Binary Art

funky PoCs & visual docs

Ange Albertini
2014/03/21

Insomni'hack
Geneva, Switzerland

reverse engineering

&

visual documentations

http://corkami.com

boring?

- file formats were supposed to be safe
  - public specs
  - open-source parsers
- {weirdness} == {exploits} ?
- software = parse, sanitize, recover

formats' diversity 1/2

no header

- COM (1982), MBR (1982)

magic signature

- none: DOL (2001)
- 2: TIFF, PE
- 4: most standard
- >4: PNG, PDF

formats’ diversity 2/2

start offset:

- archives
- range: PDF
- mostly 0

special properties

- deprecated header: PE
- variable scanning direction: PDF
- multi-versions: BMP
- scanned chunk: JPEG
- no official names: ZIP

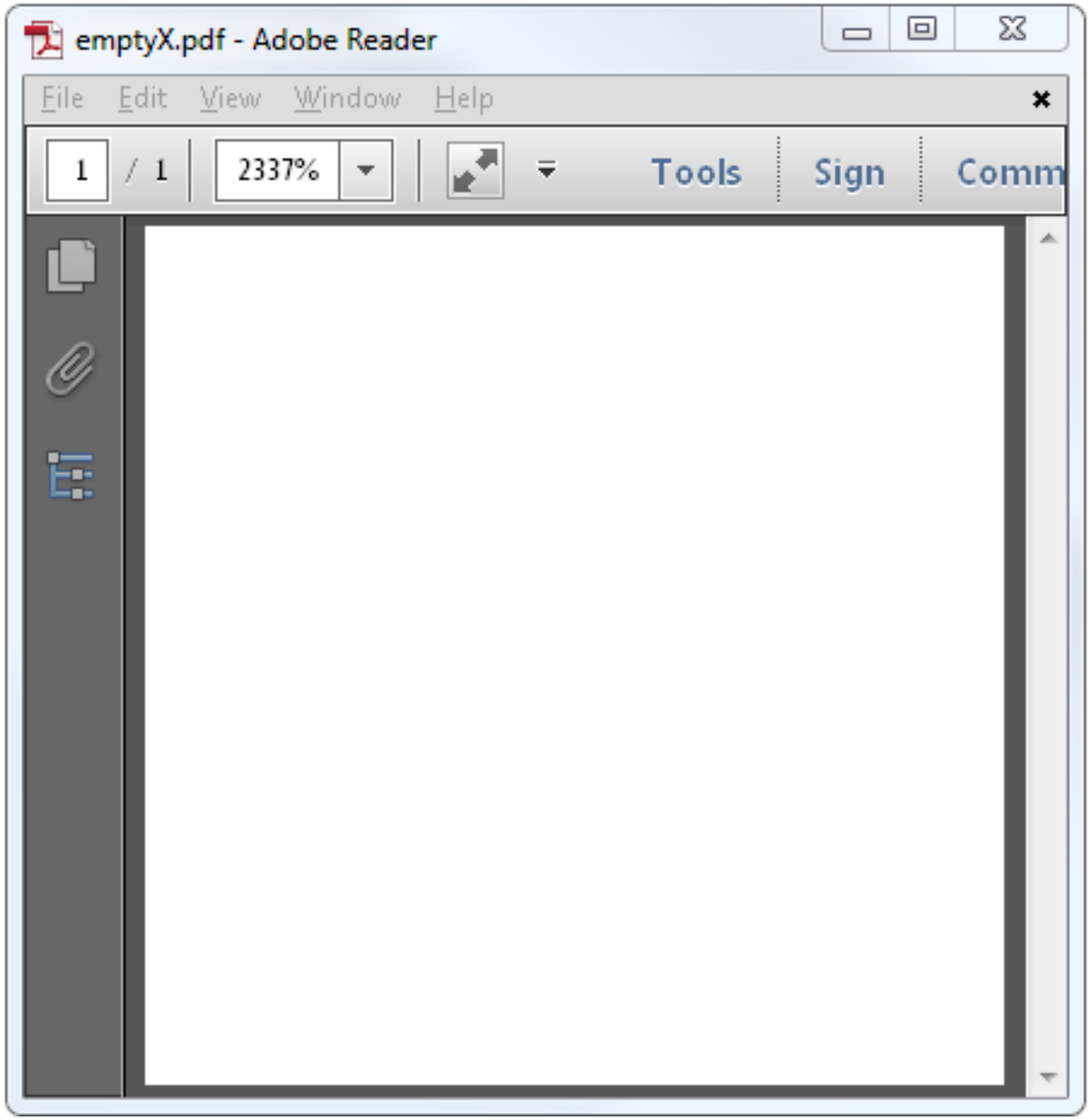
Offset	Size	Field	Description
0	2	Machine	The number that identifies the type of target machine. For more information, see section 3.3.1, “Machine Types.”
2	2	NumberOfSections	The number of sections. This indicates the size of the section table, which immediately follows the headers.
4	4	TimeDateStamp	The low 32 bits of the number of seconds since 00:00 January 1, 1970 (a C run-time time_t value), that indicates when the file was created.
8	4	PointerToSymbolTable	The file offset of the COFF symbol table, or zero if no COFF symbol table is present. This value should be zero for an image because COFF debugging information is deprecated.
12	4	NumberOfSymbols	The number of entries in the symbol table. This data can be used to locate the string table, which immediately follows the symbol table. This value should be zero for an image because COFF debugging information is deprecated.
16	2	SizeOfOptionalHeader	The size of the optional header, which is required for executable files but not for object files. This value should be zero for an object file. For a description of the header format, see section 3.4, “Optional Header (Image Only).”
18	2	Characteristics	The flags that indicate the attributes of the file. For specific flag values, see section 3.3.2, “Characteristics.”

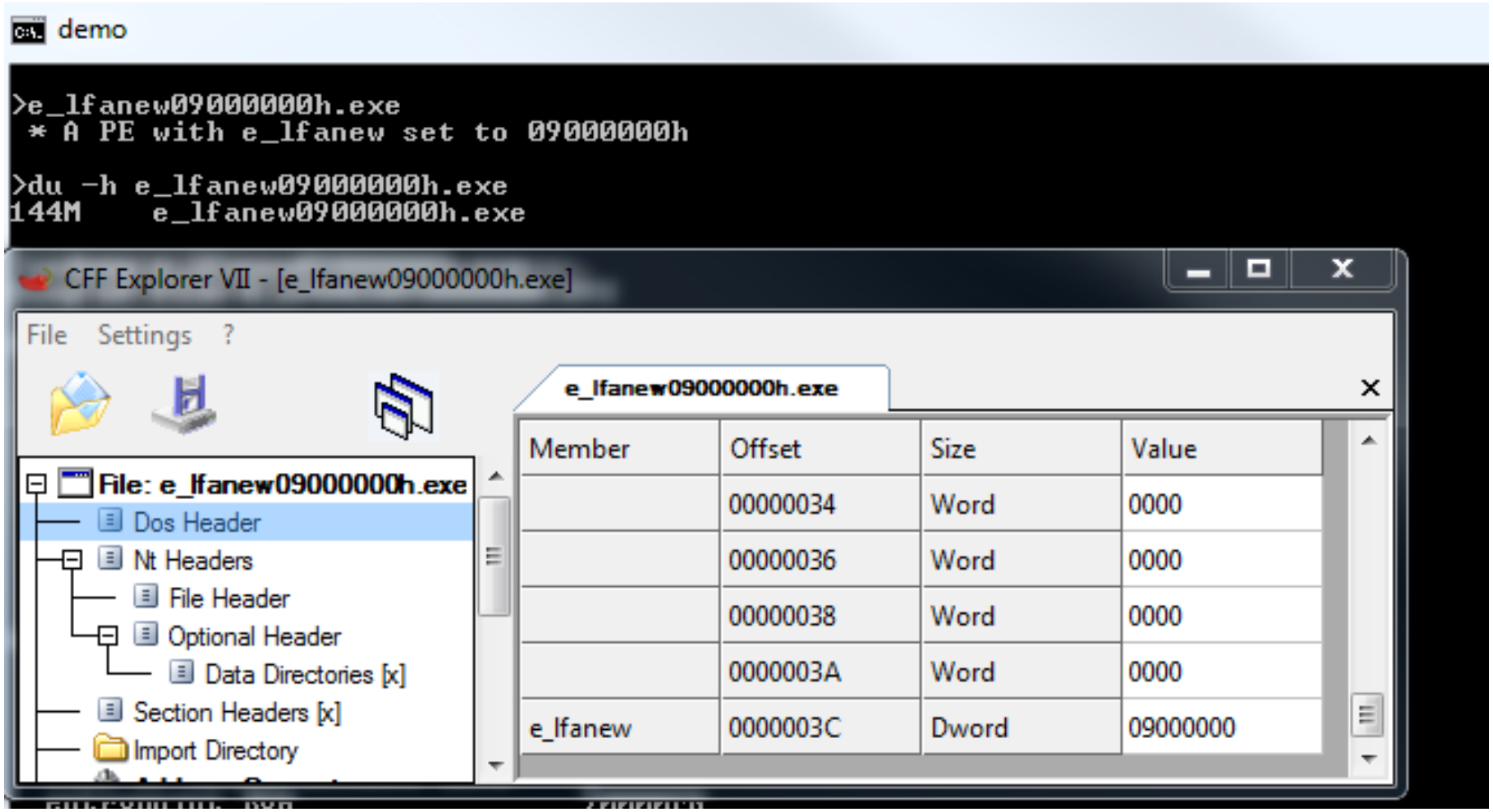
```
struc IMAGE_FILE_HEADER
  .Machine                resw 1
  .NumberOfSections       resw 1
  .TimeDateStamp          resd 1
  .PointerToSymbolTable   resd 1
  .NumberOfSymbols        resd 1
  .SizeOfOptionalHeader   resw 1
  .Characteristics        resw 1
endstruc
```

```
istruc IMAGE_FILE_HEADER
  at IMAGE_FILE_HEADER.Machine,              dw IMAGE_FILE_MACHINE_I386
  at IMAGE_FILE_HEADER.NumberOfSections,     dw NUMBEROFSECTIONS
  at IMAGE_FILE_HEADER.TimeDateStamp,        dd 04b51f504h       ; 2010/1/16 5:19pm
  at IMAGE_FILE_HEADER.SizeOfOptionalHeader, dw SIZEOFOPTIONALHEADER
  at IMAGE_FILE_HEADER.Characteristics,      dw IMAGE_FILE_RELOCS_STRIPPED |
                                                IMAGE_FILE_EXECUTABLE_IMAGE | \
                                                IMAGE_FILE_LINE_NUMS_STRIPPED | \
                                                IMAGE_FILE_LOCAL_SYMS_STRIPPED | \
                                                IMAGE_FILE_32BIT_MACHINE
iend
```

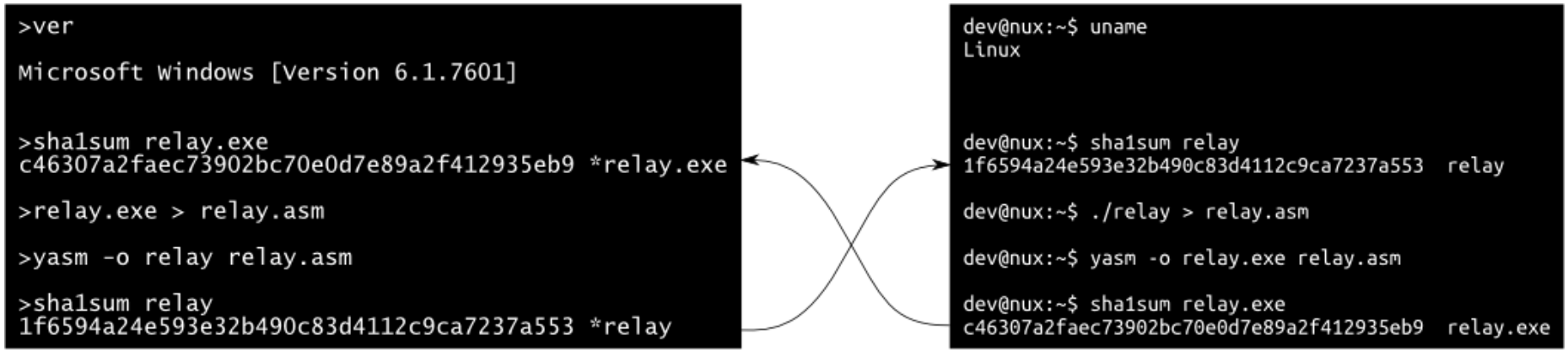
```
istruc IMAGE_FILE_HEADER
  at IMAGE_FILE_HEADER.Machine,                dw 0xffff
  at IMAGE_FILE_HEADER.NumberOfSections,       dw 0xffff
  at IMAGE_FILE_HEADER.TimeDateStamp,          dd 0xffffffff
  at IMAGE_FILE_HEADER.PointerToSymbolTable,   dd 0xffffffff
  at IMAGE_FILE_HEADER.NumberOfSymbols,        dd 0xffffffff
  at IMAGE_FILE_HEADER.SizeOfOptionalHeader,   dw SIZEOFOPTIONALHEADER
  at IMAGE_FILE_HEADER.Characteristics,        dw 0xffff
iend
```

%PDF-NULtrailer<</Root<</Pages<<>>>>>>

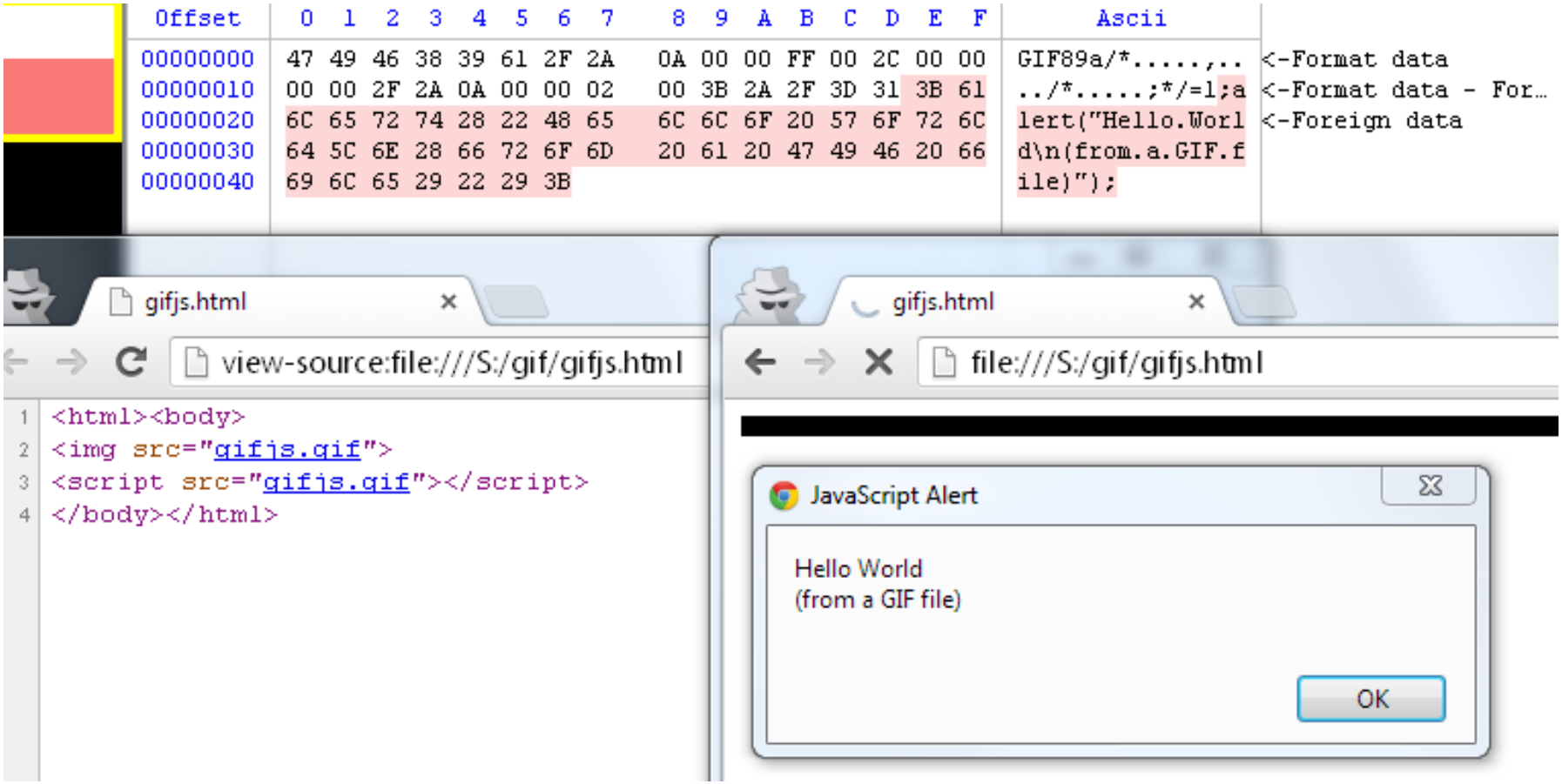



demo
>e_lfanew09000000h.exe
* A PE with e_lfanew set to 09000000h
>du -h e_lfanew09000000h.exe
144M e_lfanew09000000h.exe
CFF Explorer VII - [e_lfanew09000000h.exe]
File Settings ?
File: e_lfanew09000000h.exe
Dos Header
Nt Headers
File Header
Optional Header
Data Directories [x]
Section Headers [x]
Import Directory
e_lfanew09000000h.exe
Member Offset Size Value
00000034 Word 0000
00000036 Word 0000
00000038 Word 0000
0000003A Word 0000
e_lfanew 0000003C Dword 09000000

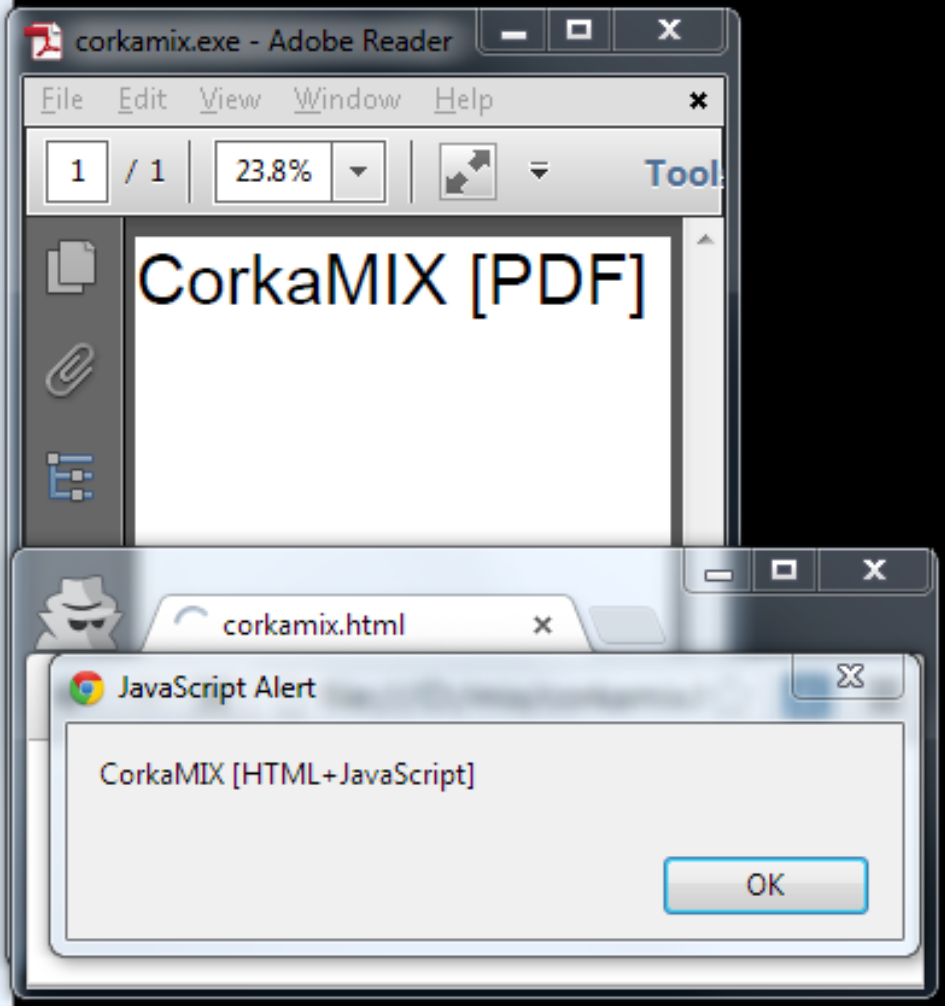
quine (relay)


>ver
Microsoft Windows [Version 6.1.7601]
>sha1sum relay.exe
c46307a2faec73902bc70e0d7e89a2f412935eb9 *relay.exe
>relay.exe > relay.asm
>yasm -o relay relay.asm
>sha1sum relay
1f6594a24e593e32b490c83d4112c9ca7237a553 *relay
dev@nux:~$ uname
Linux
dev@nux:~$ sha1sum relay
1f6594a24e593e32b490c83d4112c9ca7237a553 relay
dev@nux:~$ ./relay > relay.asm
dev@nux:~$ yasm -o relay.exe relay.asm
dev@nux:~$ sha1sum relay.exe
c46307a2faec73902bc70e0d7e89a2f412935eb9 relay.exe

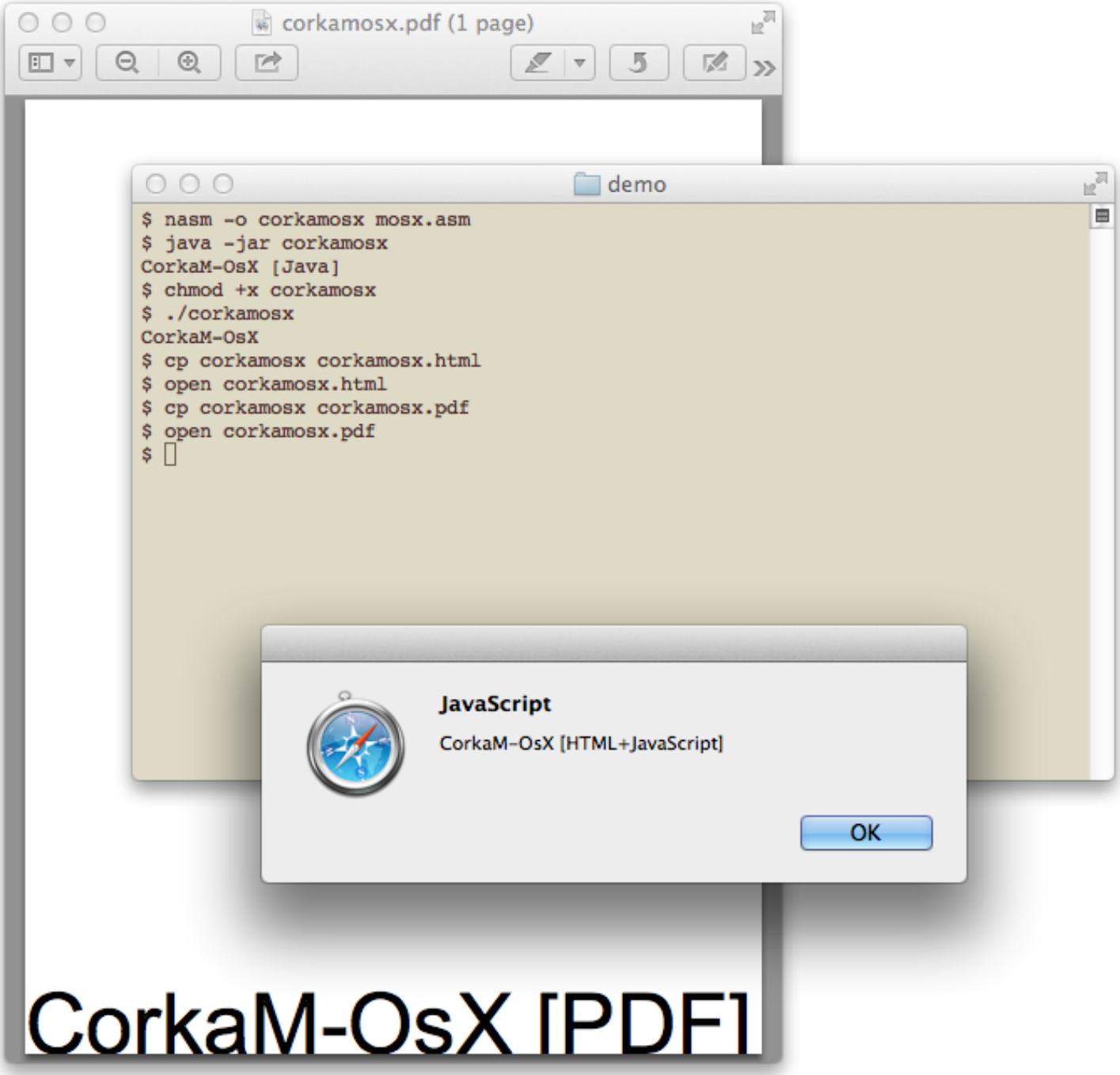
polyglot

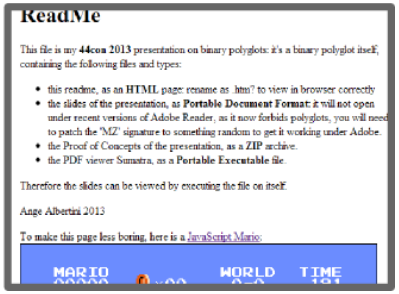

Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F Ascii
00000000 47 49 46 38 39 61 2F 2A 0A 00 00 FF 00 2C 00 00 GIF89a/*.....,..
00000010 00 00 2F 2A 0A 00 00 02 00 3B 2A 2F 3D 31 3B 61 ../*.....;*/=1;a
00000020 6C 65 72 74 28 22 48 65 6C 6C 6F 20 57 6F 72 6C lert("Hello.Worl
00000030 64 5C 6E 28 66 72 6F 6D 20 61 20 47 49 46 20 66 d\n(from.a.GIF.f
00000040 69 6C 65 29 22 29 3B ile)");
<-Format data
<-Format data - For…
<-Foreign data
gifjs.html
view-source:file:///S:/gif/gifjs.html
1 <html><body>
2 <img src="gifjs.gif">
3 <script src="gifjs.gif"></script>
4 </body></html>
gifjs.html
file:///S:/gif/gifjs.html
JavaScript Alert
Hello World
(from a GIF file)
OK

```
>corkamix.exe
CorkaMIX [PE]
>java -jar corkamix.exe
CorkaMIX [Java CLASS in JAR]

>cmp -b corkamix.exe corkamix_1b.exe
cmp: EOF on corkamix.exe

>python corkamix_1b.exe
CorkaMIX [python]

>copy corkamix.exe corkamix.html
        1 file(s) copied.
```




```
db 'MZ'
; [...]
db '%PDF-1.', 0ah
db 'obj<<>>stream', 0ah

db '<html>'
; [...]
    at IMAGE_NT_HEADERS.Signature, db 'PE',0,0
; [...]
    db 0fh, 018h, 111b << 3
    push msg
    call [__imp__printf]
    salc
; [...]
header:
    db 'PK', 3, 4
    dw 0ah ; version_needed
; [...]
_dd 0CAFEBABEh ; signature
_dw 3          ; major version
_dw 2dh        ; minor version
; [...]
            _dd 9 ; length of bytecode
                GETSTATIC 8
                LDC 14
                INVOKEVIRTUAL 16
                RETURN
            _dw 0 ; exceptions_count
            _dw 0 ; attributes_count
; [...]
```


corkamosx.pdf (1 page)
demo
$ nasm -o corkamosx mosx.asm
$ java -jar corkamosx
CorkaM-OsX [Java]
$ chmod +x corkamosx
$ ./corkamosx
CorkaM-OsX
$ cp corkamosx corkamosx.html
$ open corkamosx.html
$ cp corkamosx corkamosx.pdf
$ open corkamosx.pdf
$
JavaScript
CorkaM-OsX [HTML+JavaScript]
OK
CorkaM-OsX [PDF]



The file being studied is a Portable Executable file! More specifically, it is a PDF file for the Windows GUI subsystem.

pocorgtfo02.pdf - SumatraPDF

File View Go To Zoom Favorites Settings Help

Page: 1 / 32 Find:

Children's Bible Coloring Book of PoC || GTFO
Issue 0x02, an Epistle to the 30th CCC Congress in Hamburg

Composed by the Rt. Revd. Pastor Manul Laphroaig to put pwnage before politics.
pastor@phrack.org

Antivirus scan for 39e5658

https://www.virustotal.com/en/file/f427e8d95c0ac1

Community Statistics Documentation FAQ About

File identification

MD5	39e5658e24a08e786955af1f4d7e2852
SHA1	2434e76e2d3a4dcd36d0ada363e3a9ed59272f60
SHA256	f427e8d95c0ac15abe61d96fb75cfb55df1fd5ac9e713
ssdeep	393216:VBwNFodCfQD/l+pEfNlcY/hS2L1dUWFF1
File size	13.5 MB (14109425 bytes)
File type	PDF
Magic literal	x86 boot sector
TrID	Unknown!
Tags	pdf

VirusTotal metadata

2013-12-28 20:54:41 UTC (1 day, 12 hours ago)

2013-12-28 20:54:41 UTC (1 day, 12 hours ago)

pocorgtfo02.pdf

demo - qemu-system-i386 -fda pocorgtfo02.pdf

```
>qemu-system-i386 -fda pocorgtfo02.pdf
```

QEMU

```
 ____ ____    ___  ____
| __ ) ___| / _ \/ ___|
|  _ \___ \| | | \___ \
| |_) |__) | |_| |___) |
|____/____/ \___/|____/
Berliner Spargel Operating System
Mein Deutsch is nicht so gut, aber es ist Spargel zeit!
by Travis Goodspeed


m -- Memory Viewer
a -- About



This is a minimal operating system by Travis Goodspeed for 16-bit Real
Mode 8086 on an IBM PC.  It was written in order to learn about the
8086, and it quite likely will serve no use for you.  It is free
without any strings attached, but please give credit were credit is
due if you fork it.

Also, and this is very important, you should use the included hex viewer
to poke around this machine's memory.  The boot sector at 0000:7C000
is likely a good place to start.
Press the 'any' key to continue.
```

schizophren

44con-albertini.pdf - SumatraPDF

File View Go To Zoom Favorites Settings Help

Page: 1 / 119 Find:

Messing with binary formats

ΠΟΛΎΓΛΩΣΣΟΣ

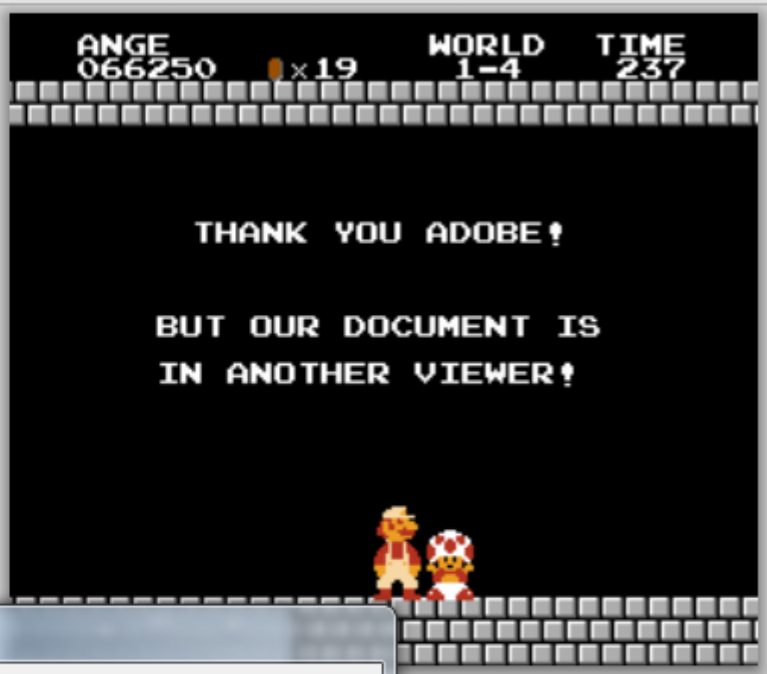
44CON

Ange Albertini

44

44con-albertini.pdf

file:///S:/44con-albertini.pdf



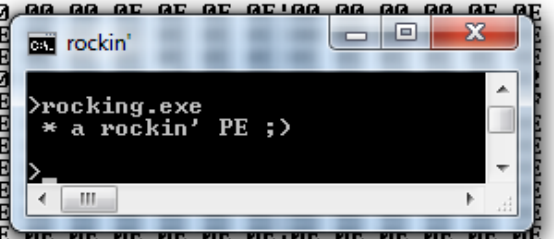
Adobe Reader

Adobe Reader could not open '44con-albertini.pdf' because it is either not a supported file type or because the file has been damaged (for example, it was sent as an email attachment and wasn't correctly decoded).

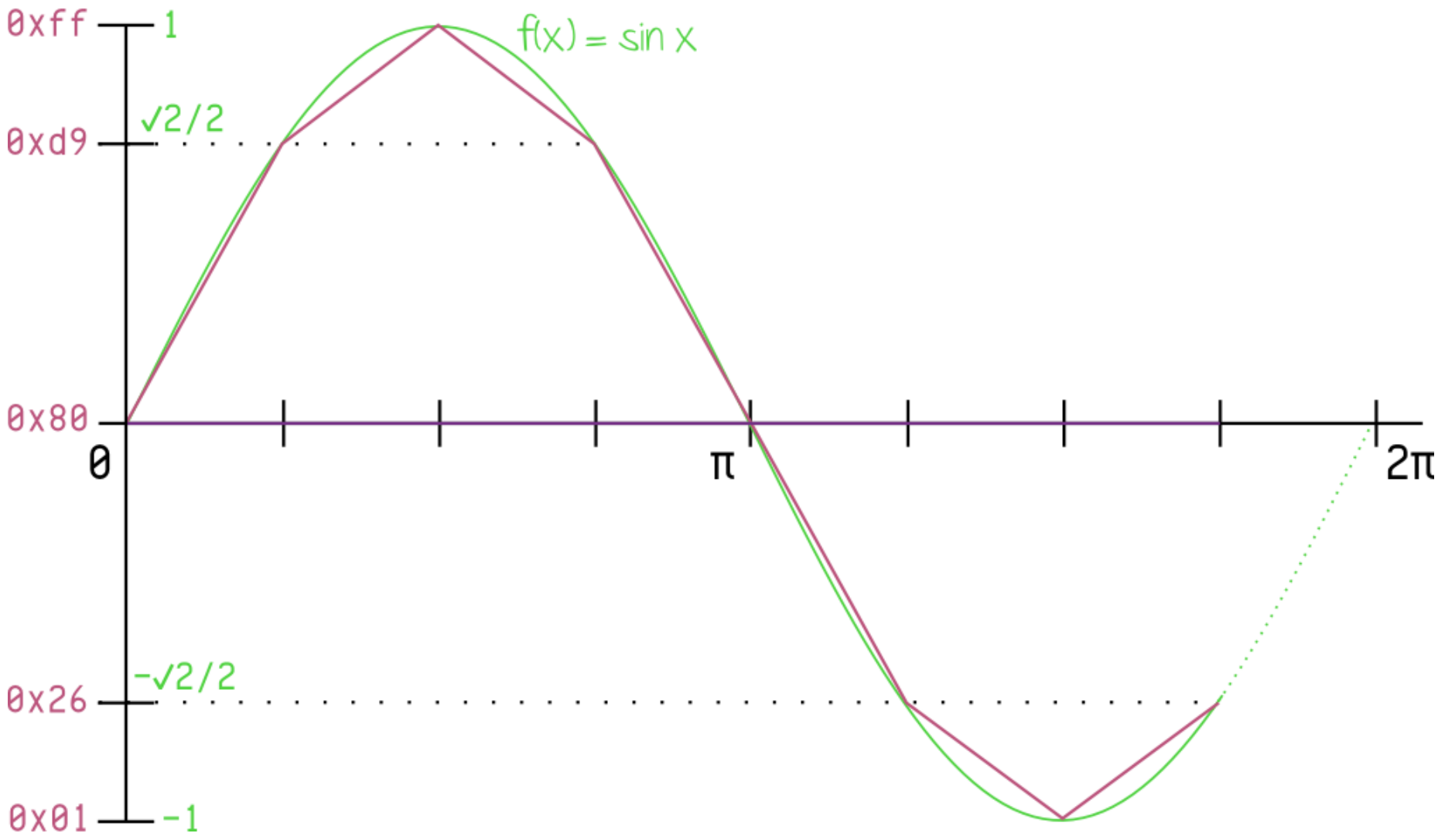
OK

misc

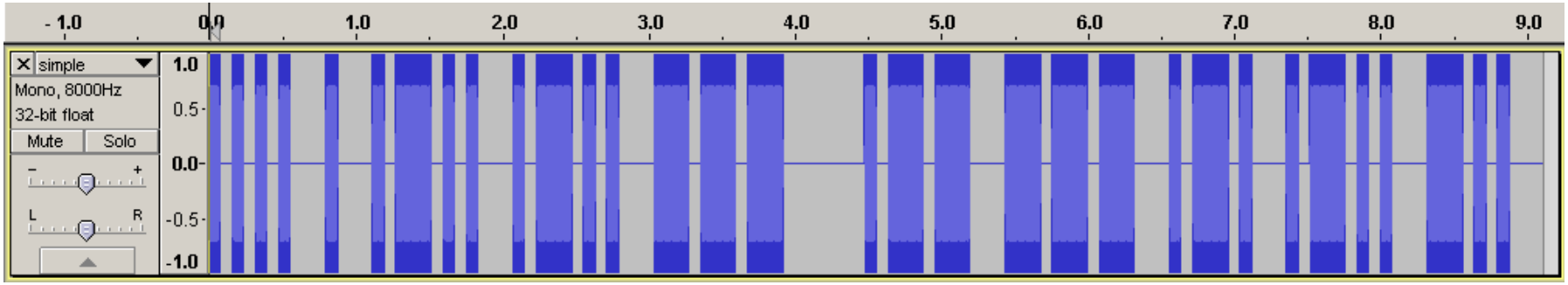
```
00000000: 4D 5A 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  | MZ
00000010: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000020: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000030: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 40 00 00 00  |             @
00000040: 50 45 00 00 4C 01 01 00:0E 0E 0E 0E 0E 0E 0E 0E  | PE  L
00000050: 0E 0E 0E 0E E0 00 0E DF:0B 01 0E 0E 0E 0E 0E 0E  |
00000060: 0E 0E 0E 0E 0E 0E 0E 0E:00 10 00 00 0E 0E 0E 0E  |
00000070: 0E 0E 0E 0E 00 00 40 00:00 10 00 00 00 02 00 00  |
00000080: 0E 0E 0E 0E 0E 0E 0E 0E:04 00 0E 0E 0E 0E 0E 0E  |
00000090: 00 20 00 00 60 01 00 00:0E 0E 0E 0E 03 00 7F EF  |
000000A0: FF FF FF 00 FF 1F 00 00:FF FF FF 00 FF 1F 00 00  |
000000B0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000000C0: 2A 10 00 00 0E 0E 0E 0E:00 00 00 00 0E 0E 0E 0E  | *
000000D0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000000E0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 00 00 00 00  |
000000F0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000100: 00 00 00 00 0E 0E 0E 0E:00 00 00 00 0E 0E 0E 0E  |
00000110: 00 00 00 00 0E 0E 0E 0E:00 00 00 00 0E 0E 0E 0E  |
00000120: 0E 0E                                           |
00000130: 0E 0E                                           |
00000140: 00 10                                           |
00000150: 0E 0E                                           |
00000160: 0E 0E                                           |
00000170: 0E 0E                                           |
00000180: 0E 0E                                           |
00000190: 0E 0E                                           |
000001A0: 0E 0E                                           |
000001B0: 0E 0E                                           |
000001C0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000001D0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000001E0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000001F0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000200: 68 16 10 40 00 FF 15 8D:10 40 00 83 C4 04 6A 00  | h  @  @  j
00000210: FF 15 85 10 40 00 20 2A:20 61 20 72 6F 63 6B 69  |    @   * a rocki
00000220: 6E 27 20 50 45 20 3B 29:0A 00 66 10 00 00 0E 0E  | n' PE ;)  f
00000230: 0E 0E 0E 0E 0E 0E 95 10:00 00 85 10 00 00 0E 0E  |
00000240: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E A2 10 00 00 8D 10  |
00000250: 00 00 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000260: 0E 0E 00 00 00 00 6E 10:00 00 00 00 00 00 0E 0E  |      n
00000270: 45 78 69 74 50 72 6F 63:65 73 73 00 0E 0E 70 72  | ExitProcess  pr
00000280: 69 6E 74 66 00 6E 10 00:00 FF FF FF FF 7C 10 00  | intf n        |
00000290: 00 00 00 00 6B 65 72 6E:65 6C 33 32 2E 64 6C     |     kernel32.dl
000002A0: 6C 00 6D 73 76 63 72 74:2E 64 6C 6C 00 0E 0E     | l msvcrt.dll
000002B0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000002C0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000002D0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000002E0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000002F0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000300: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000310: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000320: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000330: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000340: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000350: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000360: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000370: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000380: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
00000390: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000003A0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000003B0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000003C0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000003D0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000003E0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
000003F0: 0E 0E 0E 0E 0E 0E 0E 0E:0E 0E 0E 0E 0E 0E 0E 0E  |
```



```
128, 217, 255, 217,
128,  38,   1,  38
```


0xff
1
f(x) = sin x
√2/2
0xd9
0x80
0
π
2π
-√2/2
0x26
0x01
-1

```
%macro _beep 0
  times DURATION db 128, 217, 255, 217, 128, 38, 1, 38
%endmacro

%macro _silence 0
  times DURATION * 8 db 128
%endmacro

%macro _dot 0
  _beep
_silence
%endmacro

%macro _e 0
  _dot
%endmacro
```

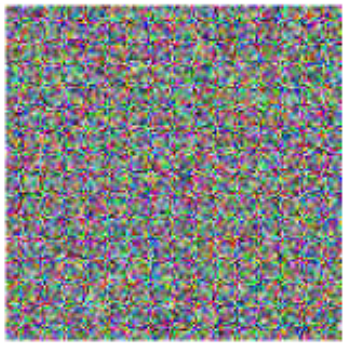
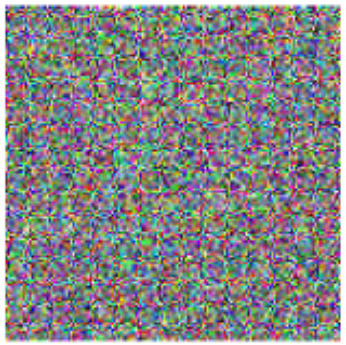


crypto-tology:

for anything crypto, ask **@veorq**

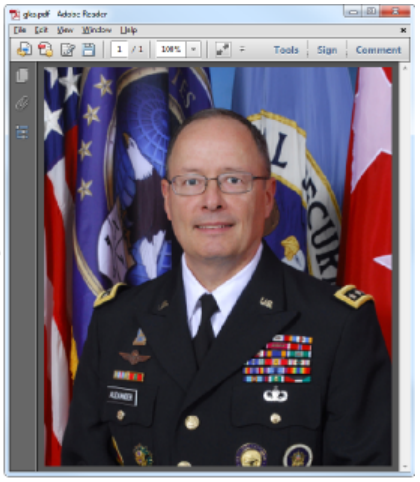
(coz he’s awesome)

THE ADOBE LOGO, ENCRYPTED WITH 3DES IN ECB MODE
(THE SAME ALGORITHM THEY USE TO STORE PASSWORDS)

AES() = 

WITH KEY = '\XE3#X\XAD\X05\XA0\X87\X8B\X1A\X83\XE8\XCA\X1D\XB8=N'

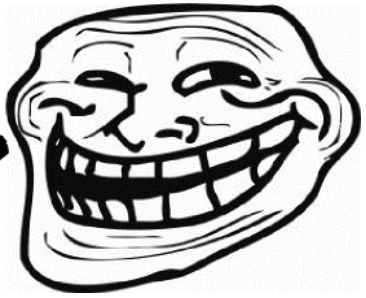
ANY RAW IMAGE WILL ENCRYPT AS A RAW IMAGE !

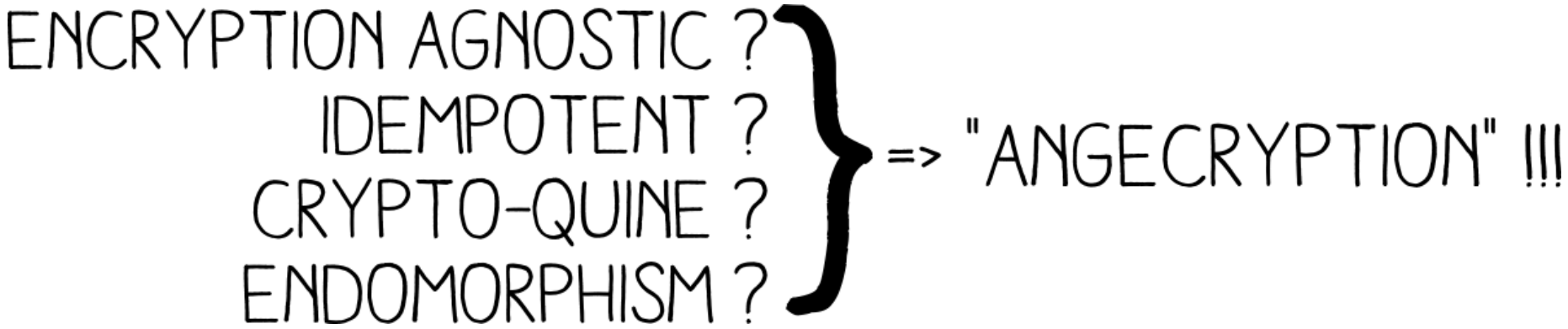


RSA SECURITY

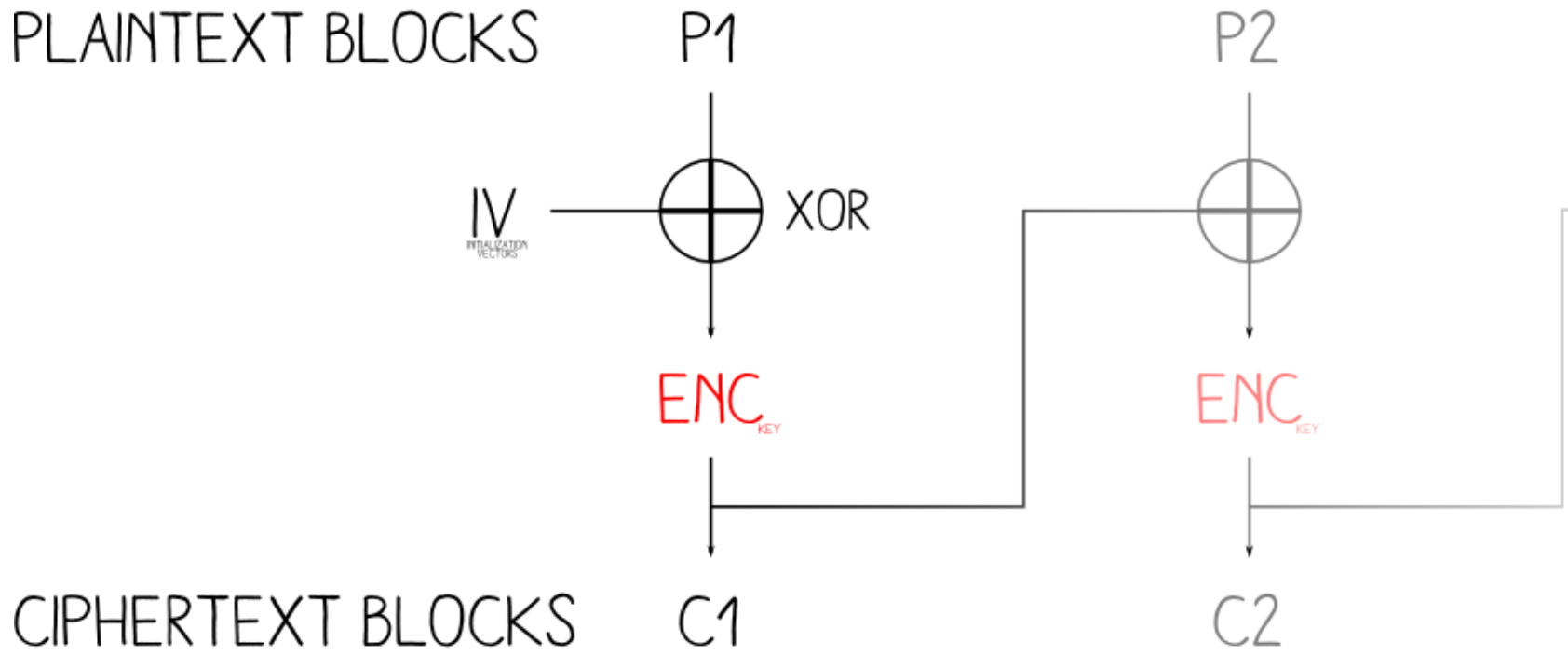
NATIONAL SECURITY AGENCY
UNITED STATES OF AMERICA

Adobe




ENCRYPTION AGNOSTIC ?
IDEMPOTENT ?
CRYPTO-QUINE ?
ENDOMORPHISM ?
=> "ANGECRYPTION" !!!

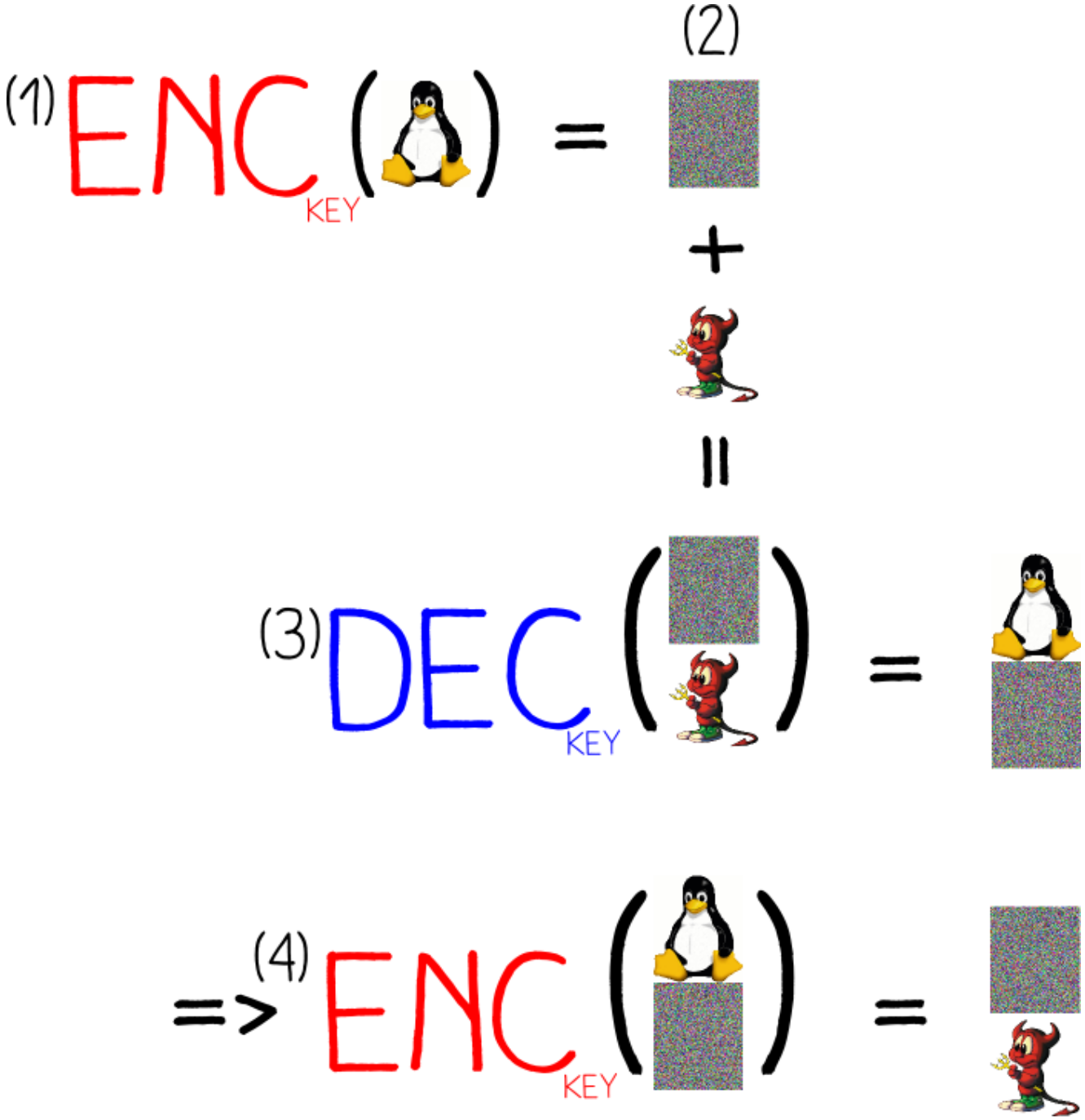
CIPHER BLOCK CHAINING



$$C1 = ENC_{KEY}(P1 \hat{} IV)$$

$$DEC_{KEY}(C1) = P1 \hat{} IV$$

$$IV = DEC_{KEY}(C1) \hat{} P1$$


(1)
ENC
KEY
(2)
+
=
(3)
DEC
KEY
=
=>
(4)
ENC
KEY
=

CONTENTS

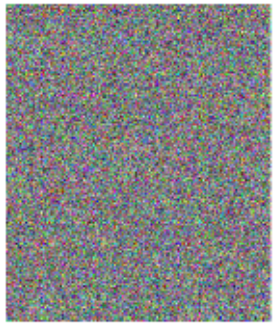
(1)

PNG SIGNATURE `89 .P .N .G 0d 0a 1a 0a`

STARTING A DUMMY CHUNK `.. .. .. .. .. .. .. .. xx xx xx xx tt tt tt tt`

CHUNK LENGTH — CHUNK TYPE

RANDOM ENCRYPTED DATA



ENDING DUMMY CHUNK `yy yy yy yy`

CHUNK CRC

(2)

STARTING CONTROLLED DATA `.. .. .. .. 00 00 00 0d .I .H .D .R`

ORIGINAL IMAGE HEADER

END OF IMAGE `...00 00 00 00 .I .E .N .D AE 42 60 82`

pocorgtfo03.pdf - Adobe Reader

File Edit View Window Help

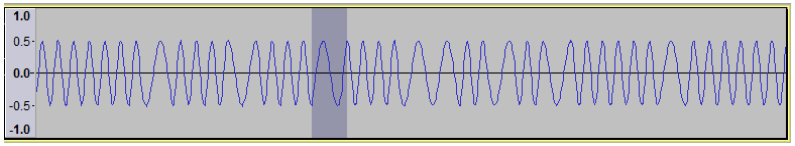
1 / 40 55.3% Tools Sign Comment

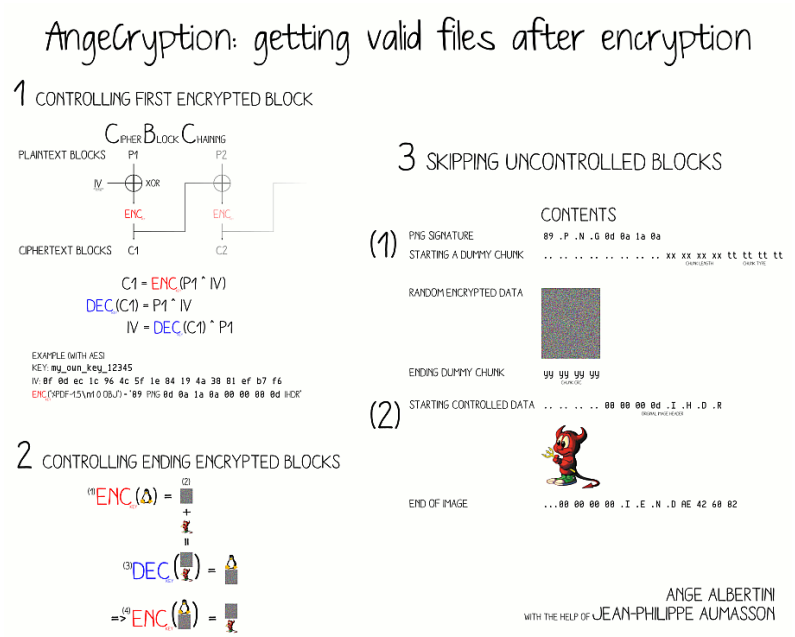
AN ADDRESS
to the
SECRET SOCIETY
of
POC || GTFO
concerning
THE GOSPEL OF THE WEIRD MACHINES
and also
THE SMASHING OF IDOLS TO BITS AND BYTES
by the Rt. Revd. Dr.
PASTOR MANUL LAPHROAIG

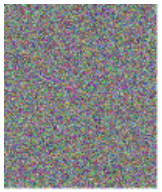
pastor@phrack.org

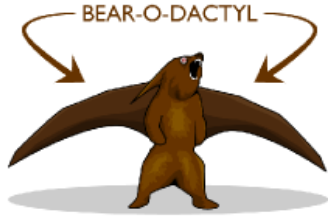
March 2, 2014

```
Archive:  pocorgtfo03.pdf
warning [pocorgtfo03.pdf]:  12224072 extra bytes at beginning or within zipfile
  (attempting to process anyway)
  Length      EAs    ACLs    Date     Time    Name
 --------     ---    ----    ----     ----    ----
     2561       0       0  02/10/14 06:23   alexander.txt
     7848       0       0  02/08/14 20:20   bochs-2.6.2.patch
     6135       0       0  02/08/14 20:21   bochs-20140203.patch
     7248       0       0  02/09/14 08:35   defusing.zip
     4830       0       0  12/01/13 15:48   despair.txt
    14892       0       0  11/27/13 19:03   lasta.txt
    26325       0       0  02/07/14 21:06   lastq.txt
   473449       0       0  02/07/14 21:06   netwatch-337f8b1.tar.gz
   131930       0       0  02/24/14 20:32   nokiacipher.png
    14645       0       0  02/17/14 18:52   packed
     2129       0       0  02/07/14 21:06   saucers.txt
     3144       0       0  02/07/14 21:06   tamadec.txt
     6227       0       0  02/07/14 21:06   tetranglix.tar.bz2
 14109425       0       0  02/07/14 21:06   pocorgtfo02.pdf
      322       0       0  03/03/14 01:28   pocorgtfo03-encrypt.py
 --------     -----   -----                   -------
 14811110       0       0                   15 files
```





 + DECOY KEY => 

 + REAL KEY => 


```
>crypto_hash *
test0.jpg 13990732b0d16c3e112f2356bd3d0dad1....
test1.jpg 13990732b0d16c3e112f2356bd3d0dad1....
```

conclusion
on binary formats

On binary formats

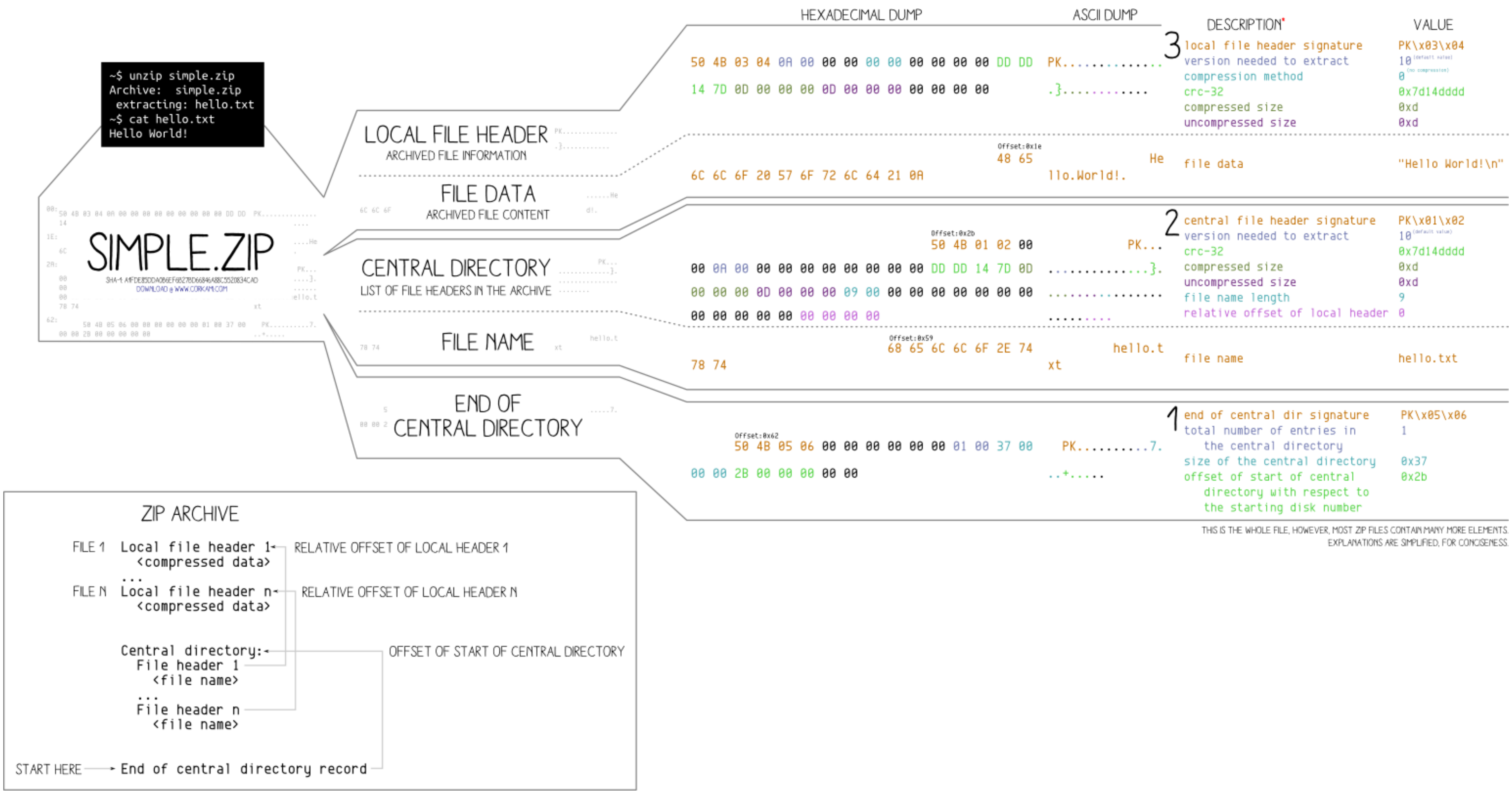
- specs far from perfect
- plenty of fun
- many consequences for infosec
  - unforeseen attack channels

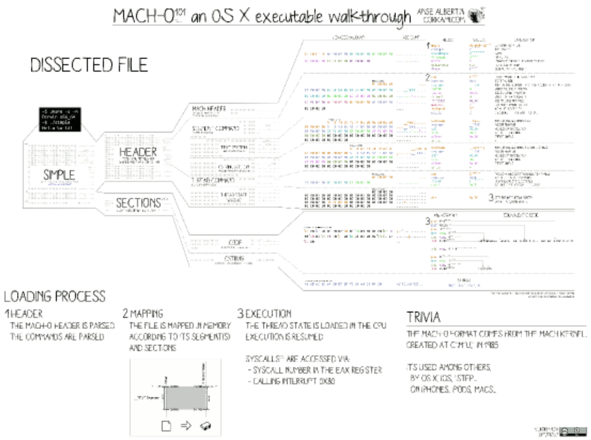
on visual docs

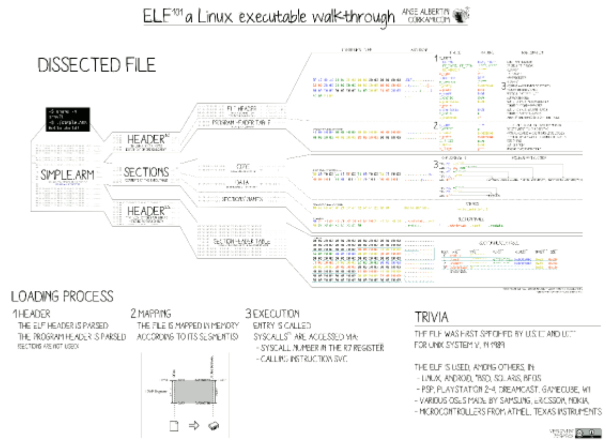
http://pics.corkami.com
http://prints.corkami.com

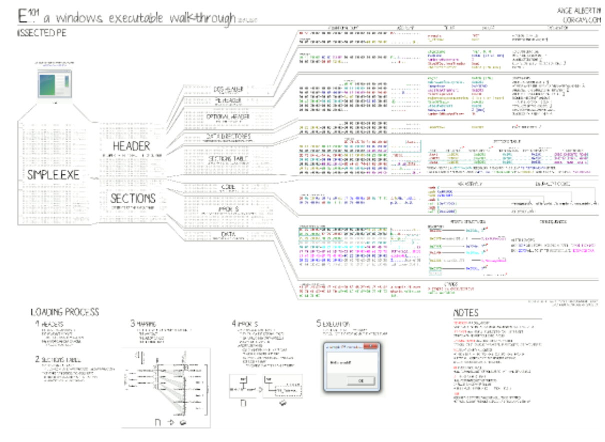
disclaimer

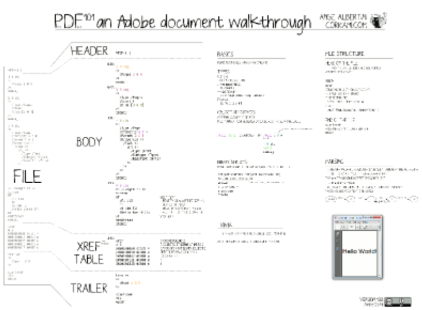
no awards, no studies

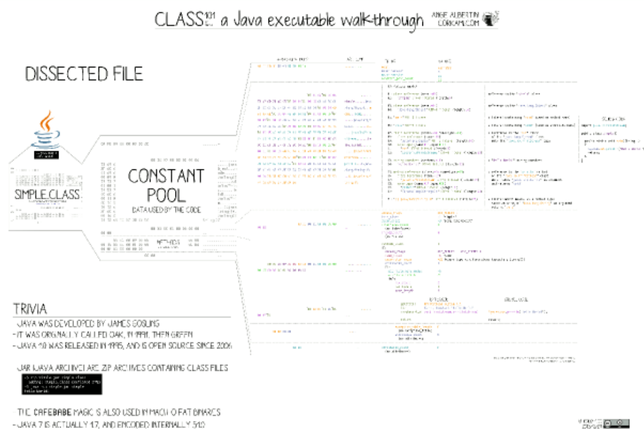

~$ unzip simple.zip
Archive: simple.zip
extracting: hello.txt
~$ cat hello.txt
Hello World!
SIMPLE.ZIP
LOCAL FILE HEADER
ARCHIVED FILE INFORMATION
FILE DATA
ARCHIVED FILE CONTENT
CENTRAL DIRECTORY
LIST OF FILE HEADERS IN THE ARCHIVE
FILE NAME
END OF CENTRAL DIRECTORY
HEXADECIMAL DUMP
ASCII DUMP
DESCRIPTION
VALUE
50 4B 03 04 0A 00 00 00 00 00 00 00 00 00 DD DD
14 7D 0D 00 00 00 0D 00 00 00 00 00 00 00
PK..............
.}..........
3
local file header signature PK\x03\x04
version needed to extract 10
compression method 0
crc-32 0x7d14dddd
compressed size 0xd
uncompressed size 0xd
Offset:0x1e
48 65
6C 6C 6F 20 57 6F 72 6C 64 21 0A
He
llo.World!.
file data "Hello World!\n"
Offset:0x2b
50 4B 01 02 00
00 0A 00 00 00 00 00 00 00 00 00 DD DD 14 7D 0D
00 00 00 0D 00 00 00 09 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00
PK...
..............}.
....
....
2
central file header signature PK\x01\x02
version needed to extract 10
crc-32 0x7d14dddd
compressed size 0xd
uncompressed size 0xd
file name length 9
relative offset of local header 0
Offset:0x59
68 65 6C 6C 6F 2E 74
78 74
hello.t
xt
file name hello.txt
Offset:0x62
50 4B 05 06 00 00 00 00 00 00 01 00 37 00
00 00 2B 00 00 00 00 00
PK..........7.
..+.....
1
end of central dir signature PK\x05\x06
total number of entries in the central directory 1
size of the central directory 0x37
offset of start of central directory with respect to the starting disk number 0x2b
THIS IS THE WHOLE FILE, HOWEVER, MOST ZIP FILES CONTAIN MANY MORE ELEMENTS.
EXPLANATIONS ARE SIMPLIFIED, FOR CONCISENESS.
ZIP ARCHIVE
FILE 1 Local file header 1
<compressed data>
...
FILE N Local file header n
<compressed data>
Central directory:
File header 1
<file name>
...
File header n
<file name>
START HERE End of central directory record
RELATIVE OFFSET OF LOCAL HEADER 1
RELATIVE OFFSET OF LOCAL HEADER N
OFFSET OF START OF CENTRAL DIRECTORY

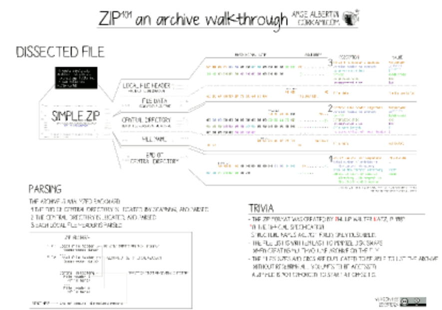

MACH-O101 an OS X executable walk-through
ANGE ALBERTINI CORKAMI.COM
DISSECTED FILE
SIMPLE
HEADER
SECTIONS
LOADING PROCESS
1 HEADER
THE MACH-O HEADER IS PARSED
THE COMMANDS ARE PARSED
2 MAPPING
THE FILE IS MAPPED IN MEMORY ACCORDING TO ITS SEGMENT(S) AND SECTIONS
3 EXECUTION
THE THREAD STATE IS LOADED IN THE CPU
EXECUTION IS RESUMED
SYSCALLS ARE ACCESSED VIA:
- SYSCALL NUMBER IN THE EAX REGISTER
- CALLING INTERRUPT 0X80
TRIVIA
THE MACH-O FORMAT COMES FROM THE MACH KERNEL CREATED AT C.M.U. IN 1985
ITS USED AMONG OTHERS, BY OS X, IOS, NSTEP...
ON IPHONES, IPODS, MACS...

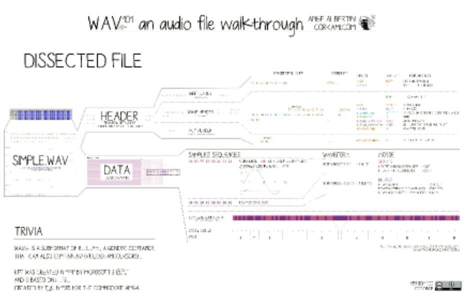

ELF101 a Linux executable walk-through
ANGE ALBERTINI CORKAMI.COM
DISSECTED FILE
SIMPLE.ARM
HEADER
SECTIONS
LOADING PROCESS
1 HEADER
THE ELF HEADER IS PARSED
THE PROGRAM HEADER IS PARSED
SECTIONS ARE NOT USED
2 MAPPING
THE FILE IS MAPPED IN MEMORY ACCORDING TO ITS SEGMENT(S)
3 EXECUTION
ENTRY IS CALLED
SYSCALLS ARE ACCESSED VIA:
- SYSCALL NUMBER IN THE R7 REGISTER
- CALLING INSTRUCTION SVC
TRIVIA
THE ELF WAS FIRST SPECIFIED BY U.S.L. AND U.I. FOR UNIX SYSTEM V, IN 1989
THE ELF IS USED, AMONG OTHERS, IN:
- LINUX, ANDROID, *BSD, SOLARIS, BEOS
- PSP, PLAYSTATION 2-4, DREAMCAST, GAMECUBE, WII
- VARIOUS OSES MADE BY SAMSUNG, ERICSSON, NOKIA,
- MICROCONTROLLERS FROM ATMEL, TEXAS INSTRUMENTS

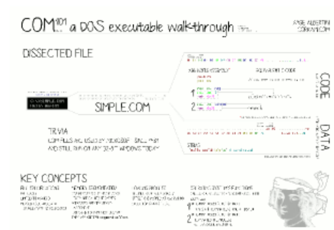

PE101 a windows executable walk-through
ANGE ALBERTINI CORKAMI.COM
DISSECTED PE
SIMPLE.EXE
HEADER
SECTIONS
LOADING PROCESS
NOTES

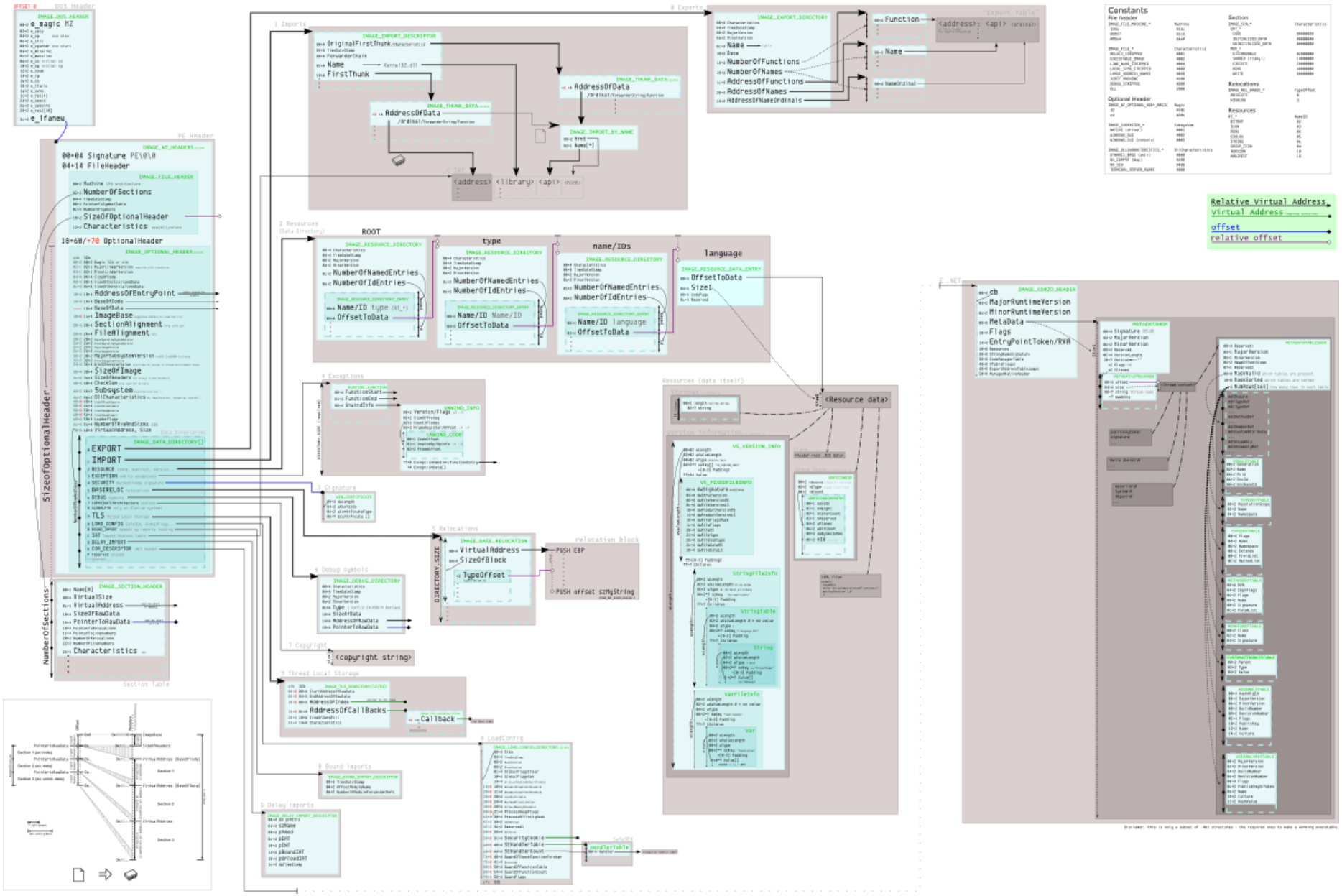

PDF101 an Adobe document walk-through
ANGE ALBERTINI CORKAMI.COM
HEADER
BODY
FILE
XREF TABLE
TRAILER
Hello World!

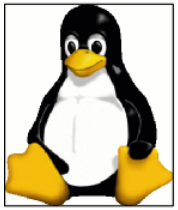

CLASS101 a Java executable walk-through
ANGE ALBERTINI CORKAMI.COM
DISSECTED FILE
SIMPLE.CLASS
CONSTANT POOL
DATA USED BY THE CODE
TRIVIA
JAVA WAS DEVELOPED BY JAMES GOSLING
- IT WAS ORIGINALLY CALLED OAK, IN 1991, THEN GREEN
- JAVA 1.0 WAS RELEASED IN 1995, AND IS OPEN SOURCE SINCE 2006
.JAR (JAVA ARCHIVE) ARE ZIP ARCHIVES CONTAINING CLASS FILES
THE CAFEBABE MAGIC IS ALSO USED IN MACH-O FAT BINARIES
- JAVA 7 IS ACTUALLY 1.7, AND ENCODED INTERNALLY 51.0


ZIP101 an archive walk-through
ANGE ALBERTINI CORKAMI.COM
DISSECTED FILE
SIMPLE.ZIP
PARSING
TRIVIA

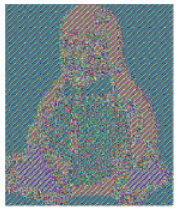

WAV101 an audio file walk-through
ANGE ALBERTINI CORKAMI.COM
DISSECTED FILE
SIMPLE.WAV
HEADER
DATA
TRIVIA

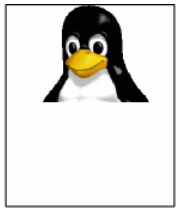

COM101 a DOS executable walk-through
ANGE ALBERTINI CORKAMI.COM
DISSECTED FILE
SIMPLE.COM
CODE
DATA
KEY CONCEPTS

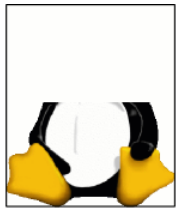

DOS Header
PE Header
SizeOfOptionalHeader
NumberOfSections
1 Imports
8 Exports
2 Resources (Data Directory)
ROOT
type
name/IDs
language
<Resource data>
4 Exceptions
5 Relocations
relocation block
6 Debug symbols
7 Copyright
<copyright string>
9 Thread Local Storage
AddressOfCallBacks
Callback
8 Bound Imports
10 LoadConfig
11 Delay Imports
Section Table
Constants
Relative Virtual Address
Virtual Address
offset
relative offset



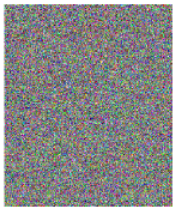
ECB MODE

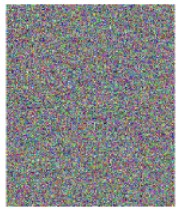




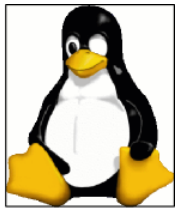


CTR MODE
SAME KEY
SAME NONCE

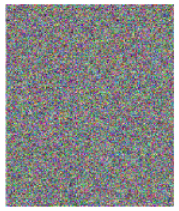


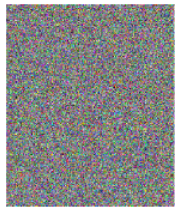


XOR

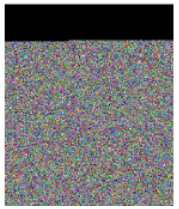


CBC MODE
SAME IV
SAME KEY





XOR



IDENTICAL UNTIL
FIRST DIFFERENCE

goal

create useful documentations

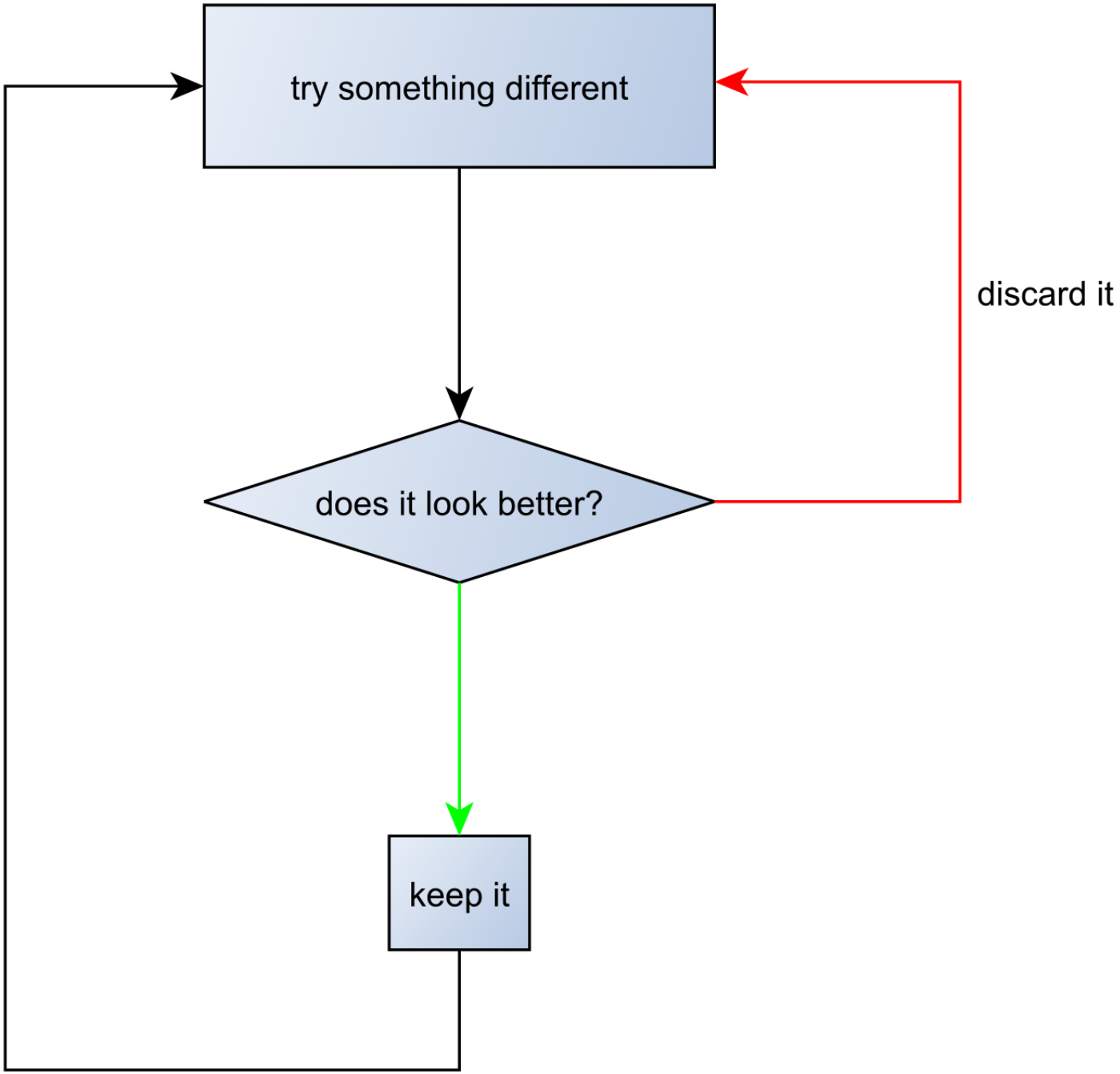
based on **reality**

posters
self-contained

- immediate ‘big picture’
- no roleplay gamebook

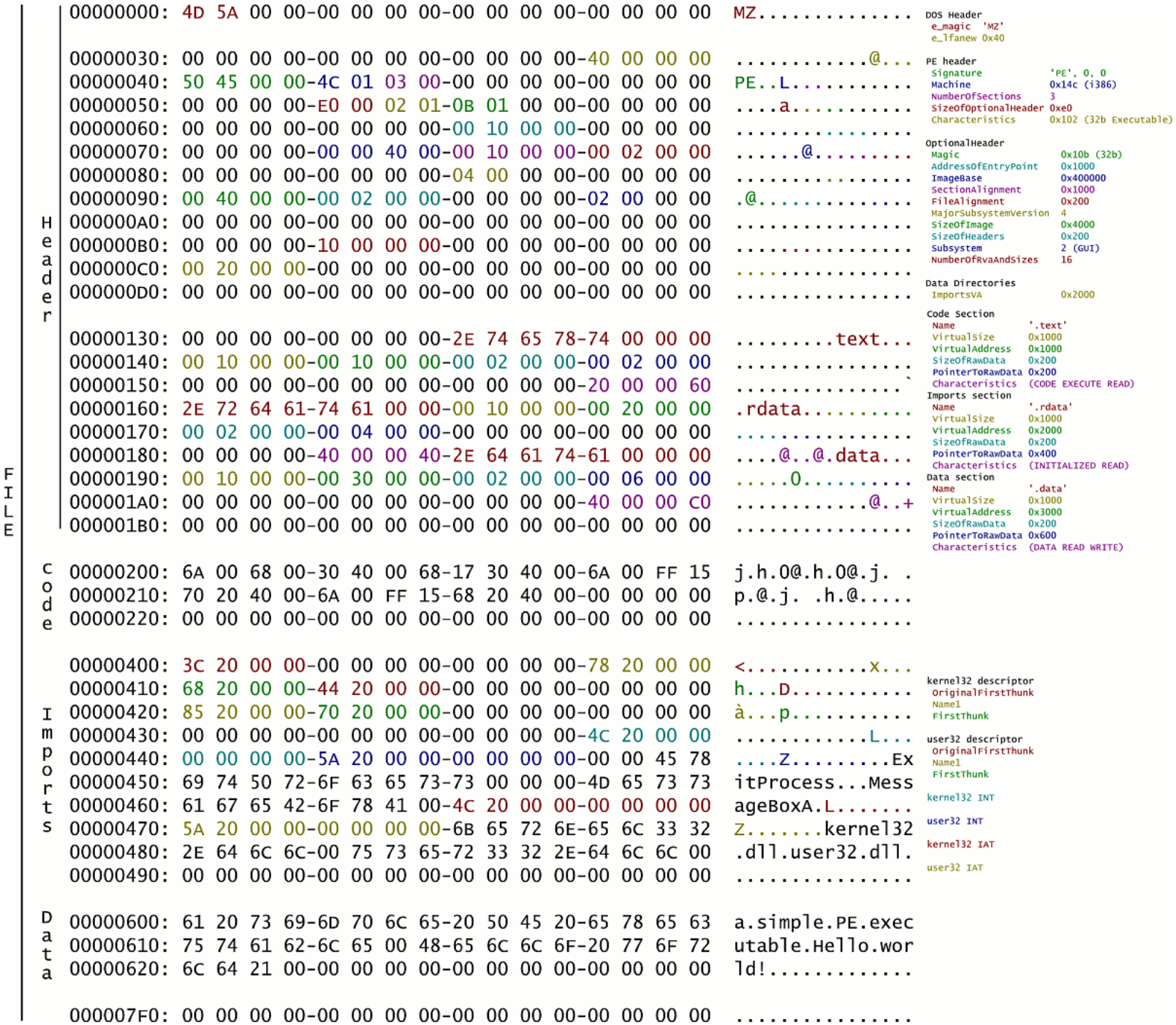
use common sense

and your own eyes

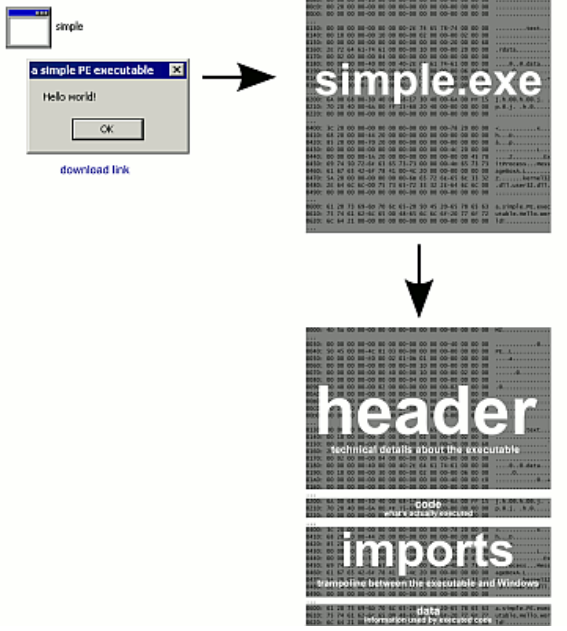

try something different
does it look better?
discard it
keep it

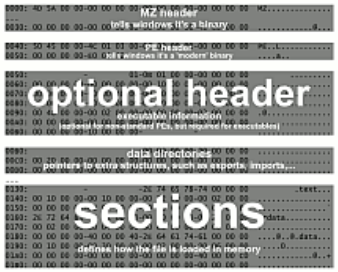
creativity?
give yourself time!

to fail!

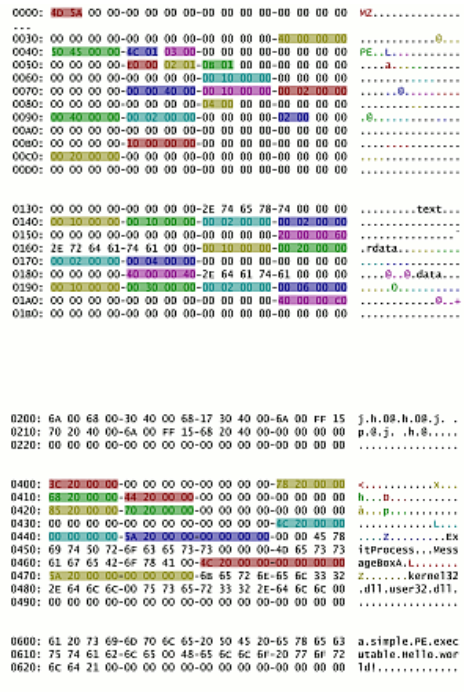

FILE
Header
code
Imports
Data
00000000: 4D 5A 00 00-00 00 00 00-00 00 00 00-00 00 00 00 MZ..............
00000030: 00 00 00 00-00 00 00 00-00 00 00 00-40 00 00 00 ....@...
00000040: 50 45 00 00-4C 01 03 00-00 00 00 00-00 00 00 00 PE..L...........
00000050: 00 00 00 00-E0 00 02 01-0B 01 00 00-00 00 00 00 ....a...........
00000060: 00 00 00 00-00 00 00 00-00 10 00 00-00 00 00 00 ....
00000070: 00 00 00 00-00 00 40 00-00 10 00 00-00 02 00 00 ......@.........
00000080: 00 00 00 00-00 00 00 00-04 00 00 00-00 00 00 00 ....
00000090: 00 40 00 00-00 02 00 00-00 00 00 00-02 00 00 00 .@..............
000000A0: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00 ....
000000B0: 00 00 00 00-10 00 00 00-00 00 00 00-00 00 00 00 ....
000000C0: 00 20 00 00-00 00 00 00-00 00 00 00-00 00 00 00 ....
000000D0: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00 ....
00000130: 00 00 00 00-00 00 00 00-2E 74 65 78-74 00 00 00 ....text...
00000140: 00 10 00 00-00 10 00 00-00 02 00 00-00 02 00 00 ....
00000150: 00 00 00 00-00 00 00 00-00 00 00 00-20 00 00 60 ....`
00000160: 2E 72 64 61-74 61 00 00-00 10 00 00-00 20 00 00 .rdata..........
00000170: 00 02 00 00-00 04 00 00-00 00 00 00-00 00 00 00 ....
00000180: 00 00 00 00-40 00 00 40-2E 64 61 74-61 00 00 00 ....@..@.data...
00000190: 00 10 00 00-00 30 00 00-00 02 00 00-00 06 00 00 .....0..........
000001A0: 00 00 00 00-00 00 00 00-00 00 00 00-40 00 00 C0 ....@..+
000001B0: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00 ....
00000200: 6A 00 68 00-30 40 00 68-17 30 40 00-6A 00 FF 15 j.h.0@.h.0@.j. .
00000210: 70 20 40 00-6A 00 FF 15-68 20 40 00-00 00 00 00 p.@.j. .h.@.....
00000220: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00 ....
00000400: 3C 20 00 00-00 00 00 00-00 00 00 00-78 20 00 00 <...........x...
00000410: 68 20 00 00-44 20 00 00-00 00 00 00-00 00 00 00 h...D...........
00000420: 85 20 00 00-70 20 00 00-00 00 00 00-00 00 00 00 à...p...........
00000430: 00 00 00 00-00 00 00 00-00 00 00 00-4C 20 00 00 ....L...
00000440: 00 00 00 00-5A 20 00 00-00 00 00 00-00 00 45 78 ....Z.........Ex
00000450: 69 74 50 72-6F 63 65 73-73 00 00 00-4D 65 73 73 itProcess...Mess
00000460: 61 67 65 42-6F 78 41 00-4C 20 00 00-00 00 00 00 ageBoxA.L.......
00000470: 5A 20 00 00-00 00 00 00-6B 65 72 6E-65 6C 33 32 Z.......kernel32
00000480: 2E 64 6C 6C-00 75 73 65-72 33 32 2E-64 6C 6C 00 .dll.user32.dll.
00000490: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00 ....
00000600: 61 20 73 69-6D 70 6C 65-20 50 45 20-65 78 65 63 a.simple.PE.exec
00000610: 75 74 61 62-6C 65 00 48-65 6C 6C 6F-20 77 6F 72 utable.Hello.wor
00000620: 6C 64 21 00-00 00 00 00-00 00 00 00-00 00 00 00 ld!.............
000007F0: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00 ....
DOS Header
e_magic 'MZ'
e_lfanew 0x40
PE header
Signature 'PE', 0, 0
Machine 0x14c (i386)
NumberOfSections 3
SizeOfOptionalHeader 0xe0
Characteristics 0x102 (32b Executable)
OptionalHeader
Magic 0x10b (32b)
AddressOfEntryPoint 0x1000
ImageBase 0x400000
SectionAlignment 0x1000
FileAlignment 0x200
MajorSubsystemVersion 4
SizeOfImage 0x4000
SizeOfHeaders 0x200
Subsystem 2 (GUI)
NumberOfRvaAndSizes 16
Data Directories
ImportsVA 0x2000
Code Section
Name '.text'
VirtualSize 0x1000
VirtualAddress 0x1000
SizeOfRawData 0x200
PointerToRawData 0x200
Characteristics (CODE EXECUTE READ)
Imports section
Name '.rdata'
VirtualSize 0x1000
VirtualAddress 0x2000
SizeOfRawData 0x200
PointerToRawData 0x400
Characteristics (INITIALIZED READ)
Data section
Name '.data'
VirtualSize 0x1000
VirtualAddress 0x3000
SizeOfRawData 0x200
PointerToRawData 0x600
Characteristics (DATA READ WRITE)
kernel32 descriptor
OriginalFirstThunk
Name1
FirstThunk
user32 descriptor
OriginalFirstThunk
Name1
FirstThunk
kernel32 INT
user32 INT
kernel32 IAT
user32 IAT

a simple PE walkthrough






```
Header
Code
 MessageBoxA(0, "a simple executable", "Hello world!", 0)
 ExitProcess(0)
Imports
 Executable            Libraries
   Imports -->       <-- Exports

 Simple.exe            user32.dll
  call xxx -->     <-- MessageBoxA

Data
 "a simple executable"
 "Hello world!"
```

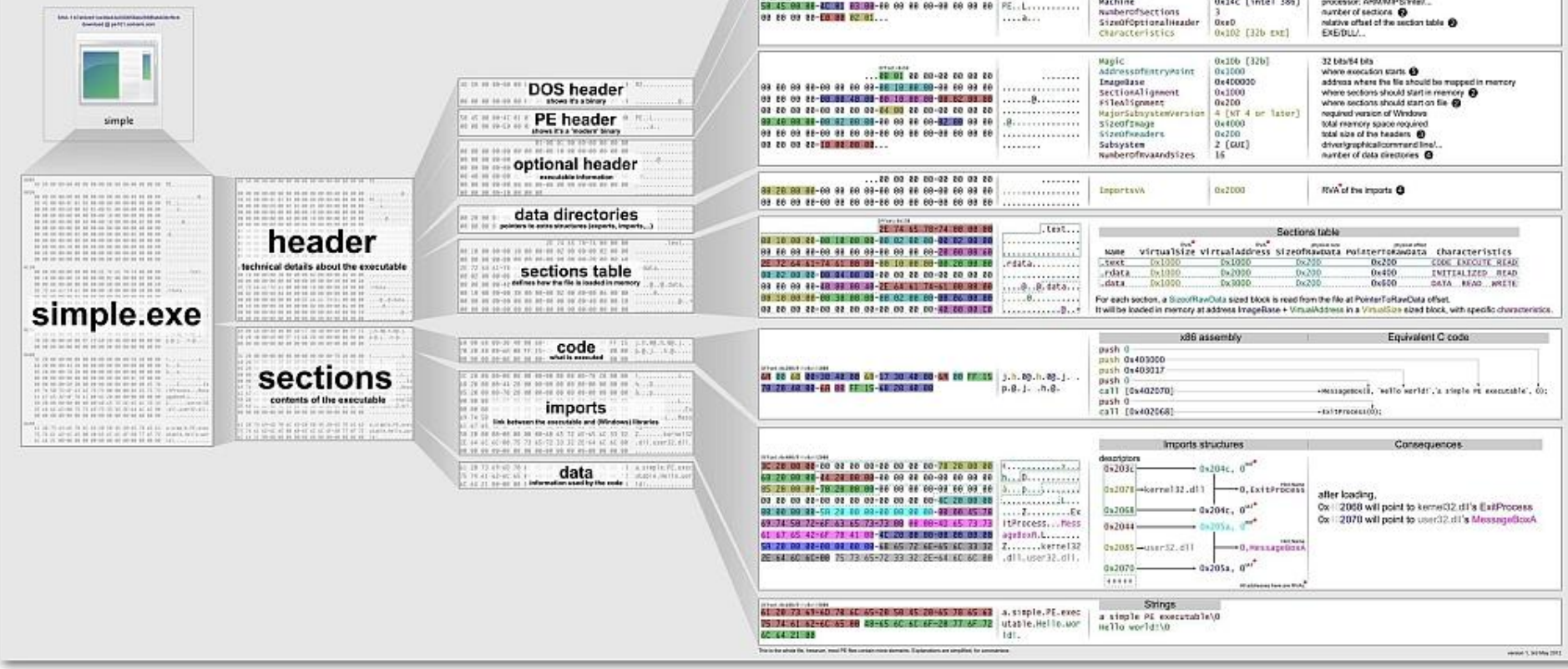



```
DOS Header
  e_magic   'MZ'
  e_lfanew  0x40

PE header
  Signature                'PE', 0, 0
  Machine                  0x14c (I386)
  NumberOfSections         3
  SizeofOptionalHeader     0xe0
  Characteristics          0x102 (32b Executable)

OptionalHeader
  Magic                    0x10b (32b)
  AddressofEntryPoint      0x1000
  ImageBase                0x400000
  SectionAlignment         0x1000
  FileAlignment            0x200
  MajorSubsystemVersion    4
  SizeOfImage              0x4000
  SizeOfHeaders            0x200
  Subsystem                2 (GUI)
  NumberOfRvaAndSizes      16
Code section '.text'
  VirtualSize              0x1000
  VirtualAddress           0x1000
  SizeOfRawData            0x200
  PointerToRawData         0x200
  Characteristics          (CODE EXECUTE READ)
Imports section '.rdata'
Data Directories
  VirtualAddress           0x2000
  SizeOfRawData            0x200
  PointerToRawData         0x400
  Characteristics          (INITIALIZED READ)
Data section '.data'
  VirtualSize              0x1000
  VirtualAddress           0x3000
  SizeOfRawData            0x200
  PointerToRawData         0x600
  Characteristics          (DATA READ WRITE)

kernel32 descriptor
  OriginalFirstThunk 0x203c
  Name1              0x2078
  FirstThunk         0x2068

user32 descriptor
  OriginalFirstThunk 0x2044
  Name1              0x2085
  FirstThunk         0x2070

kernel32 INT  0x204c, 0
user32 INT    0x205a, 0

kernel32 IAT  0x204c, 0
user32 IAT    0x205a, 0
```

PE101 a windows executable walkthrough

Ange Albertini
corkami.com

Dissected PE



Loading process

❶ Headers

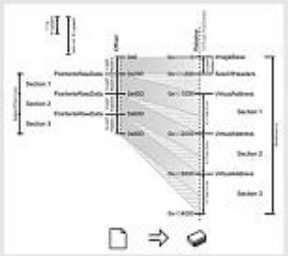
the *DOS Header* is parsed
the *PE Header* is parsed
(its offset is *DOS Header*'s e_lfanew)
the *Optional Header* is parsed
(it follows the *PE Header*)

❷ Sections table

Sections table is parsed
(it is located at: offset(*OptionalHeader*) + *SizeOfOptionalHeader*)
it contains *NumberOfSections* elements
it is checked for validity with *alignments*:
FileAlignments and *SectionAlignments*

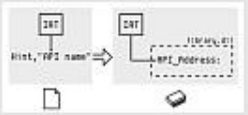
❸ Mapping

the file is mapped in memory according to:
the *ImageBase*
the *SizeOfHeaders*
the Sections table



❹ Imports

DataDirectories are parsed
they follow the *OptionalHeader*
their number is *NumOfRVAAndSizes*
imports are always #2
imports are parsed
each descriptor specifies a *DLLname*
this DLL is loaded in memory
IAT and *INT* are parsed simultaneously
for each API in *INT*
its address is written in the *IAT* entry



❺ Execution

Code is called at the *EntryPoint*
the calls of the code go via the IAT to the APIs

Notes

MZ HEADER aka DOS_HEADER
Starts with 'MZ' (initials of *Mark Zbikowski* MS-DOS developer)

PE HEADER aka IMAGE_FILE_HEADERS / COFF file header
Starts with 'PE' (Portable Executable)

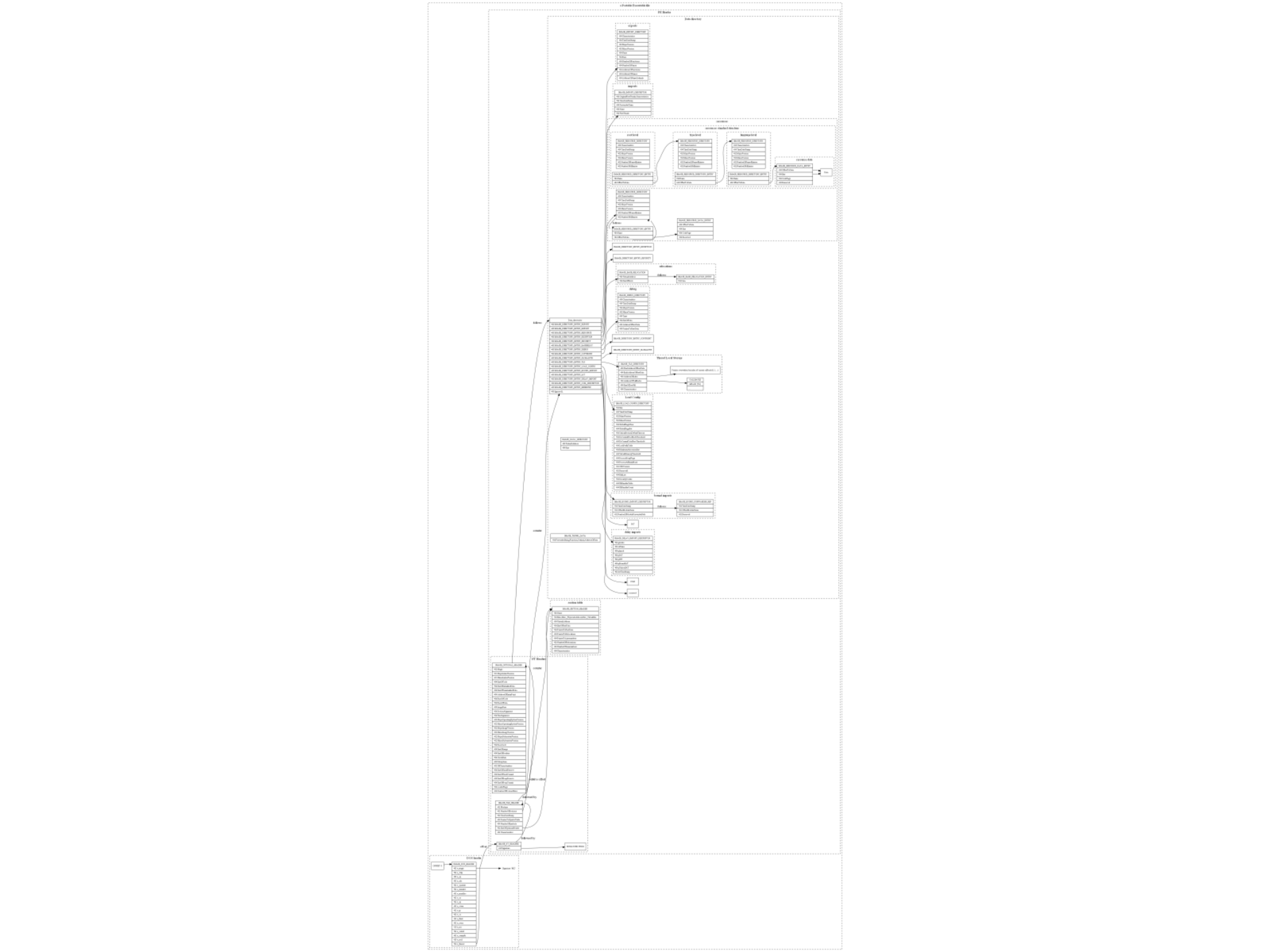
OPTIONAL HEADER aka IMAGE_OPTIONAL_HEADER
Optional only for non-standard PEs but required for executables

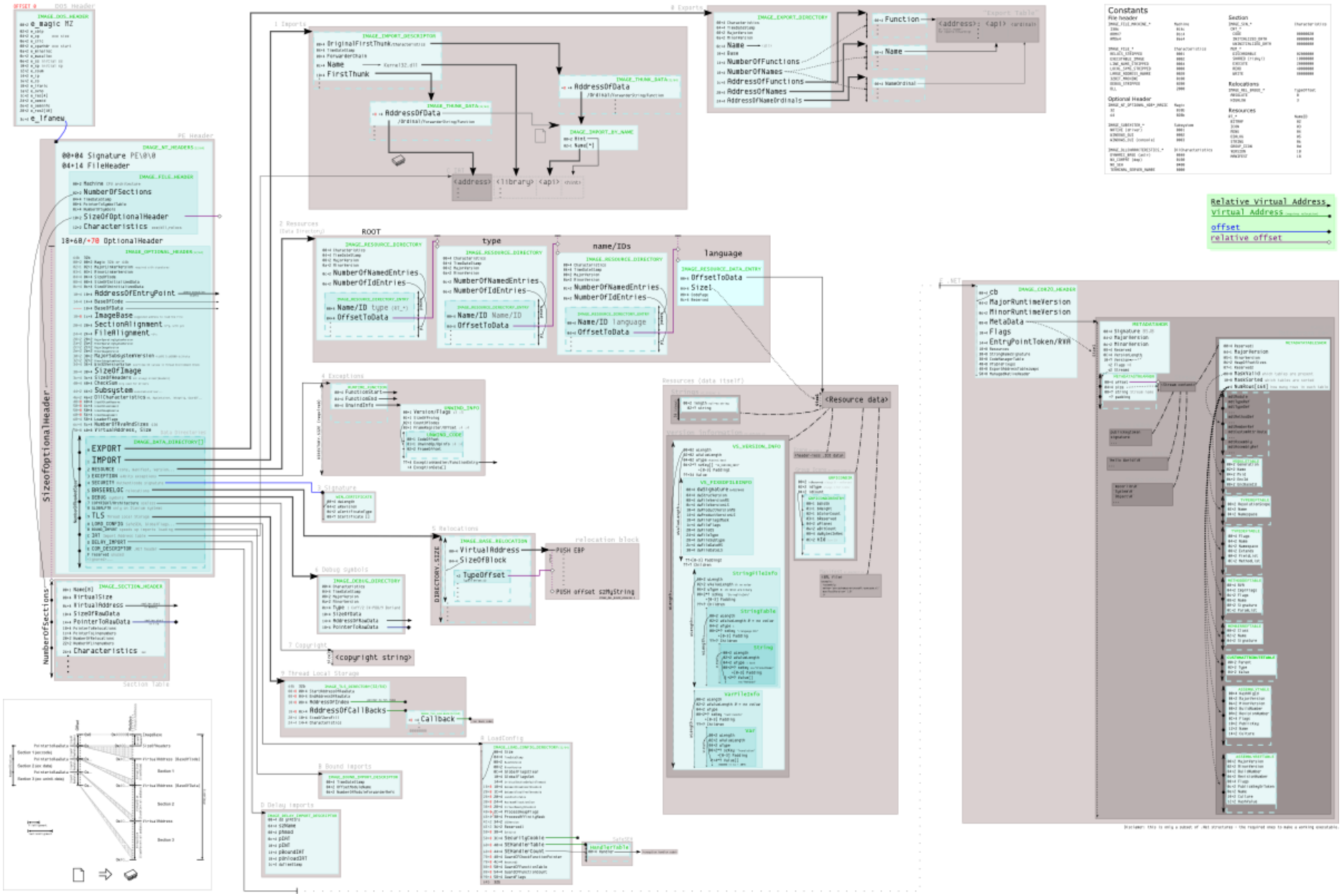
RVA Relative Virtual Address
Address relative to ImageBase (at ImageBase, RVA = 0)
Almost all addresses of the headers are RVAs
In code, addresses are *not* relative.

INT Import Name Table
Null-terminated list of pointers to Hint, Name structures

IAT Import Address Table
Null-terminated list of pointers
On file it is a copy of the INT
After loading it points to the imported APIs

HINT
Index in the exports table of a DLL to be imported
Not required but provides a speed-up by reducing look-up


x-Portable Executable file
PE Header
Data directory
signature
upgrade
security
security or attached data blobs
Import Table
Import Address Table
Thread Local Storage
Load Config
Delay Import
debug
Exception
Base Relocation Table
IAT Header
CLR Header
Export Table
EXIST
Import NT
Export NT
Section Table
.text
.rsrc
.reloc
follows
contains
follows by
offset


DOS Header
PE Header
SizeOfOptionalHeader
NumberOfSections
1 Imports
8 Exports
2 Resources (Data Directory)
ROOT
type
name/IDs
language
<Resource data>
4 Exceptions
5 Relocations
relocation block
6 Debug symbols
7 Copyright
<copyright string>
9 Thread Local Storage
AddressOfCallBacks
Callback
8 Bound Imports
10 LoadConfig
11 Delay Imports
Section Table
Constants
Relative Virtual Address
Virtual Address
offset
relative offset

define your audience

lower **and** upper limits

“you should add …”

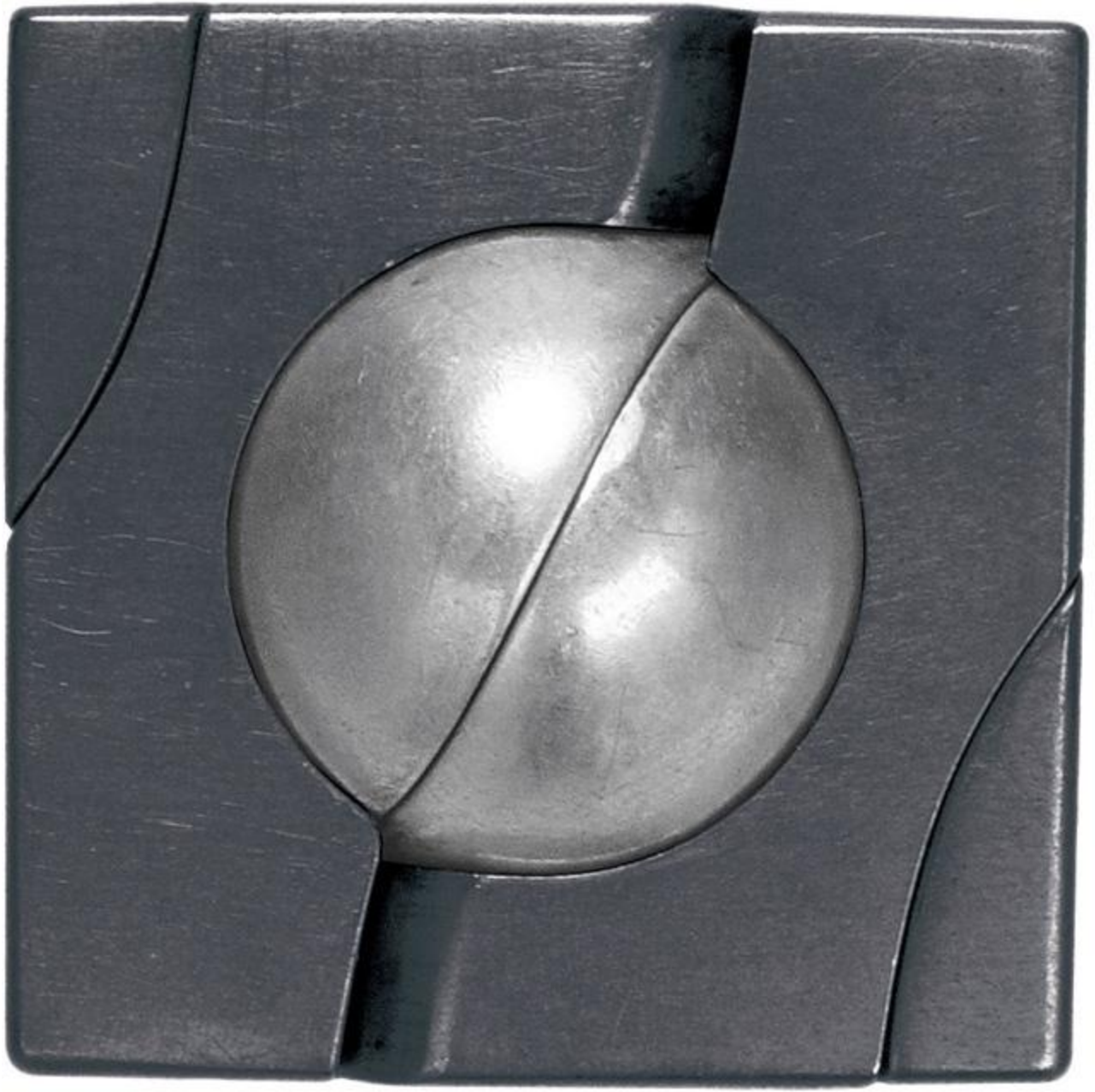
see “setting a upper limit”

“too simple/simplified”?

1/ teach others, beginners, kids
2/ no more excuses for not knowing

remove the obvious

guessing doesn’t hurt

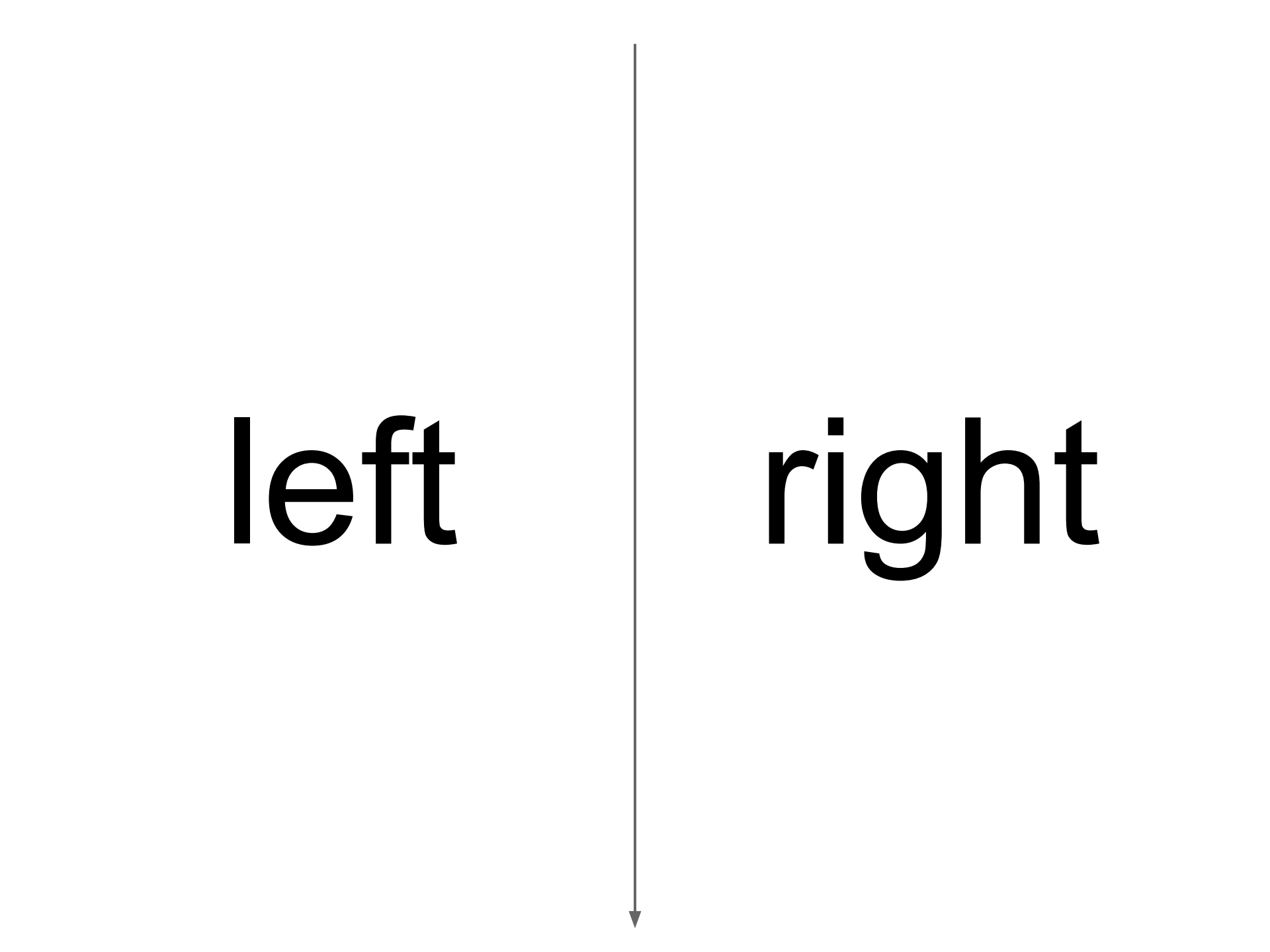


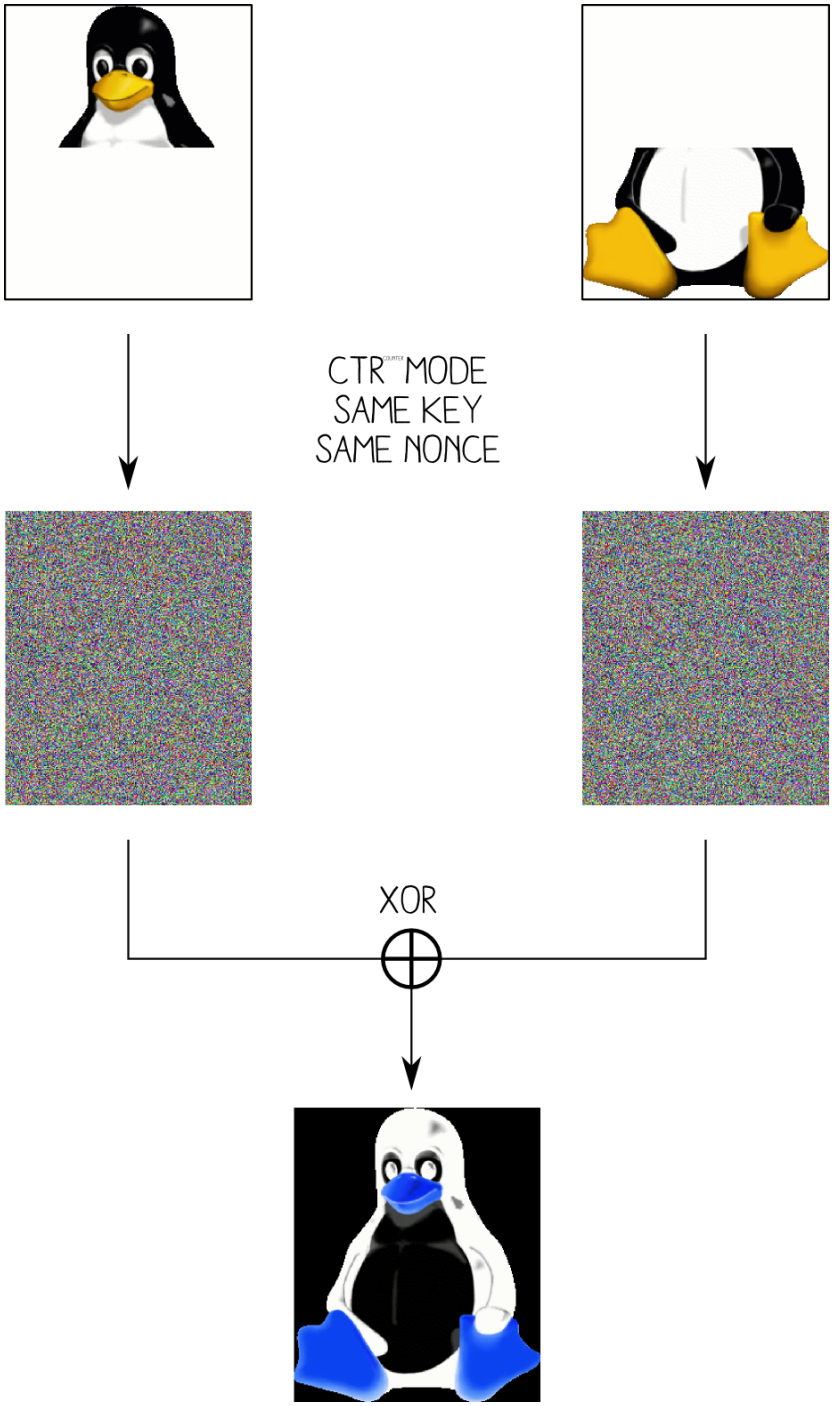
space
optimal separator

left right

left

right


left
right


CTR MODE
SAME KEY
SAME NONCE
XOR

THE ADOBE LOGO, ENCRYPTED WITH 3DES IN ECB MODE
(THE SAME ALGORITHM THEY USE TO STORE PASSWORDS)

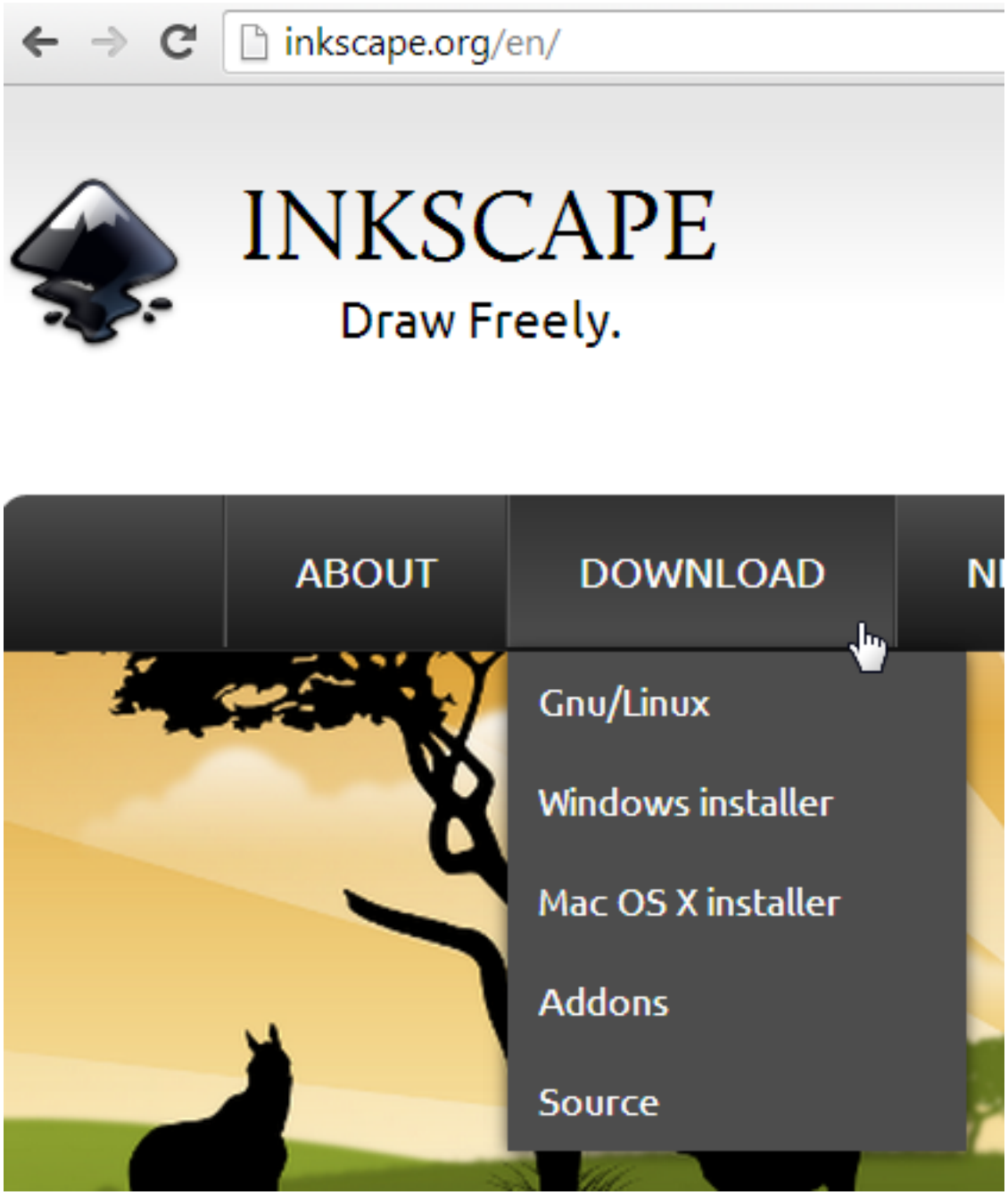
no unnecessary extras

leave doors closed
(to be opened somewhere else)

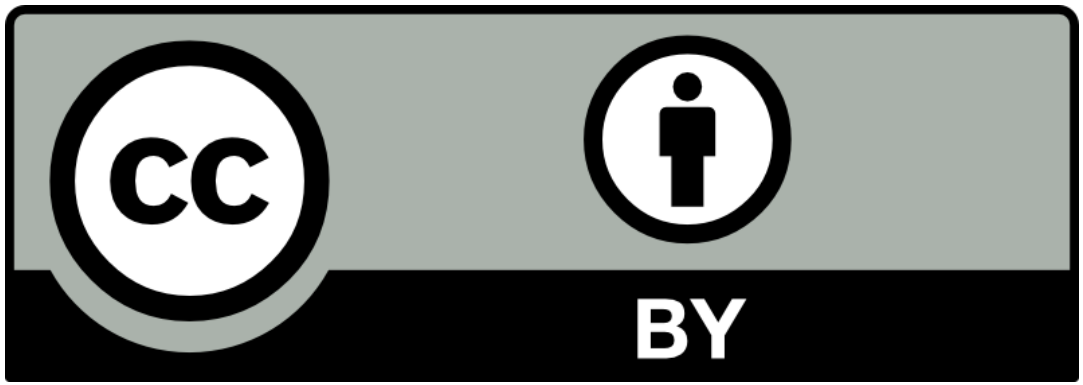
requirements

a computer

a transparent tablecloth


inkscape.org/en/
INKSCAPE
Draw Freely.
ABOUT
DOWNLOAD
Gnu/Linux
Windows installer
Mac OS X installer
Addons
Source

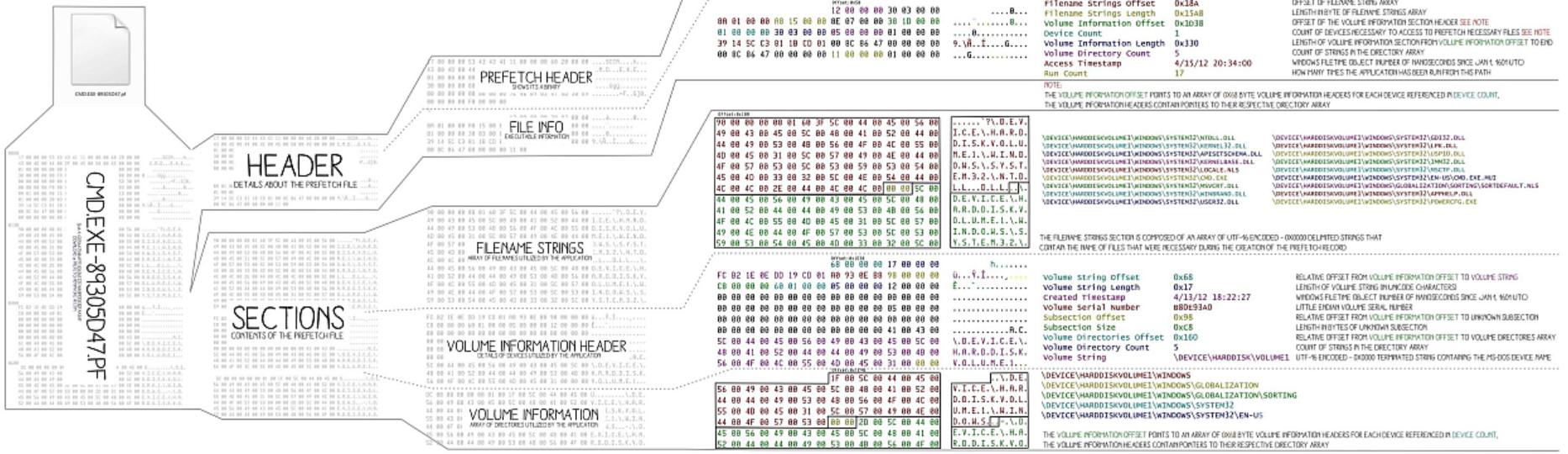
http://src.corkami.com



PREFETCH[101]: a Windows 7 Prefetch Walk-through

JARED ATKINSON
INVOKE-IR.COM

DISSECTED FILE



PREFETCHING PROCESS

THE PREFETCHER TRIES TO SPEED THE BOOT PROCESS AND APPLICATION STARTUP BY MONITORING THE DATA AND CODE ACCESSED BY BOOT AND APPLICATION STARTUPS AND USING THAT INFORMATION AT THE BEGINNING OF A SUBSEQUENT BOOT OR APPLICATION STARTUP TO READ IN THE CODE AND DATA.
-WINDOWS INTERNALS, PART 2: COVERING WINDOWS 2008 SERVER R2 AND WINDOWS 7

PREFETCH IN ACTION:

1 USER LAUNCHES PROGRAM

2 CACHE MANAGER LOOKS FOR PREFETCH FILE IN C:\WINDOWS\PREFETCH THAT CORRESPONDS WITH THE LAUNCHED APPLICATION. "NO PREFETCH FILE EXISTS"

3 THE CACHE MANAGER MONITORS THE PROCESS FOR ITS FIRST TEN SECONDS. DURING THIS TIME IT RECORDS FAULTS, AND USES THE INFORMATION GAINED TO CREATE A PREFETCH FILE FOR THE APPLICATION.

4 CACHE MANAGER CREATES PREFETCH FILE

5 PROCESS HAS STARTED

6 PROFIT

	BOOT PREFETCHING	APPLICATION PREFETCHING	HOSTING APPLICATION PREFETCHING
TIMEFRAME	WHICHEVER ELAPSES FIRST: 1) 30 SECONDS AFTER THE USER'S SHELL HAS STARTED 2) UNTIL 60 SECONDS AFTER ALL SERVICES HAVE FINISHED INITIALIZING. 3) UNTIL 120 SECONDS AFTER THE SYSTEM HAS BOOTED	FIRST 10 SECONDS AFTER AN APPLICATION LAUNCHES	FIRST 10 SECONDS AFTER AN APPLICATION LAUNCHES
FILENAME	NTOSBOOT-BOODFAAD.PF NTOSBOOT = "NEW TECHNOLOGY OPERATING SYSTEM BOOT"	APPLICATION-<HASH>.PF	APPLICATION-<HOSTEDAPPHASH>.PF
HASH	ALWAYS APPEARS AS BOODFAAD	PERFORMS AN ALGORITHM ON THE MS-DOS PATH OF THE APPLICATION	PERFORMS A HASHING ALGORITHM ON THE MS-DOS PATH OF THE HOSTED ALGORITHM

NOTES

VERSION
THREE OBSERVED VALUES:
0X11 = WINDOWS XP/2003
0X17 = WINDOWS VISTA/7/2008/2008 R2
0X1A = WINDOWS 8/8.1/2012

PREFETCH DIRECTORY
PREFETCH FILES ARE STORED IN THE '%SYSTEMROOT%\PREFETCH' DIRECTORY

MAXIMUM PREFETCH FILES
THE MAXIMUM AMOUNT OF PREFETCH FILES POSSIBLE ON A SYSTEM DEPENDS ON THE OPERATING SYSTEM
OS = WINDOWS 7/2008 R2 = 128 TOTAL PREFETCH FILES POSSIBLE.
OS = WINDOWS 7/2008 = 1024 TOTAL PREFETCH FILES POSSIBLE.

THREE TYPES OF PREFETCHING
BOOT TRACE (NTOSBOOT-BOODFAAD)
APPLICATION (EX: CMD.EXE OR MINESWEEPER.EXE)
HOSTING APPLICATION (EX: DLLHOST.EXE)

SSD PREFETCHING
PREFETCHING IS DISABLED BY DEFAULT WHEN A SOLID STATE DRIVE IS BEING USED

ENABLING PREFETCH
REGISTRY KEY:
HKEY_LOCAL_MACHINE\SYSTEM\CURRENTCONTROLSET\CONTROL\SESSION MANAGER\MEMORY MANAGEMENT\PREFETCHPARAMETERS
VALUES: 0 = DISABLED, 1 = APPLICATION ONLY, 2 = BOOT ONLY, AND 3 = APPLICATION AND BOOT

@jaredcatkinson's prefetch101

conclusion
on visual docs

On visual documentations

- it doesn’t hurt
- it’s not so hard
- requires **time**

thank YOU !

Questions ?

@angealbertini

✉ **ange@corkami.com**