Malicious Attachments

Botnet Mitigation Course

Tarmo Randel

Autumn 2018

Malicious attachments

In a nutshell ...

- Documents (Office, PDF) can contain malicious payload
  - Items are stored as objects, streams
  - Adding scripting capabilities is like paving a road to hell with red carpet
  - Payload can be hidden in remarks, comments, form elements etc etc etc ...
- Exploits can be used to execute and/or decode payload
  - Exploits have usually short lifetime, AV products and updates narrow time window of success
- Windows scripting egine will happily run .js, .jse files

Attack surface

How to use document files for delivering the attack payload?

1. Exploitable weaknesesses
2. Features (DDE)
3. Social engineering (Please click “Enable script execution“)

Document formats

Useful documentation

- MS Office files used to be undisclosed proprietary binary format. Two good news:
  - Now packed XML, 2 main standards (MS lead, Sun/Oracle lead)
  - Older formats documented and published https://msdn.microsoft.com/en-us/library/cc313105(v=office.12).aspx
- OLE documentation: https://msdn.microsoft.com/en-us/library/office/cc313094(v=office.12).aspx
- PDF documentation: https://www.adobe.com/devnet/pdf/pdf_reference_archive.html

Information sources for analyzing DOCs and PDFs

Guides, books:

- https://zeltser.com/analyzing-malicious-documents/
- ...

Some python-based tools, minimal set:

- olevba
- pdf-parser

Next task: download the tools and try them out ...

Document analysis environment

Unpacking files

- Use 7zip - it is able to resolve flaws in implementations used by malware creators
- MS proprietary format .jse
  - scrdec.c - Decoder for Microsoft Script Encoder

Preparing the document analysis environment

Extracting files

Package: mpack
Description: tools for encoding/decoding MIME messages Mpack and munpack are utilities for encoding and decoding (respectively) binary files in MIME (Multipurpose Internet Mail Extensions) format mail messages. For compatibility with older forms of transferring binary files, the munpack program can also decode messages in split-uuencoded format.

Document analysis environment

Document file parsers, helpful tools

- Install olevba
  - A: follow instructions in https://github.com/decalage2/oletools/wiki/olevba
  - B: sudo pip install oletools
- Optional: install oledump.py
  - A: apt install python-olefile
  - B: sudo apt install python-dev && sudo pip install olefile
- Optional: install Okteta (raw data editor)
  - sudo apt install okteta

Analyzing documents

MS Word files

Run olevba(.py) <filename>and explore the output

- -c : display VBA code
- -d : detailed output
- –decode : decode decodable
- –deobf : try to deobfuscate VBA expressions (beta)

Analyzing documents

PDF files

Download pdf-parser and pdfid in https://blog.didierstevens.com/programs/pdf-tools/ and place them in search path for conveninent use

1. triage the file by identifying objects with pdfid
2. explore the PDF with pdf-parser, useful switches:
   - –object : accessing the object
   - –raw : get raw data
   - –filter : apply set of useful filters

Analyzing documents

Packed files

Simple - unpack the file, but ..

- file may be password protected and password provided in the e-mail. That's why it is important to also save the e-mail ...
- file may be packed multiple times (Zip inside Zip inside Zip ...)
- Extracted file can usually be run in the MS enivronment
  - Special case in this section would be files or URLs which are handled by the Apple macOS operating system without end-users understanding of the process. For example .ZIP file is automatically unpacked and processed by the macOS. Combining this feature with URL handling registration routine we can perform successful attack against the modern macOS even without the need for the exploits.

Analyzing documents

Hiding techniques

Parts of code are hidden:

- in encoded format inside legitimate streams
- as attributes of legitimate elements, for example .Tag or .Caption of form element in Micosoft Office documents
- as payload encapsulated in another format, for example MS Word document inside PDF file

Analyzing documents

Obfuscation techniques

- Relevant pieces are injected into legitimate code with huge codebase
- Code is converted/encoded to exotic format (.js to .jse)
- Variable names are randomized
- Strings are created with Chr(x) function, may use aritmetic conversions of constant in a process
- Strings are extracted by splitting long string with custom separator
- Code is presented in backwards
- Dangerous function (WScript, CreateObject etc) names are splitted to parts
- ...

Hands-on

- Download sample: https://github.com/zyxtarmo/isecrelated/raw/master/samples/ORDER-270-8022722-7995817.docm
- Process the file with olevba.py for file triage
- If suspicious - run again with relevant set of keys in order to extract useful content from the file
- Look for CreateObject ...

Hands-on

- Download sample: https://raw.githubusercontent.com/zyxtarmo/isecrelated/master/samples/TASK.eml
- Apply mime part extraction tool to the email source
- Unpack and take a look on the script
- Hint 1: code is in reversed order, you can use Python to reverse the string (remove var and stuff after string ends, add print with string reversing directive)
- Hint 2: some characters have been “hexcoded“, you can use hexdecoder @ http://ddecode.com/hexdecoder/ and add linebreaks after ';' character to get readable code
- Hint 3: ask for an URL with relevant data from lecturer ...