Infecting files on-the-fly
About me

Red Team Leader

Hacking techniques trainer

Security researcher Offensive Security

From Spain & Equatorial Guinea
Why this talk?

VideoLAN, a project and a non-profit organization.

Downloading VLC 2.2.6 for Windows 64 bits

Thanks! Your download will start in few seconds... If not, click here. Display checksum.
Why this talk?
Why this talk?
Why this talk?

D’oh!
Why this talk? – Not all is PE

WinZip

WinZip is the world’s #1 choice when working with large files – zip, unzip, protect, share and more.

- Zip and unzip files instantly
- Protect files with banking-level AES encryption
- Share directly to iCloud Drive, Dropbox, Google Drive and ZipShare, from within WinZip

DOWNLOAD TRIAL  BUY NOW  START OF 30-DAY FREE TRIAL

TOUM: Tablón Oficial de la Universidad de Murcia

TOUM: Official Notice Board of the University of Murcia

Daniél Sánchez Martínez, Immaculada Marín López, Juan Luis Chirleaque Medrano, Juan José González Sánchez

Descargas

Hoy

SAMSUNG_USB_Driver_for_Mobile_Phone.zip
565 KB/s - 12.2 MB de 19.3 MB, Quedan 5 s

PAUSAR  CANCELAR

filetype:docx

Google

Todo  Imágenes  Noticias  Videos  Shopping  Más

Aproximadamente 24.400 resultados (0,55 segundos)

[DOC] MODELOS DE INFORMES PROFESIONALES
www4.unileon.es/trabajo_social/archivos/documentos.docx
Gamma explain what I’m going to talk about
Commercial Tools

FinFisher™
Excellence in IT Investigation

FinFly LAN

Some of the major challenges law enforcement agencies are facing are mobile targets that don’t allow any physical access to their computers and do not open any unknown files they receive. Security-aware targets are almost impossible to monitor as they keep their systems up-to-date and successfully resist common exploits or intrusion techniques.

FinFly LAN covertly deploys remote monitoring solutions on target systems in Local Area Networks (Wired and Wireless). It patches files that are downloaded by the target on-the-fly, sends fake software updates or deploys the monitoring solution into visited websites.

Hacking Team

- RCS 9.6 (stable) 11 Jul 2015 02:48 -- Folder
- Documentation 11 Jul 2015 02:48 -- Folder
- Product 11 Jul 2015 02:49 -- Folder
- Console 11 Jul 2015 02:48 -- Folder
- Injector 11 Jul 2015 02:48 -- Folder
- networkinjector-9.6.0.iso 08 Jul 2015 11:56 1.11 GB ISO Disk Image
- Server 11 Jul 2015 02:49 -- Folder
- Remote Control System 9.6 - Readme.pdf 06 Jul 2015 22:32 43 KB Adobe...cument
Free Tools

Peinjector [https://github.com/JonDoNym/peinjector](https://github.com/JonDoNym/peinjector)

Provides different ways to infect these files with custom payloads without changing the original functionality. It creates patches, which are then applied seamlessly during file transfer. It is very performant, lightweight, modular and can be operated on embedded hardware.

Features

- Full x86 and x64 PE file support.
- Open Source
- Fully working on Windows and Linux, including automated installation scripts.
- Can be operated on embedded hardware, tested on a Raspberry Pi 2.
- On Linux, all servers will be automatically integrated as service, no manual configuration required.
- Plain C, no external libraries required (peinjector).
- MITM integration is available in C, Python and Java. A sample Python MITM implementation is included.
- Foolproof, mobile-ready web interface. Anyone who can configure a home router can configure the injector server.
- Easy to use integrated shellcode factory, including reverse shells, meterpreter, ... or own shellcode. Everything is available in 32 and 64 bit with optional automated encryption. Custom shellcode can be injected directly or as a new thread.
Free Tools

Backdoor Factory Proxy  https://github.com/secretsquirrel/BDFProxy

Patch Binaries via MITM: BackdoorFactory + mitmProxy.

Features

- Full x86 and x64 PE file support.
- ELF and Mach-O
- Open Source

https://www.youtube.com/watch?v=YzeGxlurQy8
Modifying Tools issues

- Resources / icons?
- Antivirus?
- Executable signatures?
- The original code is executed?
- Executables auto-integrity checks
- UAC?
- Other not executable file formats:
  - ZIP
  - RAR
  - PDF
  - MS Office
  - ETC
Modifying Tools issues

What do you think on-the-fly means?
**Download a file vs Download a file through a proxy**

**Index of /videolan/vlc/last/win64**

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.7z</td>
<td>2017-05-24 15:09</td>
<td>28M</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.7z.asc</td>
<td>2017-05-24 15:09</td>
<td>195</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.7z.md5</td>
<td>2017-05-24 15:09</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.7z.sha1</td>
<td>2017-05-24 15:09</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.7z.sha256</td>
<td>2017-05-24 15:09</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.exe</td>
<td>2017-05-24 15:09</td>
<td>31M</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.exe.asc</td>
<td>2017-05-24 15:09</td>
<td>195</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.exe.md5</td>
<td>2017-05-24 15:09</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.exe.sha1</td>
<td>2017-05-24 15:09</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.exe.sha256</td>
<td>2017-05-24 15:09</td>
<td>86</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.xpi</td>
<td>2017-05-26 13:40</td>
<td>32M</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.xpi.asc</td>
<td>2017-05-26 13:40</td>
<td>195</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.xpi.md5</td>
<td>2017-05-26 13:40</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.xpi.sha1</td>
<td>2017-05-26 13:40</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.xpi.sha256</td>
<td>2017-05-26 13:40</td>
<td>86</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.zip</td>
<td>2017-05-26 13:40</td>
<td>51M</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.zip.asc</td>
<td>2017-05-26 13:40</td>
<td>195</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.zip.md5</td>
<td>2017-05-26 13:40</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.zip.sha1</td>
<td>2017-05-26 13:40</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>vlc-2.2.6-win64.zip.sha256</td>
<td>2017-05-26 13:40</td>
<td>86</td>
<td></td>
</tr>
</tbody>
</table>
Files Modification on-the-fly (option 1)

- Dependency of connection speed.
- Dependency of file size.
- Dependency of file format.
File Modification on-the-fly
(option 2)

- (IN)dependency of connection speed.
- (IN)dependency of file size.
- Dependency of file format.
**File Modification on-the-fly (option 2)**

- Know the file format to guess the future modification.
- From the beginning we know the original file size (original Content-Length).
- Resulting file size? (New Content-Length)
- File format with CRC?
- Headers of modified file?
- A study of all file formats is needed.
- Thanks to Ange Albertini (@angealbertini) it is easy with most common filetypes.
PE Infection
### PE Format

#### Normal Sections

- **.text**: executable code
- **.data**: global initialized data
- **.rdata**: global read-only data
- **.edata**: export tables
- **.idata**: import tables
- **.pdata**: exception handling information
- **.xdata**: exception information, free format
- **.reloc**: information for relocation of library files
- **.rsrc**: resources of the executable
- **.directive**: linker options
- **.bss**: uninitialized data, free format

PE Resource Structure

- Structure very unclear even in the specification.
- Usage of virtual relative addresses from uncertain PE points.
- Hard analysis and hard construction.
- Have to consider the alignment inside file space.
EXE Infection

- EXE = executable PE.
- The original EXE must be executed as expected while other process is executed (payload).
- The process must be transparent to the user.
- Protections:
  - File name (reputation)
  - Digital signature (S.O. Level)
  - Integrity check (code level)
**EXE Infection**
*(Option 1)*

- Add new code at the end of the file.

- Change Entry Point.

- Remove digital signature.

- The new code is executed and calls later to the original code.
EXE Infection
(Opción 1)

Original EXE

- Header
  - Entry point
  - .text
  - .data
  - .rsrc

Infected EXE

- Header
  - Entry point
  - .text
  - .data
  - .rsrc
  - .newcode
**EXE Infected**  
*(Option 1)*

- The executable may be signed a posteriori if you have a certificate.

- The auto-integrity check fails.

- Well known method by AVs

- New size = original + new code

- The resource section is preserved so the ICON is not modified
**EXE Infection**

*(Option 2)*

- Add original EXE to a section of the new malware.

- The new EXE executes its magic, copies the section to another file and runs it.
EXE Infection (Option 2)

EXE Original

- Header
- Entry point
- .data
- .text
- .rsrc

Malware + original

- Header
- Entry point
- .data
- .text
- .blob
- Exe original
- .text
- .rsrc
- .rsrc
EXE Infection (Option 2)

• The executable may be signed a posteriori if you have a valid certificate.

• The resource section is preserved so the ICON is not modified

• Size = original + malware + original resources section + alignment
Encimadelamosca (EDM)  
EXE Infection

- Analyzes original EXE’s header (1st stream).
- Modifies the new EXE’s header.
  - Add data section.
  - Sizes
  - Add resources section.
- Sends the new EXE
- Sends the original EXE
- Sends the resources section
- Content-type:

<table>
<thead>
<tr>
<th>File type</th>
<th>Content-types</th>
</tr>
</thead>
<tbody>
<tr>
<td>.exe</td>
<td>application/x-msdownload</td>
</tr>
<tr>
<td></td>
<td>application/octet-stream</td>
</tr>
</tbody>
</table>
Demo:

Infecting an EXE using EDM
ZIP files infection
PKWare ZIP Format

- General structure of non encrypted ZIP file. (simplified)

https://www.pkware.com/documents/casestudies/APPNOTE.TXT
PKWare ZIP Format

- Local file header

<table>
<thead>
<tr>
<th>0x0000</th>
<th>Signature</th>
<th>Version</th>
<th>Flags</th>
<th>Compression</th>
<th>Mod time</th>
<th>Mode date</th>
<th>Crc-32</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0010</td>
<td>Crc-32</td>
<td>Compressed size</td>
<td>Uncompressed size</td>
<td>File name len</td>
<td>Extra field len</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0030</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Extra field (variable size)

PKWare ZIP Format

• Data descriptor

https://www.pkware.com/documents/casestudies/APPNOTE.TXT
PKWare ZIP Format

- Central directory: Contains metadata about the files in the file, including the data about the files encryption. Beside the ZIP files divided in multiple files.

https://www.pkware.com/documents/casestudies/APPNOTE.TXT
# PKWare ZIP Format

- **Central directory file header**

<table>
<thead>
<tr>
<th>Field</th>
<th>Offset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
<td>0x0000</td>
<td>Contains the ZIP file signature (8 bytes)</td>
</tr>
<tr>
<td>Version</td>
<td>0x0010</td>
<td>Contains the version number (2 bytes)</td>
</tr>
<tr>
<td>Version needed</td>
<td>0x0012</td>
<td>Contains the version number of the operating system (2 bytes)</td>
</tr>
<tr>
<td>Flags</td>
<td>0x0014</td>
<td>Contains flags information (2 bytes)</td>
</tr>
<tr>
<td>Compressed size</td>
<td>0x0016</td>
<td>Contains the size of the compressed file (4 bytes)</td>
</tr>
<tr>
<td>Uncompressed size</td>
<td>0x001A</td>
<td>Contains the size of the uncompressed file (4 bytes)</td>
</tr>
<tr>
<td>File name len</td>
<td>0x001E</td>
<td>Contains the length of the file name (2 bytes)</td>
</tr>
<tr>
<td>Extra field len</td>
<td>0x0020</td>
<td>Contains the length of the extra field (2 bytes)</td>
</tr>
<tr>
<td>File comm. len</td>
<td>0x0022</td>
<td>Contains the length of the file comment (2 bytes)</td>
</tr>
<tr>
<td>Disk # start</td>
<td>0x0024</td>
<td>Contains the disk # start (2 bytes)</td>
</tr>
<tr>
<td>Internal attr.</td>
<td>0x0026</td>
<td>Contains internal attributes (2 bytes)</td>
</tr>
<tr>
<td>External attr.</td>
<td>0x0028</td>
<td>Contains external attributes (2 bytes)</td>
</tr>
<tr>
<td>Offset of local header</td>
<td>0x002A</td>
<td>Contains the offset of the local header (4 bytes)</td>
</tr>
</tbody>
</table>

- **End**

<table>
<thead>
<tr>
<th>Field</th>
<th>Offset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
<td>0x0000</td>
<td>Contains the ZIP file signature (8 bytes)</td>
</tr>
<tr>
<td>Disk number</td>
<td>0x0010</td>
<td>Contains the disk number (4 bytes)</td>
</tr>
<tr>
<td>Disk w/ocd</td>
<td>0x0014</td>
<td>Contains disk with the central directory (2 bytes)</td>
</tr>
<tr>
<td>Disk entries</td>
<td>0x0016</td>
<td>Contains the number of disk entries (2 bytes)</td>
</tr>
<tr>
<td>Total entries</td>
<td>0x0018</td>
<td>Contains the total number of entries (2 bytes)</td>
</tr>
<tr>
<td>Central directory size</td>
<td>0x001A</td>
<td>Contains the size of the central directory (4 bytes)</td>
</tr>
<tr>
<td>Offset of ad w/ starting disk</td>
<td>0x001E</td>
<td>Contains the offset of the starting disk (4 bytes)</td>
</tr>
<tr>
<td>Comment len</td>
<td>0x0022</td>
<td>Contains the length of the comment (2 bytes)</td>
</tr>
<tr>
<td>ZIP file comment</td>
<td>0x0024</td>
<td>Variable length for ZIP file comment</td>
</tr>
</tbody>
</table>

Encimadelamosca
ZIP files infection(beta)
Encimadelamosca
ZIP file infection

• Delays with big compressed files.
• Encrypted ZIP files (Being resolved).
• Size = file size + infected data + padding.
• Capability of adding files on-the-fly.
• Content-type:

<table>
<thead>
<tr>
<th>File type</th>
<th>Content-types</th>
</tr>
</thead>
<tbody>
<tr>
<td>.zip</td>
<td>application/zip</td>
</tr>
<tr>
<td></td>
<td>application/octet-stream</td>
</tr>
</tbody>
</table>
DEMO:
Infecting an ZIP using EDM
OOXML Infection
OOXML Format

- Office Open XML is a standard file format which most common extensions are .docx, .xlsx, .pptx and .ppsx.

OOXML Format

Identifies the content type in the document, such as the document’s main body, styles, configuration, and file properties.

Contains information about the files relationships (XML)

Defines the relationships between the directory tree files and the document.

Document in xml format

Document properties

Files properties
OOXML Format

```
[Dinamics:hitb Leonardo]$ unzip document.docx
Archive:  document.docx
  inflating: [Content_Types].xml
  inflating: _rels/.rels
  inflating: word/_rels/document.xml.rels
  inflating: word/document.xml
  inflating: word/theme/theme1.xml
  extracting: word/media/image1.png
  extracting: word/media/image2.png
  inflating: word/settings.xml
  inflating: word/webSettings.xml
  inflating: word/styles.xml
  inflating: word/numbering.xml
  inflating: docProps/core.xml
  inflating: word/fontTable.xml
  inflating: docProps/app.xml
Dinamics:hitb Leonardo$
```
Encimadelamosca
OOXML Infection

• Similar to ZIP file

• Content-types:

<table>
<thead>
<tr>
<th>File type</th>
<th>Content-types</th>
</tr>
</thead>
<tbody>
<tr>
<td>.docx</td>
<td>application/vnd.openxmlformats-officedocument.wordprocessingml.document application/octet-stream application/msword</td>
</tr>
<tr>
<td>.xlsx</td>
<td>application/vnd.openxmlformats-officedocument.spreadsheetml.sheet application/octet-stream</td>
</tr>
<tr>
<td>.pptx</td>
<td>application/vnd.openxmlformats-officedocument.presentationml.presentation application/octet-stream</td>
</tr>
<tr>
<td>.ppsx</td>
<td>application/vnd.openxmlformats-officedocument.presentationml.slideshow application/octet-stream</td>
</tr>
</tbody>
</table>
DEMO:
Infecting an OOOXML using EDM
Other files type Infection??
TODO

- **PDF** - How to infect on the paper, just need to code it
- **RAR** - How to infect on the paper, just need to code it
- **TGZ** - TAR + GZ - How to infect on the paper, just need to code it
- **APK** - It is just a ZIP - same technique
**TODO**

- Torrent
- Mac OS X Disk Image **DMG**
- Mac OS X Installer Package **PKG**
- Java **CLASS** (it exists inside JARs which are ZIP files too)
TODO

- C-code porting for embedded devices and HA
- New techniques of infection OTF
- PE signature implementation
EDM

EDM available on Git:

https://b1b.es/edm
The End...