Public version (in Polish): http://goo.gl/iU1aT

Ten Thousand Traps

ZIP, RAR, etc.

Gynvael Coldwind
(English version of SEConference slides)

Who?

Gynvael Coldwind

http://gynvael.coldwind.pl/

All opinions expressed in this presentation are mine alone, and not those of my neighbours / accountant / employer / etc.

Srsly :)

What's on the menu?

- Freshly squeezed ZIP juice.
- RAR in gravy.

a.k.a. ZIP analysis made be me + a note on Tavis' work on RAR

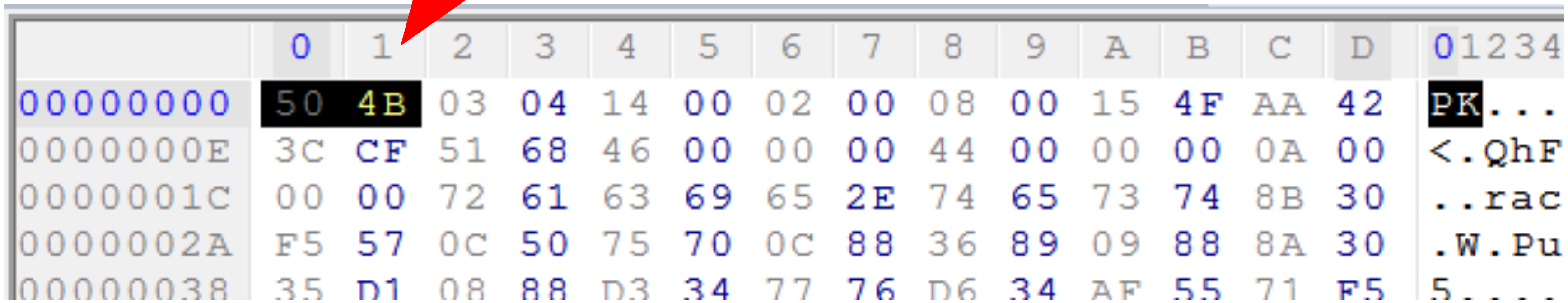
Let me show you photos I got!

Hey, it's awesome!
britney20@trustmesrsly.com sent me photos of her! Let's take a look!

DEMO

Let's start with simple stuff - the ZIP format

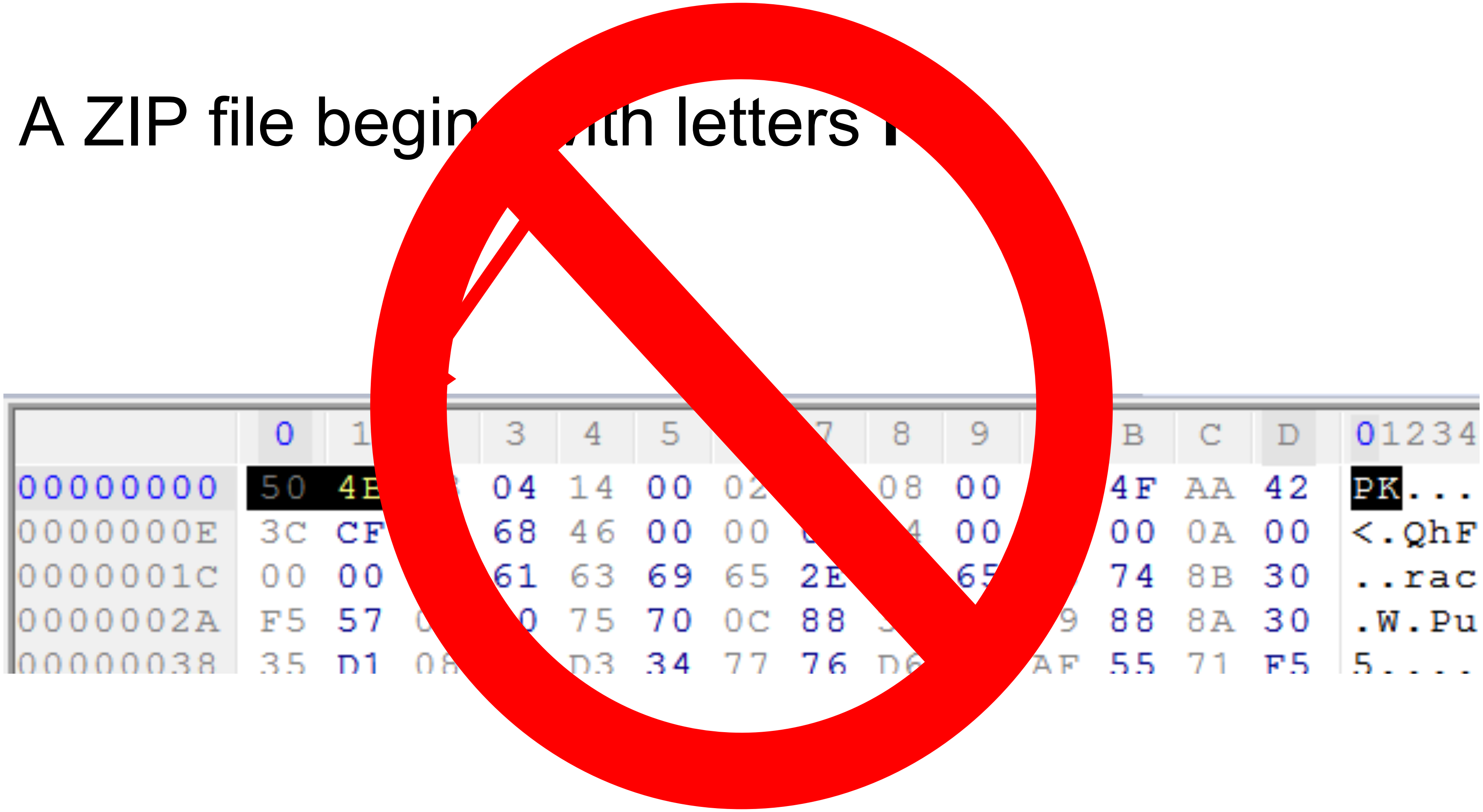
A ZIP file begins with letters **PK.**



```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D   01234
00000000  50 4B 03 04 14 00 02 00 08 00 15 4F AA 42  PK...
0000000E  3C CF 51 68 46 00 00 00 44 00 00 00 0A 00  <.QhF
0000001C  00 00 72 61 63 69 65 2E 74 65 73 74 8B 30  ..rac
0000002A  F5 57 0C 50 75 70 0C 88 36 89 09 88 8A 30  .W.Pu
00000038  35 D1 08 88 D3 34 77 76 D6 34 AF 55 71 F5  5....
```

Let's start with simple stuff - the ZIP format

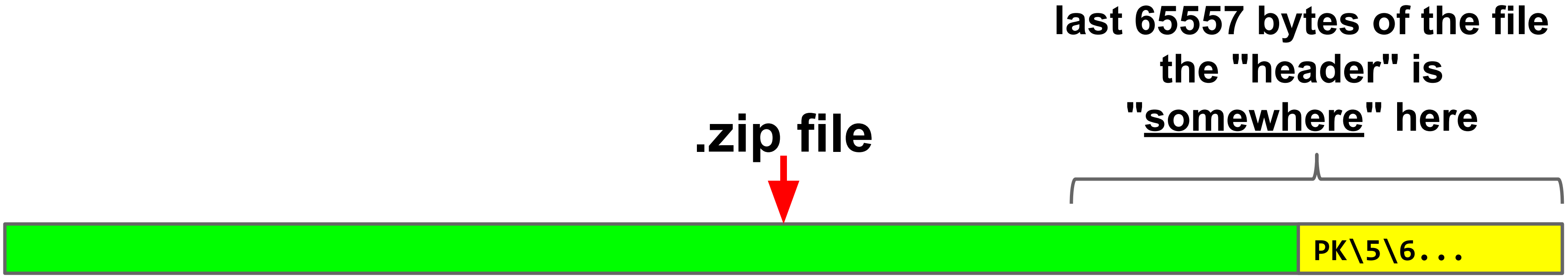
A ZIP file begin ith letters P



```
           0  1     3  4  5     7  8  9     B  C  D   01234
00000000  50 4B    04 14 00 02     08 00    4F AA 42  PK...
0000000E  3C CF    68 46 00 00        00    00 0A 00  <.QhF
0000001C  00 00    61 63 69 65 2E     65    74 8B 30  ..rac
0000002A  F5 57       75 70 0C 88           88 8A 30  .W.Pu
00000038  35 D1 08    D3 34 77 76           AF 55 71 F5  5....
```

NOPE :)

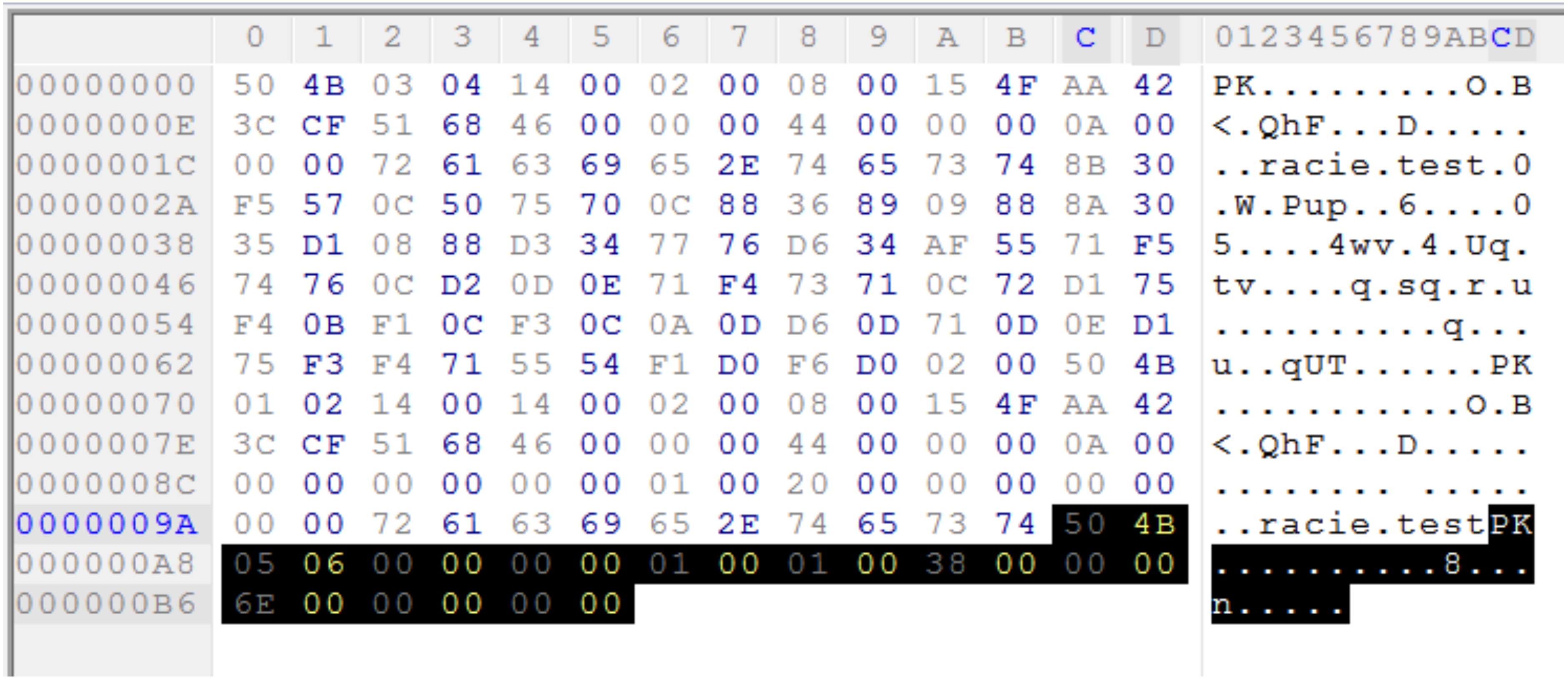
ZIP - second attempt :)



last 65557 bytes of the file the "header" is "<u>somewhere</u>" here

.zip file

PK\5\6...



```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D   0123456789ABCD
00000000  50 4B 03 04 14 00 02 00 08 00 15 4F AA 42  PK..........O.B
0000000E  3C CF 51 68 46 00 00 00 44 00 00 00 0A 00  <.QhF...D.....
0000001C  00 00 72 61 63 69 65 2E 74 65 73 74 8B 30  ..racie.test.0
0000002A  F5 57 0C 50 75 70 0C 88 36 89 09 88 8A 30  .W.Pup..6.....0
00000038  35 D1 08 88 D3 34 77 76 D6 34 AF 55 71 F5  5....4wv.4.Uq.
00000046  74 76 0C D2 0D 0E 71 F4 73 71 0C 72 D1 75  tv....q.sq.r.u
00000054  F4 0B F1 0C F3 0C 0A 0D D6 0D 71 0D 0E D1  ...........q...
00000062  75 F3 F4 71 55 54 F1 D0 F6 D0 02 00 50 4B  u..qUT......PK
00000070  01 02 14 00 14 00 02 00 08 00 15 4F AA 42  ............O.B
0000007E  3C CF 51 68 46 00 00 00 44 00 00 00 0A 00  <.QhF...D.....
0000008C  00 00 00 00 00 00 01 00 20 00 00 00 00 00  ........ .....
0000009A  00 00 72 61 63 69 65 2E 74 65 73 74 50 4B  ..racie.testPK
000000A8  05 06 00 00 00 00 01 00 01 00 38 00 00 00  ..........8...
000000B6  6E 00 00 00 00 00                          n.....
```

ZIP - "somewhere" ?!

you begin ZIP parsing from this; it MUST be at the end of the file

```
4.3.16  End of central directory record:

      end of central dir signature    4 bytes  (0x06054b50)
      number of this disk             2 bytes
      number of the disk with the
      start of the central directory  2 bytes
      total number of entries in the
      central directory on this disk  2 bytes
      total number of entries in
      the central directory           2 bytes
      size of the central directory   4 bytes
      offset of start of central
      directory with respect to
      the starting disk number        4 bytes
      .ZIP file comment length        2 bytes
      .ZIP file comment       (variable size)
```

22 bajty

$0000-$FFFF
0-65535

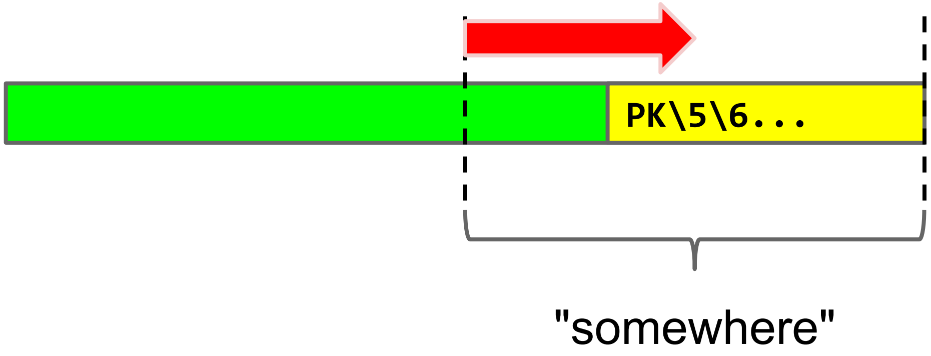
Total: from 22 to 65557 bytes

(aka: PK\5\6 magic will be somewhere between `EOF-65557` and `EOF-22`)

ZIP - looking for the "header"?

"From the START"

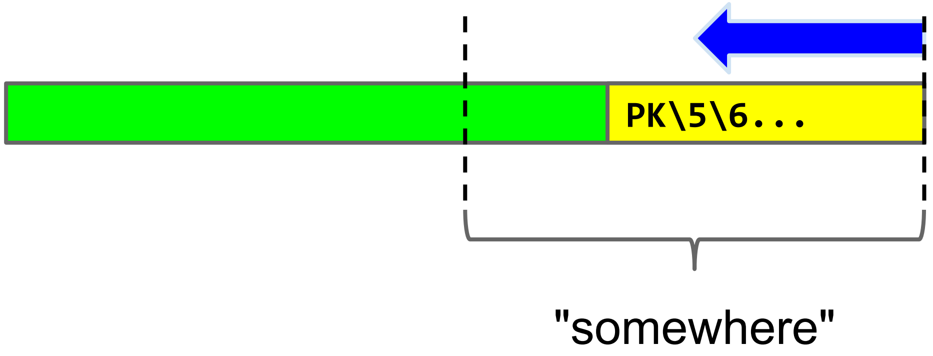
Begin at **`EOF-65557`**,
and move forward.



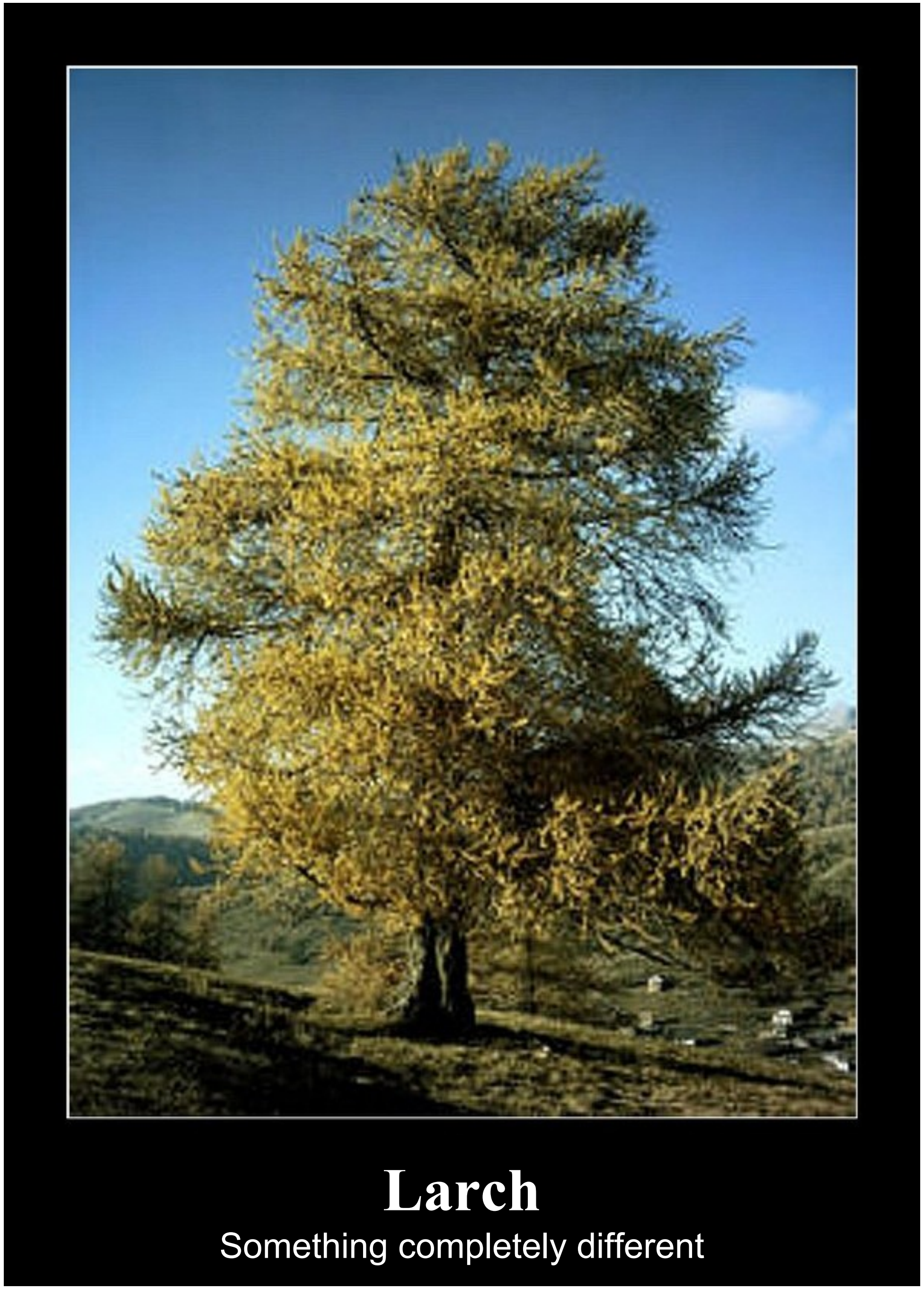
"From the END"

(ZIPs usually don't have comments)

Begin at **`EOF-22`**,
and move backward.



The show will continue in a moment.



ZIP Format - LFH

4.3.7 Local file header:

random stuff

local file header signature	4 bytes	(0x04034b50)
version needed to extract	2 bytes	
general purpose bit flag	2 bytes	
compression method	2 bytes	
last mod file time	2 bytes	
last mod file date	2 bytes	
crc-32	4 bytes	
compressed size	4 bytes	
uncompressed size	4 bytes	
file name length	2 bytes	
extra field length	2 bytes	
file name (variable size)		
extra field (variable size)		
file data (variable size)		

PK\3\4... LFH + data

Each file/directory in a ZIP has LFH + data.

ZIP Format - CDH

```
[central directory header n]

central file header signature   4 bytes  (0x02014b50)
version made by                 2 bytes
version needed to extract       2 bytes
general purpose bit flag        2 bytes
compression method              2 bytes
last mod file time              2 bytes
last mod file date              2 bytes
crc-32                          4 bytes
compressed size                 4 bytes
uncompressed size               4 bytes
file name length                2 bytes
extra field length              2 bytes
file comment length             2 bytes
disk number start               2 bytes
internal file attributes        2 bytes
external file attributes        4 bytes
relative offset of local header 4 bytes

file name (variable size)
extra field (variable size)
file comment (variable size)
```

similar stuff to LFH

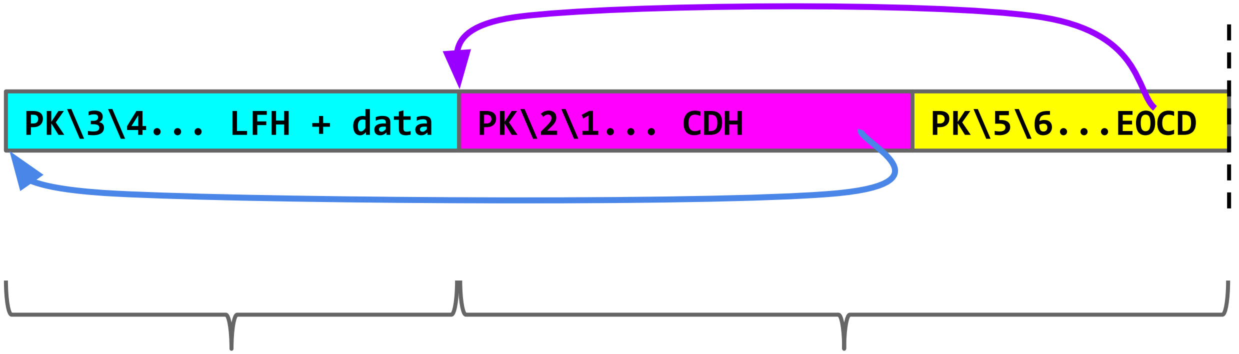
thanks to the redundancy you can recover LFH using CDH, or CDH using LFH

(xslx)

PK\2\1... CDH

Each file/directory has a CDH entry in the Central Directory

ZIP - a complete file



PK\3\4... LFH + data

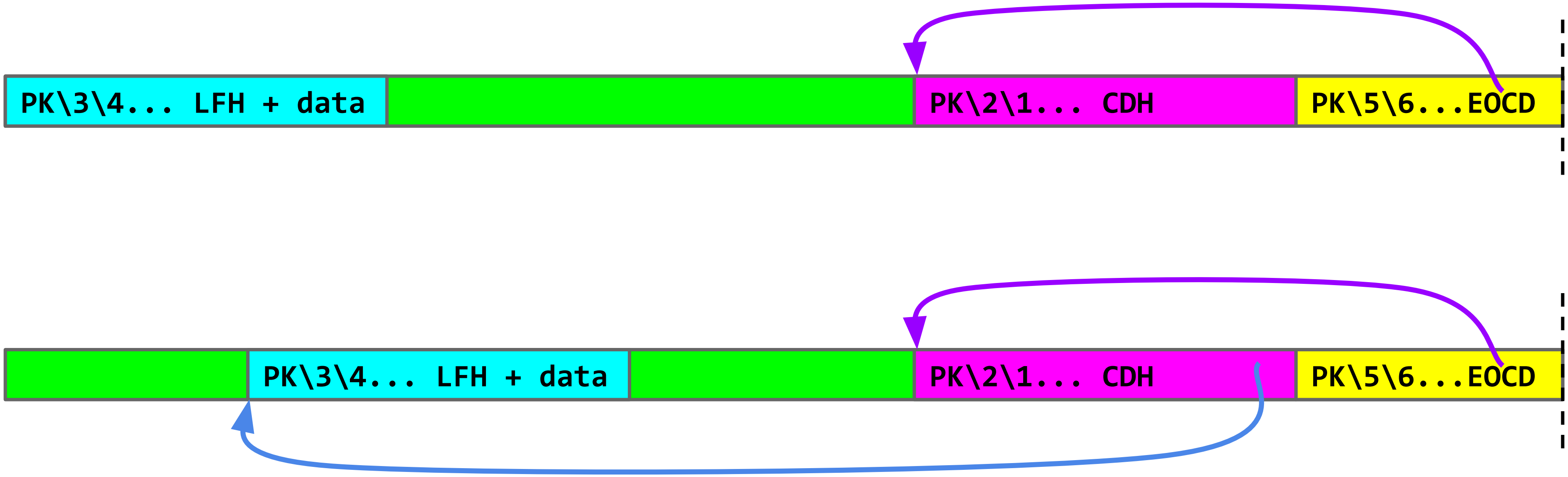
PK\2\1... CDH

PK\5\6...EOCD

Files (header+data)

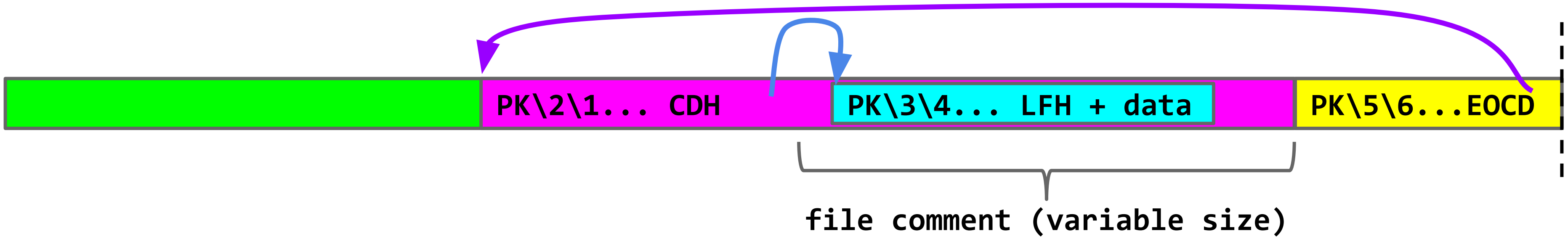
List of files
(and pointers)

ZIP - a complete file (continued)



If the list of the files has pointers to files...
... the ZIP structure can be more relaxed.

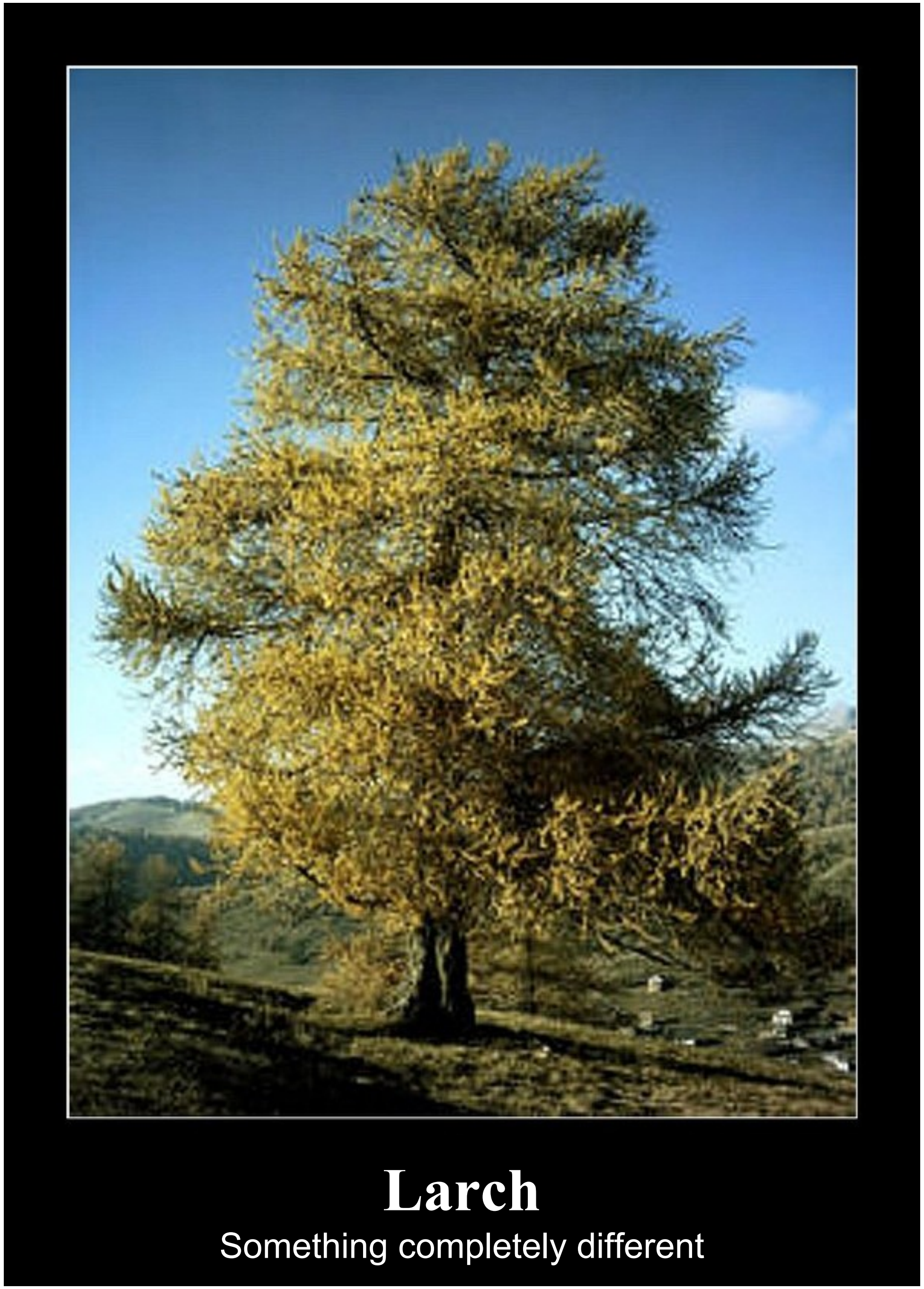
ZIP - a complete file (continued)



You can even do an "inception"
(some parsers may allow **EOCD(CHD(LFH))**)

And now back to our show!

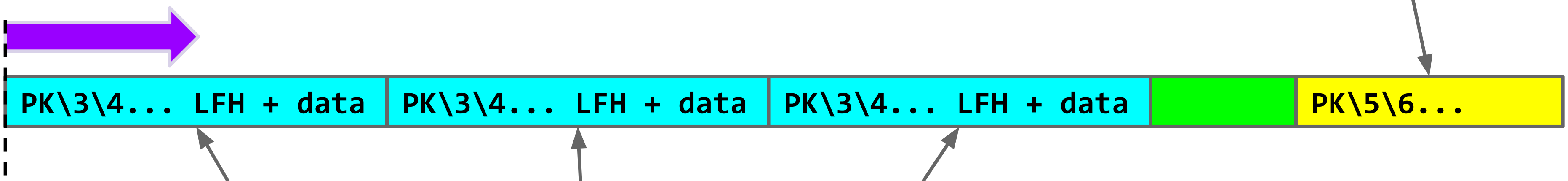
(we were looking for the EOCD)



ZIP - looking for the "header"?

"stream"
Let's ignore EOCD!

(it's sometimes faster)
(99.9% of ZIPs out there can be parsed this way)



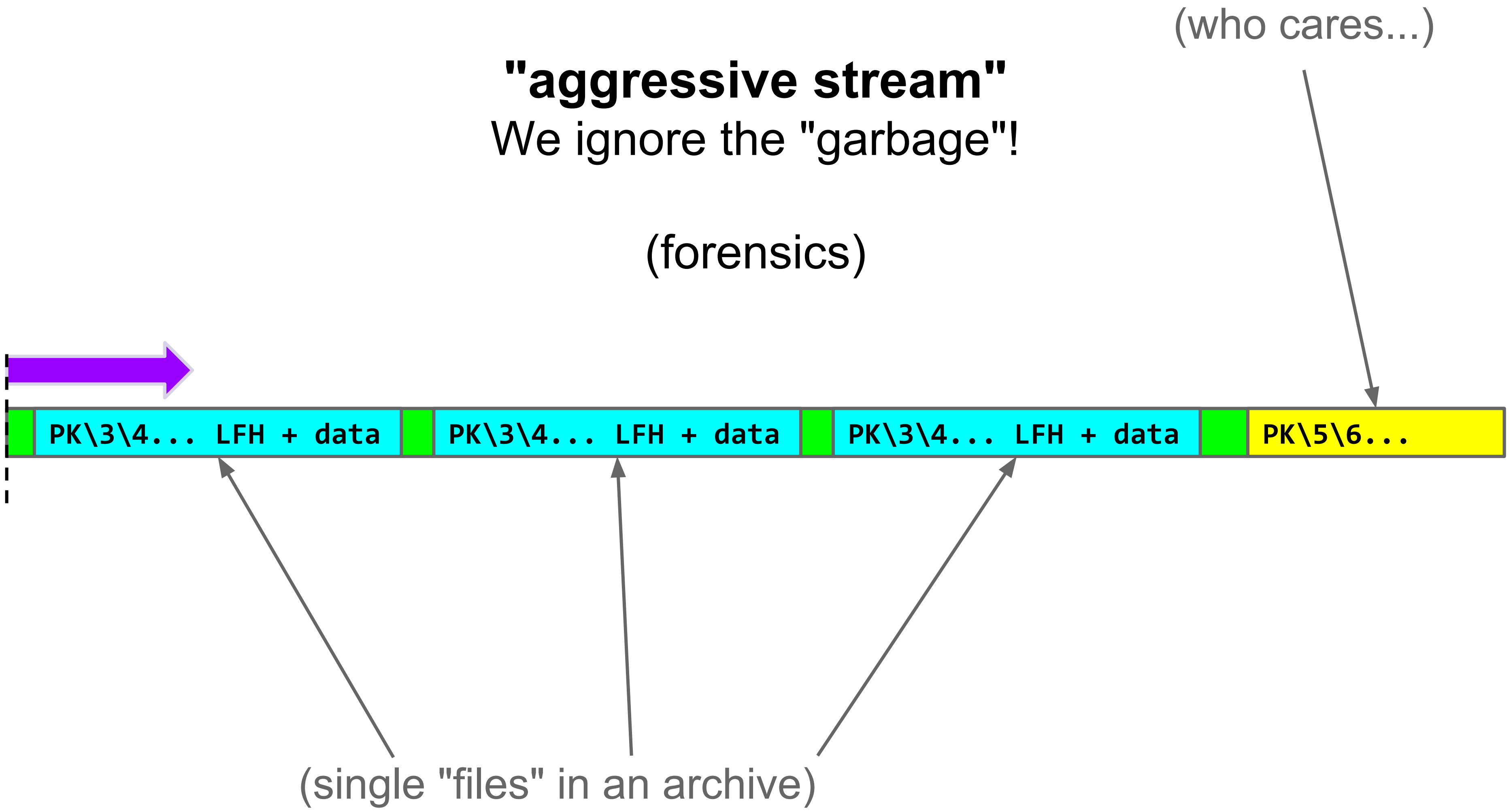
(who cares...)

(single "files" in an archive)

ZIP - looking for the "header"?

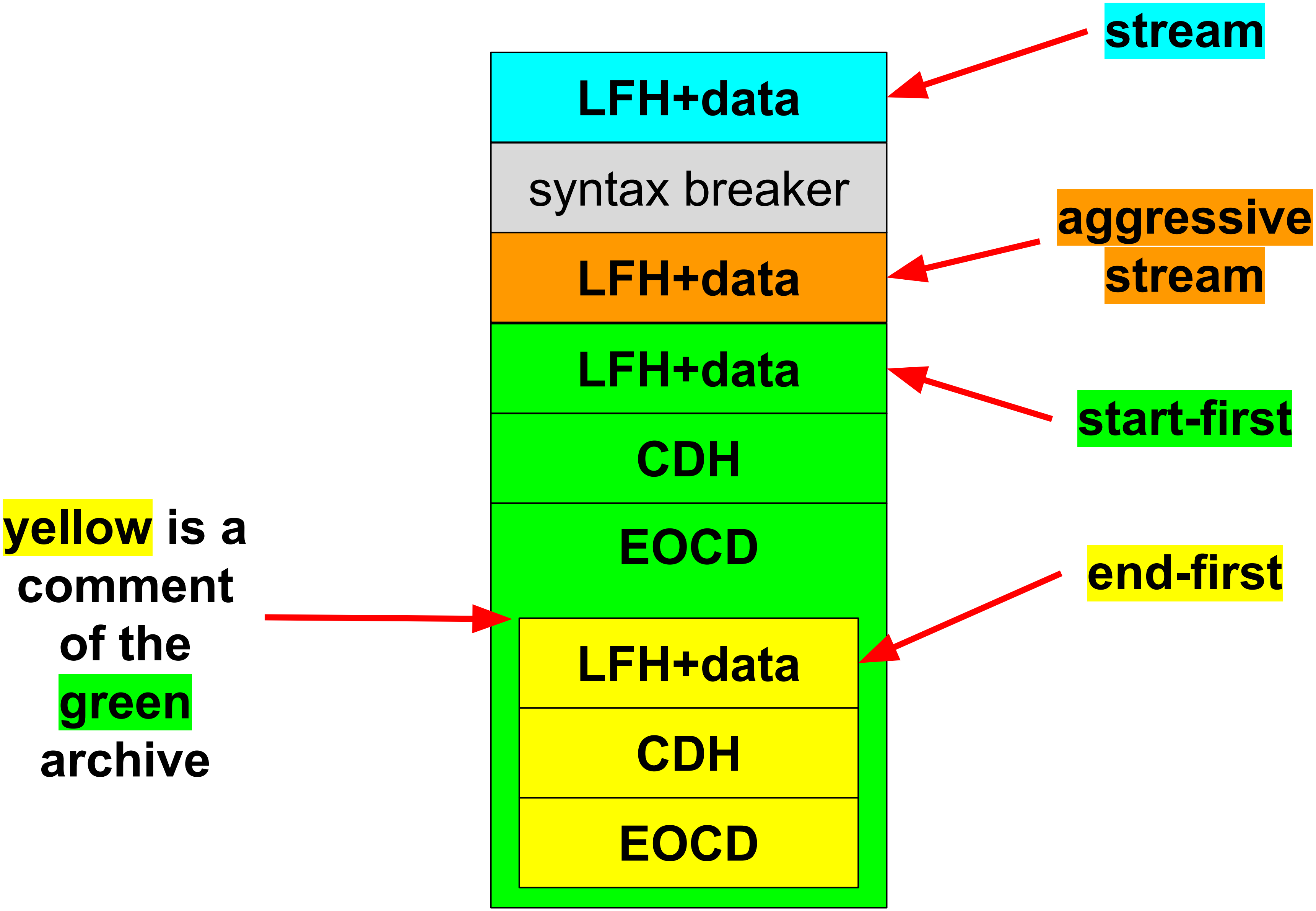
"aggressive stream"
We ignore the "garbage"!

(forensics)



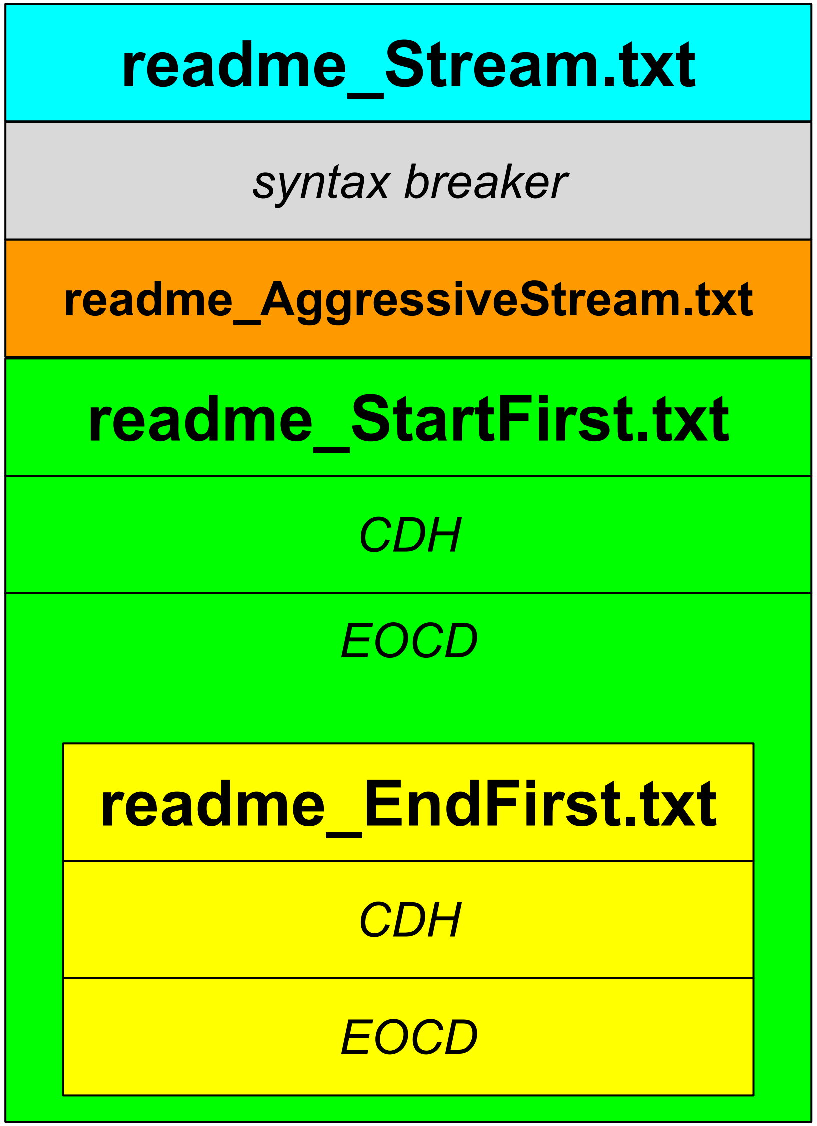
Let's test the parsers!
abstract.zip

abstract.zip



abstract.zip

abstract.zip - result summary



readme_Stream.txt

syntax breaker

readme_AggressiveStream.txt

readme_StartFirst.txt

CDH

EOCD

readme_EndFirst.txt

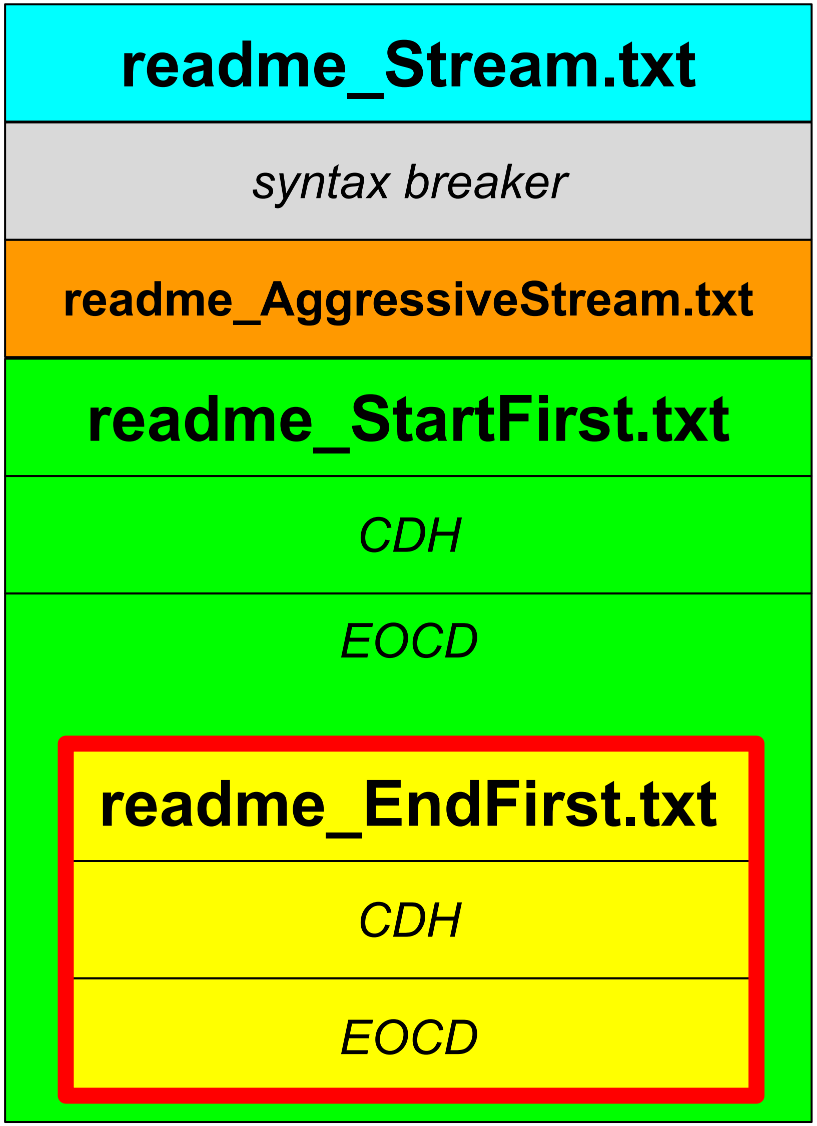
CDH

EOCD

Thanks!

- Mulander
- Felix Groebert
- Salvation
- j00ru

abstract.zip



readme_Stream.txt

syntax breaker

readme_AggressiveStream.txt

readme_StartFirst.txt

CDH

EOCD

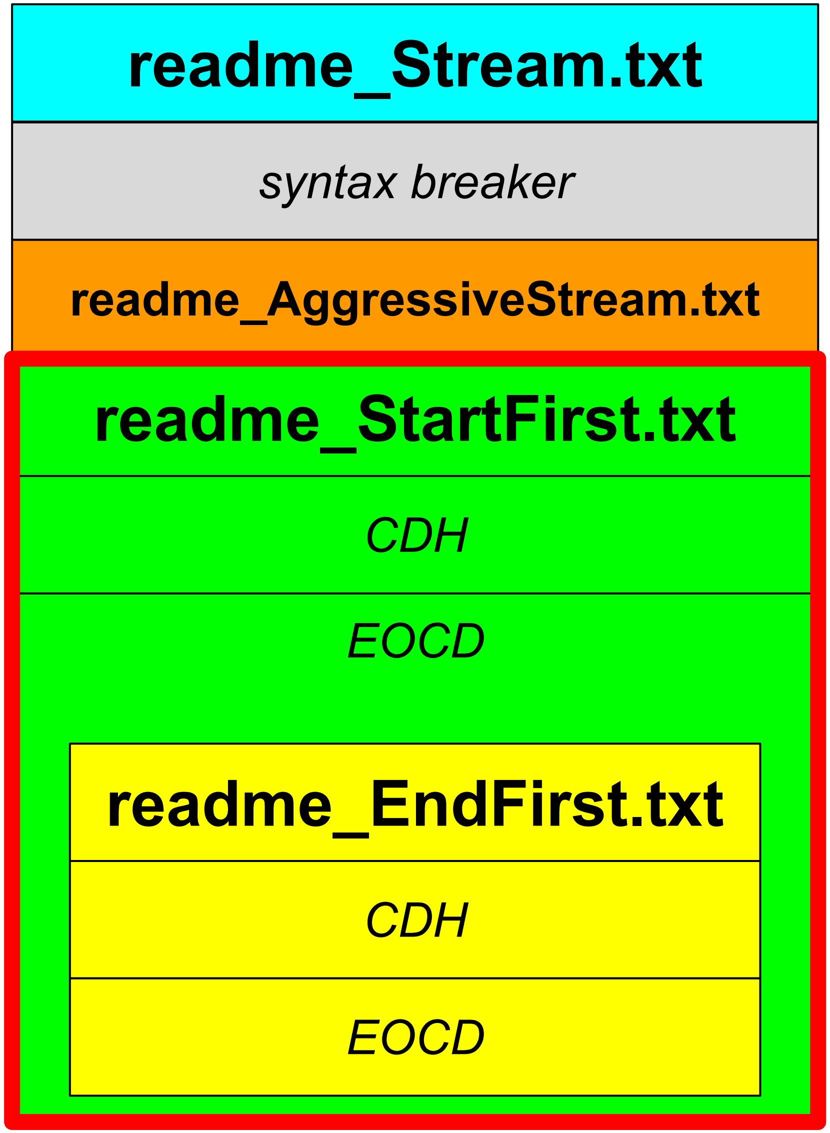
readme_EndFirst.txt

CDH

EOCD

Total Commander 8.01
UnZip 6.00 (Debian)
Midnight Commander
Windows 7 Explorer
ALZip
KGB Archiver
7-zip
b1.org
Python zipfile
JSZip
C# DotNetZip
perl Archive::Zip
Jeffrey's Exif Viewer
WOBZIP
GNOME File Roller
WinRAR
OSX UnZip
zip.vim v25
Emacs Zip-Archive mode
Ada Zip-Ada v45
Go archive/zip
Pharo smalltalk 2.0 ZipArchive
Ubuntu less
Java ZipFile

abstract.zip



readme_Stream.txt

syntax breaker

readme_AggressiveStream.txt

readme_StartFirst.txt

CDH

EOCD

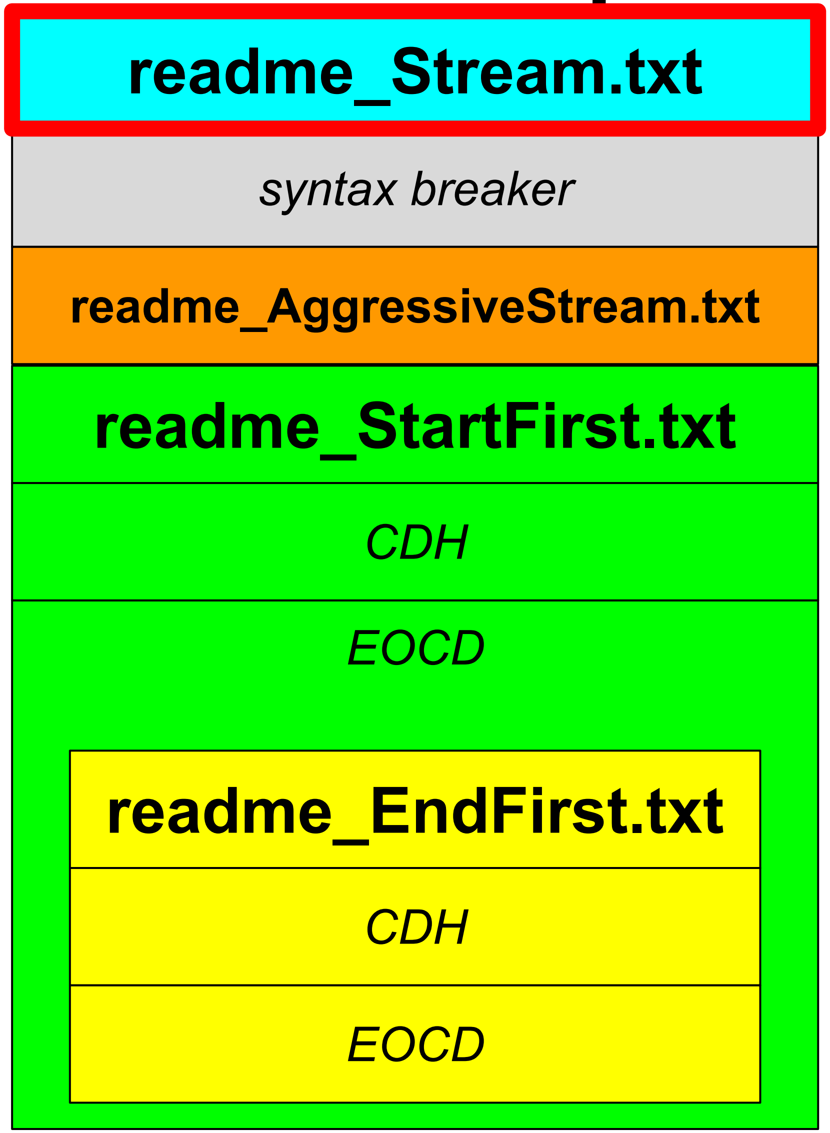
readme_EndFirst.txt

CDH

EOCD

```
PHP ZipArchive
PHP zip_open ...
PHP zip:// wrapper
tcl + tclvfs + tclunzip
```

abstract.zip



readme_Stream.txt

syntax breaker

readme_AggressiveStream.txt

readme_StartFirst.txt

CDH

EOCD

readme_EndFirst.txt

CDH

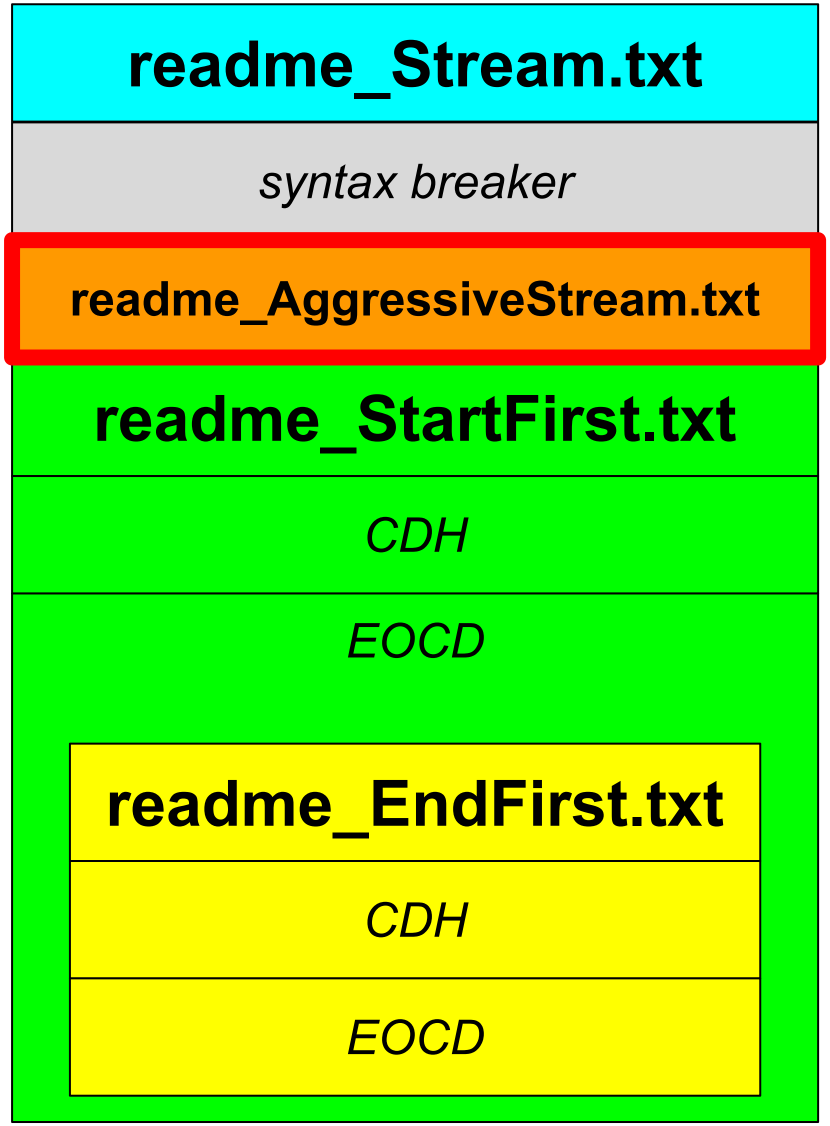
EOCD

Ruby rubyzip2

Java ZipArchiveInputStream

java.util.zip.ZipInputStream

abstract.zip



`binwalk (found all)`

abstract.zip - who cares?

From my experience:

- **verify files via End-First**
- **unpack via Stream**

Ups.

abstract.zip - AV

EICAR test results (using VT):

- most End-First
- some Aggressive
- Stream-only:
  - VBA32
  - NANO-Antivirus
  - Norman
  - F-Prot
  - Agnitum
  - Commtouch

https://docs.google.com/spreadsheet/ccc?key=0Apy5AGVPzpIOdDRPTFNJQXpqNkdjUzl4SE80c1kwdkE&usp=sharing

File names in ZIP

There are two*:

- LFH
- CDH
- Extra: Info-ZIP Unicode Path Extra Field

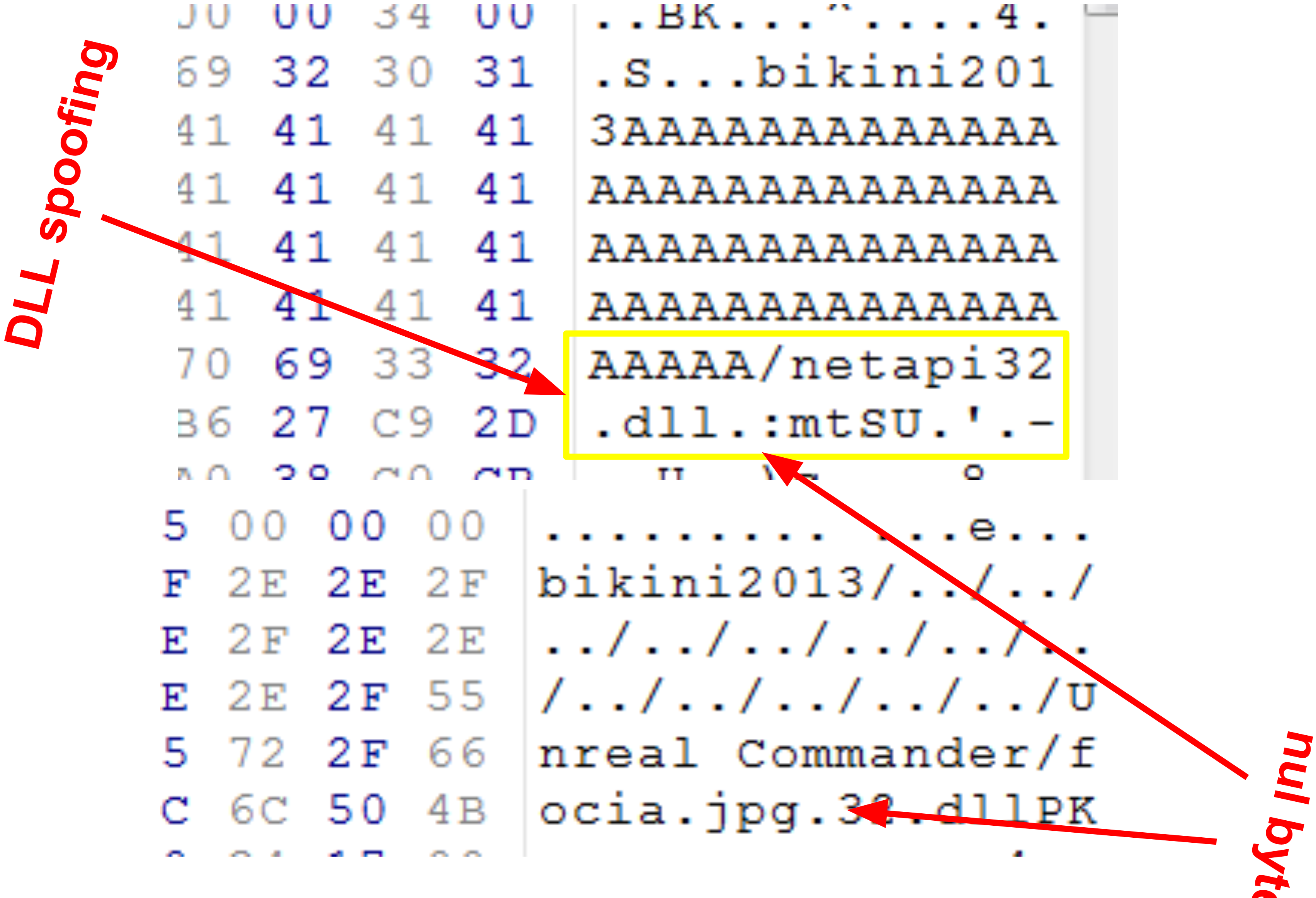
(unzip in GNU/Linux, etc)

each ZIP file can has N extra fields, both in LFH and CDH separately ;)

DEMO

** There are only two hard problems in Computer Science: naming things, cache coherency, and off-by-one errors.*

File names in ZIP - bikini



```
50 00 34 00   ..BK...^.....4.
69 32 30 31   .S...bikini201
41 41 41 41   3AAAAAAAAAAAAA
41 41 41 41   AAAAAAAAAAAAAA
41 41 41 41   AAAAAAAAAAAAAA
41 41 41 41   AAAAAAAAAAAAAA
70 69 33 32   AAAAA/netapi32
36 27 C9 2D   .dll.:mtSU.'.-
```

```
5 00 00 00   ......... ...e...
F 2E 2E 2F   bikini2013/../../
E 2F 2E 2E   ../../../../../..
E 2E 2F 55   /../../../../../U
5 72 2F 66   nreal Commander/f
C 6C 50 4B   ocia.jpg.32.dllPK
```

Path Traversal! (+ wrong permissions) (+ LFH-vs-CDE)

File names in ZIP (cont.)

A couple of files with the same name?

DEMO (if we have time)

Food for thought:

- lower-upper case
- ADS :$data

File names in ZIP (cont.)

Other ideas?

- SMB network drives?
- absolute paths?
- XSS in the name? (a common problem)
- very long names (cheers Icewall!)
- charset? (utf-8 vs OS vs ibm 437)
- unicode RTL

ZIP vs low-level

Standard ideas where the bugs could be:

- the old good buffer overflow
  - compressed size < after-unpack(data)
  - long file names?
- memory info disclosure?
  - uncompressed size > after-unpack(data)
  - uncompressed size > compressed size dla STORED

DEMO

GIFAR / Ange CorkaMIX (binary polyglots)

http://en.wikipedia.org/wiki/Gifar

https://code.google.com/p/corkami/wiki/mix

CorkaMIX, CorkaMInuX and CorkaM-OsX are respectively valid Windows, Linux and OS X binaries, and also a working PDF document, Jar (Zip + Class + manifest), and HTML + JavaScript files.

ZIP & stegano?

Sometimes appears in CTFs :)

- Office XML Steganography Tool (extra field)
- "Unused" space.
- More data than uncompressed size claims (STORED)
- Extra, comment
- Same-name files or name eq. \0
- abstract!
- Abusing compression algorithms.

Bonus - ZIP download!

Since ZIP has a list of all files and pointers to them...

... you can download a single file from an archive over HTTP using Range: field :)

```
> python zipdl.py http://example.com/example.zip
File Name                        ...      Size
readme_EndFirst.txt              ...       231
> python zipdl.py http://example.com/example.zip readme_EndFirst.txt
> ls -la readme_EndFirst.txt
-rw-r----- 1 gynvael gynvael 231 May 13 14:45 readme_EndFirst.txt
>
```

http://gynvael.coldwind.pl/n/python_zipdl

Bonus - ZIP download!

```
class MyFileWrapper:
  def __init__(self, url):
    --> HEAD ...

  def seek(self, offset, whence):

  def tell(self):

  def read(self, amount=-1):
    --> GET ...
        Range: bytes=%u,%u

z = zipfile.ZipFile(some MyFileWrapper object)
```

Oh yes... and there are packbombs.

Three types:

1. **small zip --> very big file**
   (unreal cmd uncomp size)

2. **small zip --> a couple of zips --> ... --> very large files**

3. **infinite recusion ftw!**
   http://research.swtch.com/zip
   (by Russ Cox)

EPIC!
(demo?)

Encryption

- Oldest scheme long gone and broken

- Newer scheme broken if you can predict the first 13 bytes of plaintext.
  (known-plaintext attack)

- Now it just uses AES.

Note: MOST zip compressors only encrypt data, but not file names.

(though good ones encrypt everything)

That's all about ZIP :)

Big thanks to the author of
Unreal Commandera
for not fixing any bugs that I reported in
2007 :)

http://gynvael.coldwind.pl/?id=30

APPNOTE

http://www.pkware.com/documents/casestudies/APPNOTE.TXT

Tools: **nasm + hex workshop**

A short note on RAR

It's a "packed" chunk-based format. No separation for LFH/CDH.

The header is at the beginning.

Boooooooooooooring!

A short note on RAR

It's a "packed" chunk-based format. No separation for LFH/CDH.

The header is at the beginning.

Boooooooooooooring!

But there's a turing-complete VM! (hi Tavis!)
http://blog.cmpxchg8b.com/2012/09/fun-with-constrained-programming.html

RAR VM

Looks like x86 (assembler by TavisO):

```
mov     r3, #0x1000                 ; Output buffer.
mov     [r3+#0], #0x6c6c6548        ; 'lleH'
mov     [r3+#4], #0x57202c6f        ; 'W ,o'
mov     [r3+#8], #0x646c726f        ; 'dlro'
mov     [r3+#12], #0x00000a21       ; '!\n'
mov     [VMADDR_NEWBLOCKPOS],  r3   ; Pointer
mov     [VMADDR_NEWBLOCKSIZE], #14  ; Size
call    $_success
```

RAR VM cd...

Regs: `r0-r7`

Mem: (256KB)

Addressing: `[#0x12345], [r0], [r4+#0x1234]`

Consts: `#0x12312`

And so on...

RAR VM - CRC32

CRC32(output) must be equal to CRC32 from the header!

Julien's CRC32 preimage algorithm!

https://www.cr0.org/misc/jt-securitech-06-11.pdf

What for?

CrackMe / CTF!

np: 29c3 CTF 2012

Write-up by PiggyBird CTF Team:
http://piggybird.net/?p=374

Pack bombs?

RAR - other things...

- path traversal
- xss
- etc.. all of these bug classes relate to RAR as well (if someone uses it incorrectly)

Other?

A LOT of archive formats out there:

- 7z?
- .a / .lib? (yep, these are archives as well; can we attack a build server?)
- rule of life: every gamedev must develop his own new archive format :)
  (e.g. Blizzards MPQ - sometimes archives are sent P2P between players)

The End. Questions?

gynvael@coldwind.pl

http://gynvael.coldwind.pl/