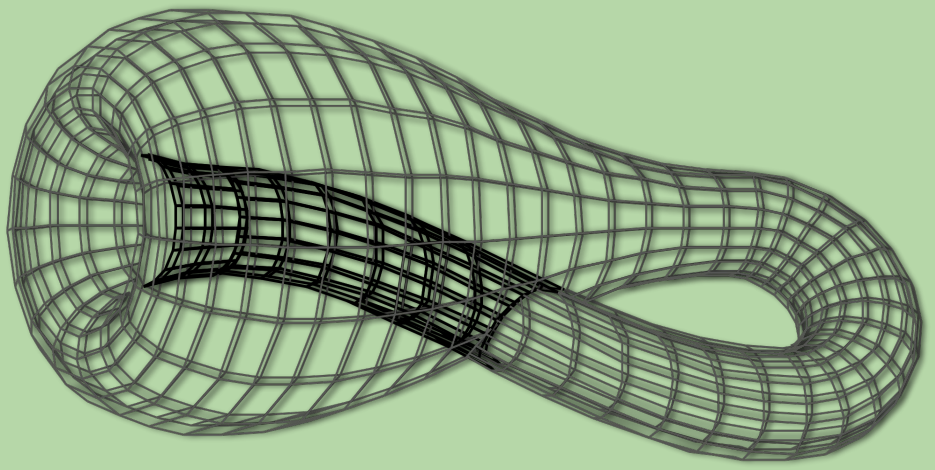
# INSIDE OUT

## Abusing archive file formats



Ange Albertini

July 2022

# ABUSING ARCHIVE FILE FORMATS

A presentation by

Ange
Albertini

A.K.A.

# About the author

My own views and opinions.

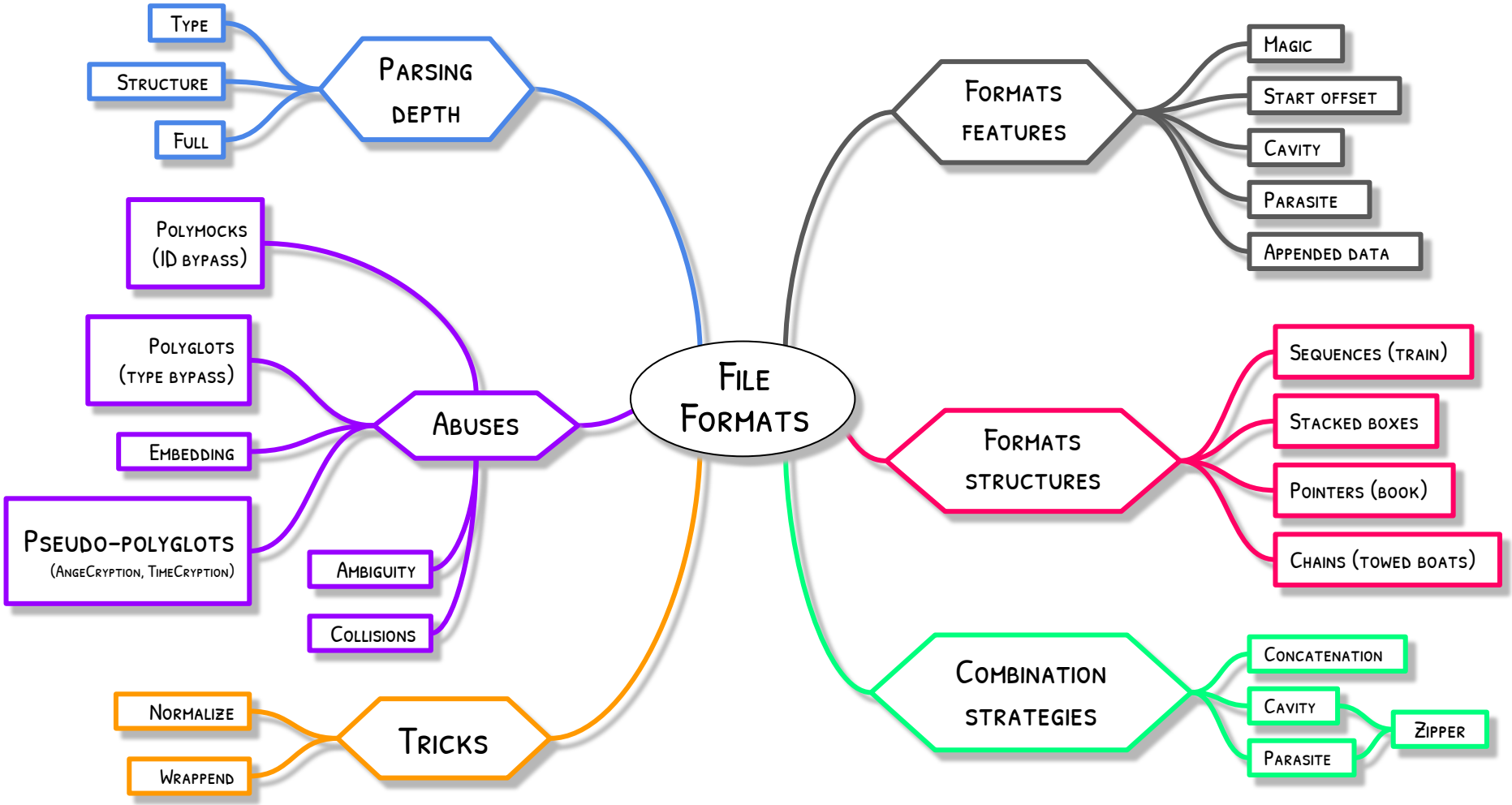
- Reverse engineering and hex viewing since the 80s.
- Author of Corkami since 2007.
- PoC or GTFO since 2013.


my license plate is a CPU.
my phone case is a PDF doc.
my resume is a Super NES/Megadrive rom.

Professionally

- Symantec, Avira, Google
- malware analyst, infosec engineer

https://github.com/angea/pocorgtfo/blob/master/README.md
https://github.com/corkami/pocs/tree/master/poly/SnesMd




This is my world...

# Contributions to hash collisions

Crypto-polyglots



2014
BsidesLV


2017
BlackHat, RWC, Crypto


2019
PTS, Hack.lu


2019 (workshop)
PTS, Hack.lu, BA...

https://github.com/corkami/collisions

docs, precomputed prefixes, scripts, pocs... (MIT licence)

THIS SLIDE IS AN

# HONEST TALK TRAILER

A CORKAMI ORIGINAL PRODUCTION

## 2021.04.20 12:06 "[Tiff] Libtiff 4.3.0 is released", by Even Rouault

Hi,

No blocking issues have been reported on rc1, so I've moved to promote it to final.

Read about this release at https://libtiff.gitlab.io/libtiff/v4.3.0.html

The release tarball can be downloaded at:

http://download.osgeo.org/libtiff/tiff-4.3.0.tar.gz
http://download.osgeo.org/libtiff/tiff-4.3.0.zip

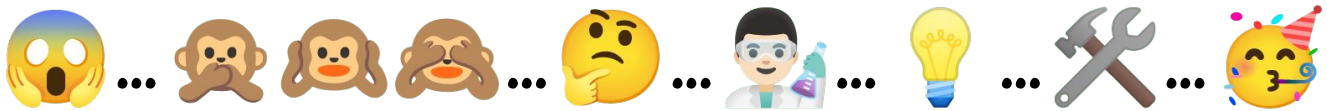
md5sum:

0a2e4744d1426a8fc8211c0cdbc3a1b3 tiff-4.3.0.tar.gz
6b792ee97cffef772206bc3ba0cd257c tiff-4.3.0.zip

The release files are also signed with my private PGP key, and the associated signature files have ".sig" extensions.

Even

http://www.spatialys.com
My software is free, but my time generally not.



https://www.asmail.be/msg0055059467.html

# Flashback

Until this research,
official Libtiff releases as ZIP and TAR.GZ
were announced with MD5 (and PGP sigs).

<Company> indexes Office files with MD5.

Office files are XML files in ZIPs

How bad is it actually?
Can we prove them wrong? *Really* wrong?

# Plot (spoilers)

No known way to easily abuse XML, TAR, GZIP or ZIP with hash collisions.

-> what about TAR.GZ and DOCX (zipped XML) ?

1. XML isn't exploitable... but ZIP comes to the rescue!
2. "Uncommon" GZIPs are actually exploitable.

# Recap on hash collisions

# Existing attacks (MD2/4/5 SHA1)

No practical pre-image attack:
can't make a file with an arbitrary hash.

Existing attack:
make 2 files with *some* arbitrary contents get the same hash.

"buy 1, get 1 free" risk:

Get $F_1$ validated, then use $F_2$ interchangeably.

# "Buy 1 get 1 free"


Get clean 'bill.pdf' file whitelisted by hash,
spread malicious 'kill.exe'.

Get benign certificate signed, enjoy full powers.

Problem: $F_1$ and $F_2$ need to be both "valid"

- with all parsers ? (compatibility)
- permanently ? (not a fixable bug)

# Actual examples

Structure of colliding files:

Everything is aligned to 64 bytes.

1. Prefix (optional)

   Either identical for both files

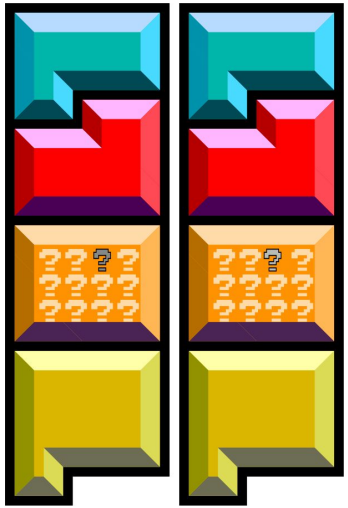
   or chosen

2. Padding

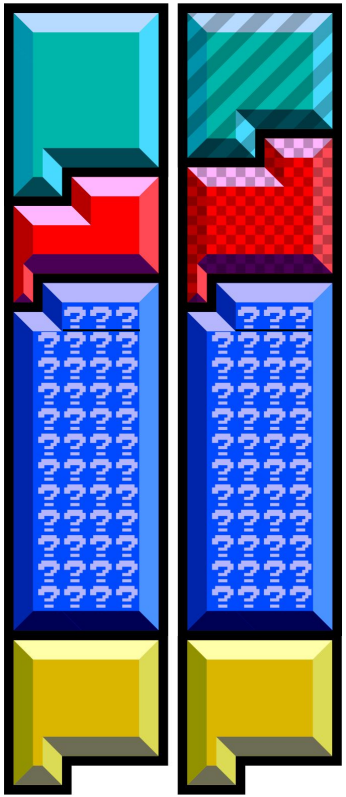
3. High entropy collision blocks

With tiny differences.

4. Identical suffix (optional)



Identical prefix collision

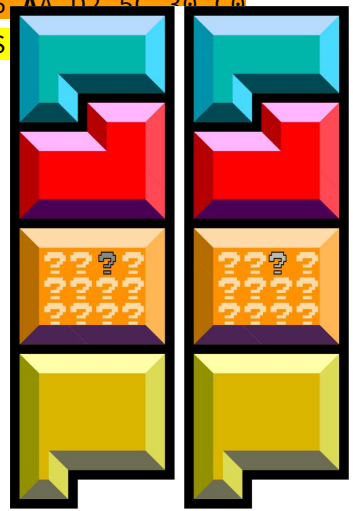


Chosen prefix collision

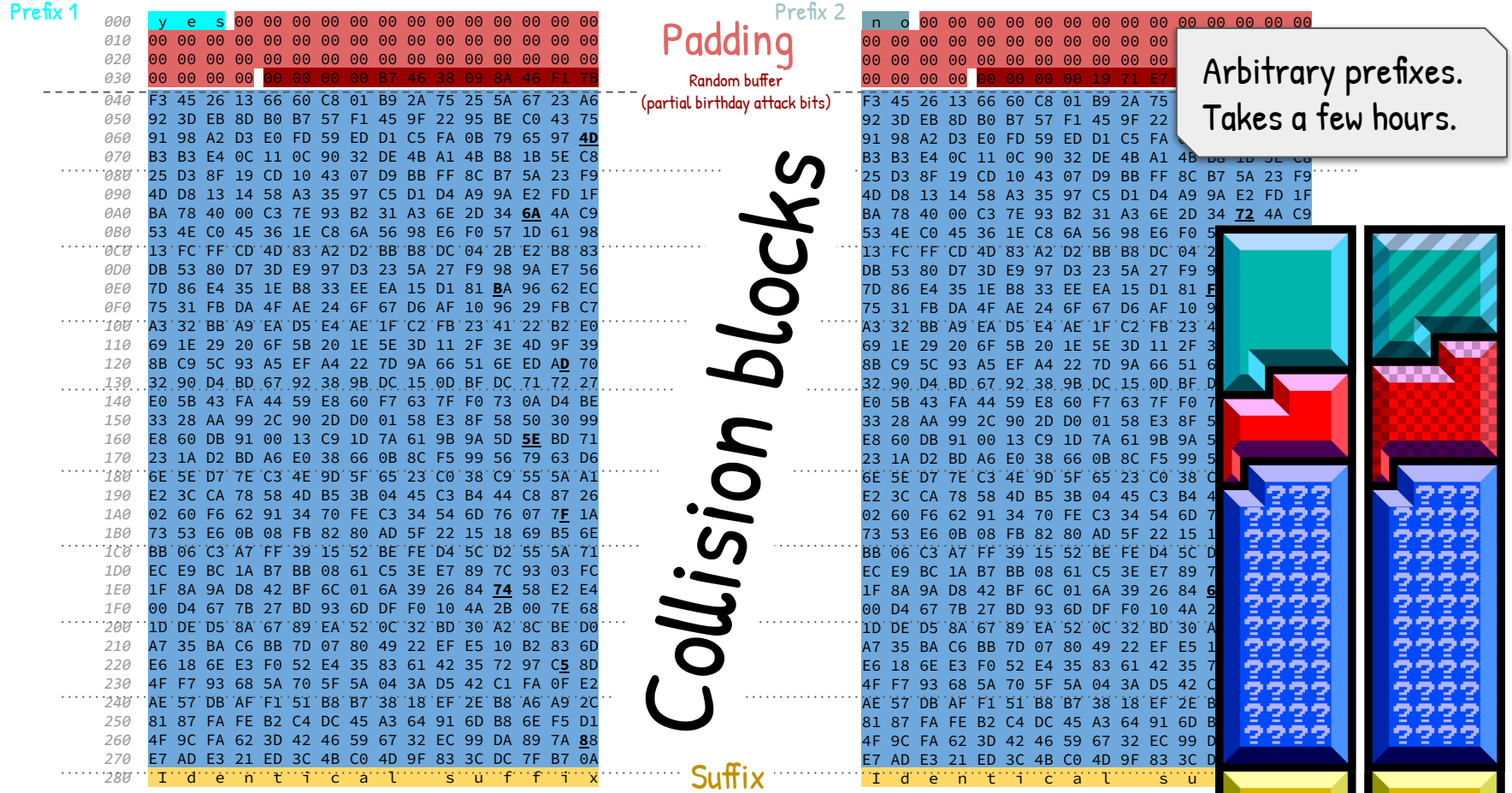
| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | H | e | r | e | | i | s | | a | | f | i | l | e | | w |
| 10 | i | t | h | | a | | f | e | w | | b | y | t | e | s | 00 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 40 | CE | 84 | 07 | 61 | 4B | BA | 7A | 3D | 3A | EA | 8A | AA | F8 | EE | 1D | E5 |
| 50 | 44 | 17 | 9B | **F**0 | 0A | E0 | D2 | 64 | 21 | E2 | 38 | E1 | 94 | 18 | 0A | F6 |
| 60 | 93 | D2 | B5 | E4 | FC | 2F | 3A | 32 | 4F | 50 | 46 | 01 | F1 | **4**B | B**F** | 02 |
| 70 | 23 | EE | EF | BF | 92 | B5 | 7C | 29 | D9 | C5 | 66 | **0**8 | 31 | 5E | 7A | 1D |
| 80 | 2F | 5A | 9C | 5C | 12 | 8E | DF | F2 | 85 | 17 | 5B | DD | 67 | 25 | 05 | 78 |
| 90 | 13 | F2 | BF | **D**6 | 64 | 59 | F2 | C8 | 8B | C3 | 00 | 6F | 8B | 5F | 88 | C6 |
| A0 | CB | 3D | 80 | E4 | 9F | 48 | 91 | 5E | 34 | 06 | D0 | 3A | 8B | **0**3 | FB | E0 |
| B0 | ED | 18 | 67 | 0F | C8 | 3A | C9 | A1 | E7 | 48 | F6 | **2**A | D2 | 5C | 30 | C0 |
| C0 | I | d | e | n | t | i | c | a | l | | S | u | f | f | i | x |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H | e | r | e | | i | s | | a | | f | i | l | e | | w |
| | i | t | h | | a | | f | e | w | | b | y | t | e | s | 00 |
| | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| | CE | 84 | 07 | 61 | 4B | BA | 7A | 3D | 3A | EA | 8A | AA | F8 | EE | 1D | E5 |
| | 44 | 17 | 9B | **7**0 | 0A | E0 | D2 | 64 | 21 | E2 | 38 | E1 | 94 | 18 | 0A | F6 |
| | 93 | D2 | B5 | E4 | FC | 2F | 3A | 32 | 4F | 50 | 46 | 01 | F1 | **C**B | B**E** | 02 |
| | 23 | EE | EF | BF | 92 | B5 | 7C | 29 | D9 | C5 | 66 | **8**8 | 31 | 5E | 7A | 1D |
| | 2F | 5A | 9C | 5C | 12 | 8E | DF | F2 | 85 | 17 | 5B | DD | 67 | 25 | 05 | 78 |
| | 13 | F2 | BF | **5**6 | 64 | 59 | F2 | C8 | 8B | C3 | 00 | 6F | 8B | 5F | 88 | C6 |
| | CB | 3D | 80 | E4 | 9F | 48 | 91 | 5E | 34 | 06 | D0 | 3A | 8B | **8**3 | FB | E0 |
| | ED | 18 | 67 | 0F | C8 | 3A | C9 | A1 | E7 | 48 | F6 | **A**A | D2 | 5C | 30 | C0 |
| | I | d | e | n | t | i | c | a | l | | S | | | | | |

Takes a few seconds.



# 1/3 An Identical Prefix Collision



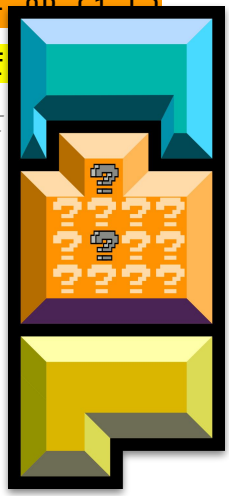

# 2/3 A Chosen Prefix Collision

| | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | H | e | r | e | | i | s | | m | y | | p | r | e | f | i |
| 10 | x | ! | ! | \n | 85 | 33 | 77 | E3 | 4E | 2D | B4 | F7 | 33 | 52 | CD | 17 |
| 20 | 63 | F0 | 24 | 11 | 8E | 42 | EE | 0D | 6D | 73 | 1D | 18 | FA | BA | 3F | 2D |
| 30 | 53 | C6 | C3 | 9E | 17 | F6 | 86 | 5F | 44 | EB | 71 | C4 | 24 | FB | 67 | 10 |
| 40 | 53 | 75 | 43 | D7 | 3B | 33 | 9A | FE | E7 | B8 | ED | BD | AE | A8 | 07 | B9 |
| 50 | F4 | 49 | FA | 94 | 34 | 01 | 54 | DB | BE | 87 | 3C | 39 | AF | CD | A1 | 82 |
| 60 | C4 | EA | 3A | F8 | 9B | 7C | BA | D3 | AC | AF | 3D | 47 | A1 | 03 | 0D | 34 |
| 70 | 7F | FF | 0C | 58 | 92 | BC | 2B | 8A | A4 | 31 | 53 | EE | 2F | 9B | C1 | F2 |
| 80 | I | d | e | n | t | i | c | a | l | | S | u | f | f | i | x |

| | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | H | e | r | e | | i | s | | m | z | | p | r | e | f | i |
| 10 | x | ! | ! | \n | 85 | 33 | 77 | E3 | 4E | 2D | B4 | F7 | 33 | 52 | CD | 17 |
| 20 | 63 | F0 | 24 | 11 | 8E | 42 | EE | 0D | 6D | 73 | 1D | 18 | FA | BA | 3F | 2D |
| 30 | 53 | C6 | C3 | 9E | 17 | F6 | 86 | 5F | 44 | EB | 71 | C4 | 24 | FB | 67 | 10 |
| 40 | 53 | 75 | 43 | D7 | 3B | 33 | 9A | FE | E7 | B7 | ED | BD | AE | A8 | 07 | B9 |
| 50 | F4 | 49 | FA | 94 | 34 | 01 | 54 | DB | BE | 87 | 3C | 39 | AF | CD | A1 | 82 |
| 60 | C4 | EA | 3A | F8 | 9B | 7C | BA | D3 | AC | AF | 3D | 47 | A1 | 03 | 0D | 34 |
| 70 | 7F | FF | 0C | 58 | 92 | BC | 2B | 8A | A4 | 31 | 53 | EE | 2F | | | |
| 80 | I | d | e | n | t | i | c | a | l | | S | u | f | | | |

+1

+1 on the 10th byte
of the collision block.
Takes a few **minutes**.



# 3/3 Unicoll: an IPC with a predictable difference

# Formats and hash collisions (in general)

Most formats...

✔ are parsed top-down.

✔ tolerate appended data
(of any length and content).

-> trivial chosen-prefix collision of a single pair:
Run Hashclash on both files. Done.
(collision blocks will be ignored by parsers).

https://github.com/cr-marcstevens/hashclash

# Formats and hash collisions (exceptions)

Notable exceptions:

- ZIP is parsed bottom-up.
- No appended data for XML & GZIP (GZIP -> warning).
- ZIP only works with 64 kb of appended data at most.

ZIP, XML, GZIP aren't hash collision friendly.

(otherwise this talk wouldn't make sense)

# Increased impact via file formats tricks

MD5 with standard case via chosen prefix:
70h*core. Repeat for every pair of files.

One-time collision
EXIT ONLY

MD5+file tricks and pre-computed prefixes:
70h*core. Needed only once.
Then less than 1 second of file manipulations.

Reusable prefixes
FAST LANE

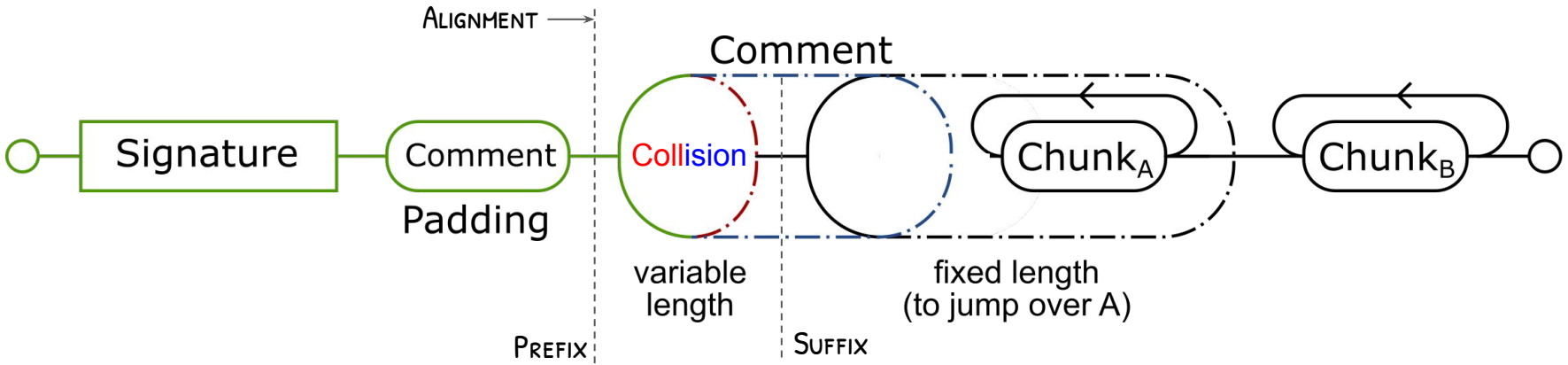
For more info ->


https://speakerdeck.com/ange/colltris

# Layout of a reusable collision

A sequence of 3 'comment' blocks:

1. Padding for alignment
2. Variable length by collision
3. Covering first file contents - toggled by comment #2



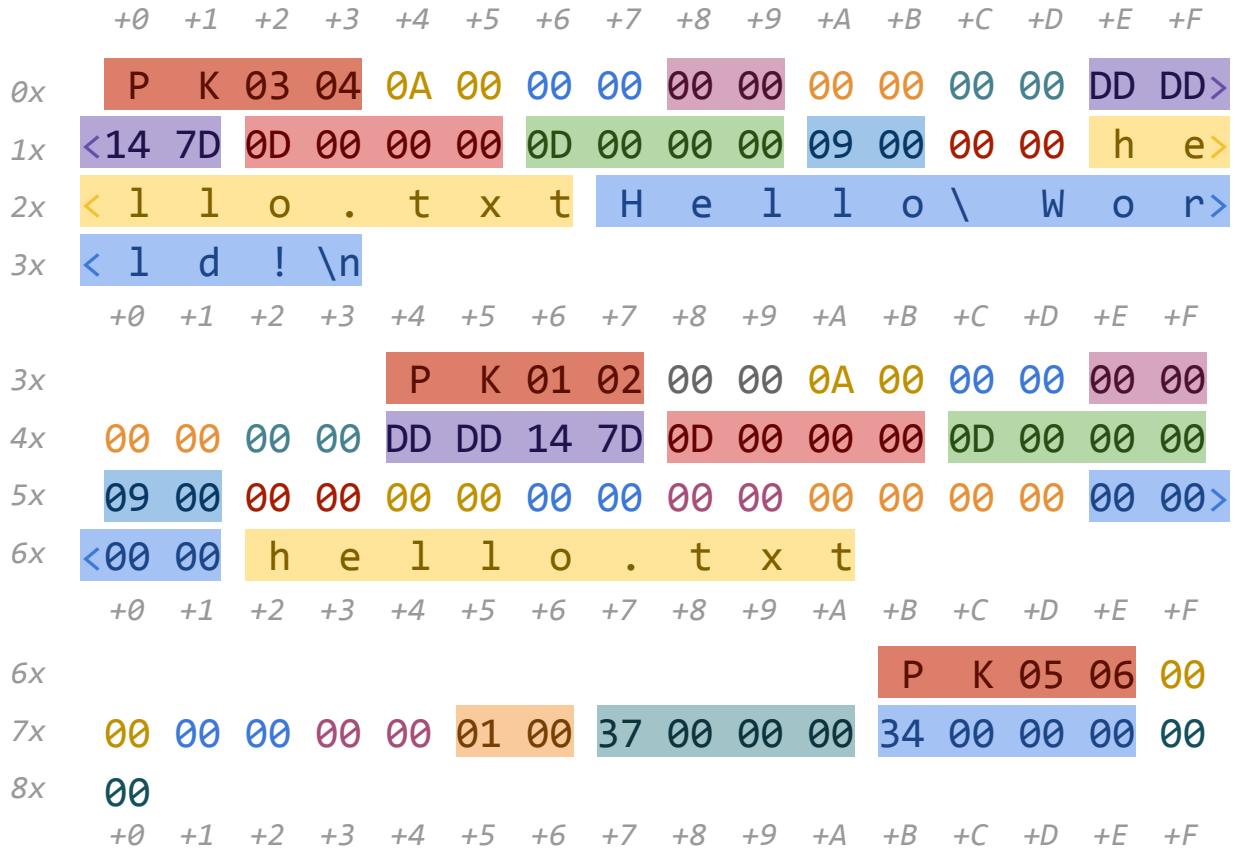
# Collisions of ZIP archives

Easy / Hard / Impossible?

Single use / Reusable?

```
    +0  +1  +2  +3  +4  +5  +6  +7  +8  +9  +A  +B  +C  +D  +E  +F
0x   P   K 03 04 0A 00 00 00 00 00 00 00 00 00 DD DD
1x  14 7D 0D 00 00 00 0D 00 00 00 09 00 00 00  h  e
2x   l   l   o   .   t   x   t   H   e   l   l   o  \   W   o   r
3x   l   d   ! \n   P   K 01 02 00 00 0A 00 00 00 00 00
4x  00 00 00 00 DD DD 14 7D 0D 00 00 00 0D 00 00 00
5x  09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6x  00 00  h  e  l  l  o  .  t  x  t  P  K 05 06 00
7x  00 00 00 00 00 01 00 37 00 00 00 34 00 00 00 00
8x  00
    +0  +1  +2  +3  +4  +5  +6  +7  +8  +9  +A  +B  +C  +D  +E  +F
```

# A simple ZIP archive




# A dissected Zip archive

## Local File Header

| | | | |
|---|---|---|---|
| 00 | Signature | 4 | PK\3\4 |
| 04 | Needed Version | 2 | 10 |
| 06 | Flags | 2 | None |
| 08 | CompMethod | 2 | 0=Store |
| 20A | ModTime | 2 | 00:00 |
| 20C | ModDate | 2 | 0/0/1980 |
| 0E | CRC32 | 4 | 0x7D14DDDD |
| 12 | CompressSize | 4 | 13 |
| 16 | UncompSize | 4 | 13 |
| 1A | FileNameLen | 2 | 9 |
| 1C | ExtraFieldLen | 2 | 0 |
| 1E | FileName | ? | hello.txt |
| 27 | ExtraField | ? | n/a |
| 27 | Content | ? | Hello World\n |

## Central Directory

| | | | |
|---|---|---|---|
| 34 | Signature | 4 | PK\1\2 |
| 38 | Made Version | 2 | 0 |
| 3A | Needed Version | 2 | 10 |
| 3C | Flags | 2 | None |
| 3E | CompMethod | 2 | 0=Store |
| 40 | ModTime | 2 | 00:00 |
| 42 | ModDate | 2 | 0/0/1980 |
| 44 | CRC32 | 4 | 0x7D14DDDD |
| 48 | CompressSize | 4 | 13 |
| 4C | UncompSize | 4 | 13 |
| 50 | FileNameLen | 2 | 9 |
| 52 | ExtraFieldLen | 2 | 0 |
| 54 | FileCommentLen | 2 | 0 |
| 56 | DiskNumberStart | 2 | 0 |
| 58 | InternalAttr | 2 | 0 |
| 5A | ExternalAttr | 4 | 0 |
| 5E | LFHOffset | 4 | 0 |
| 62 | FileName | ? | hello.txt |
| 6A | ExtraField | ? | n/a |
| 6A | FileComment | ? | n/a |

## End of Central Directory

| | | | |
|---|---|---|---|
| 6C | Signature | 4 | PK\5\6 |
| 6E | ThisDiskNumber | 2 | 0 |
| 71 | StartDiskNumber | 2 | 0 |
| 73 | ThisDiskEntries | 2 | 0 |
| 75 | StartDiskEntries | 2 | 1 |
| 77 | Size | 4 | 37 |
| 7B | CDOffset | 4 | 34 |
| 7F | CommentLen | 2 | 0 |
| 81 | Comment | ? | n/a |

```
    +0  +1  +2  +3  +4  +5  +6  +7  +8  +9  +A  +B  +C  +D  +E  +F
0x   P   K 03 04  0A  00  00  00  00  00  00  00  00  00  DD  DD
1x  14  7D  0D  00  00  00  0D  00  00  00  09  00  00  00   h   e
2x   l   l   o   .   t   x   t   H   e   l   l   o   \   W   o   r
3x   l   d   !  \n

    +0  +1  +2  +3  +4  +5  +6  +7  +8  +9  +A  +B  +C  +D  +E  +F
3x                   P   K 01 02  00  00  0A  00  00  00  00  00
4x  00  00  00  00  DD  DD  14  7D  0D  00  00  00  0D  00  00  00
5x  09  00  00  00  00  00  00  00  00  00  00  00  00  00 00 00>
6x <00 00   h   e   l   l   o   .   t   x   t

    +0  +1  +2  +3  +4  +5  +6  +7  +8  +9  +A  +B  +C  +D  +E  +F
6x                                           P   K 05 06  00
7x  00  00  00  00  00  01  00 37 00 00 00 34 00 00 00  00
8x  00

    +0  +1  +2  +3  +4  +5  +6  +7  +8  +9  +A  +B  +C  +D  +E  +F
```

# A bottom-up chain: EoCD -> [CD] -> [LFH]

### 3. Local File Header

| | | | |
|---|---|---|---|
| 00 | Signature | 4 | PK\3\4 |
| 04 | NeededVersion | 2 | 10 |
| 06 | Flags | 2 | None |
| 08 | CompMethod | 2 | 0=Store |
| 20A | ModTime | 2 | 00:00 |
| 20C | ModDate | 2 | 0/0/1980 |
| 0E | CRC32 | 4 | 0x7D14DDDD |
| 12 | CompressSize | 4 | 13 |
| 16 | UncompSize | 4 | 13 |
| 1A | FileNameLen | 2 | 9 |
| 1C | ExtraFieldLen | 2 | 0 |
| 1E | FileName | ? | hello.txt |
| 27 | ExtraField | ? | n/a |
| 27 | Content | ? | Hello World\n |

### 2. Central Directory

| | | | |
|---|---|---|---|
| 34 | Signature | 4 | PK\1\2 |
| 38 | MadeVersion | 2 | 0 |
| 3A | NeededVersion | 2 | 10 |
| 3C | Flags | 2 | None |
| 3E | CompMethod | 2 | 0=Store |
| 40 | ModTime | 2 | 00:00 |
| 42 | ModDate | 2 | 0/0/1980 |
| 44 | CRC32 | 4 | 0x7D14DDDD |
| 48 | CompressSize | 4 | 13 |
| 4C | UncompSize | 4 | 13 |
| 50 | FileNameLen | 2 | 9 |
| 52 | ExtraFieldLen | 2 | 0 |
| 54 | FileCommentLen | 2 | 0 |
| 56 | DiskNumberStart | 2 | 0 |
| 58 | InternalAttr | 2 | 0 |
| 5A | ExternalAttr | 4 | 0 |
| 5E | LFHOffset | 4 | 0 |
| 62 | FileName | ? | hello.txt |
| 6A | ExtraField | ? | n/a |
| 6A | FileComment | ? | n/a |

### 1. End of Central Directory

| | | | |
|---|---|---|---|
| 6C | Signature | 4 | PK\5\6 |
| 6E | ThisDiskNumber | 2 | 0 |
| 71 | StartDiskNumber | 2 | 0 |
| 73 | ThisDiskEntries | 2 | 0 |
| 75 | StartDiskEntries | 2 | 1 |
| 77 | Size | 4 | 37 |
| 7B | CDOffset | 4 | 34 |
| 7F | CommentLen | 2 | 0 |
| 81 | Comment | ? | n/a |

# ZIP PARSING METHODS

LFHs are written first, then CDs, then EoCD at the end.

Existing algorithms in the wild:
- Locate EoCD from the last 64kb, Parse CDs, Parse LFHs.
- Locate EoCD from the end, Parse CDs, Parse LFHs.
- Parse LFHs (they're at the top of the file).

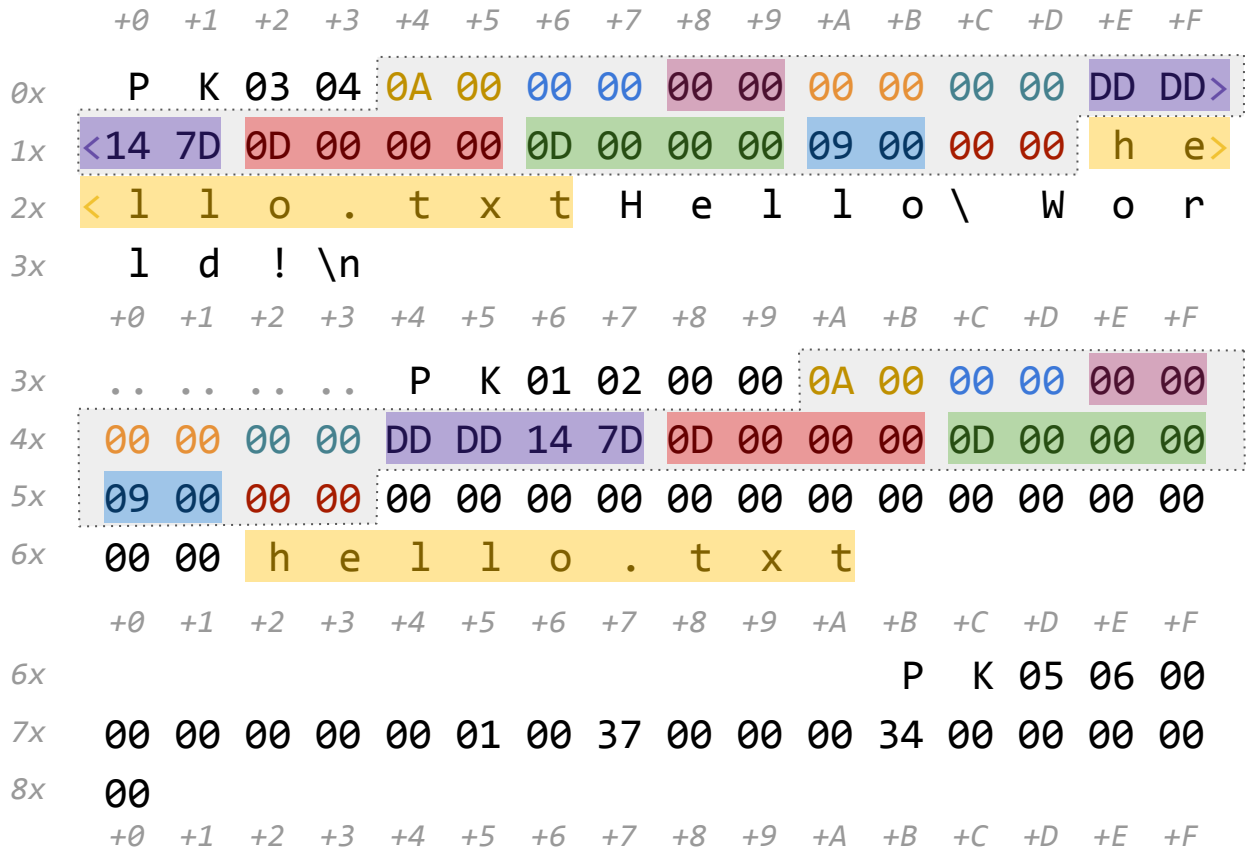
-> Can't abuse ZIP structure and stay fully compatible.

# Arbitrary Zip collision?

Zip parsers are tolerant,
but the EoCD is parsed in the last 64kb.

If the file size difference exceeds this limit,
one file will not be valid: *EoCD not found*.

Bottom-up formats are naturally "collision resistant".




## Local File Header

| | | | |
|---|---|---|---|
| 00 | Signature | 4 | PK\3\4 |
| 04 | Needed Version | 2 | 10 |
| 06 | Flags | 2 | None |
| 08 | CompMethod | 2 | 0=Store |
| 0A | ModTime | 2 | 00:00 |
| 0C | ModDate | 2 | 0/0/1980 |
| 0E | CRC32 | 4 | 0x7D14DDDD |
| 12 | CompressSize | 4 | 13 |
| 16 | UncompSize | 4 | 13 |
| 1A | FileNameLen | 2 | 9 |
| 1C | ExtraFieldLen | 2 | 0 |
| 1E | FileName | ? | hello.txt |
| 27 | ExtraField | ? | n/a |
| 27 | Content | ? | Hello World\n |

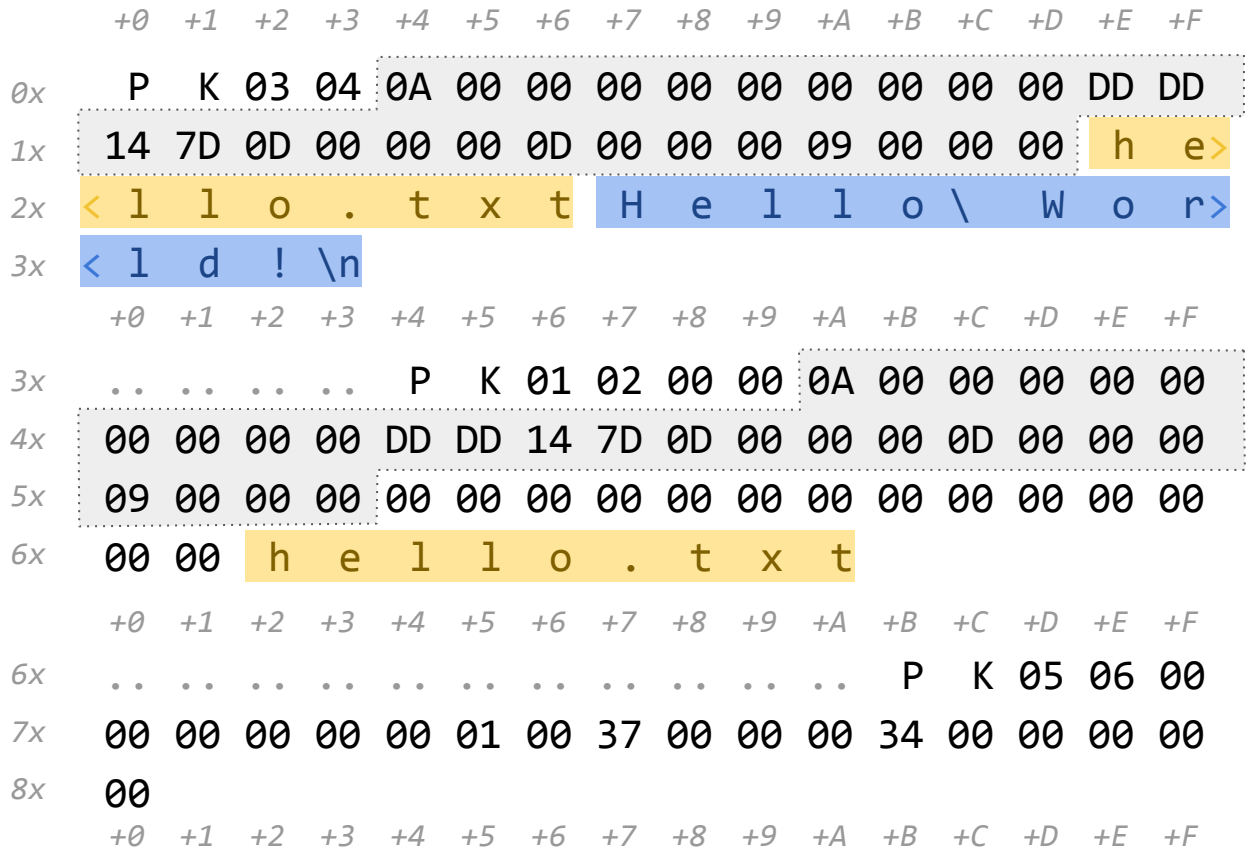
## Central Directory

| | | | |
|---|---|---|---|
| 34 | Signature | 4 | PK\1\2 |
| 38 | MadeVersion | 2 | 0 |
| 3A | Needed Version | 2 | 10 |
| 3C | Flags | 2 | None |
| 3E | CompMethod | 2 | 0=Store |
| 40 | ModTime | 2 | 00:00 |
| 42 | ModDate | 2 | 0/0/1980 |
| 44 | CRC32 | 4 | 0x7D14DDDD |
| 48 | CompressSize | 4 | 13 |
| 4C | UncompSize | 4 | 13 |
| 50 | FileNameLen | 2 | 9 |
| 52 | ExtraFieldLen | 2 | 0 |
| 54 | FileCommentLen | 2 | 0 |
| 56 | DiskNumberStart | 2 | 0 |
| 58 | InternalAttr | 2 | 0 |
| 5A | ExternalAttr | 4 | 0 |
| 5E | LFHOffset | 4 | 0 |
| 62 | FileName | ? | hello.txt |
| 6A | ExtraField | ? | n/a |
| 6A | FileComment | ? | n/a |

## End of Central Directory

| | | | |
|---|---|---|---|
| 6C | Signature | 4 | PK\5\6 |
| 6E | ThisDiskNumber | 2 | 0 |
| 71 | StartDiskNumber | 2 | 0 |
| 73 | ThisDiskEntries | 2 | 0 |
| 75 | StartDiskEntries | 2 | 1 |
| 77 | Size | 4 | 37 |
| 7B | CDOffset | 4 | 34 |
| 7F | CommentLen | 2 | 0 |
| 81 | Comment | ? | n/a |

# A LOT OF DATA IS DUPLICATED




## Local File Header

| | | | |
|---|---|---|---|
| 00 | Signature | 4 | PK\3\4 |
| 04 | NeededVersion | 2 | 10 |
| 06 | Flags | 2 | None |
| 08 | CompMethod | 2 | 0=Store |
| 0A | ModTime | 2 | 00:00 |
| 0C | ModDate | 2 | 0/0/1980 |
| 0E | CRC32 | 4 | 0x7D14DDDD |
| 12 | CompressSize | 4 | 13 |
| 16 | UncompSize | 4 | 13 |
| 1A | FileNameLen | 2 | 9 |
| 1C | ExtraFieldLen | 2 | 0 |
| 1E | FileName | ? | hello.txt |
| 27 | ExtraField | ? | n/a |
| 27 | Content | ? | Hello World\n |

## Central Directory

| | | | |
|---|---|---|---|
| 34 | Signature | 4 | PK\1\2 |
| 38 | MadeVersion | 2 | 0 |
| 3A | NeededVersion | 2 | 10 |
| 3C | Flags | 2 | None |
| 3E | CompMethod | 2 | 0=Store |
| 40 | ModTime | 2 | 00:00 |
| 42 | ModDate | 2 | 0/0/1980 |
| 44 | CRC32 | 4 | 0x7D14DDDD |
| 48 | CompressSize | 4 | 13 |
| 4C | UncompSize | 4 | 13 |
| 50 | FileNameLen | 2 | 9 |
| 52 | ExtraFieldLen | 2 | 0 |
| 54 | FileCommentLen | 2 | 0 |
| 56 | DiskNumberStart | 2 | 0 |
| 58 | InternalAttr | 2 | 0 |
| 5A | ExternalAttr | 4 | 0 |
| 5E | LFHOffset | 4 | 0 |
| 62 | FileName | ? | hello.txt |
| 6A | ExtraField | ? | n/a |
| 6A | FileComment | ? | n/a |

## End of Central Directory

| | | | |
|---|---|---|---|
| 6C | Signature | 4 | PK\5\6 |
| 6E | ThisDiskNumber | 2 | 0 |
| 71 | StartDiskNumber | 2 | 0 |
| 73 | ThisDiskEntries | 2 | 0 |
| 75 | StartDiskEntries | 2 | 1 |
| 77 | Size | 4 | 37 |
| 7B | CDOffset | 4 | 34 |
| 7F | CommentLen | 2 | 0 |
| 81 | Comment | ? | n/a |

# File content is stored between the 2 copies

# Data is duplicated

Before and after compressed data:

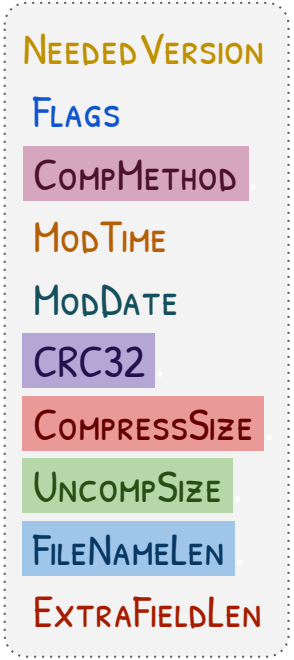
-> prevents generic hash collisions

For maximum compatibility,

these fields have to be:

- Set in both headers
- Constant across colliding files

Zip structure prevent generic reuse.



# What's a Docx file ?

An archive of several files with subdirectories.

XML, PNG, JPG...

A root file: _rels/.rels, pointing to the main doc file.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Id="rId3" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/extended-properties" Target="docProps/app.xml"/>
    <Relationship Id="rId2" Type="http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties" Target="docProps/core.xml"/>
    <Relationship Id="rId1" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument" Target="word/document.xml"/>
</Relationships>
```

# Abusing the document structure

- ✔ Make 2 documents co-exist in the same archive.
- ✔ Point to each document via the root.
- ✔ Constant Root file length.
- ? Hide collision blocks in a compatible way (without CRC dependency).
- ? Constant Root file CRC.

# .XML collisions?

Comments are defined, but encoding is enforced.

All collisions produce blocks with a high entropy

```
000  4D C9 68 FF 0E E3 5C 20 95 72 D4 77 7B 72 15 87  M╔h ▀ π\ òr╘w{r ç
010  D3 6F A7 B2 1B DC 56 B7 4A 3D C0 78 3E 7B 95 18  ╙o° ▒ ▄V╖J=└x>{ò
020  AF BF A2 00 A8 28 4B F3 6E 8E 4B 55 B3 5F 42 75  »¬ ó ¿(K≤nÄKU│_Bu
030  93 D8 49 67 6D A0 D1 55 5D 83 60 FB 5F 07 FE A2  ô╪Igmá╤U]â`√_ ▪ó
```

```
000  4D C9 68 FF 0E E3 5C 20 95 72 D4 77 7B 72 15 87  M╔h ▀ π\ òr╘w{r ç
010  D3 6F A7 B2 1B DC 56 B7 4A 3D C0 78 3E 7B 95 18  ╙o° ▒ ▄V╖J=└x>{ò
020  AF BF A2 02 A8 28 4B F3 6E 8E 4B 55 B3 5F 42 75  »¬ ó ¿(K≤nÄKU│_Bu
030  93 D8 49 67 6D A0 D1 D5 5D 83 60 FB 5F 07 FE A2  ô╪Igmá╤╒]â`√_ ▪ó
```

Even the simplest collision (single block) has a lot of non-ASCII characters.

-> No collision blocks can be stored in a valid XML file.

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
<![CDATA[comment]]>
</tag>
```

<- Abusing this...

...will trigger this ->

**This page contains the following errors:**

error on line 4 at column 10: Encoding error

**Below is a rendering of the page up to the first error.**

https://www.w3.org/TR/REC-xml/#sec-cdata-sect
https://marc-stevens.nl/research/md5-1block-collision/md5-1block-collision.pdf

# In another archived file ?

✔ Constant length of the blocks

- Files' CRC has to be present after the data too.

-> Use a dummy file

Extra Field is stored **before** file contents

and store the contents in the "Extra Field" (No CRC)

✔ Hide collision blocks in a compatible way.

-> Declare dummy file in `[Content_Types].xml`

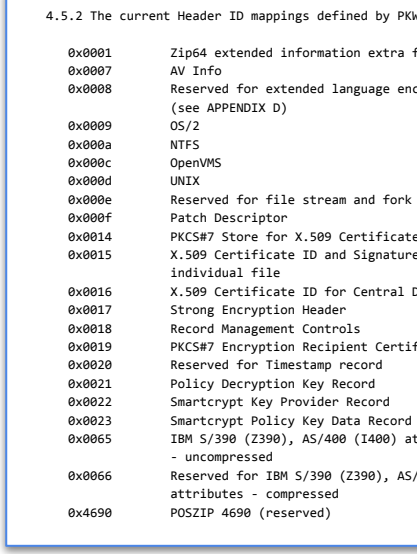
# Extra Field in ZIP

Standard: defined in LFHs since v1.0 in 1990.
Commonly used.

Extends the format for all kinds of use.

Each field uses an ID. Unsupported IDs are just ignored.

-> Perfect for our use case.



```
4.5.2 The current Header ID mappings defined by PKW

    0x0001        Zip64 extended information extra f
    0x0007        AV Info
    0x0008        Reserved for extended language enc
                  (see APPENDIX D)
    0x0009        OS/2
    0x000a        NTFS
    0x000c        OpenVMS
    0x000d        UNIX
    0x000e        Reserved for file stream and fork
    0x000f        Patch Descriptor
    0x0014        PKCS#7 Store for X.509 Certificate
    0x0015        X.509 Certificate ID and Signature
                  individual file
    0x0016        X.509 Certificate ID for Central D
    0x0017        Strong Encryption Header
    0x0018        Record Management Controls
    0x0019        PKCS#7 Encryption Recipient Certif
    0x0020        Reserved for Timestamp record
    0x0021        Policy Decryption Key Record
    0x0022        Smartcrypt Key Provider Record
    0x0023        Smartcrypt Policy Key Data Record
    0x0065        IBM S/390 (Z390), AS/400 (I400) at
                  - uncompressed
    0x0066        Reserved for IBM S/390 (Z390), AS/
                  attributes - compressed
    0x4690        POSZIP 4690 (reserved)
```

https://pkware.cachefly.net/webdocs/APPNOTE/APPNOTE-1.0.txt
https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT

# Constant CRC for the root file

Bruteforced CRC 💥 enforced encoding

CRChack by *resilar* (public domain)
specify the bits, forge a CRC - in 0.3s

Via 6 ASCII characters

```
$ cat ASCII
<!--ABCDEF-->
$ crchack -b 4.0:+.8*6:1 \
          -b 4.1:+.8*6:1 \
          -b 4.2:+.8*6:1 \
          -b 4.3:+.8*6:1 \
          -b 4.4:+.8*6:1 \
          -b 4.5:+.8*2:1 \
          ASCII 0xdeadf00d
<!--tuI_\Y-->
```

Via 32 letters

```
$ cat CASE
<!--THESEKINDSOFCRCAREVERYIMPRESSIVE-->
$ crchack -b 4.5:+.8*32:.8 CASE 0xcafebabe
<!--thEsEKIndsOFcRcAReVEryiMPREssIVe-->
```

https://github.com/resilar/crchack

# XML + CRCHack

Pair of different root files

with constant size and CRC, ASCII-only

Perfect for generic ZIP collision. 👌

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Id="rId3" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/extended-properties" Target="docProps/app.xml"/>
    <Relationship Id="rId2" Type="http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties" Target="docProps/core.xml"/>
    <Relationship Id="rId1" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument" Target="word1/document.xml"/>
    <!-- aAAaaAAaAAAaaaAaAaaAAaAaAaaaaaaA -->
</Relationships>
```

Same CRC
0xCAFEBABE

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Id="rId3" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/extended-properties" Target="docProps/app.xml"/>
    <Relationship Id="rId2" Type="http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties" Target="docProps/core.xml"/>
    <Relationship Id="rId1" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument" Target="word2/document.xml"/>
    <!-- bbBbbbbBbBBBBbBBBbbbbbBBBBbbBBBb -->
</Relationships>
```

# Collision ~~prefixes~~ pre-archives !?

```
$ md5sum docx*zip
6c33d52590ff0bb0cc8cdafe6aa5153b *docx1.zip
6c33d52590ff0bb0cc8cdafe6aa5153b *docx2.zip
$ zip -oXll docx1.zip zinsider.py
  adding: zinsider.py (deflated 64%)
$ zip -oXll docx2.zip zinsider.py
  adding: zinsider.py (deflated 64%)
$ md5sum docx*zip
d12044feee801ad0530a911fa7f18db5 *docx1.zip
d12044feee801ad0530a911fa7f18db5 *docx2.zip
$ zip -d docx1.zip zinsider.py
deleting: zinsider.py
$ zip -d docx2.zip zinsider.py
deleting: zinsider.py
$ md5sum docx*zip
6c33d52590ff0bb0cc8cdafe6aa5153b *docx1.zip
6c33d52590ff0bb0cc8cdafe6aa5153b *docx2.zip
```

```
CLI options:
-d  --delete
-ll --from-crlf
-o  --latest-time
-X  --strip-extra
```

Typically, a prefix is an invalid file:
a header without a body.

These prefixes can be used as valid archives:
MD5 equality is maintained with identical operations
(just be cautious with timestamps).

-> reproducible collision PoCs via standard tools !

https://github.com/corkami/collisions/tree/397e1f0504dc4301a4d122017d2f66068bb7730c/scripts

# Zinsider

(Python, MIT licence)

Combines a pair of ZIP(XML) format.

Requires a pair of pre-computed prefix for each format.

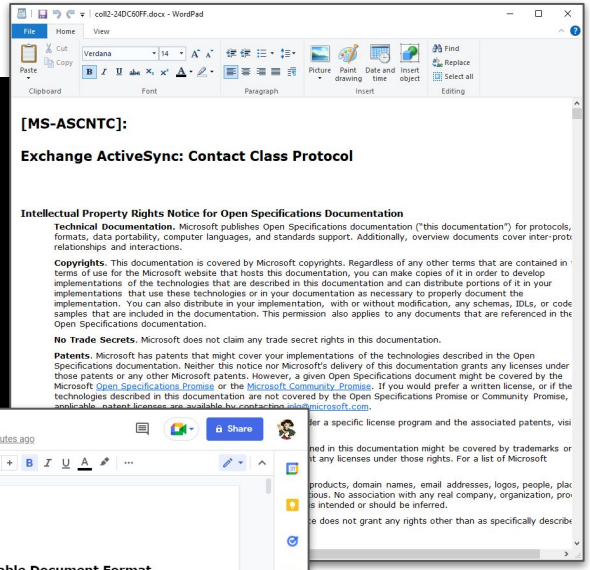
No special setting.

Instant reusable collision.

```
zinsider.py -h
usage: zinsider.py [-h] file1 file2

Generate MD5 collisions of zip+xml file formats.

positional arguments:
  file1       First input file.
  file2       Second input file.

optional arguments:
  -h, --help  show this help message and exit
```

https://github.com/corkami/collisions/blob/master/scripts/zinsider.py

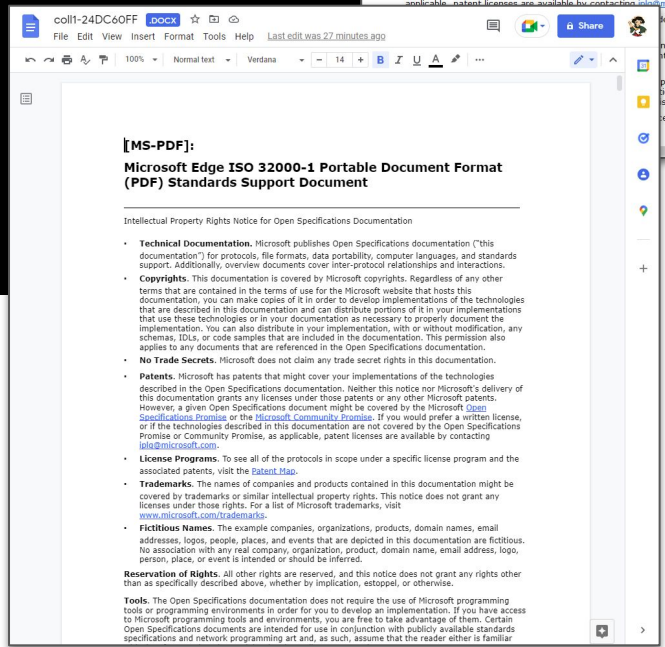
```
$ time ./zinsider.py "[MS-PDF]-180828.docx" "[MS-ASCNTC]-220429.docx"
Common file type: docx
Merging archived files
Copying content types
Merging content types
Adding collision block exclusion
Merging suffix with prefix pair
Suffix: 39 file(s)
Verifying and saving
Common md5: 24dc60ff914906c08897a3f1dbe9bdcb
Success!

real  0m0.164s
user  0m0.132s
sys   0m0.036s
```


coll2-24DC60FF.docx - WordPad

[MS-ASCNTC]:

Exchange ActiveSync: Contact Class Protocol

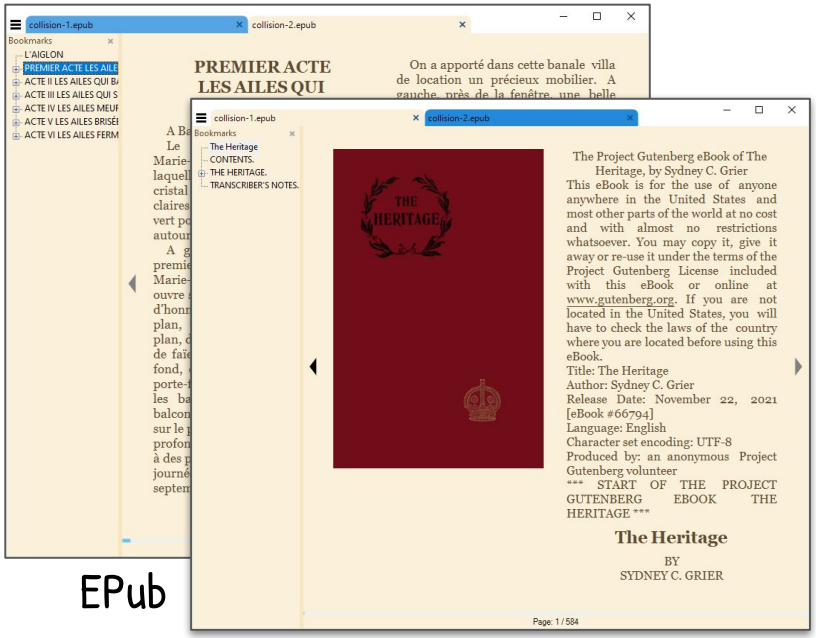
Intellectual Property Rights Notice for Open Specifications Documentation


coll1-24DC60FF .DOCX

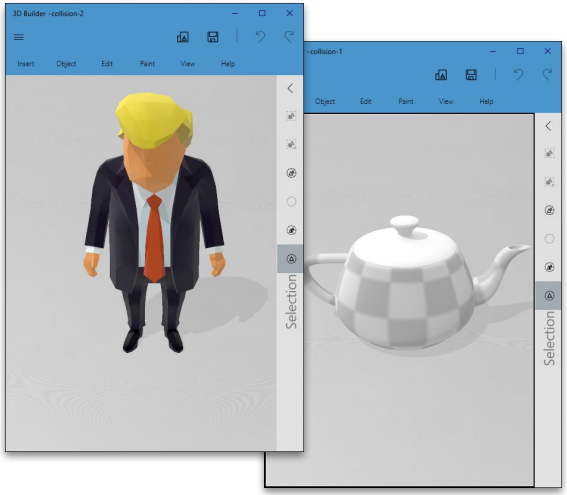
[MS-PDF]:

Microsoft Edge ISO 32000-1 Portable Document Format (PDF) Standards Support Document

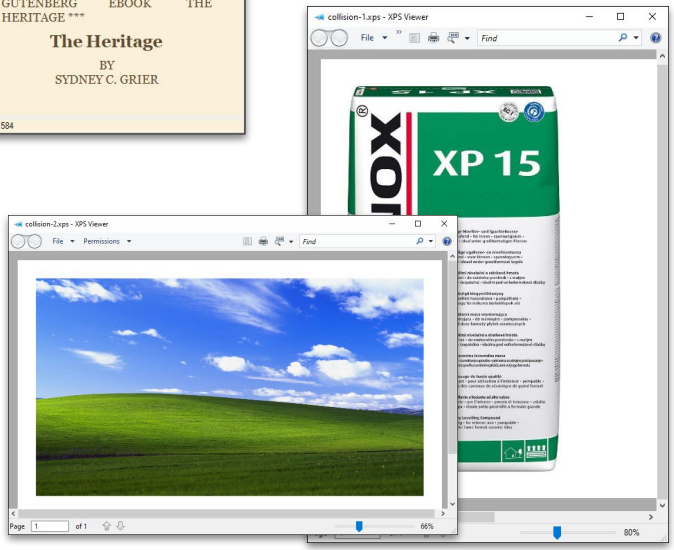
Intellectual Property Rights Notice for Open Specifications Documentation




EPub




3D Manufacturing Format
(open source standard)




XML
Paper
Specification

# Other formats

# Supported formats

- Office Open XML: docx / pptx / xlsx
- Open Container Format: epub
- Open Packaging Conventions:
  - 3D manufacturing format: 3mf
    XML Paper Specification: xps / oxps

Extensible to other ZIP(Root.xml) format.
Requires a pre-computed prefix pair.

# Unsupported ZIP(XML) formats

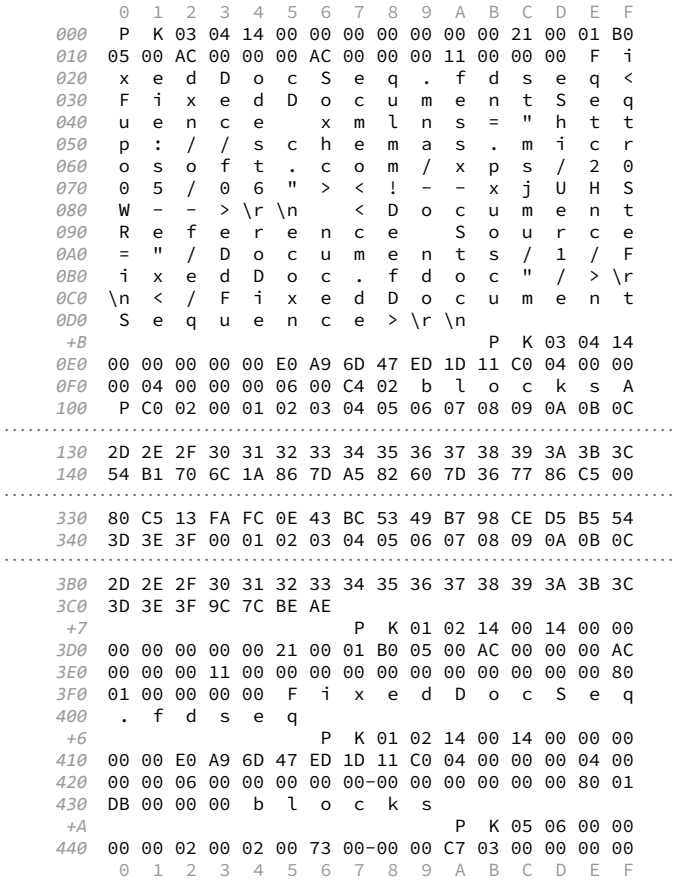
Quake PK3: no root file to abuse.

Open Document Format:

META-INF/manifest.xml has to mention every other file.

-> not generic.

APK, JAR, XPI: like ODF, but also with files' hashes !!

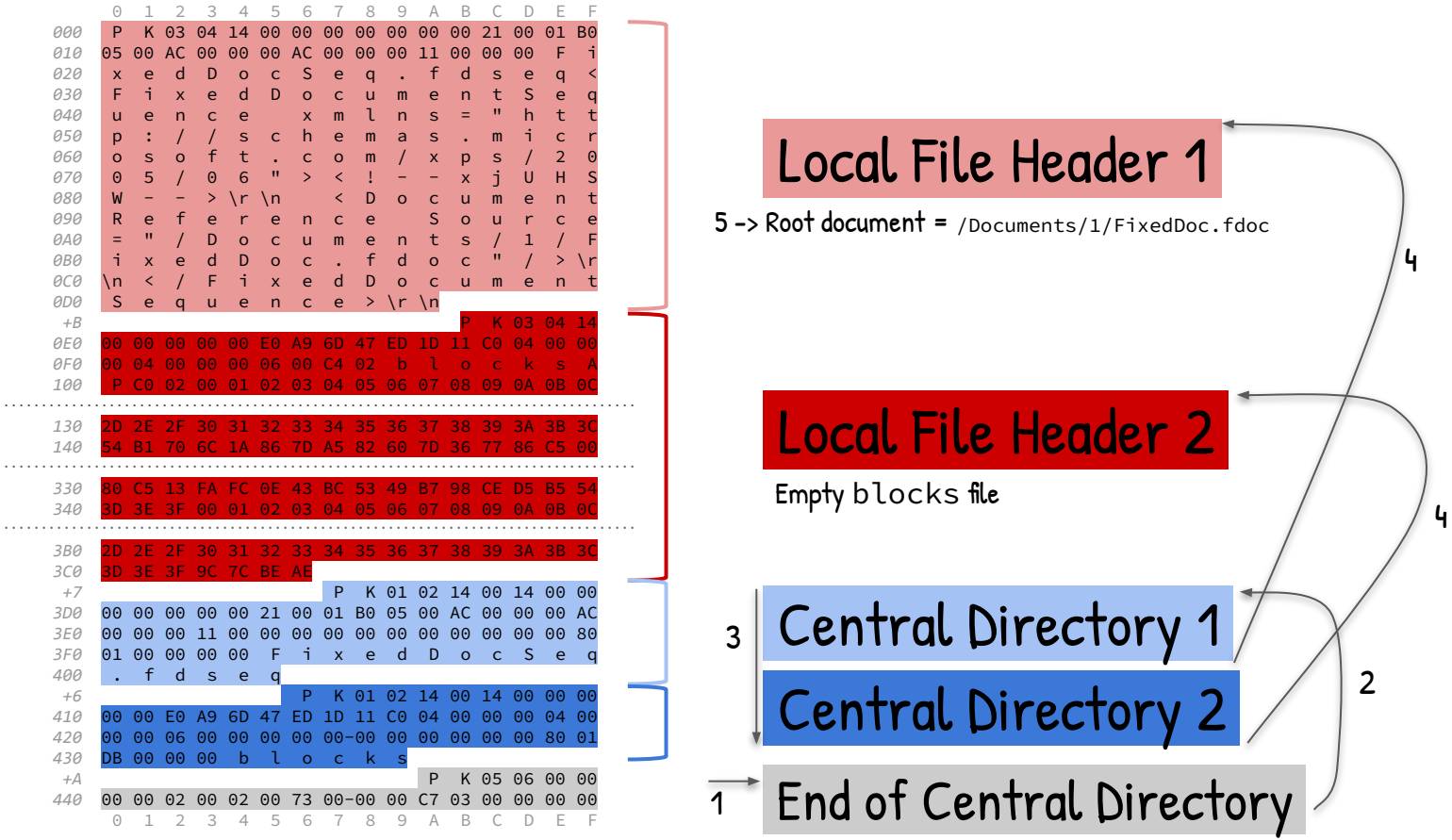



A ZIP archive containing a root XML file.

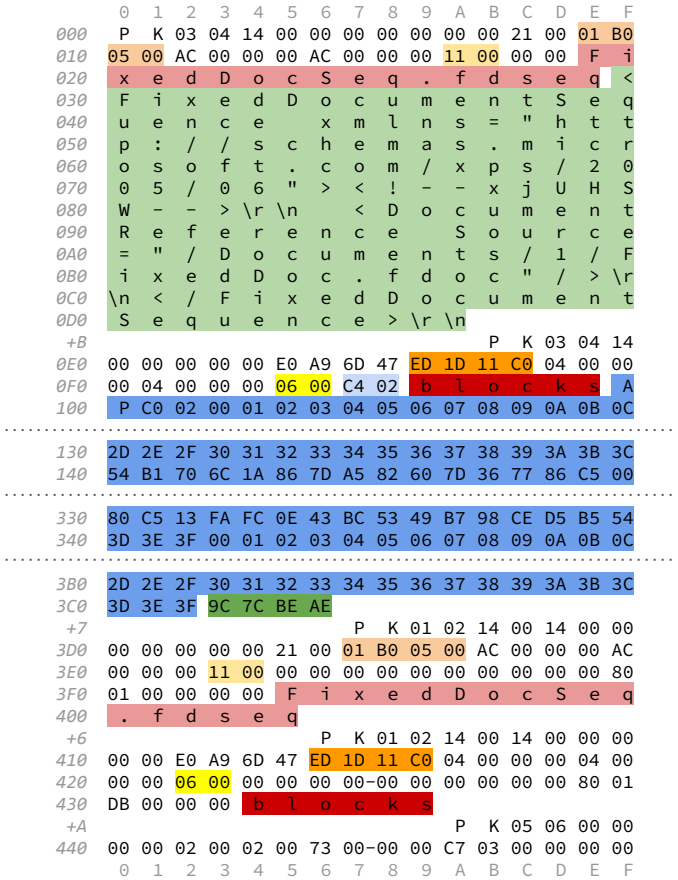
Slightly different XML content.

Same MD5.

# Overview of a Zinsider pre-archive




# Bottom-up parsing flow




File 1: FixedDocSeq.fdseq

File name (length)

File contents CRC32

File 2: blocks

File name (length)

Extra fields (length)

File contents CRC32

Duplicated: file names, contents size, CRC32.

Extra fields have no CRC32.

# Structure

```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
000  P  K 03 04 14 00 00 00 00 00 00 00 21 00 01 B0
010 05 00 AC 00 00 00 AC 00 00 00 11 00 00 00  F  i
020  x  e  d  D  o  c  S  e  q  .  f  d  s  e  q  <
030  F  i  x  e  d  D  o  c  u  m  e  n  t  S  e  q
040  u  e  n  c  e     x  m  l  n  s  =  "  h  t  t
050  p  :  /  /  s  c  h  e  m  a  s  .  m  i  c  r
060  o  s  o  f  t  .  c  o  m  /  x  p  s  /  2  0
070  0  5  /  0  6  "  >  <  !  -  -  x  j  U  H  S
080  W  -  -  >  \r \n     <  D  o  c  u  m  e  n  t
090  R  e  f  e  r  e  n  c  e     S  o  u  r  c  e
0A0  =  "  /  D  o  c  u  m  e  n  t  s  /  1  /  F
0B0  i  x  e  d  D  o  c  .  f  d  o  c  "  /  >  \r
0C0 \n  <  /  F  i  x  e  d  D  o  c  u  m  e  n  t
0D0  S  e  q  u  e  n  c  e  >  \r \n
 +B                                    P  K 03 04 14
0E0 00 00 00 00 00 E0 A9 6D 47 ED 1D 11 C0 04 00 00
0F0 00 04 00 00 00 06 00 C4 02  b  l  o  c  k  s  A
100  P C0 02 00 01 02 03 04 05 06 07 08 09 0A 0B 0C
.....
130 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C
140 54 B1 70 6C 1A 86 7D A5 82 60 7D 36 77 86 C5 00
.....
330 80 C5 13 FA FC 0E 43 BC 53 49 B7 98 CE D5 B5 54
340 3D 3E 3F 00 01 02 03 04 05 06 07 08 09 0A 0B 0C
.....
3B0 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C
3C0 3D 3E 3F 9C 7C BE AE
 +7                       P  K 01 02 14 00 14 00 00
3D0 00 00 00 00 00 21 00 01 B0 05 00 AC 00 00 00 AC
3E0 00 00 00 11 00 00 00 00 00 00 00 00 00 00 00 80
3F0 01 00 00 00 00  F  i  x  e  d  D  o  c  S  e  q
400  .  f  d  s  e  q
 +6                    P  K 01 02 14 00 14 00 00 00
410 00 00 E0 A9 6D 47 ED 1D 11 C0 04 00 00 00 04 00
420 00 00 06 00 00 00 00 00-00 00 00 00 00 00 80 01
430 DB 00 00 00  b  l  o  c  k  s
 +A                                P  K 05 06 00 00
440 00 00 02 00 02 00 73 00-00 00 C7 03 00 00 00 00
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
```

Prefix (with **differences**)

<- CRC32 manipulation

| x | j | U | H | S | W |
|---|---|---|---|---|---|
| W | u | A | ^ | Q | A |

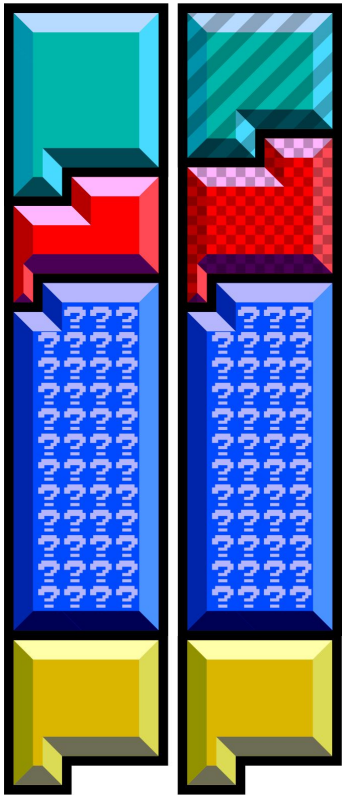
<- change XML root path 1

2

Padding

Collision blocks

Suffix



Root file: constant CRC, length.

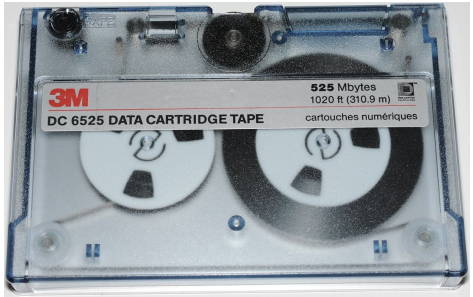
Collision file: constant MD5, no content change

(blocks are stored in Extra Field)

# Collision structure

# Abusing TAR.GZ archives

# TAR archive




a 3M QIC tape (525 Mb)

"Tape Archive" (1979)

A sequence of file header + file contents
(no compression).

Everything is aligned to 512-byte blocks.

2 empty blocks of 512 bytes at the end
(not enforced, but it makes any appended data ignored).

# a TAR file

- hardcoded offsets
- integers in octal

```
      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
00x   h  e  l  l  o  .  t  x  t  00 00 00 00 00 00 00
...   [...]
06x   00 00 00 00 0  0  0  6  4  4  00 00 00 00 00 00
07x   00 00 00 00 00 00 00 00 00 00 00 00 0  0  0  0
08x   0  0  0  0  0  0  3  00 1  3  6  4  4  3  3  3
09x   4  2  2  00 0  0  0  6  3  2  5  00 30 00 00 00
...   [...]
10x   00 u  s  t  a  r  20 20 00 00 00 00 00 00 00 00

1Fx   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20x   <file contents>
```

Filename

File mode

File size

Timestamp

Checksum

Magic

Header starts with filename (exploitable, but tiny)

Magic is at offset 0x101, and is enforced.

Header checksum is **enforced**.

The end of the header should be empty. (cf libmagic)

https://github.com/file/file/blob/master/magic/Magdir/archive#L13

# Collision and TAR

Top-down format with appended data.
-> Compatible with chosen-prefix collisions.

No supported comments, hardcoded offsets.
-> no reusable collisions.

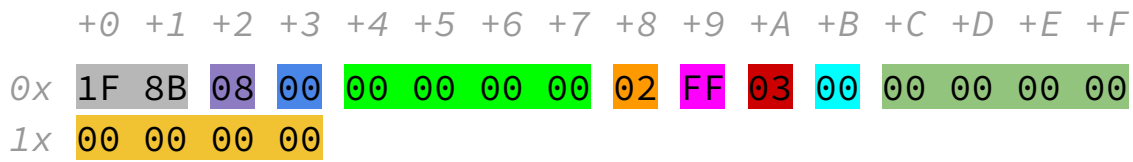
# What's a TAR.GZ file?

A TAR archive in a GZIP.

The TAR ignores what's happening at the GZIP layer.

Abusing the GZIP won't interfere
as long as the TAR is decompressed fine..

# A minimal GZIP archive




Magic 1F 8B

Method 8 = Deflate

Flags Filename, Extra Field...

ModTime Flags

Extra Flags

OS

Deflate data:

- Last Block Set
- Length No block content

CRC32

lenUncomp

This archive is empty.

Compression method is always 08 (Deflate),
so the minimal data is 03 00.

# Notes on GZIP

GZIP only uses Deflate. Unlike ZIP, it cannot store file contents as-is.
A Deflate non-compressed block is at most 64kb.

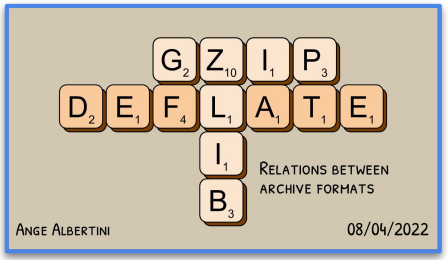
Padding possible with empty non-compressed blocks (always 5 bytes):

`00` `00 00` `FF FF`

Contents can't be skipped.

-> Collision blocks can't be abusing compressed data.

Clarification ->



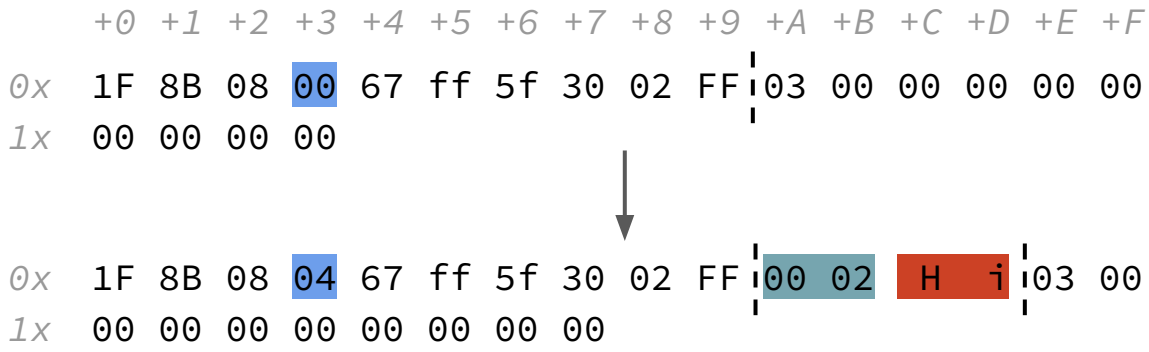

https://speakerdeck.com/ange/gzip-equals-zip-equals-zlib-equals-deflate

# An extra field 1/2

Set bit 4 in the **Flags**.

The Extra Field comes after the OS flag.

It starts with its data **Length**, then its **Data** - no CRC.




Flags

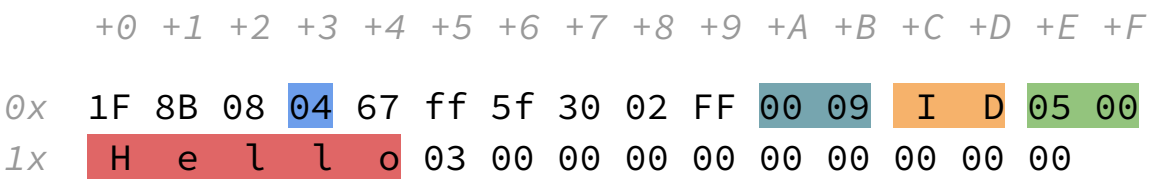
Extra Field:
- Length
- Data

# An extra field 2/2

The Extra Field is supposed to be a sequence of subfields:

Not really enforced!

- ID (2 alphanum chars) Ex: AP = Apollo file type information.
- SubLength
- Data




Flags

Extra field
- Length
- SubFields:
  - ID
  - SubLength
  - Data

# Extra Fields in Gzip

Standard, but rarely used

Single official use case: Apollo Computer (in the 80s)

Notable use: bgzip ("BGZF" blocks)

| SI1 | SI2 | Data |
| --- | --- | --- |
| 0x41 ('A') | 0x70 ('P') | Apollo file type information |

https://en.wikipedia.org/wiki/Apollo_Computer
http://www.htslib.org/doc/bgzip.html
https://www.rfc-editor.org/rfc/rfc1952#page-8

# Collision blocks in Extra Field

Give Extra Field a variable length via collision blocks.
-> get different Deflate data parsed or skipped.

Reusable header, but limited to 64kb length.

It works, but it's limiting.

(This is exactly the same constraint of size of JPEG in the Shattered exploitation).

# What's a GZIP file?

Gzip specs: RFC 1952


The `1F 8B … length` structure is called a "member".

While most GZIP files are made of a single member,

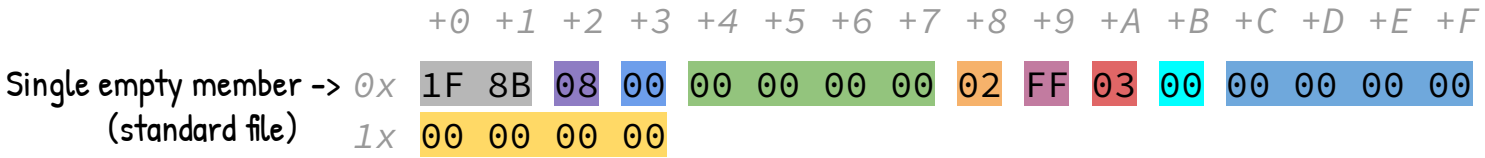
members can be concatenated and
data will be silently decompressed and concatenated.

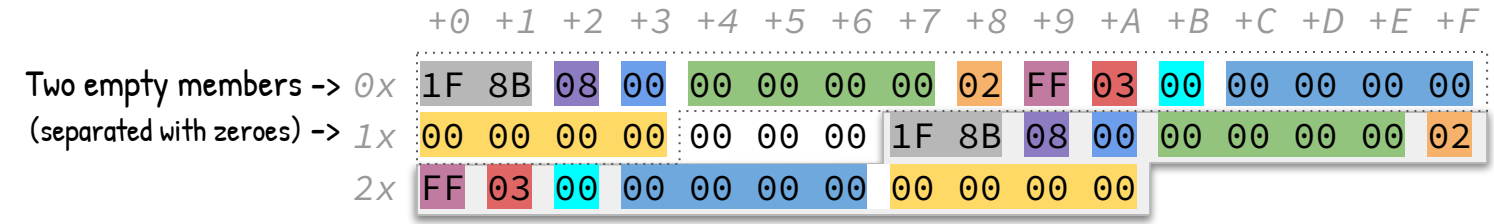
| | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A Gzip member -> 0x | 1F | 8B | 08 | 00 | 00 | 00 | 00 | 00 | 02 | FF | 03 | 00 | 00 | 00 | 00 | 00 |
| 1x | 00 | 00 | 00 | 00 | | | | | | | | | | | | |

Magic
Method
Flags
ModTime
Extra flags
OS

CompData:
- Last Block
- Length
CRC32
lenUncomp

https://datatracker.ietf.org/doc/html/rfc1952




# These 2 files are equivalent (and both empty) 🤔

# Abusing several members

Members may contain empty compressed data,
but still store information via Extra Field.

Unknown types of Extra Field are ignored.

-> empty members are treated like classic "comments".

-> classic collision exploitation is possible.

# 1. "Header"

```
          +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
0x        1F 8B 08 04 00 00 00 00 02 FF
```

Magic
Method
Flags (extra field set)
ModTime
Extra Flags
OS

# 2. "Body"

```
+A        .. .. .. .. .. .. .. .. .. .. 08 00  E  F 00 04
1x        D  a  t  a
```

Extra Field:
- Length
- SubFields:
  - ID
  - SubLength
  - <data>

# 3. "Footer"

Empty data body

```
10+len(Data) .. .. .. .. 03 00 00 00 00 00 00 00 00 00
```

CompData:
- Last Block
- Length

CRC32
lenUncomp

# A very unusual kind of "comment"

# Gzip exploitation

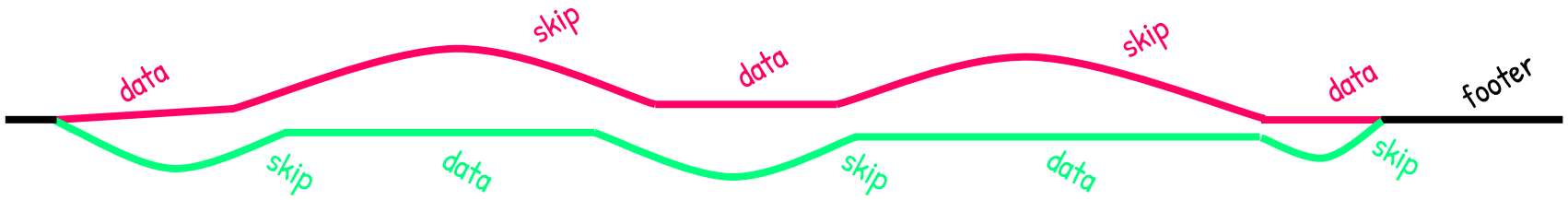
Insert members with no data
as comments to skip other members

Split data in members (members are limited to 64kb).

Alternate data members and skip members.

Make both chains end on a member's footer (to avoid warnings).

-> 2 chains of valid members with different contents.



# Chosen prefix collision?

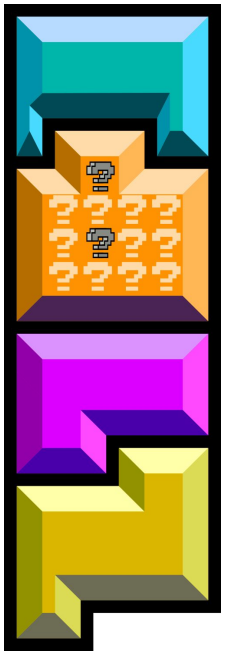
**Unicoll**
+1 on the 10th byte
of the collision block.
Takes a few minutes.

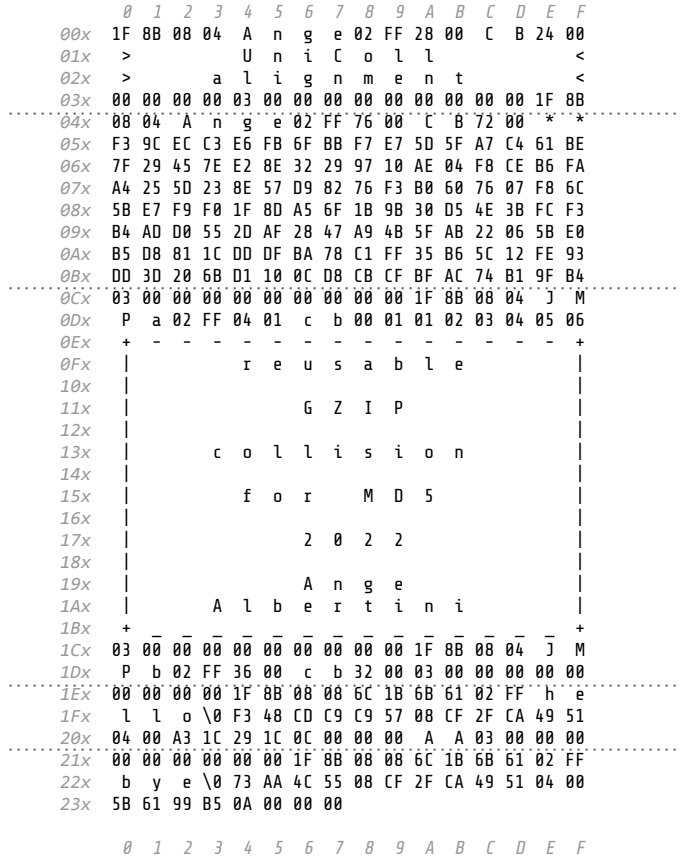
Unicoll can be used:

- Extra Field length is 2 bytes, little endian
- declared before its contents.

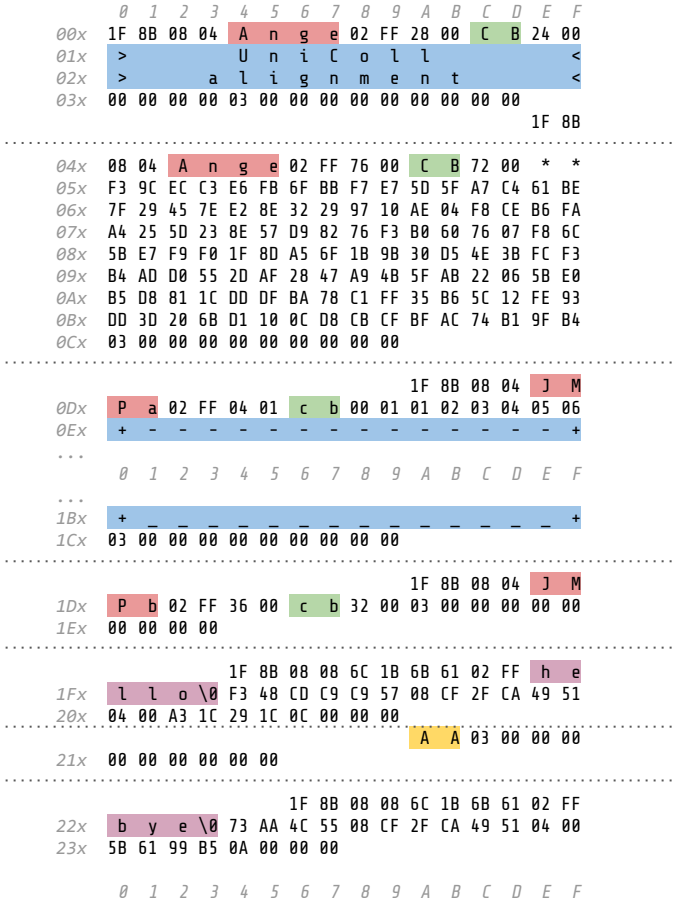
1 member for unicoll alignment.

1 member declared at the start of the Unicoll blocks.






# A complete Unicoll-based GZIP collision




TimeStamp

Extra Field ID

Filling text

File name

Marker

# Role of ASCII strings

```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00x 1F 8B 08 04  A  n  g  e 02 FF 28 00  C  B 24 00
01x  >           U  n  i  C  o  l  l              <
02x  >        a  l  i  g  n  m  e  n  t           <
03x 00 00 00 00 03 00 00 00 00 00 00 00 00 00
                                              1F 8B
.....................................................
04x 08 04  A  n  g  e 02 FF 76 00  C  B 72 00  *  *
05x F3 9C EC C3 E6 FB 6F BB F7 E7 5D 5F A7 C4 61 BE
06x 7F 29 45 7E E2 8E 32 29 97 10 AE 04 F8 CE B6 FA
07x A4 25 5D 23 8E 57 D9 82 76 F3 B0 60 76 07 F8 6C
08x 5B E7 F9 F0 1F 8D A5 6F 1B 9B 30 D5 4E 3B FC F3
09x B4 AD D0 55 2D AF 28 47 A9 4B 5F AB 22 06 5B E0
0Ax B5 D8 81 1C DD DF BA 78 C1 FF 35 B6 5C 12 FE 93
0Bx DD 3D 20 6B D1 10 0C D8 CB CF BF AC 74 B1 9F B4
0Cx 03 00 00 00 00 00 00 00 00 00
.....................................................
                                        1F 8B 08 04  J  M
0Dx  P  a 02 FF 04 01  c  b 00 01 01 02 03 04 05 06
0Ex  +  -  -  -  -  -  -  -  -  -  -  -  -  -  -  +
...
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
...
1Bx  +  _  _  _  _  _  _  _  _  _  _  _  _  _  _  +
1Cx 03 00 00 00 00 00 00 00 00 00
                                        1F 8B 08 04  J  M
1Dx  P  b 02 FF 36 00  c  b 32 00 03 00 00 00 00 00
1Ex 00 00 00 00
.....................................................
                 1F 8B 08 08 6C 1B 6B 61 02 FF  h  e
1Fx  l  l  o \0 F3 48 CD C9 C9 57 08 CF 2F CA 49 51
20x 04 00 A3 1C 29 1C 0C 00 00 00
                                           A  A 03 00 00 00
21x 00 00 00 00 00 00
.....................................................
                    1F 8B 08 08 6C 1B 6B 61 02 FF
22x  b  y  e \0 73 AA 4C 55 08 CF 2F CA 49 51 04 00
23x 5B 61 99 B5 0A 00 00 00

     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
```

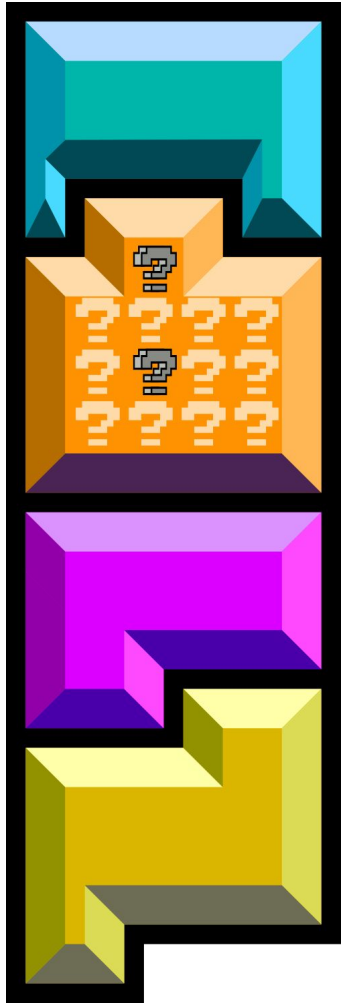
Identical prefix

Unicoll blocks
(with early chosen text)

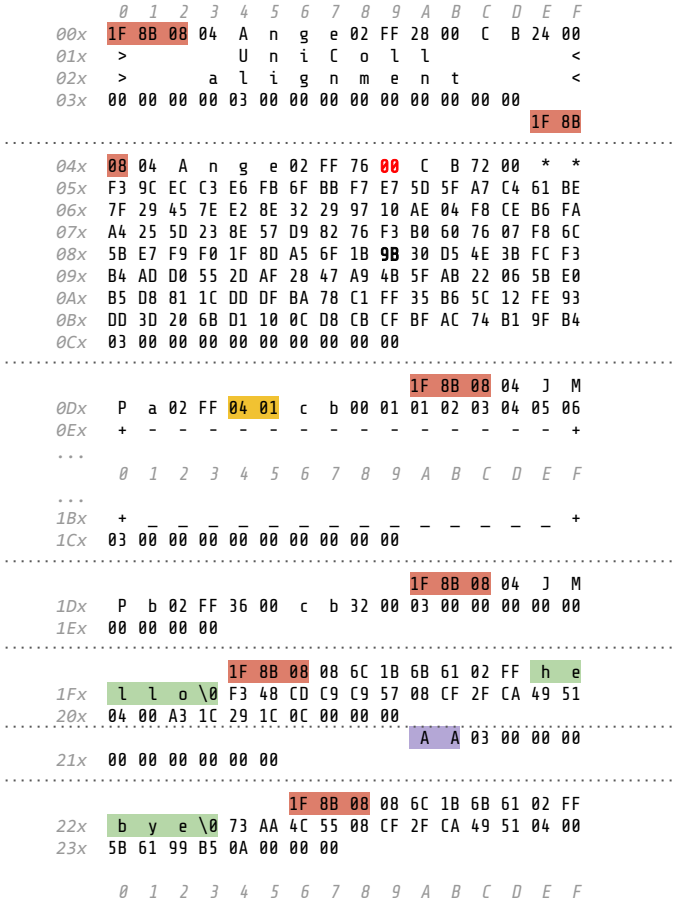
Post-Unicoll trampoline

File 1

File 2



# UniColl structure




Member for UniColl alignment

Member with variable length via Unicoll blocks
Length = 0x0076 / 0x0176

Member to skip over 0x100 bytes (due to UniColl)

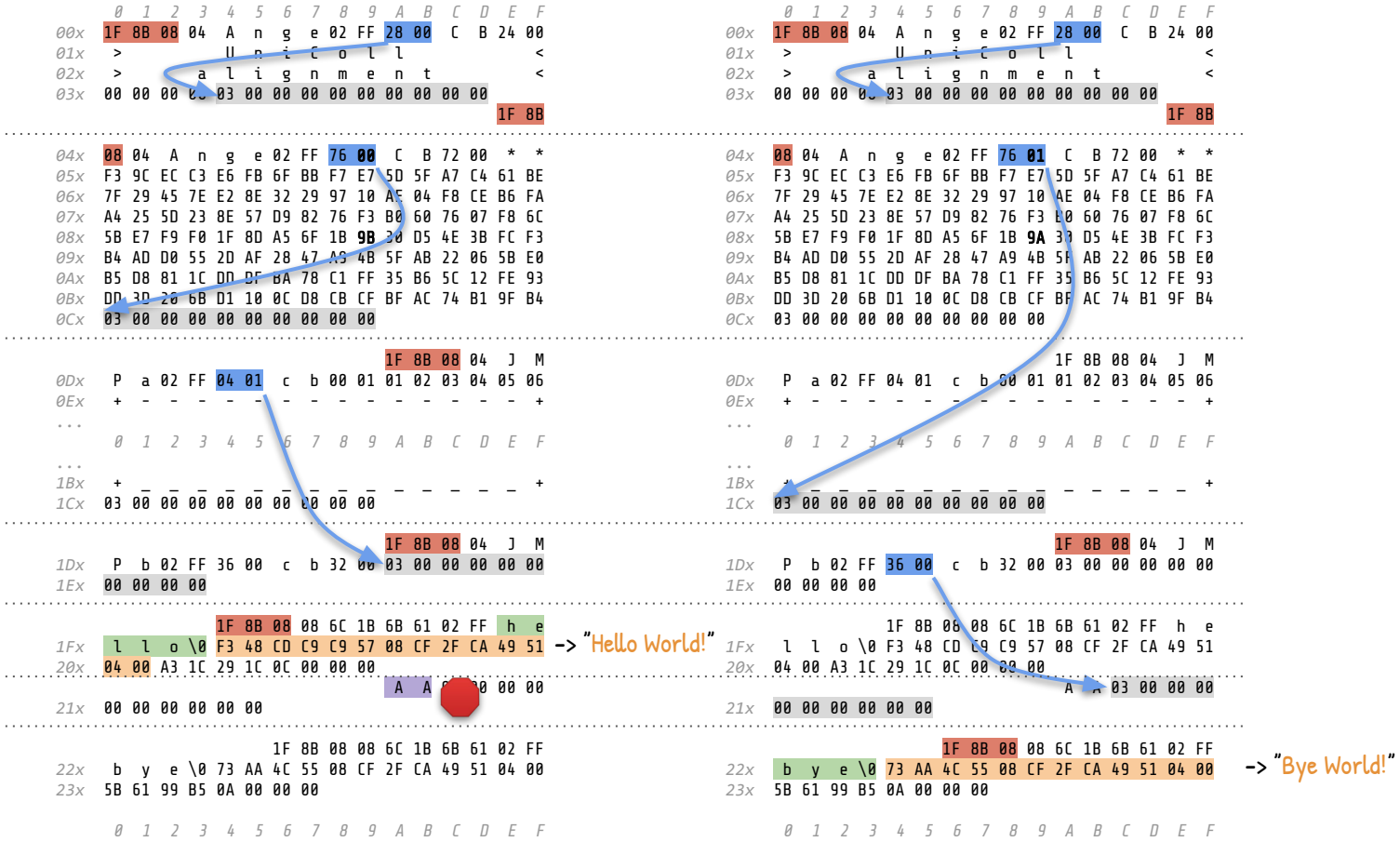
Member to jump over first data member.

Data member ("hello" file containing "Hello World!")

Terminator

Data member (" bye" file containing "Bye World!")

# GZIP STRUCTURE




# Different parsing of colliding GZIP pairs

```
$ ./gz.py libjpeg-turbo-2.1.3.tar.gz tiff-4.4.0rc1.tar.gz
libjpeg-turbo-2.1.3.tar.gz (2260756 bytes): split in 78 members
tiff-4.4.0rc1.tar.gz (2841082 bytes): split in 78 members
Success!
22fb3b1171cc1bb9969b093e77f69e7c
coll-1.gz => libjpeg-turbo-2.1.3.tar.gz
coll-2.gz => tiff-4.4.0rc1.tar.gz
```

Takes 1s…

```
$ tar tvf coll-1.gz
drwxrwxr-x root/root         0 2022-02-25 19:53 libjpeg-turbo-2.1.3/
-rw-rw-r-- root/root     24927 2022-02-25 19:53 libjpeg-turbo-2.1.3/BUILDING.md
[...]
-rw-rw-r-- root/root     10840 2022-02-25 19:53 libjpeg-turbo-2.1.3/wrppm.c
-rw-rw-r-- root/root
```

```
$ tar tvf coll-2.gz
drwxrwxr-x even/even         0 2022-05-20 18:13 tiff-4.4.0/
-rw-rw-r-- even/even      1146 2021-03-05 14:01 tiff-4.4.0/COPYRIGHT
[...]
-rw-rw-r-- even/even      1520 2022-02-19 16:33 tiff-4.4.0/contrib/addtiffo/Makefile.am
-rw-rw-r-- even/even     20907 2022-05-20 18:11 tiff-4.4.0/contrib/addtiffo/Makefile.in
-rw-rw-r-- even/even     33511 2022-05-20 18:11 tiff-4.4.0/Makefile.in
```

# Works with any GZIP pair.

# Conclusion

# From "no collision" to "instant collision"

Instant MD5 colliding pair of arbitrary:

- GZIP, including TAR.GZ and many others.
- ZIP(XML) docs:
  - Office Open XML: DOCX / PPTX / XLSX
  - Open Container Format: EPUB
  - Open Packaging Conventions:
    - 3D manufacturing format: 3MF
    - XML Paper Specification: XPS / OXPS

## Office exploitation

- Abusing root XML document inside the archive.
- Storing collision blocks in dummy file via extra fields for generic reuse.
- dummy file ignored via content types.
- Keeping length and CRC constants for generic reuse.
- Merge of 2 documents in different paths.

Same archive, 2 different root files, with both sets of files together.

## TAR.GZ exploitation

- Abusing GZIP structure to deliver different TAR archives.
- Abusing empty members as comments with data in extra field.
- interleaving archives contents via two chains of skip+data link.

2 different archives of independent TAR files in the same file.

# Two very different exploitation strategies

## ZIP

Extra field:
fully supported and preserved.

DOCX Root:
mostly supported (Office, GDocs).

Standard collision PoCs:
-> incremental update
via standard tools!

## GZIP

Extra field:
fully supported and preserved.

Extra members:
mostly supported. Likely unpreserved as such.

Very crafty collision PoCs:
-> any modification will break the collision.

# Tricks and compatibility

# Only for MD5!?

These tricks will work for SHA1 and SHA2
(same Merkle-Damgård construct).

And at least, experimenting with MD5 is easier/cheaper:
Sha1tered: 11k USD / Shambles: 45k USD


https://twitter.com/realhashbreaker/status/838409756742156289

# Fix or prevention ?

Both tricks rely on "Extra fields".
Standard and documented, commonly skipped, no scrutiny (no bug to fix).

They can be scanned or removed (no needed recompression).
-> check known IDs, length and entropy.

Multiple members in Gzip: detectable - but standard.

**2021.04.20 12:06 "[Tiff] Libtiff 4.3.0 is released", by Even Rouault**

Hi,

No blocking issues have been reported on rc1, so I've moved to promote it to final.

Read about this release at https://libtiff.gitlab.io/libtiff/v4.3.0.html

The release tarball can be downloaded at:

http://download.osgeo.org/libtiff/tiff-4.3.0.tar.gz
http://download.osgeo.org/libtiff/tiff-4.3.0.zip

md5sum:

0a2e4744d1426a8fc8211c0cdbc3a1b3 tiff-4.3.0.tar.gz
6b792ee97cffef772206bc3ba0cd257c tiff-4.3.0.zip

The release files are also signed with my private PGP key, and the associated signature files have ".sig" extensions.

Even

http://www.spatialys.com
My software is free, but my time generally not.

->

**2022.05.20 16:38 "[Tiff] libtiff v4.4.0 RC1 available", by Even Rouault**

Hi,

I've prepare a release candidate for libtiff v4.4.0:

- https://download.osgeo.org/libtiff/tiff-4.4.0rc1.tar.gz
- https://download.osgeo.org/libtiff/tiff-4.4.0rc1.tar.gz.sig
- https://download.osgeo.org/libtiff/tiff-4.4.0rc1.tar.xz
- https://download.osgeo.org/libtiff/tiff-4.4.0rc1.tar.xz.sig
- https://download.osgeo.org/libtiff/tiff-4.4.0rc1.zip
- https://download.osgeo.org/libtiff/tiff-4.4.0rc1.zip.sig

Release notes at https://libtiff.gitlab.io/libtiff/v4.4.0.html

I'll promote it to final next week if no serious blocking issues are reported.

Best regards,

http://www.spatialys.com
My software is free, but my time generally not.

# LibTiff: no more MD5 mentions (only OpenPGP signatures)

https://www.asmail.be/msg0055059467.html
https://www.asmail.be/msg0055222537.html

Corkami's Collisions repository on Github

MIT Licence. Docs, pre-computed prefixes, scripts.

PII-free/copyright-free minimal PoCs.

Covered collisions:
FastColl, UniColl, Hashclash, Shattered, Shambles.

Covered formats:
GIF, GZ, JPG, MP4, PDF, PE, PNG, ZIP, ZIP(XML).

DOCX PPTX XSLX
3MF EPUB XPS

https://github.com/corkami/collisions

# Don't play with fire.
# Don't rely on MD5.

No matter your threat model,
a stronger algorithm guarantees
that no one can play tricks.

MD5 To Be Considered Harmful Someday - Dan Kaminsky 2004
https://eprint.iacr.org/2004/357

# On a personal note

Some formats aren't exploitable alone.
They can be exploited when combined with others.

I was stuck. I was helped/pushed.

Format or researcher:
Failing alone. Successful together.

# Thank you!

Questions, suggestions...

Special thanks to:
Philippe Teuwen, Marc Stevens, Gaëtan Leurent,
Philippe Lagadec, Yann Droneaud, Hans Wennborg.

Welcome to the

# Bonus slides

# Interference of other Extra Fields

ZIP: EF enforced only for the collision block file.
Other files are not affected.

GZIP: Depends if it's per file or per member.
At least, UniColl is cheaper to compute.

"Block gzip"

# BGZIP: GZIP-based with Extra Field

Uses concatenated members on 64b blocks.

Stores index in "BC" Subfield for each member.

```
      x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 xA xB xC xD xE xF
0x    1F 8B 08 04 00 00 00 00 00 FF
                                     06 00  B  C 02 00
1x    1b 00
      .. .. 03 00 00 00 00 00 00 00 00 00
      x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 xA xB xC xD xE xF
```

# Other GZIP-based formats

- Ableton Live Set
- EMZ - Enhanced MetaFile
- LiveSwif / Gnumeric spreadsheet
- RData
- SVGZ (multiple members not supported by Inkscape)

https://inkscape.gitlab.io/inkscape/doxygen/ziptool_8cpp_source.html#l01704

# Other archive formats

Are they exploitable?

# BZIP2

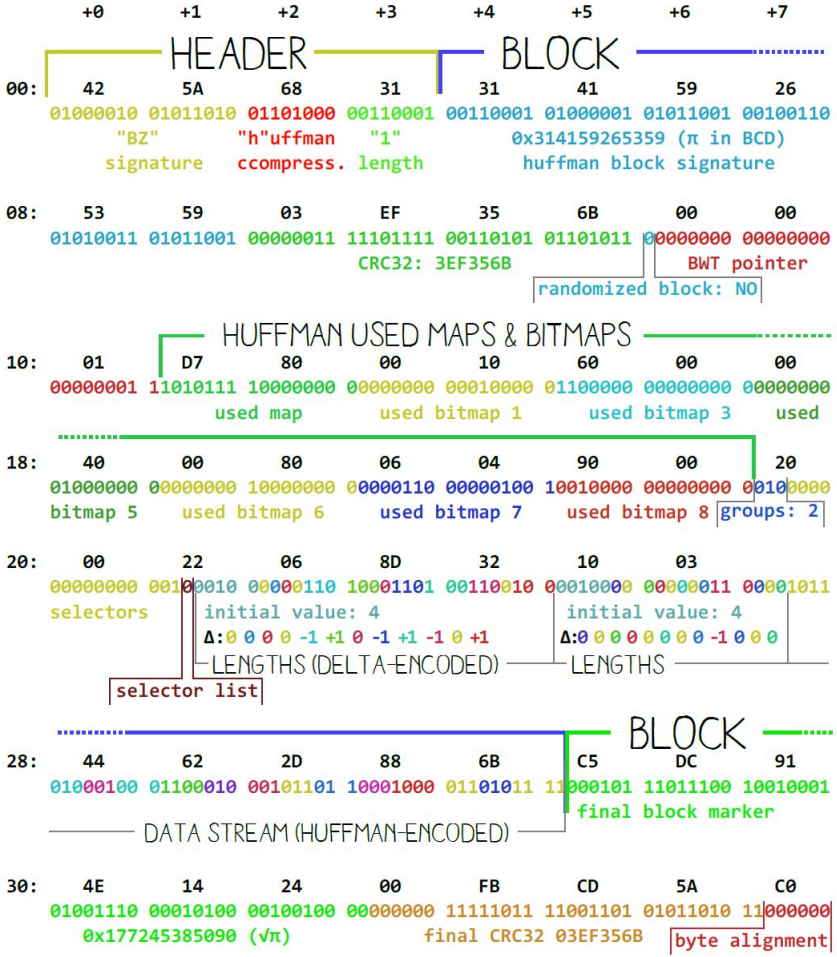
Pure compressor.

Bit-based format.

Bit alignment - not byte.

No padding.

No comment.

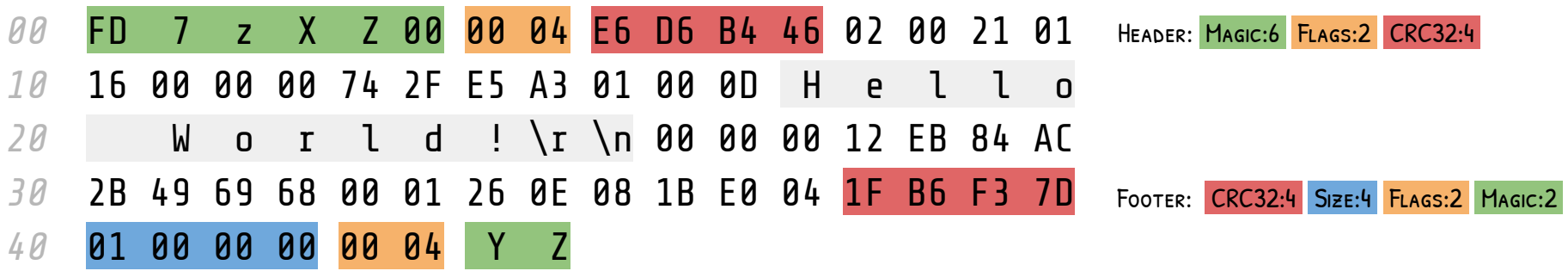



https://github.com/dsnet/compress/blob/master/doc/bzip2-format.pdf

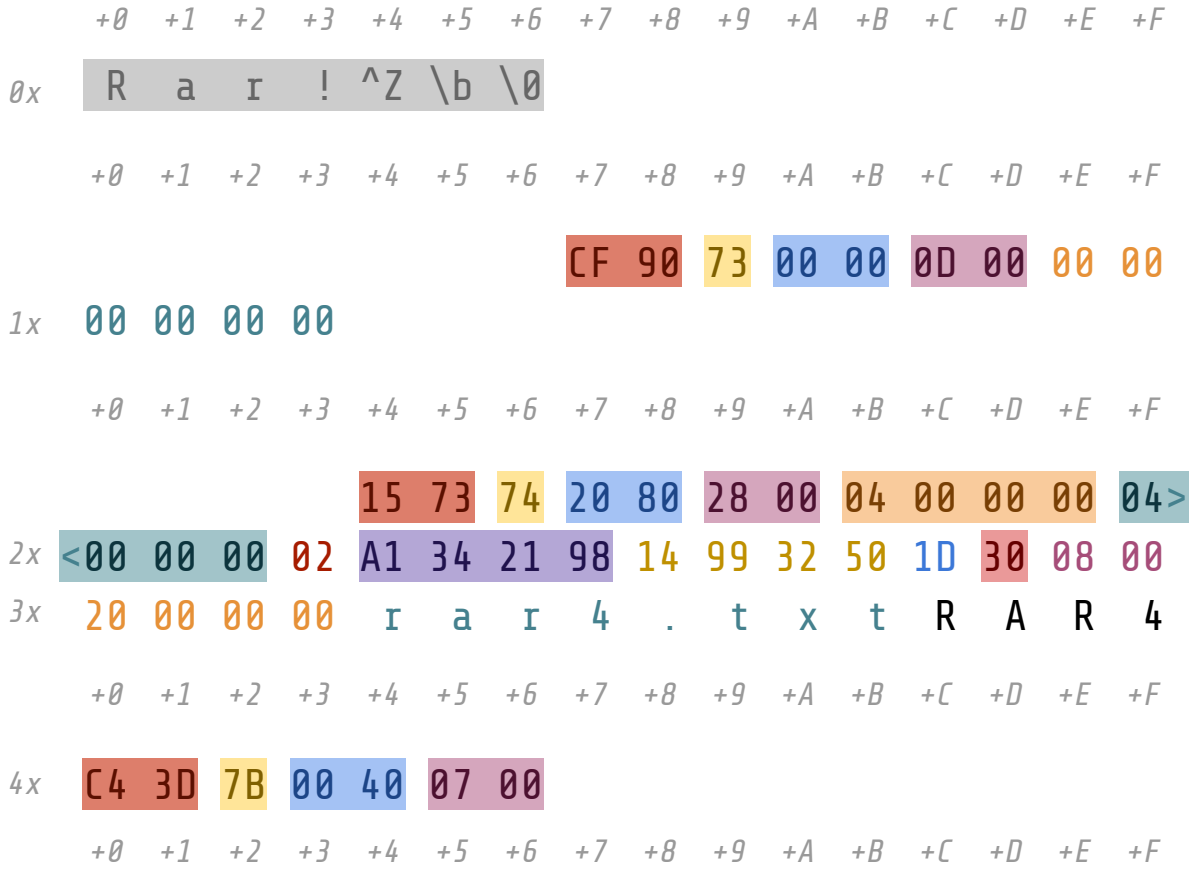
# XZ: A FORMAT WITH NO POLYGLOTS

Sequence of streams w/ enforced header **and** footer!

No fancy feature - no comment, no filename, no storage




https://tukaani.org/xz/xz-file-format.txt




# A simple Rar archive

## Magic

| | | | |
|---|---|---|---|
| 00 | Magic | 6 | Rar! ^Z \b \0 |

## Archive block

| | | | |
|---|---|---|---|
| 07 | CRC16 | 2 | 0x90CF |
| 09 | Type | 1 | 0x73 (Archive Header) |
| 0A | Flags | 2 | 0 |
| 0C | Size | 2 | 0x000D |
| 0E | Reserved2 | 2 | 0 |
| 10 | Reserved4 | 4 | 0 |

## File block

| | | | |
|---|---|---|---|
| 14 | CRC16 | 2 | 0x7315 |
| 16 | Type | 1 | 0x74 (File Header) |
| 17 | Flags | 2 | 0x8020 (Dict=128k) |
| 19 | Size | 2 | 0x0028 |
| 1B | Pack size | 4 | 4 |
| 1F | Unp size | 4 | 4 |
| 23 | Host OS | 1 | 2 (Win) |
| 24 | File CRC | 4 | 0x982134A1 |
| 28 | Ftime | 4 | 0x50329914 |
| 2C | Unp Ver | 1 | 0x1D |
| 2D | Method | 1 | 0x30 (Store) |
| 2E | Name size | 2 | 8 |
| 30 | Attr | 4 | 0x00000002 |
| 34 | File name | ? | rar4.txt |
| 3C | Contents | ? | RAR4 |

## Archive end

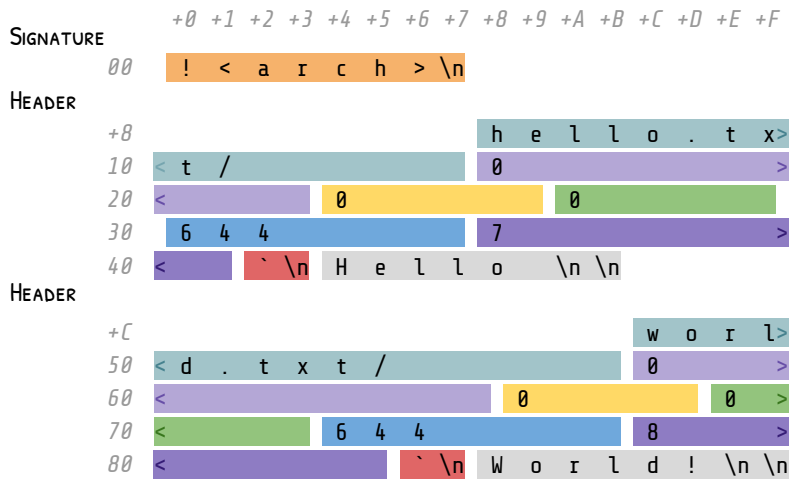
| | | | |
|---|---|---|---|
| 40 | CRC16 | 2 | 0x3DC4 |
| 42 | Type | 1 | 0x7B (Terminator) |
| 43 | Flags | 2 | 0x0400 |
| 45 | Size | 2 | 0x0007 |

# RAR:

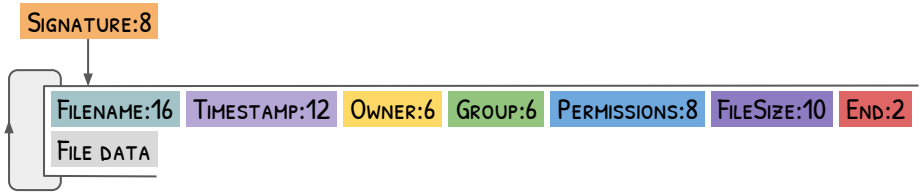
✔ Top-down parsed
✔ Appended data

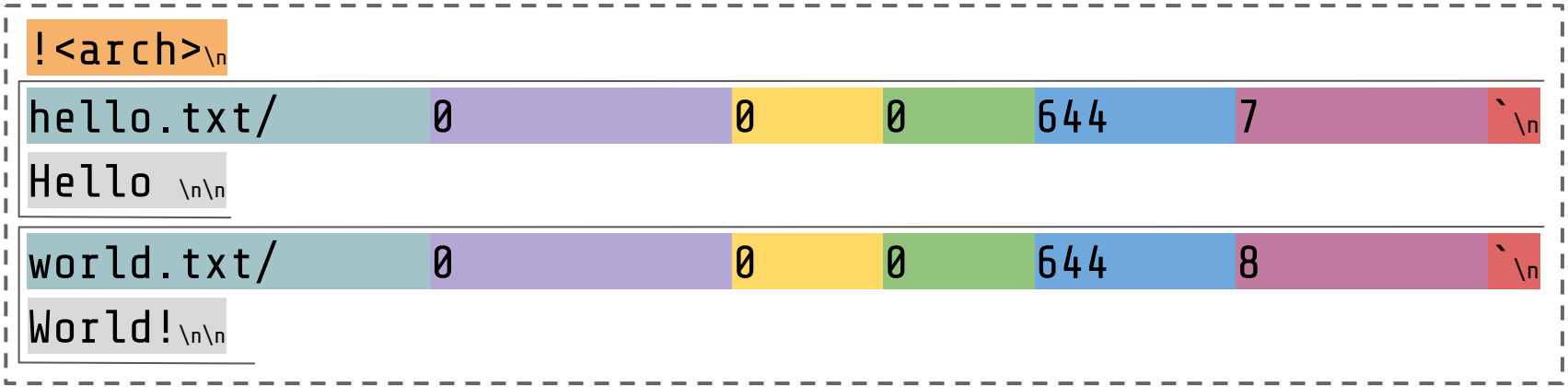
CRC16 for each header -> no UniColl.

Standard generic exploitation via Hashclash ?
Poorly documented format - proprietary.




A magic signature, then
a sequence of a fixed-size header
and file contents.



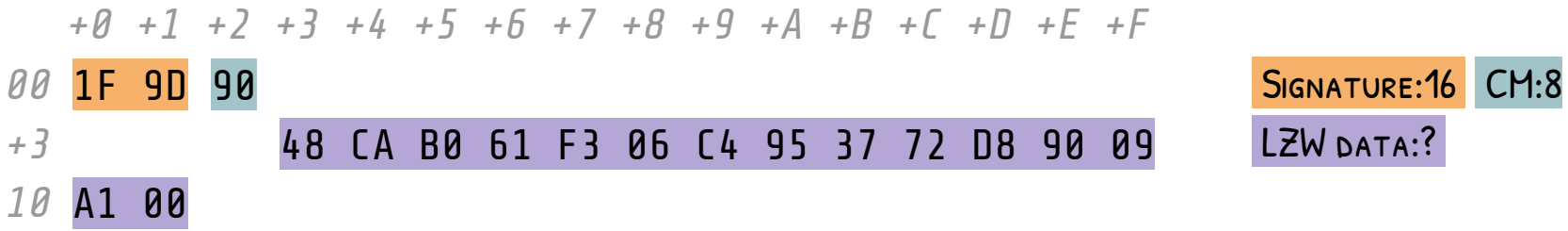




# ARchive (.a / .lib / .ar): too simple for abuse

https://en.wikipedia.org/wiki/Ar_(Unix)

A magic signature, then a maxbit/block byte, then LZW data.

HelloWorld.Z




# Compress (.Z): way too simple

https://en.wikipedia.org/wiki/Compress

Bonus

# Ambiguous Office files

Wordpad ignores the root file.

# WordPad

Included in Windows, default handler of DOCX.

Ignored root file -> collisions are not working.

"Valid" doc files w/ just 2 XML files.

2 files, ~ 600 bytes

```
Archive:  mini.docx
  Length      Date    Time    Name
---------  ---------- -----   ----
      265  06/12/2022 15:07   [Content_Types].xml
      260  06/12/2022 15:06   doc.xml
---------                     -------
      525                     2 files
```

https://www.virustotal.com/gui/file/3134ff057c1e7b7384ed6eaaa1acd7f9ac4c35b045f4a11f28622278d8dcc380

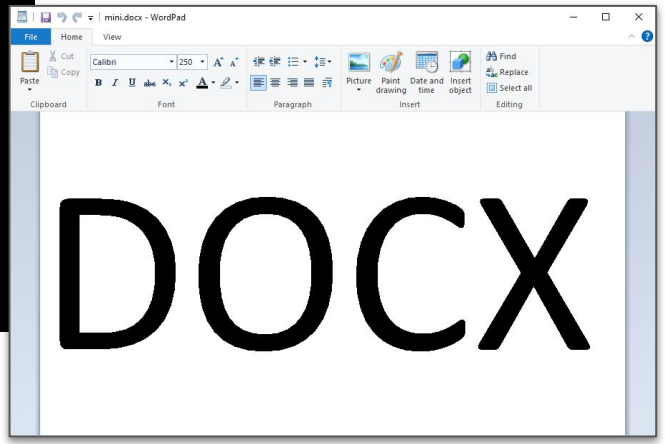
[Content_Types].xml

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Types
  xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
  <Override
    ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml"
    PartName="/doc.xml"/>
</Types>
```

doc.xml

Only referenced in the content types file!?

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<w:document
  xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
  <w:body>
    <w:p>
      <w:r>
        <w:rPr>
          <w:sz w:val="500"/>
        </w:rPr>
        <w:t xml:space="preserve">DOCX</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

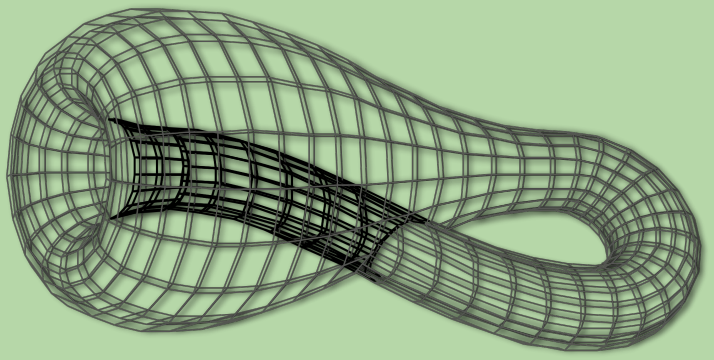



# Contents of a minimal WordPad Docx

ANGE ALBERTINI
reverse engineering
VISUAL DOCUMENTATIONS
@angealbertini
ange@corkami.com
http://www.corkami.com

# INSIDE OUT

## Abusing archive file formats



</slides>