

FEARSOME FILE FORMATS

3803

ILLEGAL

INSTRUCTIONS

ANGE ALBERTINI



FEARSOME FILE FORMATS

ANGE
ALBERTINI



38C3
28/12/2024

ABOUT THE AUTHOR

- Looking at hex editors for 35 years.
- Malware analyst for 20 years: Symantec, Avira, Google.
- Corkami: posters, PoCs, tools, tutorials (15k ★).
- CPS2Shock, PoC||GTFO...

*My own views
and opinions.*



PIXEL ART BY SQUIBLYDOO (2023)



cps2shock.emu-france.info

PoC || GTFO
roo f oncept or et he uck ut

[github / angea / pocorgtfo](https://github.com/angea/pocorgtfo)

LET'S LOOK AT A FILE...

...what do you think?

(YES, IT'S EMPTY)...

CAN AN EMPTY FILE BE USEFUL?

Can one find purpose in emptiness ? 🤔

Besides:

- crashing code in production,
- stopping malware installation,
- shutting down botnets,

...

"THERE'S NO WAY
THE EMPTY FILE
COULD BE USED
IN STANDARD!"

/bin/true

USED TO BE EMPTY.

An empty shell script. Standard in every system.
It always works, and saves space.
It even became copyrighted despite its empty payload.

Nowadays, /bin/true is an ELF binary.

```
$ touch test
$ chmod +x test
$ ./test
$ echo $?
0
```

In Doom WADs,
empty files are used
as map index
in the archive table:
E1M1, ...

VOID IS HERE

HEADER

Magic PWAD Patch
Count
0xF0 →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	06	00	00	00	F0	00	00	00	00	00	00	00	00
010	00	00	00	07	00	00	00	01	00	01	00	00	00	00	00	00
020	00	00	00	02	00	01	00	00	00	00	00	00	00	01	00	00
030	00	00	00	03	00	01	00	00	00	00	00	02	00	FF	FF	00
040	00	00	00	04	00	00	00	00	00	03	00	FF	FF	00	00	00
050	00	00	00	05	00	00	00	00	00	00	00	00	00	00	00	00
060	00	00	00	06	00	00	00	00	00	00	00	00	00	00	00	00
070	00	00	00	07	00	00	00	00	00	00	00	00	00	00	00	00
080	00	00	00	08	00	00	00	00	00	00	00	00	00	00	00	00
090	00	00	00	09	00	00	00	00	00	00	00	00	00	00	00	00
0A0	00	00	00	0A	00	00	00	00	00	00	00	00	00	00	00	00
0B0	00	00	00	0B	00	00	00	00	00	00	00	00	00	00	00	00
0C0	00	00	00	0C	00	00	00	00	00	00	00	00	00	00	00	00
0D0	00	00	00	0D	00	00	00	00	00	00	00	00	00	00	00	00
0E0	00	00	00	0E	00	00	00	00	00	00	00	00	00	00	00	00
0F0	00	00	00	0F	00	00	00	00	00	00	00	00	00	00	00	00
100	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00	00
110	00	00	00	11	00	00	00	00	00	00	00	00	00	00	00	00
120	00	00	00	12	00	00	00	00	00	00	00	00	00	00	00	00
130	00	00	00	13	00	00	00	00	00	00	00	00	00	00	00	00
140	00	00	00	14	00	00	00	00	00	00	00	00	00	00	00	00

→ DIRECTORY

Offset	0xC	0xC	0x16	0x4E	0xC6	0xD6
Size	0	0xA	0x38	0x78	0x10	0x1A
Name	E1M1	THINGS	LINEDEFS	SIDEDEFS	VERTEXES	SECTORS

Marker →

THINGS

X 0x40
Y 0x40
Angle 0 0:East
Type 1 1:Pistart
Flag 7 1:Easy 2:Medium 3:Hard

LINEDEFS

Vortex start 0 1 2 3
Vortex end 1 2 3 0
Flags 1 1 1 1 1:Impassable
Line type 0 0 0 0
Sector tag 0 0 0 0
Side right 0 1 2 3
Side left -1 -1 -1 -1

SIDEDEFS

X 0 0 0 0
Y 0 0 0 0
Upper texture - - - -
Lower texture - - - -
Middle texture BIGDOOR2 COMPUTE1 TEKWALL5 SWIBRCOM
Sector 0 0 0 0

VERTEXES

X 0 0x80 0x80 0
Y 0x80 0x80 0 0

SECTORS

Floor height 0
Ceiling height 0x40
Floor texture FLOOR4_8
Ceiling texture CEIL4_3
Light level 0xA0
Special 0 Unoriginal
Tag 0

LUMPS

BIGDOOR2
0x80
COMPUTE1
0x80
TEKWALL5
FLOOR4_8
CEIL4_3

The **WAD** file format from DOOM in 1993.

ANGE ALBERTINI
2020 <http://www.corkami.com>

```
The IBM Personal Computer DOS
Version 1.00 (C)Copyright IBM Corp 1981
```

```
A>DEBUG EMPTY.COM
File not found
-w
-q
```

```
A>DIR EMPTY.COM
EMPTY      COM          0  01-01-80
```

```
A>_
```

```
A>DIR TIME.COM
TIME      COM          250  08-04-81
```

```
A>TIME
Current time is 18:24:41.81
Enter new time:
```

```
A>DIR EMPTY.COM
EMPTY      COM          0  01-01-80
```

```
A>EMPTY
Current time is 18:24:53.27
Enter new time:
```

```
A>_
```

VOID IS LAST

UNDER IBM PC-DOS 1.0 AND CP/M,

LAUNCHING AN EMPTY FILE WILL JUST RE-RUN THE LAST ONE:

THE MEMORY WASN'T CLEARED BETWEEN EXECUTIONS.

```
CP/M 2.2 - Amstrad Consumer Electronics
plc
```

```
A>ED EMPTY.COM
```

```
NEW FILE
: *e
```

```
A>STAT EMPTY.COM
```

```
Recs Bytes Ext Acc
0 0k 1 R/W A:EMPTY.COM
Bytes Remaining On A: 5k
```

```
A>EMPTY EMPTY.COM
```

```
Recs Bytes Ext Acc
0 0k 1 R/W A:EMPTY.COM
Bytes Remaining On A: 5k
```

```
A>█
```

SO THE EMPTY FILE IS... 

- A standard system shell script that always executes successfully.
- An index in Doom archives.
- A commercial (!) DOS executable that repeats the last command.

(among possibly many other things)

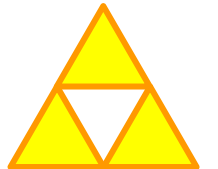
No content, and yet...

THE TYPE, CONTEXT AND PURPOSE
ARE ALREADY UNCLEAR.

A file is more than its content.
Context and metadata are important.

Let's look at something else...

WHAT ARE 'FILES'?



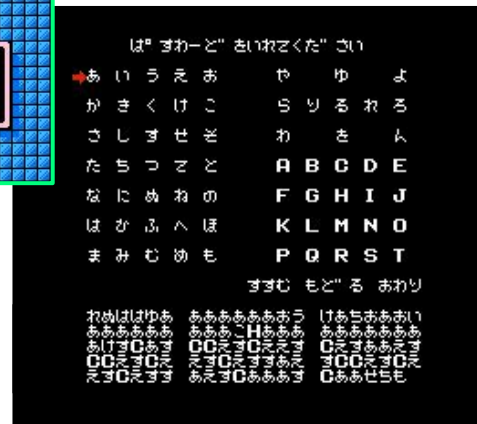
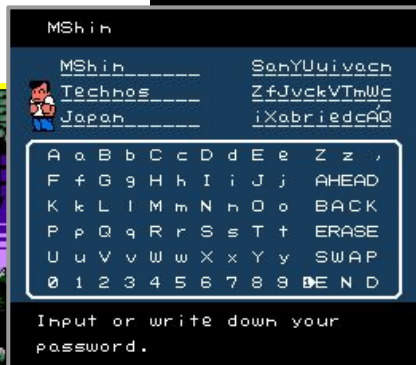
WHEN STORAGE WAS TOO EXPENSIVE,
GAMES USED TO RELY ON
LONG PASSWORDS
TO SAVE YOUR DATA!

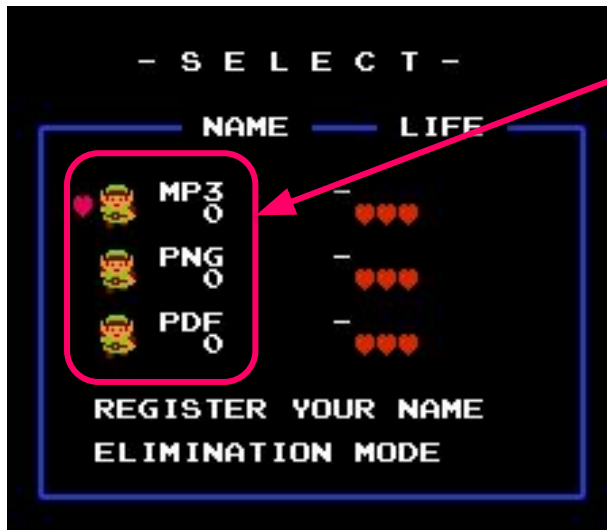
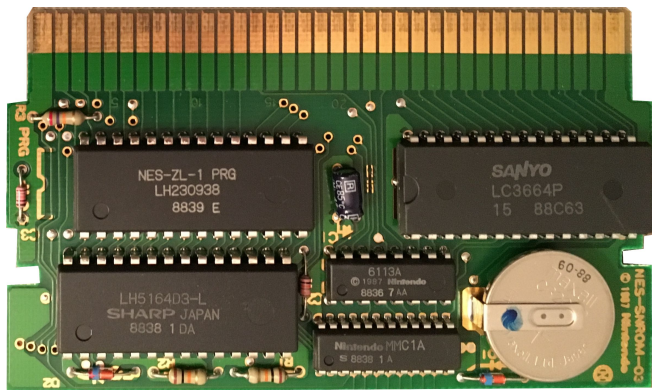
SOME AREN'T EVEN IN TEXT!

PASS WORD PLEASE

NARPAS SWORDO
000000 000000

1 2 3 4 5 6 7 8 9 A B C
D E F G H I J K L M N O P
Q R S T U V W X Y Z a b c
d e f g h i j k l m n o p
q r s t u v w x y z ? -





00-09: 0-9
10-24: A-Z

00:	16	19	03	\$.\$.\$.\$.\$
08:	19	17	10	\$.\$.\$.\$.\$
10:	19	0D	0F	\$.\$.\$.\$.\$
18:	00	00	00	00	00	00	00	00
20:	00	00	00	00	00	00	00	00
28:	00	00	00	00	00	00	00	00
30:	22	FF	00	00	00	00	00	00
38:	00	00	00	00	00	08	00	00
40:	00	00	00	00	00	00	00	00
48:	00	00	00	00	00	00	00	00
50:	00	00	00	00	00	00	00	00
58:	22	FF	00	00	00	00	00	00
60:	00	00	00	00	00	08	00	00
68:	00	00	00	00	00	00	00	00
70:	00	00	00	00	00	00	00	00
78:	00	00	00	00	00	00	00	00
80:	22	FF	00	00	00	00	00	00
88:	00	00	00	00	00	08	00	00

SAVING GAMES IN 1986: HARDCODED OFFSETS IN SRAM.



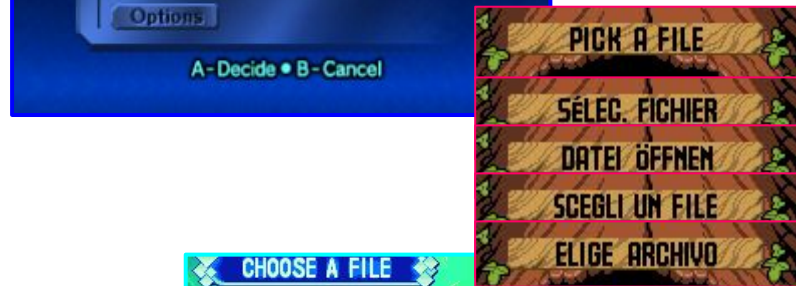
1986
the Legend of Zelda



1998: Ocarina of Time



1987
the Adventure of Link



2001: Oracles of...



1991
A Link to the Past



1993
Link's Awakening.

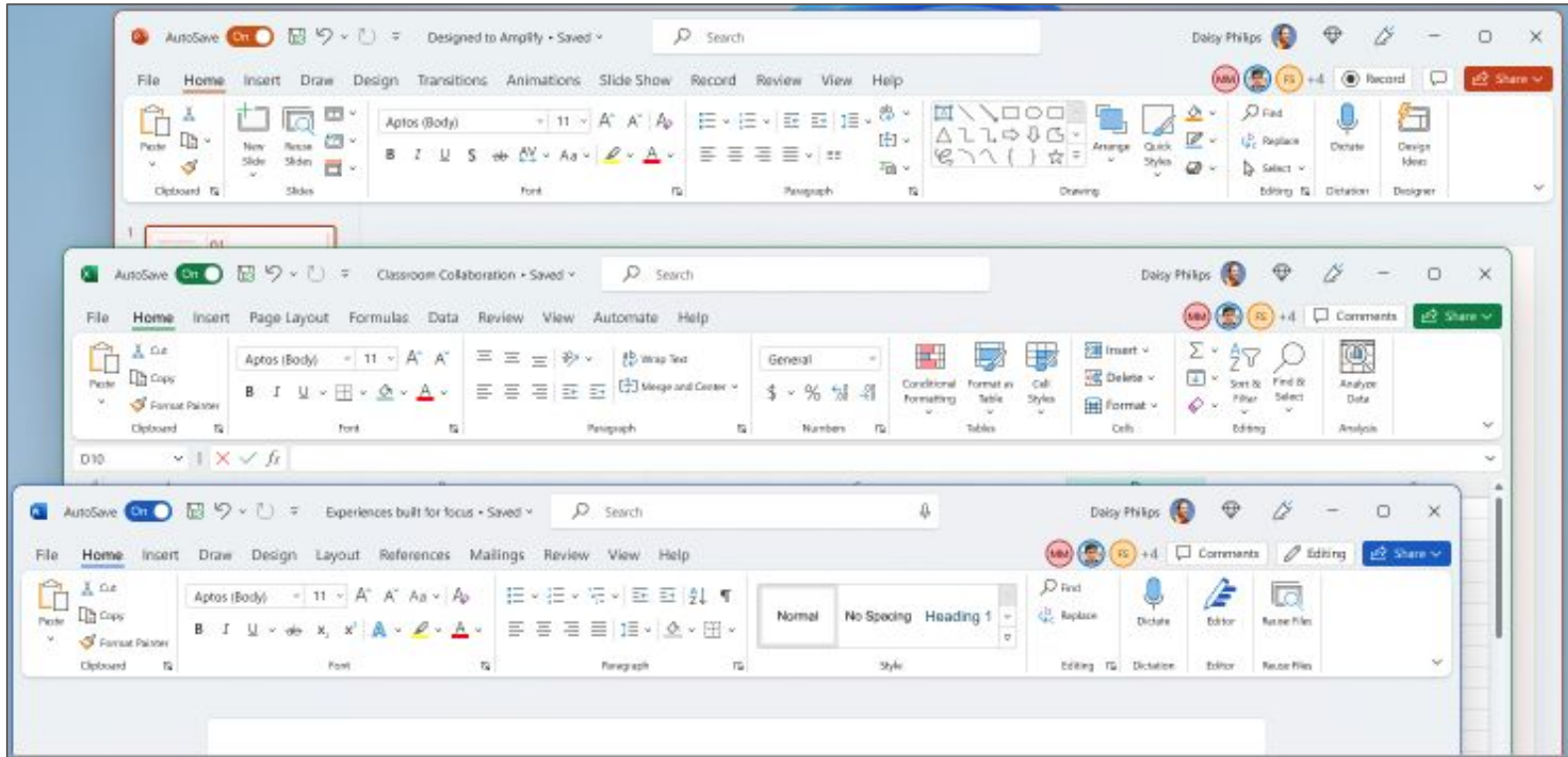
2004
The Minish Cap



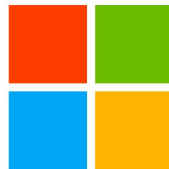
FROM "PLAYER" TO "FILE".

WHAT'S A FILE WITHOUT A FORMAT?

What does that even mean?
Why would you do that?

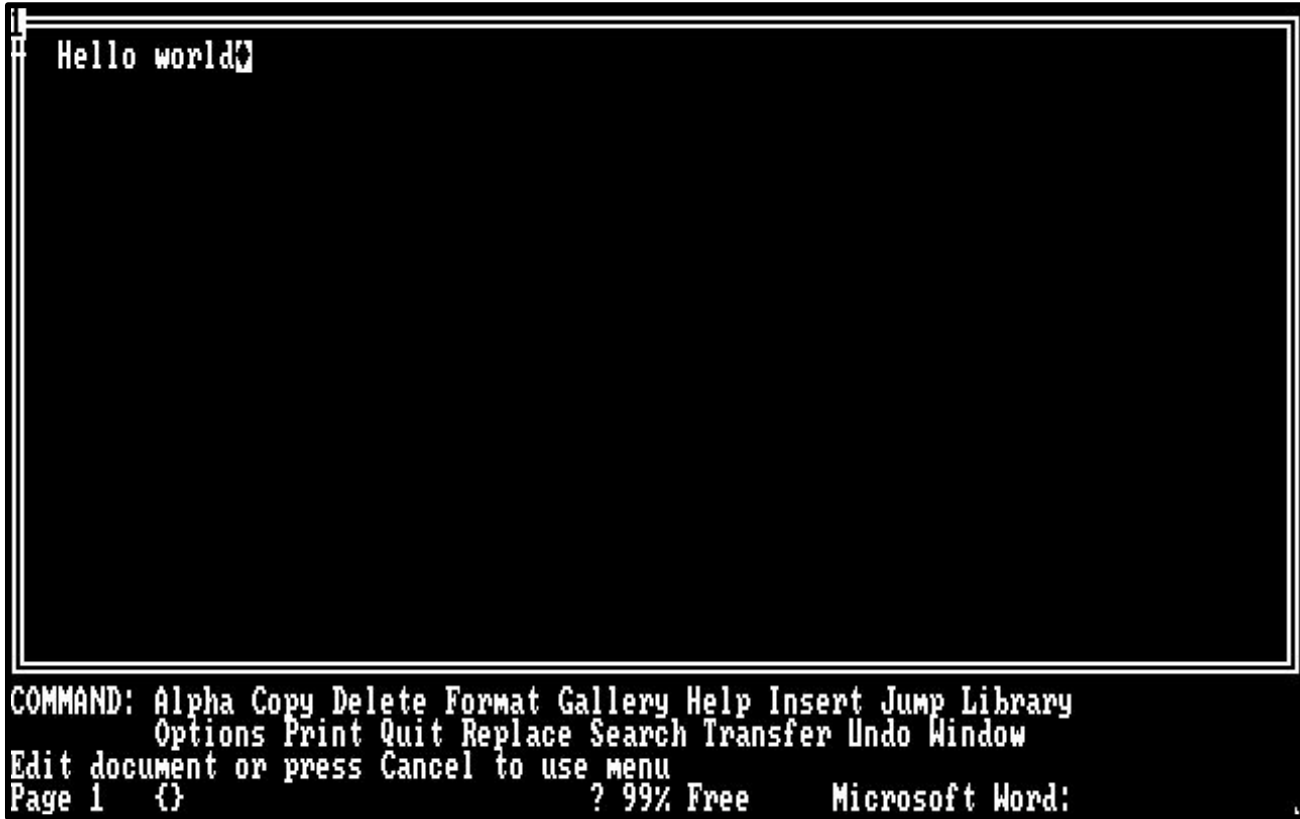


MICROSOFT OFFICE...



Microsoft

THE LOGO SINCE 2012.



(1982-1987)

MICROSOFT OFFICE WORD, IN 1984...

MICROSOFT

A sign of the times!

...DIDN'T USE A FILE FORMAT!

The whole memory page
was saved as a file...
...with whatever else in memory!

Who needs standardization
when you're just on your own?
It was just faster to snapshot
the memory range.

```

0000: 31 BE 00 00 00 AB 00 00 00 00 00 00 00 00 8C 00 1↓ ½ î
0010: 00 00 03 00 04 00 04 00 04 00 04 00 04 00 4E 4F NO
0020: 52 4D 41 4C 2E 53 54 59 00 00 00 00 00 00 00 00 RMAL.STY
0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080: 48 65 6C 6C 6F 20 57 6F 72 6C 64 21 64 2E 3E 00 Hello World!d.>
0090: 80 00 46 80 76 61 72 69 61 6E 74 3A 20 20 63 68 Ç FÇvariant: ch
00A0: 6F 6F 73 65 20 61 20 6C 65 74 74 65 72 20 6F 72 oose a letter or
00B0: 20 6E 75 6D 62 65 72 20 74 6F 20 69 64 65 6E 74 number to ident
00C0: 69 66 79 20 74 68 69 73 20 73 74 79 6C 65 20 61 ify this style a
00D0: 73 20 61 20 75 6E 69 71 75 65 46 44 82 76 61 72 s a uniqueFDévar
00E0: 69 61 74 69 6F 6E 20 6F 66 20 75 73 61 67 65 20 iation of usage.
00F0: 6E 61 6D 65 2E 20 50 72 65 73 73 20 61 20 64 69 name. Press a di
0100: 80 00 00 00 8C 00 00 00 FF FF 00 00 00 00 00 00 Ç î
0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 F6 ÷
0140: 03 00 F0 07 00 05 00 EA F6 00 00 22 AE 01 00 ≡ Ω÷ "«
0150: 17 00 C2 9C 00 80 00 F2 F6 06 80 03 04 00 00 T£ Ç ≥÷ Ç
0160: 80 00 80 00 FF 00 17 00 2C 9C 00 00 0C F7 03 00 Ç Ç ,£ ≈
0170: 32 05 34 03 17 00 16 00 18 00 C2 9C C2 9C 06 01 2 4 T£T£
0180: 80 00 00 00 8D 00 00 FF FF 00 00 00 00 00 00 Ç i
0190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 F6 ÷
01C0: 03 00 F0 07 00 05 00 EA F6 00 00 22 AE 01 00 ≡ Ω÷ "«
01D0: 17 00 C2 9C 00 80 00 F2 F6 06 80 03 04 00 00 T£ Ç ≥÷ Ç
01E0: 80 00 80 00 FF 00 17 00 2C 9C 00 00 0C F7 03 00 Ç Ç ,£ ≈
01F0: 32 05 34 03 17 00 16 00 18 00 C2 9C C2 9C 06 01 2 4 T£T£

```

HELLOW.DOC

(512 BYTES FOR 12 BYTES OF TEXT!)

How do you reliably handle such files?

...

"What a mess!"

mustangstromboneheadlinefeedbackhandrailroadsideshowdownturnoverbookcaseworkshop

File format landscape 101:
FULL OF NASTY SURPRISES,
EXCEPTIONS, ODDITIES,
FOR HISTORICAL OR
TECHNICAL REASONS.

We need to preserve file formats in a better way...

*What's the **NAME** of file formats? 🤔*

A decade later...



SOME THINGS HAVEN'T CHANGED...

Ambiguous files (aka **werewolves** aka **parser differentials** aka **schizophrenic** files) are still there.

No reference parser, no test corpus.

Expensive specifications? -> devs don't pay for them!

And often, no real/serious specifications.

A simple example...

"ANGE"

How do you pronounce this name ?

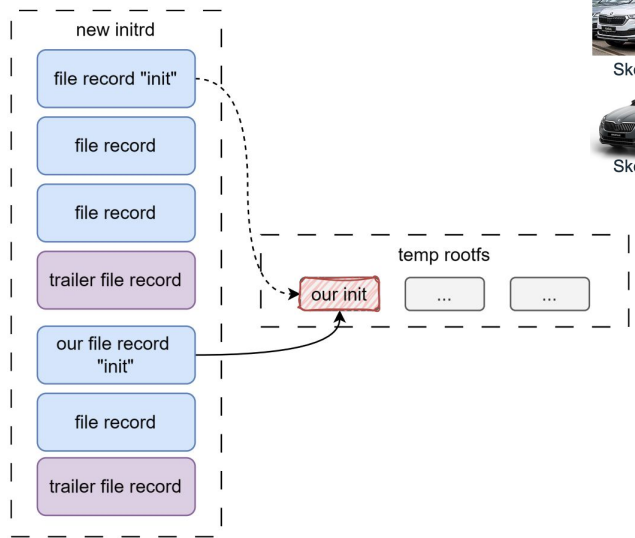
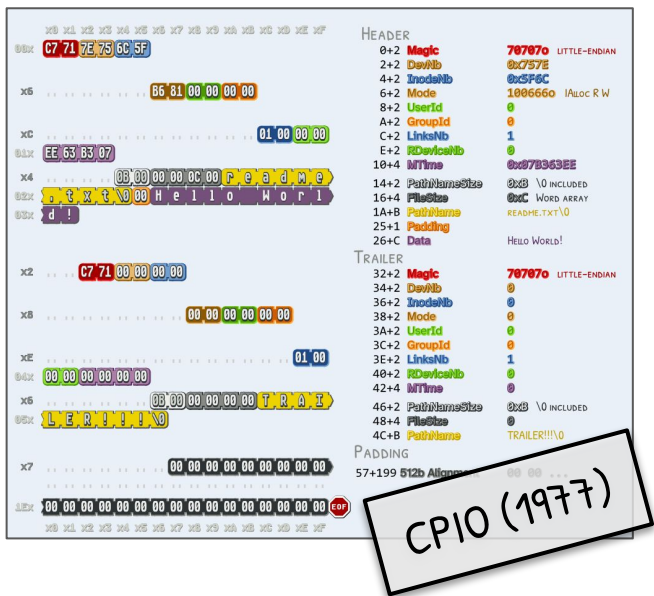
Anje, Enn-ji, An-gé, Anzu (杏), Enn-ré, Až...

Male or female?

How many names are 'unpronounceable' ?
Without references, things quickly get messy.

CONCATENATION STILL WORKS!

Duplicate file entry in a CPIO archive used to hack cars over the air, in 2024.



blackhat
EUROPE 2024
DECEMBER 11-12, 2024
BRIEFINGS

Over the Air
Compromise of Modern Volkswagen Group Vehicles

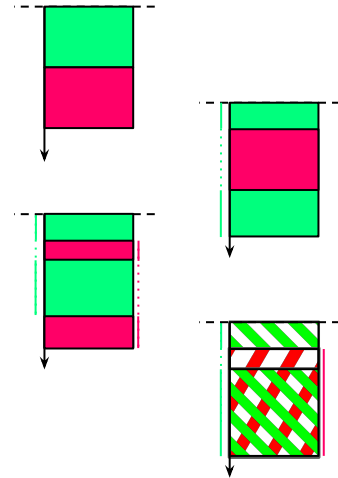
Speaker(s):
Artem Ivachev
Danila Pamishchev

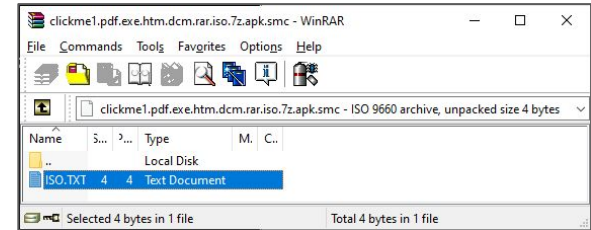
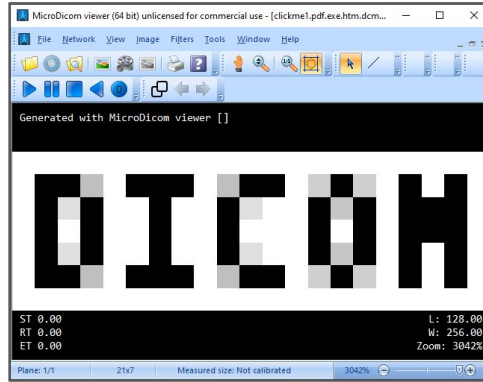


Multi-type / chameleon files, a.k.a.

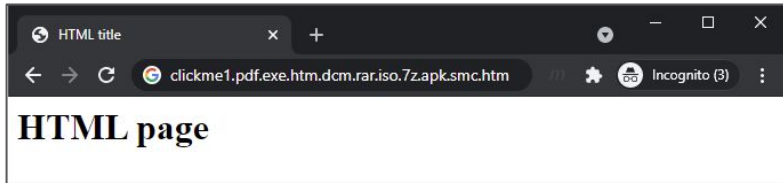
POLYGLOTS

1. **Concatenation** (appended data)
2. **Parasite** (comment)
3. **Zipper** (mutual comments)
4. **Chimera** (shared data)





```
>clickme1.pdf.exe.htm.dcm.rar.iso.7z.apk.smc.exe  
32-bit PE
```



```
> unrar v clickme1.pdf.exe.htm.dcm.rar.iso.7z.apk.smc  
UNRAR 5.40 beta 2 x64 freeware Copyright (c) Alexander Roshal  
Archive: clickme1.pdf.exe.htm.dcm.rar.iso.7z.apk.smc  
Details: RAR 4, SFX  


| Attributes | Size | Packed Ratio | Dr  | Time | Checksum | Name |
|------------|------|--------------|-----|------|----------|------|
| ..A....    | 4    | 4 100%       | 202 |      |          |      |
|            | 4    | 4 100%       |     |      |          |      |


```



[CLICKME](#) (.PDF.EXE.HTM.DCM.RAR.ISO.7Z.APK.SMC)

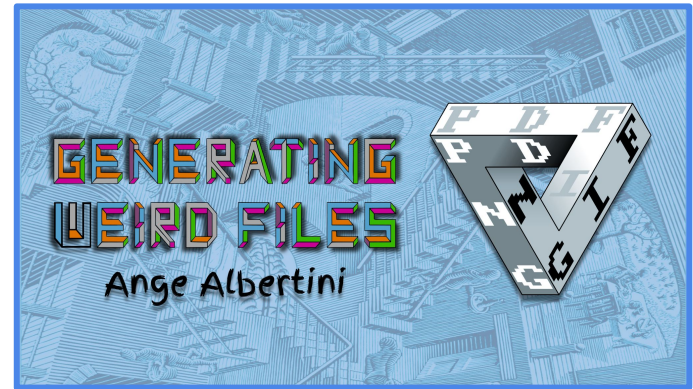
MITRA

<https://github.com/corkami/mitra>

☆ Star 1.1k

Identify file types, make space,
combine and adjust data.

It **should** keep the files valid:
no deep parsing, just the minimum.

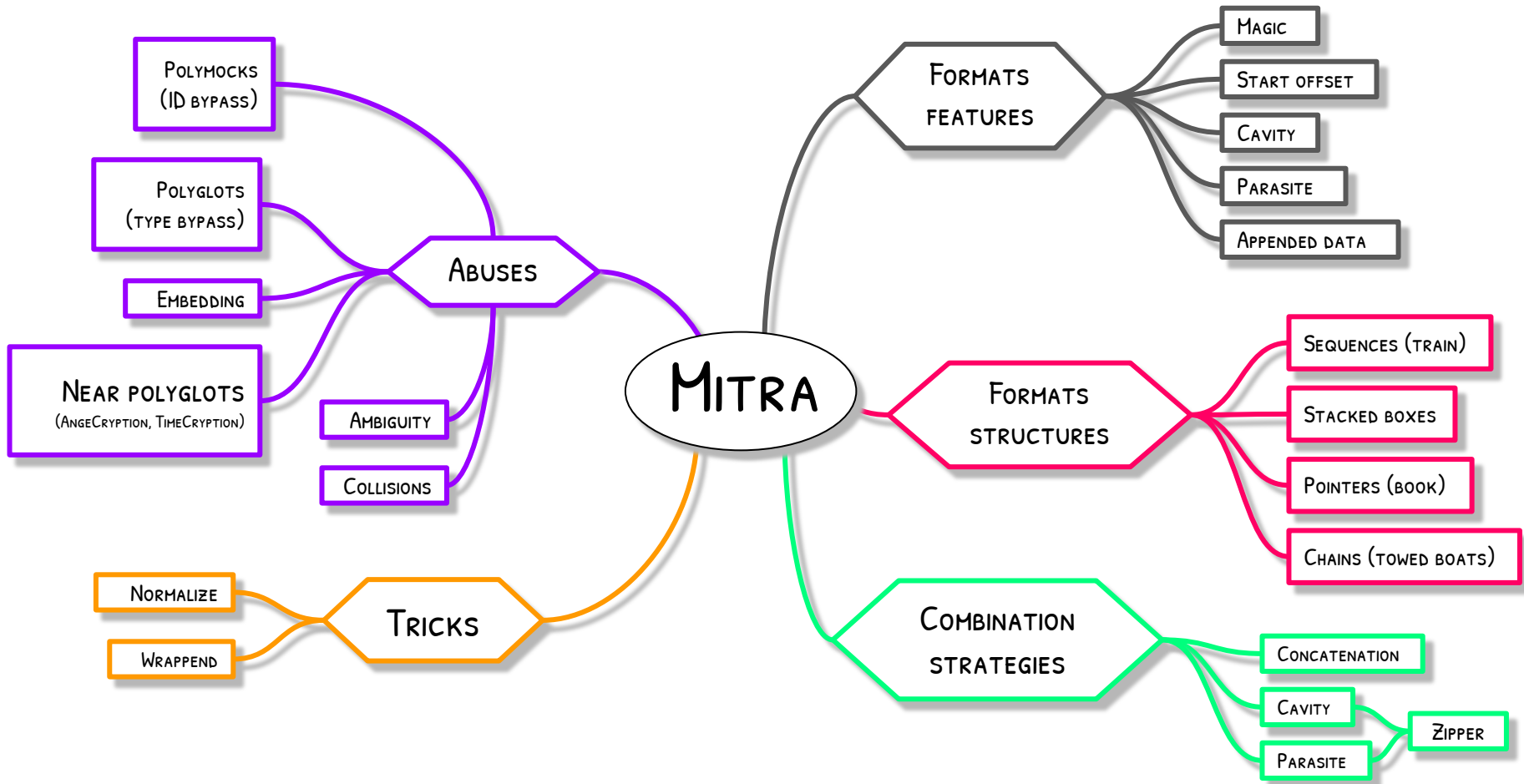


```
$ mitra.py dicom.dcm png.png
dicom.dcm
File 1: DICOM / Digital Imaging and Communications in Medicine
png.png
File 2: PNG / Portable Network Graphics

Zipper Success!
Zipper: interleaving of File1 (type DCM) and File2 (type PNG)
```



Named after [Mithridates](#)
(a famous polyglot)



EMBEDDING PAYLOADS

```
$ mitra.py png.png script.js -f
```

```
png.png
```

```
File 1: PNG / Portable Network Graphics
```

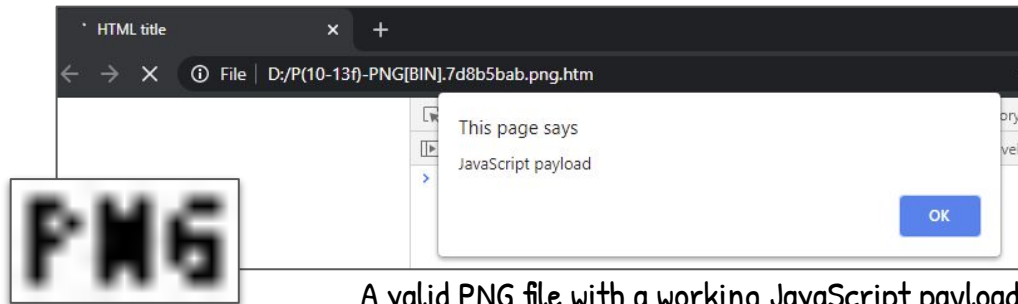
```
script.js
```

```
File 2: binary blob
```

```
Stack: concatenation of File1 (type PNG) and File2 (type BIN)
```

```
Parasite: hosting of File2 (type BIN) in File1 (type PNG)
```

```
000: 89 P N G \r \n ^Z \n 00 00 01 38 c O M M
010: - - > \r \n < d i v _ _ i d = ' m y
020: p a g e ' > \r \n < h 1 > H T M L
030: _ _ p a g e < / h 1 > \r \n < s c r
040: i p t _ _ l a n g u a g e = j a v
050: a s c r i p t _ _ t y p e = " t e
060: x t / j a v a s c r i p t " > _
070: \r \n d o c u m e n t . d o c u m
080: e n t E l e m e n t . i n n e r
090: H T M L _ _ = _ _ d o c u m e n t .
0A0: g e t E l e m e n t B y I d ( '
0B0: m y p a g e ' ) . i n n e r H T
0C0: M L ; \r \n d o c u m e n t . t i
0D0: t l e _ _ = _ _ ' H T M L _ _ t i t l
0E0: e ' ; \r \n a l e r t ( " J a v a
0F0: S c r i p t _ _ p a y l o a d " )
100: ; \r \n c o n s o l e . l o g ( "
110: J a v a S c r i p t _ _ p a y l o
120: a d " ) ; \r \n < / s c r i p t >
130: \r \n < / d i v > \r \n < ! - - _ _ 2E
140: DA DC 65 00 00 00 0D I H D R 00 00 00 0D 00
150: 00 00 07 01 03 00 00 00 E9 BE 55 59 00 00 00 06
160: P L T E FF FF FF 00 00 00 55 C2 D3 7E 00 00
170: 00 1B I D A T 08 1D 63 00 82 54 03 86 70 07
180: 86 F4 02 06 F7 00 06 57 03 06 06 06 00 21 1A 03
190: 10 32 6A 0B 48 00 00 00 00 I E N D AE 42 60
1A0: 82
```



A valid PNG file with a working JavaScript payload

```
-->
<div id='mypage'>
<h1>HTML page</h1>
<script language=javascript type="text/javascript">
document.documentElement.innerHTML =
document.getElementById('mypage').innerHTML;
document.title = 'HTML title';
alert("JavaScript payload");
console.log("JavaScript payload");
</script>
</div>
<!--
```

Parasite code


```
$ mocky.py --combined input/jpg.jpg
```

```
Filetype: JFIF / JPEG File Interchange Format
```

```
Parasite-combined sig(s): unicos / Symbian / snd / wdk / SoundFont / icc / VICAR / netbsd_ktraces / SoundFX / VirtualBox / ScreamTracker / Plot84 / ezd / dicom / Tar(checksum) / ds / CCP4 / DRDOS / pif / mbr 25676
```

```
> Combined Mock: mA-jpg.jpg
```

Add any possible signature with Mocky

```
$ file mA-jpg.jpg
```

```
mA-jpg.jpg: tar archive
```

← FILE sees it as a TAR file!
(valid TAR signature + checksum)

Many detected file types

```
$ file mA-jpg.jpg --keep-going --raw
```

```
mA-jpg.jpg: tar archive
- DR-DOS executable (COM)
- JPEG image data, baseline, precision 8, 104x56, components 1
- Windows Program Information File for acsp
- VICAR label file
- DOS/MBR boot sector
- Nintendo DS ROM image: "◆◆◆◆" (SNDH, Rev.107) (homebrew)
- Plot84 plotting file
- DOS/MBR boot sector
- sfArk compressed Soundfont
- Old EZD Electron Density Map
- Symbian installation file
- Scream Tracker Sample mono 8bit
- SNDH Atari ST music
- SoundFX Module sound file
- DICOM medical imaging data
- CCP4 Electron Density Map
- VirtualBox Disk Image (◆◆◆◆), 5715999566798081280 bytes
- unicos (cray) executable
- data
```

Still a perfectly valid JPEG!
(with an extra COMment segment stuffed with signatures)

```
$ identify -verbose ./mA-jpg.jpg
```

```
Image:
```

```
Filename: ./mA-jpg.jpg
Format: JPEG (Joint Photographic Experts Group JFIF format)
Mime type: image/jpeg
Class: PseudoClass
Geometry: 104x56+0+0
Resolution: 36x36
Print size: 2.88889x1.55556
Units: PixelsPerCentimeter
Colorspace: Gray
```

```
[...]
```

USING **MOCKY** TO BYPASS file IDENTIFICATION

A POLYMOCK - A 190-IN-1 YET EMPTY FILE

The file is mostly empty!
It only contains magics
to fake file types.

<https://github.com/corkami/pocs/tree/master/polymocks>

```
multi: Windows Program Information File for \030(0\001
- MAR Area Detector Image,
- Linux kernel x86 boot executable RW-rootFS,
- ReiserFS V3.6
- Files-11 On-Disk Structure (ODS-52); volume label is '
- DOS/MBR boot sector
- Game Boy ROM image (Rev.00) [ROM ONLY], ROM: 256Kbit
- Plot84 plotting file
- DOS/MBR boot sector
- DOSFONT2 encrypted font data
- Kodak Photo CD image pack file , landscape mode
- SymbOS executable v., name: HNR00\334\247\304\375\034\236\243
- ISO 9660 CD-ROM filesystem data (raw 2352 byte sectors)
- Nero CD image at 0x4B000 ISO 9660 CD-ROM filesystem data
- High Sierra CD-ROM filesystem data
- Old EZD Electron Density Map
- Apple File System (APFS), blocksize 24061976
- Zoo archive data, modify: v78.88+
- Symbian installation file
- 4-channel Fasttracker module sound data Title: "MZ`\352\210\360'\315!"
- Scream Tracker Sample adlib drum mono 8bit unpacked
- Poly Tracker PTM Module Title: "MZ`\352\210\360'\315!"
- SNDH Atari ST music
- SoundFX Module sound file
- D64 Image
- Nintendo Wii disc image: "NXSB\030(0\001
- Nintendo 3DS File Archive (CFA)
- Unix Fast File system [v1] (little-endian), ...
- Unix Fast File system [v2] (little-endian), ...
- Unix Fast File system [v2] (little-endian), ...
- ISO 9660 CD-ROM filesystem data (raw boot sector)
- F2FS filesystem, UUID=00000000-0000-0000-0000-000000000000, volume name ""
- DICOM medical imaging data
- Linux kernel ARM boot executable zImage (little-endian)
- CCP4 Electron Density Map
- Ultrix core file from 'X50!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIR...
- VirtualBox Disk Image (MZ`\352\210\360'\315!), 5715999566798081280 bytes
- MS Compress archive data
- AMUSIC Adlib Tracker MS-DOS executable, MZ for MS-DOS COM executable for DOS
- JPEG 2000 image
- ARJ archive data
- unicos (cray) executable
- IBM OS/400 save file data
- data
```

```
00 .M .Z 60 EA .j .P 01 07 19 04 00 10 .S .N .D .H
10 .N .R .0 .0 DC A7 C4 FD 5D 1C 9E A3 .R .E .~ .^
20 .N .X .5 .B 18 28 6F 01 .P .K 03 04 .P .T .M .F
30 .S .y .m .E .x .e .7 .z BC AF 27 1C .S .0 .N .G
40 7F 10 DA BE 00 00 CD 21 .P .K 01 02 .S .C .R .S
50 .R .a .r .! ^Z 07 01 00 .L .R .Z .I .P .L .O .T
60 .% .% .8 .4 .R .a .r .! ^Z 07 00 00 00 .M .A .P
70 .( FD .7 .z .X .Z 00 04 22 4D 18 03 21 4C 18
80 .D .I .C .M .% .P .D .F .- .1 . . .4 . .o .b .j
```

This file is simultaneously detected as:

- DOS EXE, COM and MBR
- Zoo, ARJ, VirtualBox, MS Compress, 3DS
- ISO, RAW ISO, Nero, PhotoCD
- FastTracker, ScreamTracker, Adlib tracker, Polytracker, SoundFX
- Apple, IBM, HP, Linux, Ultrix, Raid, ODS, Nintendo, Kodak
- EZD, CCP4, Plot84, MAR, Dicom

Many magics are
at the start of the file.

output from
file --keep-going

```
0 0x0 Gameboy ROM,, [ROM ONLY], ROM: 256Kbit
80 0x50 RAR archive data, version 5.x
88 0x58 lzrz compressed data
89 0x59 rzrp compressed data - versio
114 0x72 xz compressed data
120 0x78 LZ4 compressed data
...
```

output (150 sigs) from
Binwalk

EACH FORMAT CHARACTERISTIC ENABLES MORE POSSIBILITIES

Formats enforcing magics at offset zero

Footers

Valid at any offset

Formats with cavities
(->zippers)

Format	Signature	Offset
Zip	.XXXX XXXX XX	41
7Z	X.XX XXXX XX	41
Arj	XX.X XXXX XX	41
RAR	XXX.X XXXX XX	41
PDF	XXXX .XXXX	41
ISO	XXXX X.XX	41
DCM	XXXX XX.	37
TAR	XXXX XX.	30
PS	XXXX XXXX	8
MP4	XXXX XXXX	8
AR	XXXX XXXX	8
BMP	XXXX XXXX	7
BZ2	XXXX XXXX	7
CAB	XXXX XXXX	8
CP10	XXXX XXXX	8
EBML	XXXX XX	6
ELF	XXXX XXXX	7
FLV	XXXX XXXX	8
Flac	XXXX XXXX	8
GIF	XXXX XXXX	7
GZ	XXXX XXXXX	8
ICC	XXXX XX	6
ICO	XXXX XXXX	8
ID3v2	XXXX XXXX	8
ILDA	XXXX XXXX	8
JP2	XXXX XXXX	8
JPG	XXXX XXXX	8
NES	XXXX XXXX	7
OGG	XXXX XXXX	8
PSD	XXXX XXXX	8
LNK	XXXX XX	6
PE	XXXX XXXX	7
PNG	XXXX XXXX	8
RIFF	XXXX XXXX	8
RTF	XXXX XXXX	8
TIFF	XXXX XXXX	8
WAD	XXXX XXXX	8
BPG	XXXX XXXX	8
Java	XXXX XXXX	7
PCAP	XXXX XXXX	8
PCAPNG	XXXX XXXX	8
WASM	XXXX XXXX	8
ID3v1		0
XZ		0



```
000: 2031 3233 3435 3637 3839 3031 3233 3435 123456789012345
010: 0a25 5044 462d 312e 3425 2020 2020 2020 .%PDF-1.4%
020: 3031 3233 3435 3637 3839 3031 3233 3435 0123456789012345
030: 0a31 2030 206f 626a 3c3c 2520 2020 2020 .1 0 obj<<%
040: 3031 3233 3435 3637 3839 3031 3233 3435 0123456789012345
050: 0a2f 5479 7065 2f43 6174 616c 6f67 2520 ./Type/Catalog%
060: 3031 3233 3435 3637 3839 3031 3233 3435 0123456789012345
070: 0a2f 5061 6765 7320 3220 3020 5225 2020 ./Pages 2 0 R%
080: 3031 3233 3435 3637 3839 3031 3233 3435 0123456789012345
090: 0a3e 3e65 6e64 6f62 6a0a 2520 2020 2020 .>>endobj.%
...
640: 3031 3233 3435 3637 3839 3031 3233 3435 0123456789012345
650: 0a74 7261 696c 6572 203c 3c25 2020 2020 .trailer <<%
660: 3031 3233 3435 3637 3839 3031 3233 3435 0123456789012345
670: 0a2f 526f 6f74 2031 2030 2052 3e3e 2520 ./Root 1 0 R>>%
680: 3031 3233 3435 3637 3839 3031 3233 3435 0123456789012345
```



A CUSTOM BINARY LASAGNA:

ABUSING LINE COMMENTS AND

INTERLEAVE PDF STATEMENTS W/ ARBITRARY DATA.

```

00: 50 4B 03 04 00 00 00 08 00 00 00 00 00 00 95 19 PK
10: 85 1B 0C 00 00 00 0C 00 00 00 08 00 2E 00 4C 46 . LF
20: 48 20 4E 61 6D 65 75 70 11 00 01 BE A1 2C A5 55 H Nameup , U
30: 6E 69 63 6F 64 65 20 4E 61 6D 65 05 26 15 00 5A nicode Name & Z
40: 50 49 54 08 4D 61 63 20 4E 61 6D 65 5A 49 50 20 PIT Mac NameZIP
50: 53 49 54 78 48 65 6C 6C 6F 20 77 6F 72 6C 64 21 SITxHello world!
60: 50 4B 01 02 00 00 00 00 00 00 00 00 00 00 00 00 PK
70: 95 19 85 1B 0C 00 00 00 0C 00 00 00 09 00 00 00
80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 44 CD
90: 46 48 20 4E 61 6D 65 50 4B 05 06 00 00 00 00 01 FH NamePK
a0: 00 01 00 37 00 00 00 60 00 00 00 00 00

```

DUPLICITY IN ZIPs:

4 NAMES FOR THE SAME ARCHIVED FILE VIA OLDER STRUCTURES.



From near-polyglots to crypto-polyglots

SWAP THE OVERLAP VIA [CRYPTOGRAPHIC] OPERATIONS

En-/de-ryption with specific parameters (IV, Nonce):

Bruteforcing may be required.

Each payload is [partially] hidden
when the other is in clear.

```

00:  B M 3C 00 00 00 00 00 00 00 20 00 00 00 0C 00
89  P N G \r \n ^Z \r 00 00 00 2C c O M M
10:  00 00 0D 00 07 00 01 00 01 00 FF FF FF 00 00 00
20:  00 00 00 00 65 40 00 00 55 40 00 00 67 60 00 00
30:  57 50 00 00 65 60 00 00 00 00 00 00 00 00 00 00
40:  1D 44 05 DC 00 00 00 0D I H D R 00 00 00 0D
50:  00 00 00 07 01 03 00 00 00 00 E9 BE 55 59 00 00 00
60:  06 P L T E FF FF FF 00 00 00 55 C2 D3 7E 00
70:  00 00 1B I D A T 08 1D 63 00 82 54 03 86 70
80:  07 86 F4 02 06 F7 00 06 57 03 06 06 06 00 21 1A
90:  03 10 32 6A 0B 48 00 00 00 00 I E N D AE 42
A0:  60 82

```

```

B M 3C 00 00 00 00 00 00 00 20 00 00 00 0C 00

```



```

89 P N G \r \n ^Z \n 00 00 00 2C c O M M

```



```
mitra.py bmp.bmp png.png --overlap
```

Generates `o(10-40)-PNG[BMP]{424D3C0000000000000002000000000C00}.1965e270.png.bmp`

A BMP/PNG NEAR POLYGLOT, WITH 16 BYTES OF OVERLAP.

A valid BMP is AES-CBC encrypted as a PNG with a special IV to encrypt the first block as expected (AngeCryption).



AngeCryption works with
ECB, CBC, CFB, OFB

```

00:  B  M 3C 00 00 00 00 00 00 20 00 00 00 0C 00
10:  00 00 0D 00 07 00 01 00 01 00 FF FF FF 00 00 00
20:  00 00 00 00 65 40 00 00 55 40 00 00 67 60 00 00
30:  57 50 00 00 65 60 00 00 00 00 00 00 00 00 00 00
40:  00 A1 3B E2 E0 64 F0 A7 AE 5E 21 64 BC 44 5F 09
50:  E3 67 D3 10 19 AF 09 F1 99 1A 33 B3 BF 28 EF 9E
60:  71 3D 87 79 EC 73 A9 60 82 74 1B EB 08 B4 4E B7
70:  E5 9E 16 A9 CE BC 1B 71 99 E7 F8 E8 FA 8C C0 6C
80:  6B 85 4B 56 73 7D 22 BD 46 DE AC 3F BF EE 8B 96
90:  AB 74 55 5F 21 B7 10 1B D6 96 18 45 6E E5 B0 3C
A0:  7C 22 99 87 EA FE 1F 4D FF C8 52 C0 24 C7 AD A8
    
```

AES-CBC
→

```

89  P  N  G \r \n ^Z \n 00 00 00 30 c O M M
71 2F D8 C7 79 C1 EB CF 63 B0 22 2B 0A 6D E3 2D
24 49 57 B1 9B BB C2 FA 94 8A 8C 53 9E A1 30 63
30 C9 41 75 EA AF 75 EE 95 7C 57 E9 16 4F F7 3B
1D 44 05 DC 00 00 00 0D I H D R 00 00 00 0D
00 00 00 07 01 03 00 00 00 E9 BE 55 59 00 00 00
06  P  L  T  E FF FF 00 00 00 55 C2 D3 7E 00
00 00 1B I D A T 08 1D 63 00 82 54 03 86 70
07 86 F4 02 06 F7 00 06 57 03 06 06 06 00 21 1A
03 10 32 6A 0B 48 00 00 00 00 I E N D AE 42
60 82 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    
```

BMP

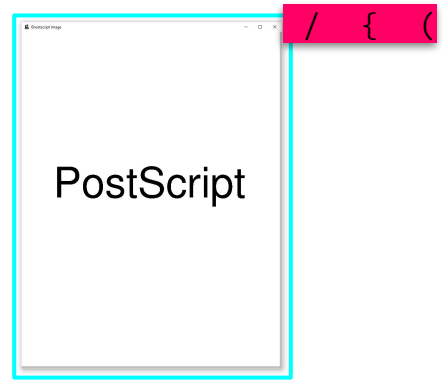
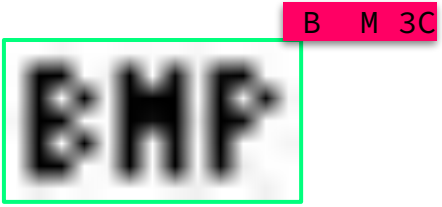
PNG

```
mitra/utls/cbc$ angecrypt.py "0(10-40)-PNG[BMP]{424D3C00000000000000200000000C00}.1965e270.png.bmp" bmp-png.cbc
```

```

00:  / { ( 00 00 00 00 00 00 00 20 00 00 00 0C 00
10:  00 00 0D 00 07 00 01 00 01 00 FF FF FF 00 00 00
20:  00 00 00 00 65 40 00 00 55 40 00 00 67 60 00 00
30:  57 50 00 00 65 60 00 00 00 00 00 00  ) } % !
40:  P S \r \n / N i m b u s S a n s -
50:  R e g u l a r 1 0 0 s e l e
60:  c t f o n t \r \n 7 5 4 0 0 m
70:  o v e t o \r \n ( P o s t S c r i
80:  p t ) s h o w \r \n s h o w p a
90:  g e \r \n s t o p \r \n 00 00 00 00 00 00

```



```
mitra.py postscript.ps bmp.bmp --overlap
```

Generates O(3-3c)-PS[BMP]{424D3C}.209881aa.ps.bmp

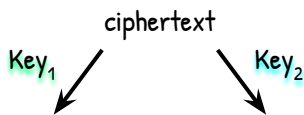
A BMP/PS NEAR POLYGLOT WITH 3 BYTES OF OVERLAP.



TimeCrypton works with
CTR, OFB, GCM, GCM-SIV, OCB3

Both files are decrypted via GCM from the same **ciphertext** but via different keys.

The nonce is bruteforced to generate the right overlap with either key.



```

00:  B M 3C 00 00 00 00 00 00 00 20 00 00 00 0C 00
10:  00 00 0D 00 07 00 01 00 01 00 FF FF FF 00 00 00
20:  00 00 00 00 65 40 00 00 55 40 00 00 67 60 00 00
30:  57 50 00 00 65 60 00 00 00 00 00 00 B7 EB 32 E8
40:  16 D6 9E 76 AC 20 9C 8C 9F 06 6F 55 3F 96 0E 09
50:  04 24 41 5D 22 7C A6 E5 0E AC ED 1C 04 65 BE E6
60:  E8 AB E4 D2 C6 B6 CD 9F AB 85 E1 CE 03 C5 A5 85
70:  70 B5 09 EB EB CB D1 2F 7C 4D B0 09 35 38 D9 B7
80:  82 31 BB 87 96 22 C8 4E C0 EC 89 C3 CB 97 63 D3
90:  A0 28 47 5B 71 C2 95 EC 12 E2 52 B0 6F B1 EE 61

```

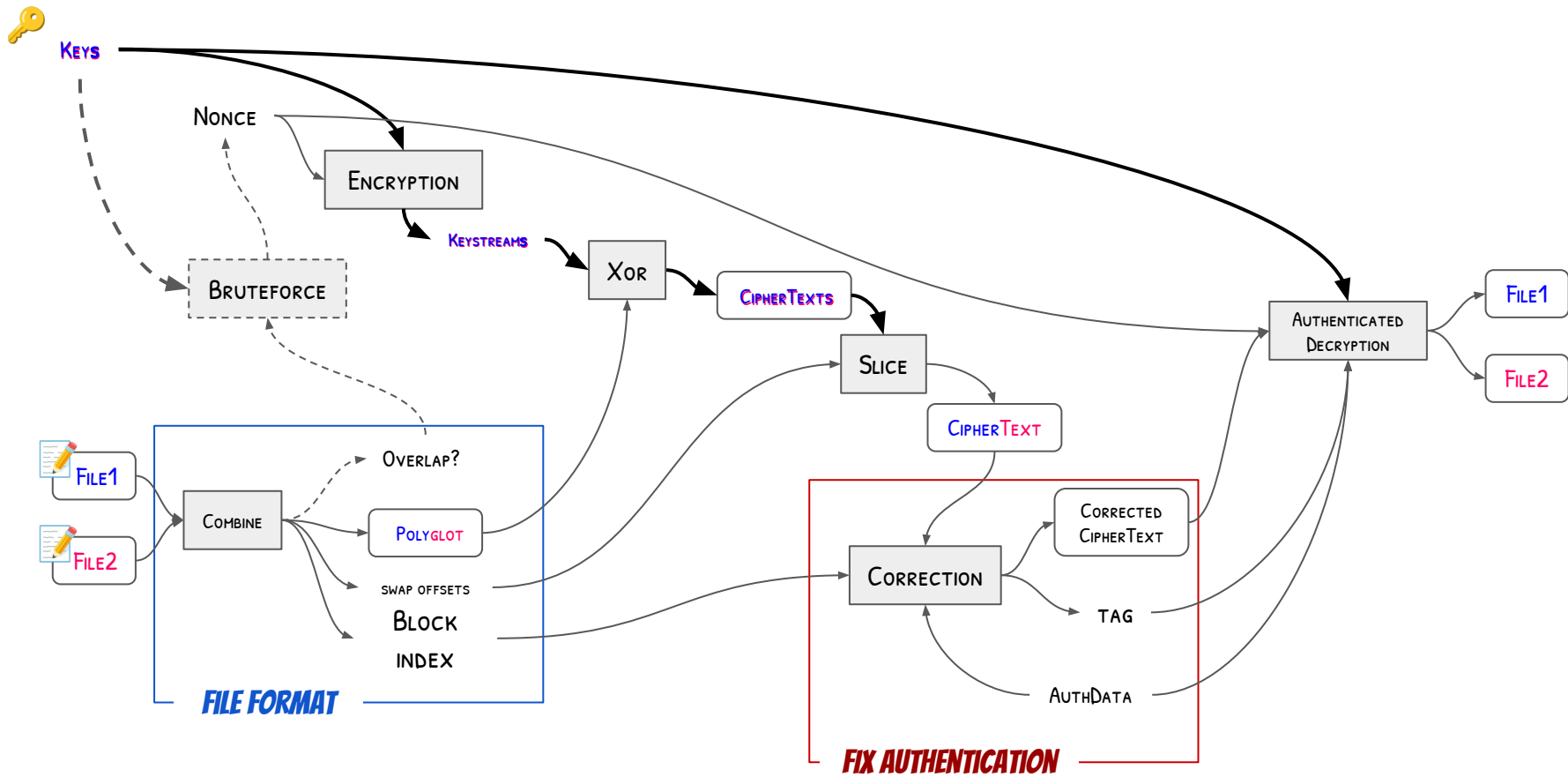
```

/ { ( 07 3A 14 40 E5 3E EC AE A2 AD 87 AA 38
11 C4 5D 5A 35 2D EB EC 47 CC A7 B5 63 22 90 B7
5F D7 41 7B FD 6D 53 DB 78 9F AA A6 2B 22 61 AD
BB 38 48 4A 5C A7 D5 E4 63 4F 4D 7B ) } % !
P S \r \n / N i m b u s S a n s -
R e g u l a r 1 0 0 s e l e
c t f o n t \r \n 7 5 4 0 0 m
o v e t o \r \n ( P o s t S c r i
p t ) s h o w \r \n s h o w p a
g e \r \n s t o p \r \n 00 00 00 00 00 00

```



```
mitra/utils/gcm$ meringue.py "0(3-3c)-PS[BMP]{424D3C}.209881aa.ps.bmp" bmp-ps.gcm
```



UNDER THE HOOD - CHECK [MITRA](#) & [KEYCOM](#).

```
$ wget https://eprint.iacr.org/2020/1456.pdf
[...]

$ openssl enc -in 1456.pdf -out crypted \
  -aes-128-ctr -iv 00000000000000000000e7c600000002 \
  -K 4e6f773f000000000000000000000000

$ openssl enc -in crypted -out viewer.exe \
  -aes-128-ctr -iv 00000000000000000000e7c600000002 \
  -K 4c347433722121210000000000000000

$ wine viewer.exe 1456.pdf
```

OUR PDF ARTICLE FILE IS ALSO
A PDF VIEWER EXECUTABLE!
VIA AUTHENTICATED ENCRYPTION.

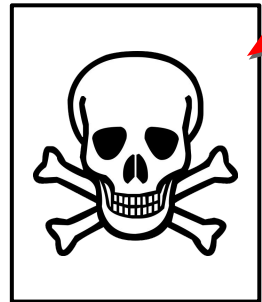
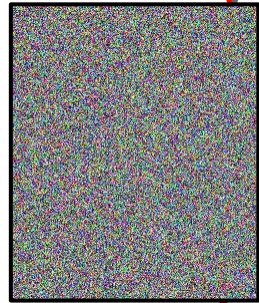
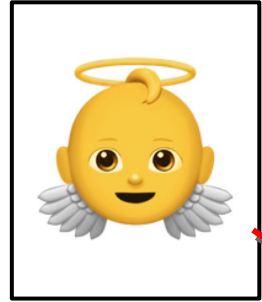


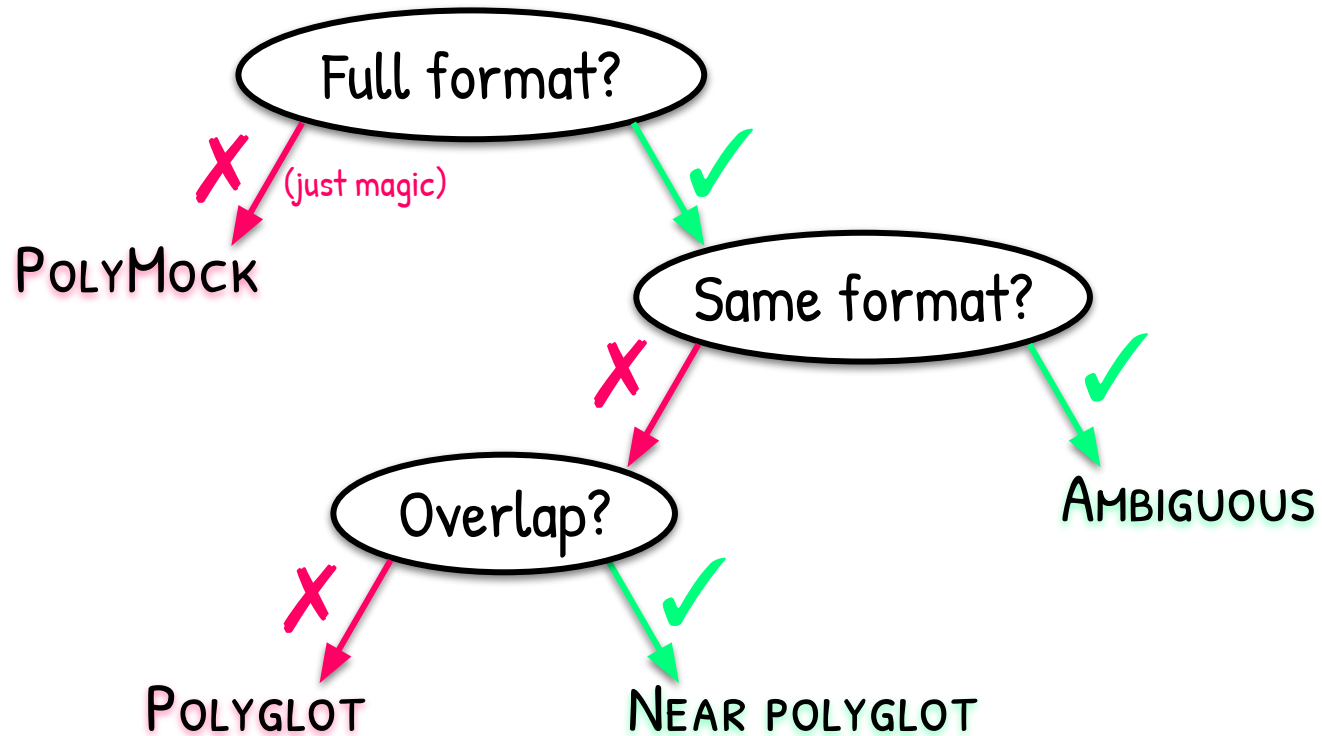
~~TIMECRYPTION~~

Without key commitment,
a ciphertext can be crafted to decrypt **with authentication**
to different payloads.

Vulnerabilities @ Facebook, Amazon, Google...

With key management: friendly today, evil tomorrow.





A HIERARCHY OF WEIRD FILES

File formats challenges in 2024

PRIVATE INFORMATION
CAN LEAK AT CLOUD'S SCALE.

Leaked credentials are abused within minutes.

Keys, login/passwords, cookies.

A single "minor" bug can affect billions of users!

ACROP ALYPSE

Just standard PNG files.

Cropped by the user.

The smaller file is kept with trailing data leftovers.

-> major leak of information for users.



by Simon Aarons and David Buchanan.

SQLITE: DATA LEAKS IN PLAIN SIGHT

Magic: "SQLite format 3\0" (16 bytes)

-> very strong identification.

No easy subtype-identification: the Application ID is rarely used.

Is it a standard assets index ? A mountable filesystem?

Cookies / web history / credit cards / bitcoin wallet ?

-> Identification tool: [sqlbuddy.py](https://github.com/0x00sec/sqlbuddy.py)

HASH COLLISIONS

Collisions in 2024?

SOME AVs DETECT THE EICAR FILE BY CRC32!

Who needs
cryptographic hashes
for collisions?

X5O!P%@AP[4\PZX54(P^)7CC)7}\$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!\$H*H*

or

DpVRUX<=EICAR CRC collision? Use Shake128/Kangaroo12/Blake3 instead!

Same CRC32

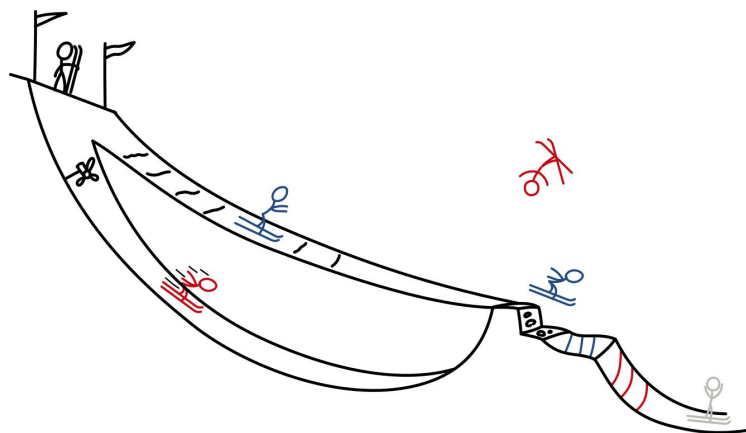
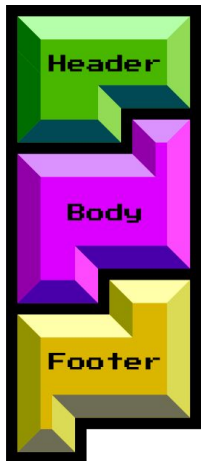
Script: mycar.sh

12 / 63
Community Score

⚠️ 12/63 security vendors flagged this file as malicious

6534dd9e2904be289ccaa8c26f409bce1851940975e22714681f90e6e4ed571d
mycar.zip
zip

CRYPTOGRAPHIC HASH COLLISIONS REQUIRE ACROBATIC CONSTRUCTS.



TROPHY WALL



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)

☆ Star 3.1k

DETECTING COLLISIONS W/ SIGNATURES

DetectColl can detect any MD5 or SHA1 hash collision.

```
$ detectcoll_unsafe flame.der | ./logparse.py  
flame.der  
block: 11, collision: Flame
```

Flame's unique collision.

```
$ detectcoll 13-shambles1.bin | ./logparse.py  
13-shambles1.bin  
block: 9, collision: SHattered/Shambles
```

Newest SHA1's: Shambles

HASHQUINES

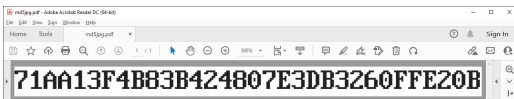
Chain 128-512 collisions
to change the displayed hash
but keep the file hash constant.

PoC||GTFO
ISSUE 0X14

- 14: 02 Z-RING PHREAKING
- 14: 03 DESERT STUDIES
- 14: 04 FLUSH+RELOAD ATTACKS
- 14: 05 ANTI-KEYLOGGING
- 14: 06 RANDOM NOPS ON ARM
- 14: 07 ETHERNET OVER GDB
- 14: 08 CONTROL PANEL VULNS
- 14: 09 MDS POSTSCRIPT
- 14: 10 MDS PDF
- 14: 11 MDS GIF
- 14: 12 YOU'RE LOOKING AT IT

GOTT BEWAHRE MICH VOR JEMAND,
DER NUR EIN
BÜCHLEIN GELESEN HAT;
370 CAMH3AAT

MDS: DB 66 9E 2F 3E B6 26 15
B7 B8 0E 6D 86 2C 58 22



PoC||GTFO

**PASTOR LAPHROAIG SCREAMS
HIGH TO THE HEAVENS
AS THE WHOLE WORLD GOES UNDER**

- 1400 Z-Ring Phreaking
- 1400 Covering Desert Studies
- 1400 Flush+Reload Side-Channel Attacks
- 1400 Anti-Keyboard with Random Nops
- 1400 Random NOPS on ARM
- 1407 Ethernet Over GDB

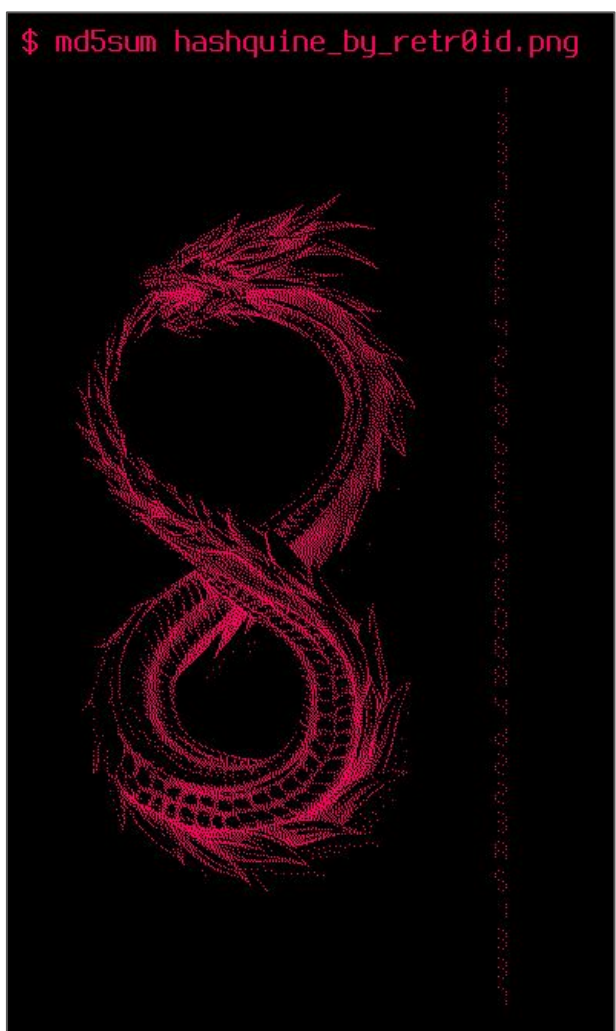
- 1408 Control Panel Vulnerabilities
- 1409 MDS Postscript
- 1410 MDS PDF
- 1411 MDS GIF
- 1412 This PDF is an NES MDS Quine

Gott bewahre mich vor jemand, der nur ein Büchlein gelesen hat: 370 camh3aat.
The MDS hash of this PDF is 58AF0028C1423258A51A50B126746C. March 20, 2017.
€ 0, 80 USD, 80 AUD, 10s 6d GBP, 0 BSD, 0 SEK, 850 CAD, 6 × 10⁹ Prang (3 × 10⁸ Adiposus).

Ghostscript Image

768d9d89d2bc825a319eb8962ad30580

R GIF MD5 hashquine
Copyheart Rogdham, 2017
22d058dd8aad588cadedf33e6c9977e



RETR0ID'S HASHQUINE ARCHIVE (2023)

A generic tar file contains a hash list with the hash of the whole archive.

```
$ tar -xvf self.tar.zst
x hash.md5
x hello.txt
```

```
$ md5sum -c hash.md5
self.tar.zst: OK
hello.txt: OK
```

The file is "building" a Tar header via 653 MD5 collisions abusing ZStandard frames.

Explanations on [github / corkami / collisions / hashquines](#)

AND YET...

"IF IT'S NOT BROKEN IN PRACTICE...
...IT MUST BE GOOD ENOUGH!"

23 October 2023

Expires: 25 April 2024

Deprecating Insecure Practices in RADIUS

While MD5 has been broken, it is a **testament** to the design of RADIUS that there have been (as yet) no attacks on RADIUS Authenticator signatures which are stronger than brute-force.

<https://www.ietf.org/archive/id/draft-dekok-radext-deprecating-radius-05.txt>



New MD5 attack - June 2024

FORMATS ARE COMPLEX. FILES ARE LAYERED.

Archive, stack, encapsulation, compressions...

One format may be robust against some attacks,
but its inner/outer format or side format
might make the whole system vulnerable.

SHATTERED



First SHA1 collision on PDF files.

It wasn't a PDF collision

as PDF parsers can't be reliably collided with SHA1.

-> Abuse JPG in PDF as JPG can be collided reliably.

This would likely work in any format based on JPG.

File 1

File 2

Identical prefix

Collision blocks

Suffix

```

000: 2550 4446 2d31 2e33 0a25 e2e3 cfd3 0a0a %PDF-1.3%.
010: 0a31 2030 206f 626a 0a3c 3c2f 5769 6474 .1 0 obj.<</Wid
020: 6820 3220 3020 522f 4865 6967 6874 2033 h 2 0 R/Height 3
030: 2030 2052 2f54 7970 6520 3420 3020 522f 0 R/Type 4 0 R/
040: 5375 6274 7970 6520 3520 3020 522f 4669 Subtype 5 0 R/Fi
050: 6c74 6572 2036 2030 2052 2f43 6f6c 6f72 lter 6 0 R/Color
060: 5370 6163 6520 3720 3020 522f 4c65 6e67 Space 7 0 R/Leng
070: 7468 2038 2030 2052 2f42 6974 7350 6572 th 8 0 R/BitsPer
080: 436f 6d70 6f6e 656e 7420 383e 3e0a 7374 Component 8>>.st
090: 7265 616d 0aff d8ff eam.....$SHA-1
0a0: 2069 7320 6465 6164 is dead!!!!!./
0b0: 0923 3975 9c39 b1a1 c63c 4c97 e1ff fe01 #9u.9...<L....
0c0: 7f46 dc93 a6b6 7e01 3b02 9aaa 1db2 560b F....~.;....V.
0d0: 45ca 67d6 88c7 f84b 8c4c 791f e02b 3df6 l.g....K.Ly..+=.
0e0: 14f8 6db1 6909 01c5 6b45 c153 0afe dfb7 .m.i....kE.S...
0f0: 6038 e972 722f e7ad 728f 0e49 04e0 46c2 8.rr./...I..F.
100: 3057 0fe9 d413 98ab e12e f5bc 942b e335 0W.....+.5
110: 42a4 802d 98b5 d70f 2a33 2ec3 7fac 3514 B.....*3....5.
120: e74d dc0f 2cc1 a874 cd0c 7830 5a21 5666 .M....t..x0Z!Vd
130: 6130 9789 606b d0bf 3f98 cda8 0446 2911 a0...`k..?....F).
-----
230: 0000 fffe 012d 0000 0000 0000 0000 ffe0 .....
240: 0010 4a46 4946 0001 0101 0048 0048 0000 ..JFIF.....H.H..
3a0: e9d6 d667 a7b0 7e65 1299 e39d 39c0 c7ff ...g...~e....9...
3b0: d92d 2d2d 2dff e000 104a 4649 4600 0101 -----JFIF...
3c0: 0100 4800 4800 00ff db00 4300 0101 0101 ..H.H.....C.....
4e0: 4b14 97f7 7f39 fcd7 f1ff d90a 656e 6473 K...9.....ends
4f0: 7472 6561 6d0a 656e 646f 626a 0a0a 3220 tream.endobj..2
500: 3020 6f62 6a0a 380a 656e 646f 626a 0a0a 0 obj.8.endobj..
840: 3e0a 0a73 7461 7274 7872 6566 0a31 3830 >..startxref.180
850: 380a 2525 454f 460a 8.%%EOF.

```

```

PDF header 2550 4446 2d31 2e33 0a25 e2e3 cfd3 0a0a %PDF-1.3%.
0a31 2030 206f 626a 0a3c 3c2f 5769 6474 .1 0 obj.<</Wid
6820 3220 3020 522f 4865 6967 6874 2033 h 2 0 R/Height 3
2030 2052 2f54 7970 6520 3420 3020 522f 0 R/Type 4 0 R/
image object 5375 6274 7970 6520 3520 3020 522f 4669 Subtype 5 0 R/Fi
declaration 6c74 6572 2036 2030 2052 2f43 6f6c 6f72 lter 6 0 R/Color
5370 6163 6520 3720 3020 522f 4c65 6e67 Space 7 0 R/Leng
7468 2038 2030 2052 2f42 6974 7350 6572 th 8 0 R/BitsPer
436f 6d70 6f6e 656e 7420 383e 3e0a 7374 Component 8>>.st
JPG header and 7265 616d 0aff d8ff eam.....$SHA-1
comment length: 0x0173 is dead!!!!!./
comment length: 0x0173
2069 7320 6465 6164 is dead!!!!!./
0923 3975 9c39 b1a1 c63c 4c97 e1ff fe01 #9u.9...<L....
7f46 dc91 66b6 7e11 8f02 9ab6 21b2 560f sF..f.....!V.
f9ca 67cc a8c7 f85b a84c 7903 0c2b 3de2 .g....[.Ly..+=.
18f8 6db3 a909 01d5 df45 c141 26fe dfb3 .m.....E.0k...
dc38 e96a c22f e7bd 728f 0e45 bce0 46d2 .8.j./...E..F.
3c57 0feb 1413 98bb 552e f5a0 a82b e331 <W.....U....+1
fea4 8037 b965 d71f 0e33 2edf 93ac 3500 ...7.....3....5.
eb4d dc0d ecc1 a864 790c 782c 7621 5660 .M....dy.x,v!V
dd30 97f1 d06b d0af 3f98 cda4 bc46 29b1 .0...k..?....F).
-----
0000 fffe 012d 0000 0000 0000 0000 ffe0 .....
0010 4a46 4946 0001 0101 0048 0048 0000 ..JFIF.....H.H..
comments' chain
e9d6 d667 a7b0 7e65 1299 e39d 39c0 c7ff ...g...~e....9...
d92d 2d2d 2dff e000 104a 4649 4600 0101 -----JFIF...
0100 4800 4800 00ff db00 4300 0101 0101 ..H.H.....C.....
first image data (ignored)
e9d6 d667 a7b0 7e65 1299 e39d 39c0 c7ff ...g...~e....9...
d92d 2d2d 2dff e000 104a 4649 4600 0101 -----JFIF...
0100 4800 4800 00ff db00 4300 0101 0101 ..H.H.....C.....
second image data (ignored)
4b14 97f7 7f39 fcd7 f1ff d90a 656e 6473 K...9.....ends
7472 6561 6d0a 656e 646f 626a 0a0a 3220 tream.endobj..2
3020 6f62 6a0a 380a 656e 646f 626a 0a0a 0 obj.8.endobj..
PDF footer
3e0a 0a73 7461 7274 7872 6566 0a31 3830 >..startxref.180
380a 2525 454f 460a 8.%%EOF.

```

same hash at this point



SHATTERED FILES LAYOUT: A NORMAL PDF WITH A FUNKY JPG.

INSIDE OUT

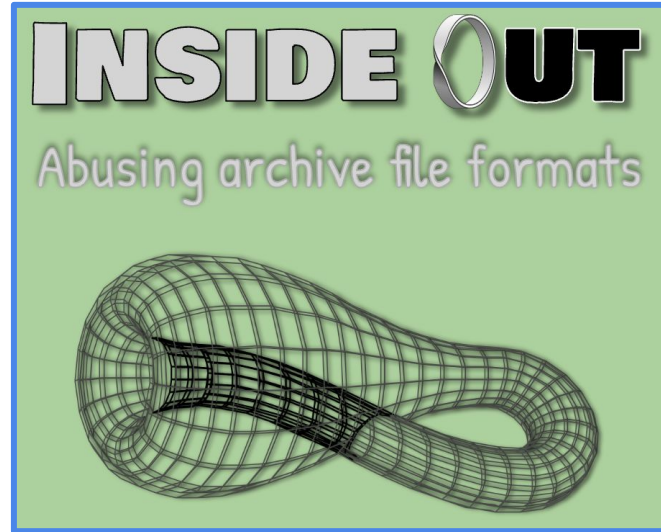
Abusing .docx:

- XML can't tolerate collision blocks.
- ZIPs can't be collided generically.
- > Abuse XML in Zips via Zip structures.

Abusing .tar.gz:

(Tar can't be collided generically)

- > Abuse Gzip structure to show different Tar contents.



TAR HASHQUINES

Tar can't be collided generically:

-> Abuse GZip/L4/Zstandard archive structure
to present different archived file contents
to external parsers.

Let's talk **seriously** about...

AI

No hype, no fake,
no empty promises.



FILE FORMATS... AND AI ?

"Who needs AI to check a magic signature?"
"It won't catch polyglots anyway."

But...

What about source files?

Or *any* kind of attachments?

...

MAGIKA

200+ formats: text **and** binary.

Small model: runs on CPU, needs 1 Mb.

Fast: <5ms per file. Used in production on 100s of billion files weekly.
Used in 150+ projects.

Open-source: Python, Go, Rust, JavaScript.

<https://github.com/google/magika>

☆ Star 8.2k

Paper (ICSE 25) <https://arxiv.org/abs/2409.13768>

Non-generative AI: no copyright infringement, just a detection verdict.



THE 'MAGIKA IN PRODUCTION' EFFECT...

How many file formats overall? Who knows... 🤯🙄

Each community have its weirdnesses, overlaps,
do's and don'ts.

"What a mess 🤞"

NO SILVER BULLET

It doesn't scan the whole file (only the first & last 2Kbs).

Not enough samples to train on many formats.

Standard AI limitations: no editing / omitting.

May fail on weird files 😊

May catch corrupted/spoofed files:
-> useful for carving, recovery-abuse.

-> Remove the first 16 bytes, then re-scan.

MAGIKA ON CORRUPTED FILES

A ZIP with invalid signatures:

An invalid file recovered by applications.

-> scanning bypass.

```
$ file badsigns.zip
```

```
badsigns.zip:      data
```

```
$ magika badsigns.zip -s
```

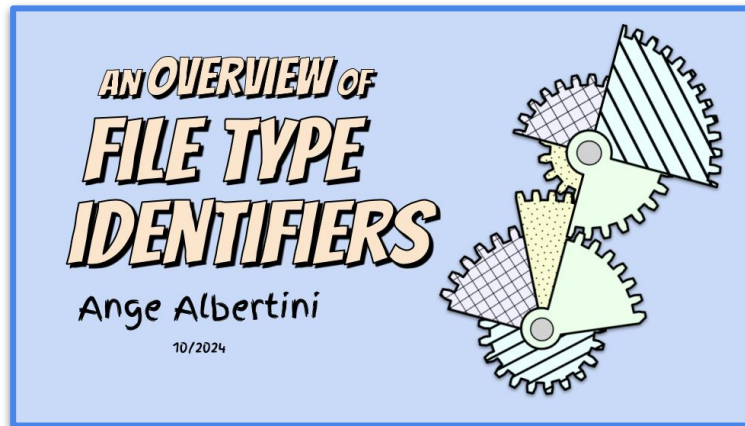
```
badsigns.zip: Zip archive data (archive) 98%
```

```
00  .B .K \3 \4 0a 00 00 00 00 00 00 00 00 00 23 8e
10  5a 6b 05 00 00 00 05 00 00 00 07 00 00 00 .z .i
20  .p .. .t .x .t .Z .I .P \r \n .B .K 01 02 1f 00
30  0a 00 00 00 00 00 00 00 00 00 23 8e 5a 6b 05 00
40  00 00 05 00 00 00 07 00 00 00 00 00 00 00 00
50  00 00 00 00 00 00 00 00 00 .z .i .p .. .t .x .t .B
60  .K 05 06 00 00 00 00 01 00 01 00 35 00 00 00 2a
70  00 00 00 00 00
```

MAGIKA IS NEW & DIFFERENT, AND USEFUL IN ITS OWN WAY.

Planning to make a new engine?

-> Investigate all existing ones,
then give a talk on the topic ->



FOOL AI IDENTIFICATION?

Some formats give you full control over the first X bytes.
Most make it possible to insert exploitable contents early.

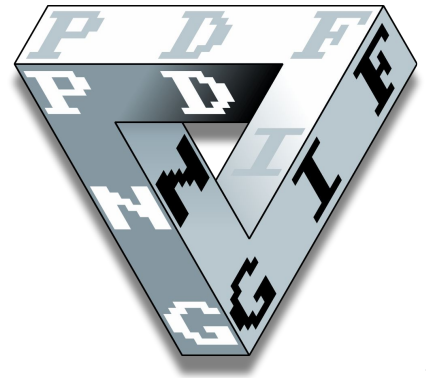
Use Mitra to insert 1 kb of free space in your file:

```
mitra.py <inputfile> /dev/null --pad 1 -f
```

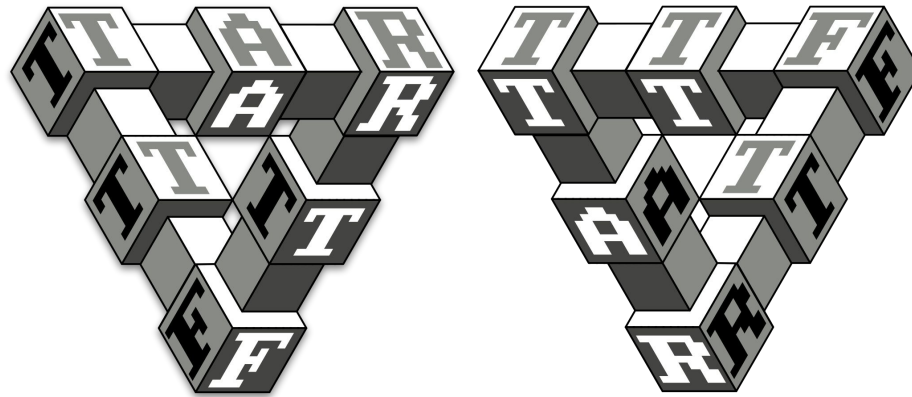
Use Mocky to insert dummy signatures:

```
mocky.py <inputfile> --combined
```

Mocky & Mitra @ [Github corkami/mitra](https://github.com/corkami/mitra)



CONCLUSION



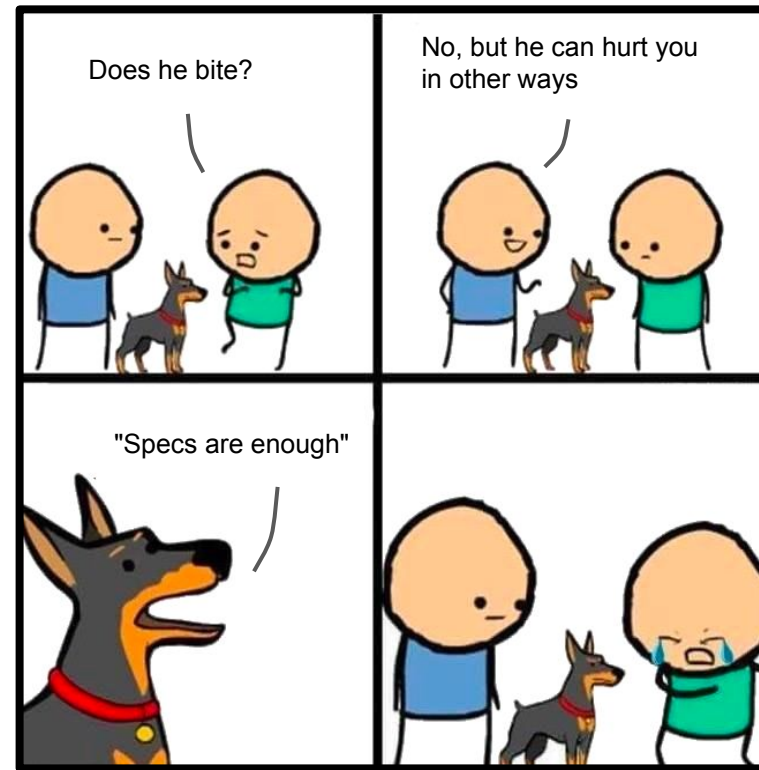
In 2024...

Many old tricks still work.

Specifications can still be naive or laughable.

No reference code, no test cases.

No incentive to fix anything if it's not a security bug.



-> back to the eternal: "let's check wikipedia..." ?

FROM FUNKY PoCs TO FEARSOME TOOLS.

Working at scale with new tools:

- 100s of collisions possibilities
- 1000s of polyglot combinations
- 100s of billions of scanned files by AI.

AI & FILE FORMATS

- Many AI formats are vulnerable.
- Magika brings something new to file format processing.
- Mitra can be used to inject arbitrary data in formats (and fool AI).

ROOM FOR IMPROVMENT

- Specifications writing and updating.
- Sample crafting and sharing.
- Format identification and heuristics.
- Format classifying and rating.

Commandments of a good file format

Magic at offset zero

fast identification, no bypass

Clear chunk structure

forward compatibility, easy parsing/cleanup

Version number

Forward thinking

No duplicity

Duplicity → discrepancy 🦴

No "constant" variables

Ossification → hardcoding

Up-to-date specs

Reflect reality

Samples set

Theory isn't enough

Extensibility

Your format will evolve in unknown ways

Keep the spirit

*Don't reuse formats for different intent
without trivial distinction*

Perfect is the enemy of good

*Shortcuts will be taken
to avoid over-complexity.*

THANKS FOR YOUR ATTENTION!

ACKNOWLEDGEMENTS:

MARC STEVENS, PHILIPPE TEUWEN, STEFAN KÖLBL, ATUL LUYKX, DANIEL BLEICHENBACHER, DAVID BUCHANAN, SOPHIE SCHMIEG, YANICK FRATANTONIO, AND THE FABIANIS.

ANGE ALBERTINI
reverse engineering
VISUAL DOCUMENTATIONS

[@angealbertini](https://twitter.com/angealbertini)
ange@corkami.com
<http://www.corkami.com>

