 0x00007ffab9900000 -> C:\Windows\System32\WLDAP32.dll.
EXCEL.EXE: Monitoring Module 0x00007ffa5f050000 -> C:\Users\user\Desktop\explo\orig\[redacted].dll.
EXCEL.EXE: Monitoring Module 0x00007ffaa2fa0000 -> C:\Windows\SYSTEM32\ACTIVEDS.dll.
```

PRACTICAL EXPLOITATION – EXCEL BEHAVIOUR

If XLL has loading issues (e.g. DLL imports) then Excel generates an error and XLL content is shown in Excel


Microsoft Excel

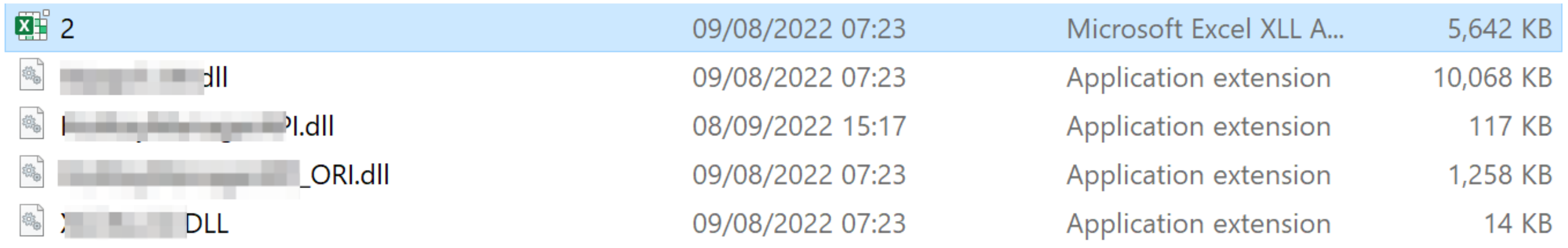
The file format and extension of '2.xll' don't match. The file could be corrupted or unsafe. Unless you trust its source, don't open it. Do you want to open it anyway?

Solution:

- Function proxying
- exportstoc - https://github.com/michaellandi/exportstoc

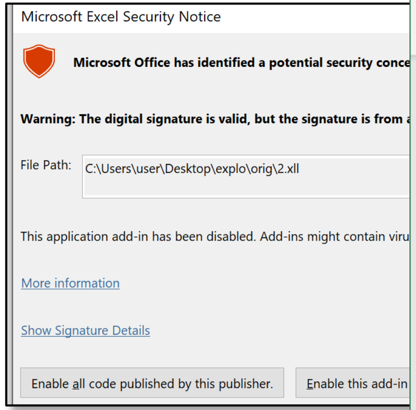
PRACTICAL EXPLOITATION

1: User gets an XLL and a bunch of DLL's (e.g. via zip, iso, etc)



2	09/08/2022 07:23	Microsoft Excel XLL A...	5,642 KB
[redacted]dll	09/08/2022 07:23	Application extension	10,068 KB
[redacted]l.dll	08/09/2022 15:17	Application extension	117 KB
[redacted]_ORI.dll	09/08/2022 07:23	Application extension	1,258 KB
[redacted]DLL	09/08/2022 07:23	Application extension	14 KB

2: Starts add-in and e



Microsoft Excel Security Notice

Microsoft Office has identified a potential security conce

Warning: The digital signature is valid, but the signature is from a

File Path: C:\Users\user\Desktop\explo\orig\2.xll

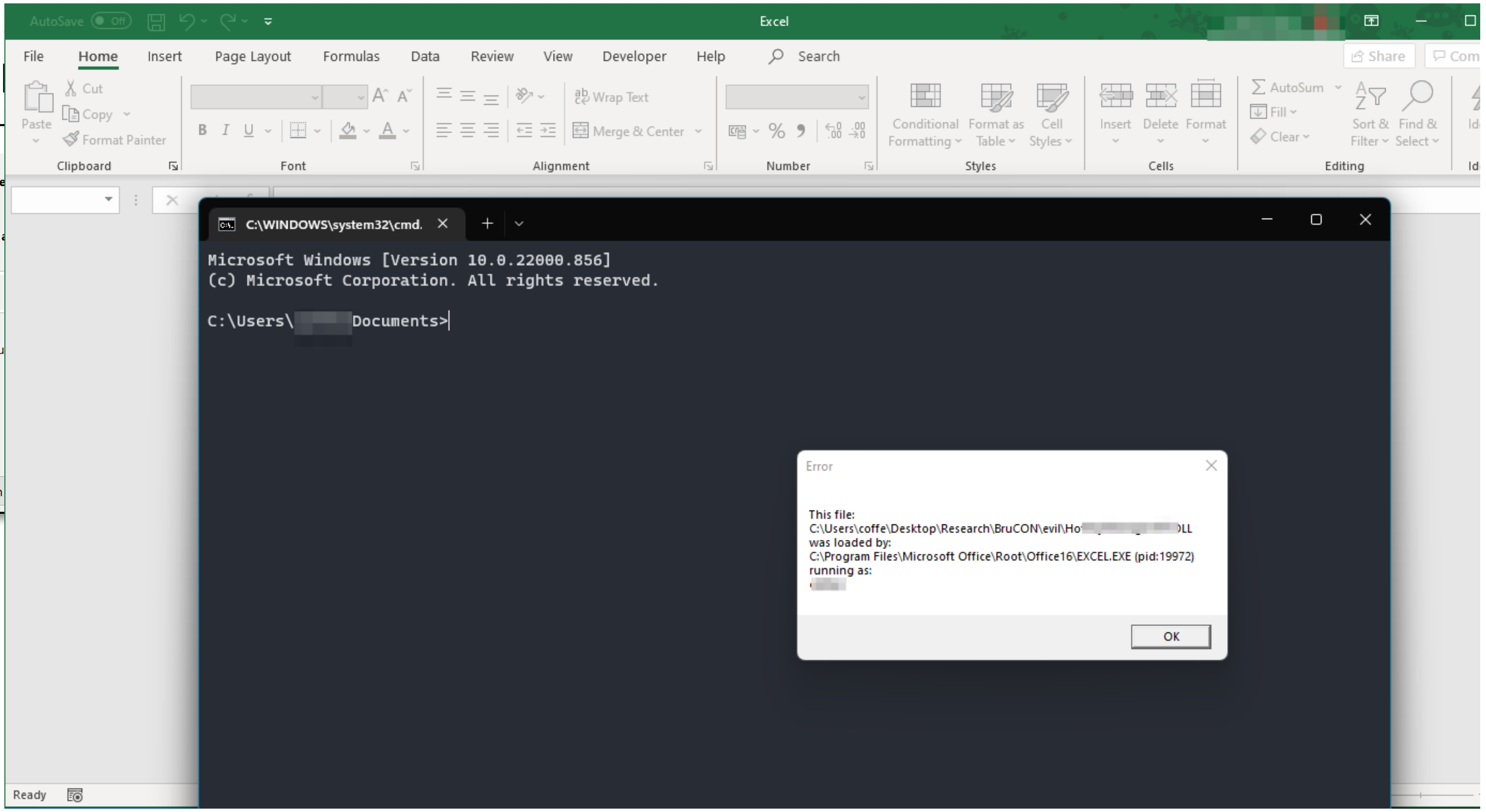
This application add-in has been disabled. Add-ins might contain viru

More information

Show Signature Details

Enable all code published by this publisher. | Enable this add-in

3: Got Shell..



C:\WINDOWS\system32\cmd.

```
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\[redacted]Documents>
```

Error

This file:
C:\Users\coffe\Desktop\Research\BruCON\evil\Ho[redacted]DLL
was loaded by:
C:\Program Files\Microsoft Office\Root\Office16\EXCEL.EXE (pid:19972)
running as:
[redacted]

OK

MITIGATION

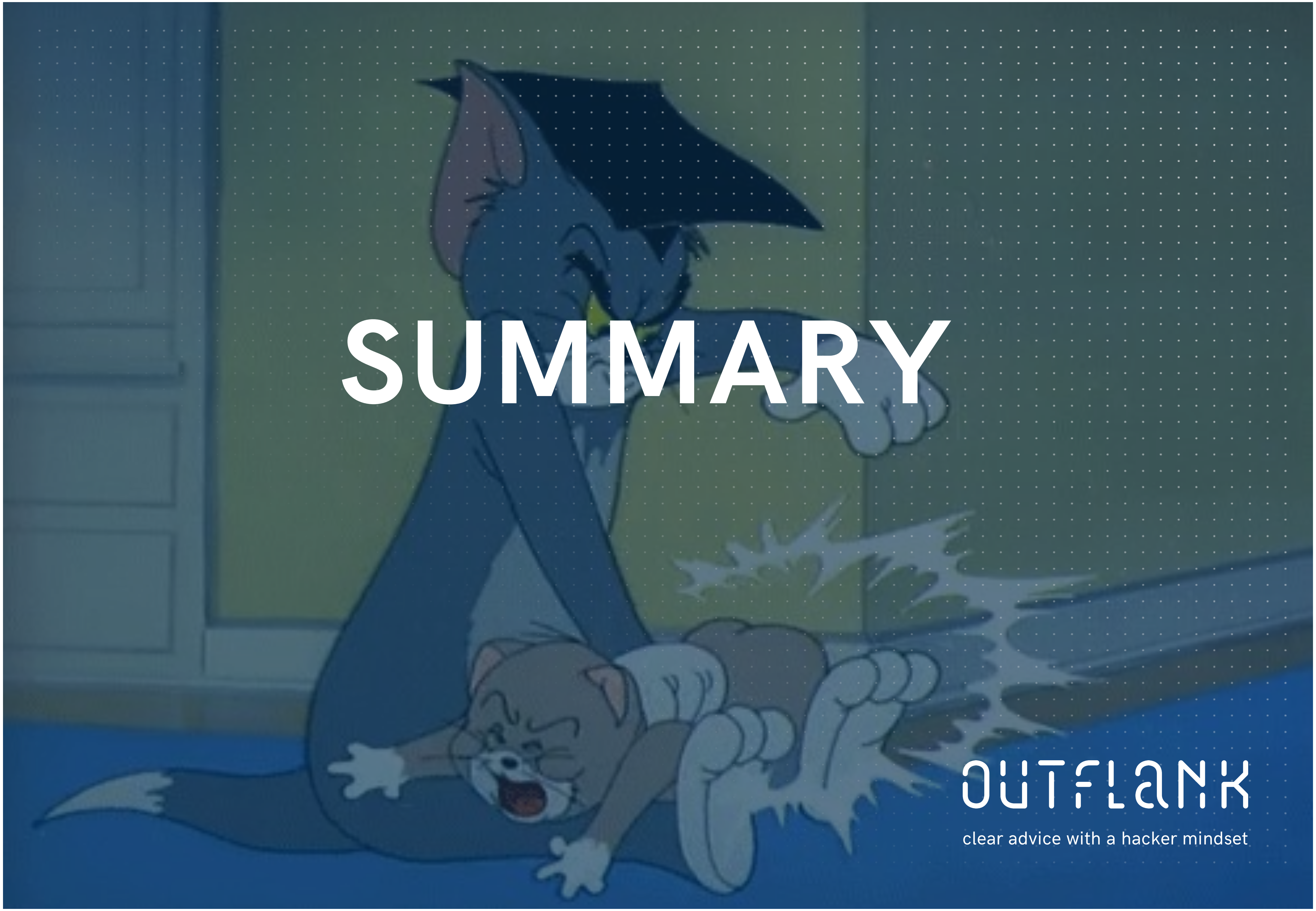
- Full paths for DLL's when possible
  (Not feasible on user selected paths/install dirs)

- Configure DLL loading via linker configuration e.g.

  ```
  LOAD_LIBRARY_SEARCH_APPLICATION_DIR
  LOAD_LIBRARY_REQUIRE_SIGNED_TARGET
  ```

- Configure search order via `SetDefaultDllDirectories` function before `loadlibrary` calls (or linker `/DELAYLOAD` (Delay Load Import))

https://docs.microsoft.com/en-us/windows/win32/dlls/dynamic-link-library-security



SUMMARY

OUTFLANK

clear advice with a hacker mindset

COMMON VULNERABLE CODE PATTERNS

	Microsoft signed	Public samples signed	Public code / stack overflow
1. Code flow depending on cell contents			
2. VBA Declare Ghost DLL hijack			
3. Document loading			
4. XLL Ghost DLL hijack			

TAKE AWAYS

Blue teams

- Office Security settings are complex...
- LOLDocs: An attacker can use signed files out of application/installer context
- Reconsider your trusted publishers
- Code review prior to signing VBA code and XLLs and consider the listed attacks
- To timestamp or not? Revoking strategy?

TAKE AWAYS

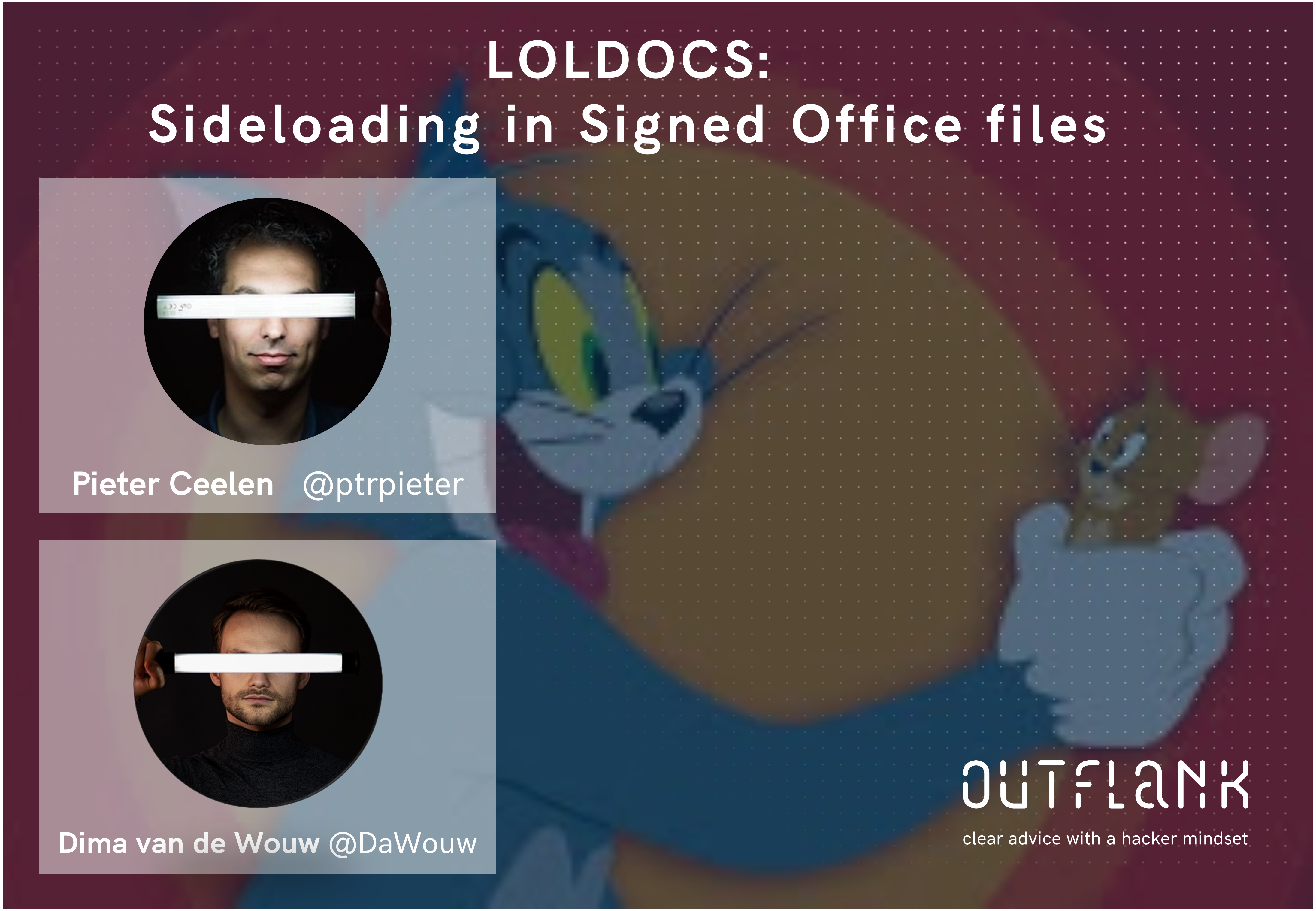
Red teams

- There are many vulnerable public signed samples out there
- Revoking these files is hard! Typically no AMSI (runtimescanscope)
- When inside, check trusted publishers, download signed files

Initial access, lateral, long term persistence!

MS Office: The product that keeps on giving!

LOLDOCS: Sideloading in Signed Office files



Pieter Ceelen @ptrpieter

Dima van de Wouw @DaWouw

OUTFLANK

clear advice with a hacker mindset