



# Fast fingerprinting of OLE2 files:

heuristics for detection of exploited OLE2 files based on specification non-conformance

# Fast fingerprinting of OLE2 files

Authors

- Stephen Edwards, SophosLabs UK
- Paul Baccas, SophosLabs UK
- {stephen.edwards, paul.baccas}@sophos.com

# Fast fingerprinting of OLE2 files

## Menu

- What were we trying to achieve?
- How we went about it?
- What we found?
- What happened next?

# Fast fingerprinting of OLE2 files

What were we trying to achieve?

- Troj/DocDrop-S
- Binary specification
- Prototyped a non-conformance scanner

# Fast fingerprinting of OLE2 files

What were we trying to achieve?

- Fast method to hash on files that we wanted to spend time on.
- Differentiate different threats

# Fast fingerprinting of OLE2 files

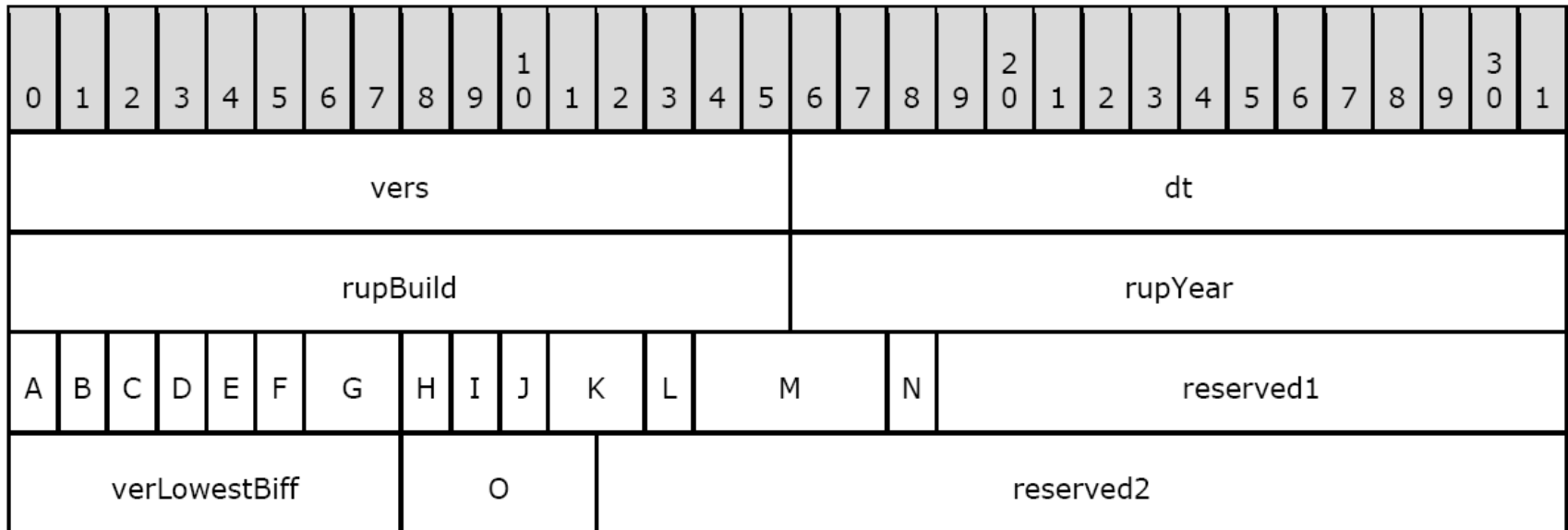
How we went about it?

- Read the Microsoft Specifications
- Initially we implemented the detection as an internal detection
- Implemented it in python

# Fast fingerprinting of OLE2 files

How we went about it?

## BOF record – 128 bits



Source: <http://msdn.microsoft.com/en-us/library/cc313154%28v=office.12%29.aspx>

# Fast fingerprinting of OLE2 files

How we went about it?

**BOF record** – 128 bits : 104 can violate the specification

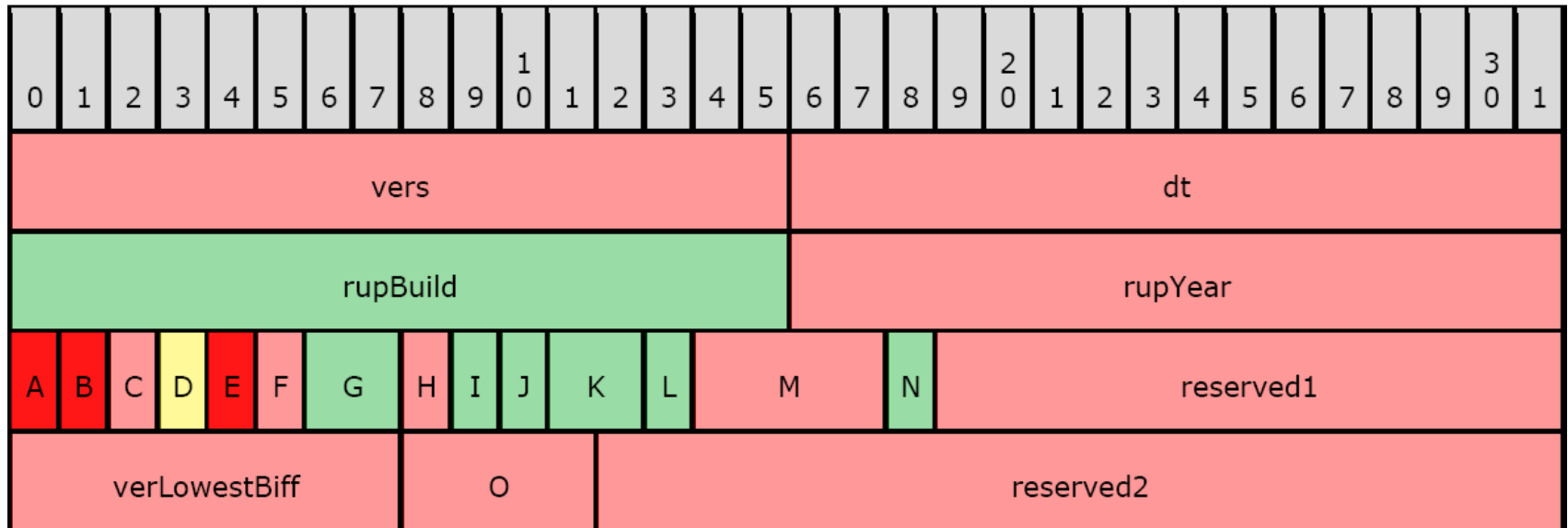


Source: <http://msdn.microsoft.com/en-us/library/cc313154%28v=office.12%29.aspx>



# Fast fingerprinting of OLE2 files

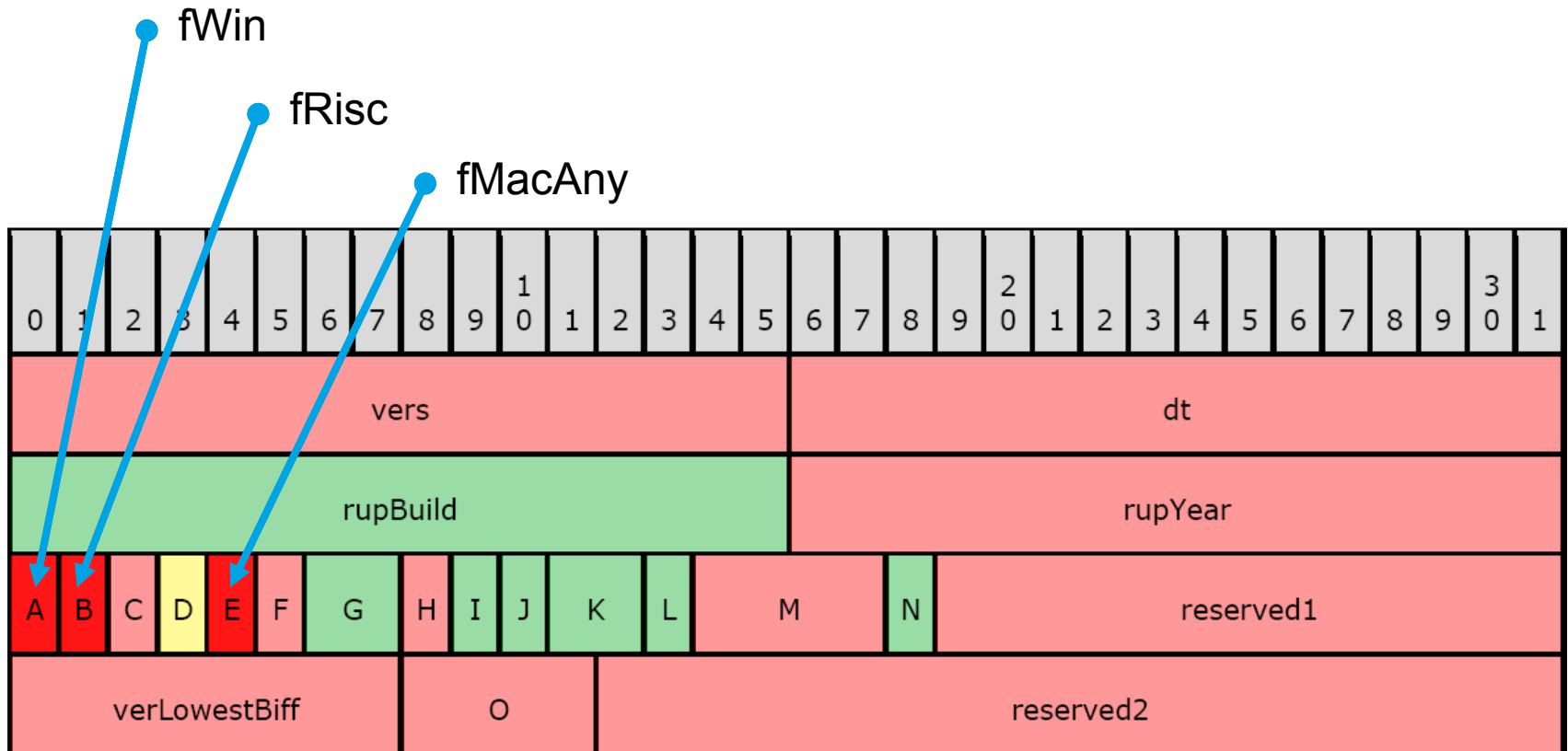
How we went about it?



Source: <http://msdn.microsoft.com/en-us/library/cc313154%28v=office.12%29.aspx>

# Fast fingerprinting of OLE2 files

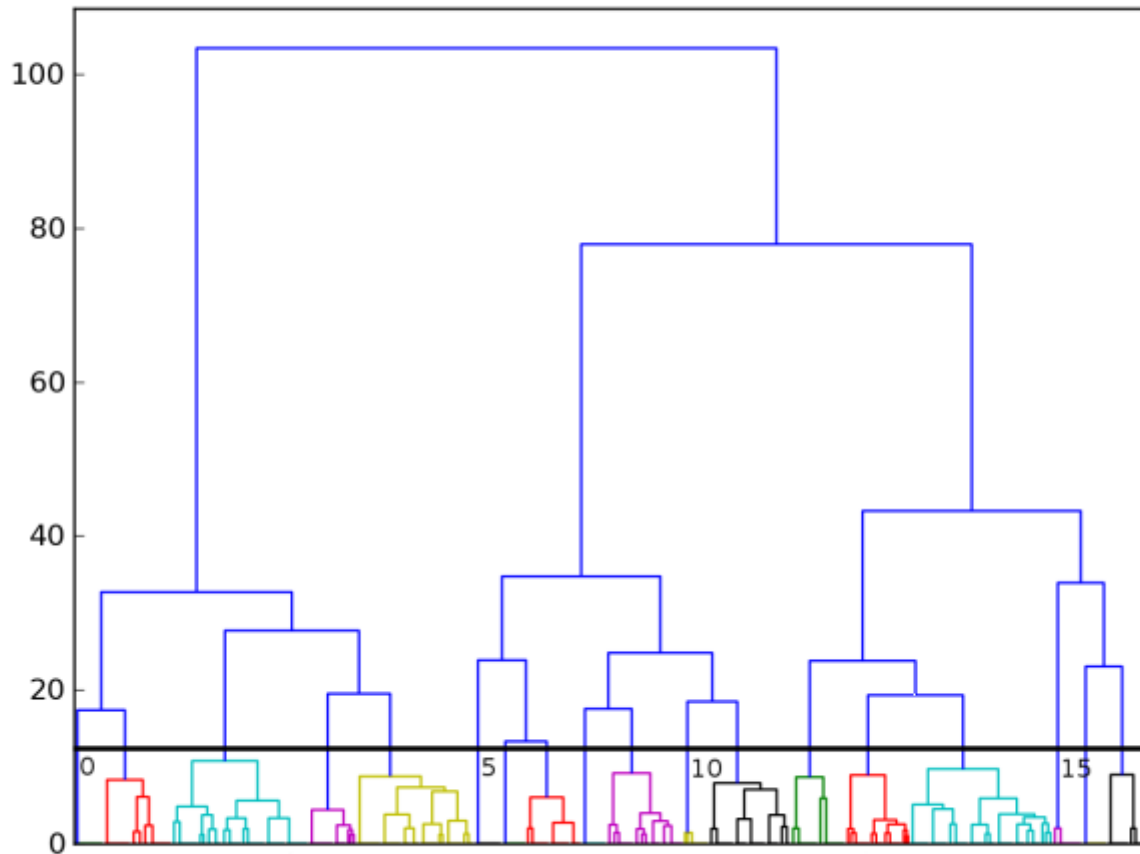
How we went about it?



Source: <http://msdn.microsoft.com/en-us/library/cc313154%28v=office.12%29.aspx>

# Fast fingerprinting of OLE2 files

What we found? Clustering



# Fast fingerprinting of OLE2 files

What we found?

- Troj/DocDrop-S grouped well
- XM97/Hidemod-A also grouped

# Fast fingerprinting of OLE2 files

What we found? Troj/DocDrop-S

- [CVE-2009-3129](#)
- [MS09-067](#)
- Interesting [sample](#) names
- Some more on my [blog](#)

F

Wh

00000a00	E0	07	00	00	53	4D	54	4A	00	00	00	00	10	00	D0	07	...	SMTJ	...
00000a10	48	00	50	00	20	00	4C	00	61	00	73	00	65	00	72	00	H.P.	L.a.s.e.r.	
00000a20	4A	00	65	00	74	00	20	00	50	00	32	00	30	00	31	00	J.e.t.	P.2.0.1.	
00000a30	35	00	20	00	53	00	65	00	72	00	69	00	65	00	73	00	5.	S.e.r.i.e.s.	
00000a40	20	00	50	00	43	00	4C	00	20	00	35	00	65	00	00	00		P.C.L.	5.e...
00000a50	49	6E	70	75	74	42	69	6E	00	46	4F	52	4D	53	4F	55	InputBin	FORMSOU	
00000a60	52	43	45	00	52	45	53	44	4C	4C	00	55	6E	69	72	65	RCE	RESDLL	Unire
00000a70	73	44	4C	4C	00	48	50	50	43	6F	6D	70	6F	73	69	74	sDLL	HPPComposit	
00000a80	65	55	53	42	47	65	74	53	65	72	69	61	6C	4E	75	6D	eUSBGetSerialNum		
00000a90	62	65	72	00	68	70	70	64	76	71	30	31	2E	64	6C	6C	ber	hppdvq01.dll	
00000aa0	5F	67	65	74	44	65	76	69	63	65	53	65	72	69	61	6C	_getDeviceSerial		
00000ab0	4E	75	6D	62	65	72	57	00	48	50	50	49	D6	77	13	00	NumberW	HPPI.w...	
00000ac0	B2	77	13	00	DE	77	13	00	BA	77	13	00	E6	77	13	00	.w...		
00000ad0	EB	1A	71	30	31	2E	64	6C	D6	77	13	00	B2	77	13	00	..q01.dl.w...		
00000ae0	DE	77	13	00	BA	77	13	00	E6	77	13	00	EB	22	57	00	.w...	"W	
00000af0	52	65	73	6F	6C	75	74	69	6F	6E	00	36	30	30	64	70	Resolution	600dp	
00000b00	69	00	4F	72	69	65	6E	74	61	74	69	6F	6E	00	50	4F	i.Orientation	PO	
00000b10	33	C0	66	B8	9D	01	91	EB	12	00	33	C0	5A	B0	FF	49	3.f	.....3.Z..I	
00000b20	30	04	0A	FE	C8	85	C9	75	F6	EB	06	FC	E8	E9	FF	FF	0	.....u	.....
00000b30	FF	36	EF	89	E7	8B	48	68	6A	6B	E7	99	86	30	71	71	.6	...Hhjk...0qq	
00000b40	72	F8	8C	9D	B2	77	78	79	F3	7D	F5	23	7A	80	B6	E9	r	...wxy}.#z...	
00000b50	2F	18	F9	5A	6E	63	88	89	8A	02	CA	85	71	B9	F8	B8	/..Znc	.....q...	
00000b60	80	55	C2	7D	43	97	98	99	13	DD	90	62	A8	F7	F9	9D	.U}	C.....b	.....
00000b70	A5	D8	4C	63	A6	A7	A8	20	EC	BB	53	DA	AA	50	C6	B9	..Lc	.....S..P..	
00000b80	5A	D5	B4	B5	B6	3E	FE	AD	D0	BB	D6	BD	D4	BF	AA	C5	Z	....>	.....
00000b90	A8	C3	3B	B3	D2	38	9E	C5	43	8D	D4	A7	CE	A5	D0	BB	...	8..C	.....
00000ba0	D2	B9	D2	2A	A0	CF	27	8F	CA	52	9A	C1	67	CB	E4	E1	...*	...'R.g...	
00000bb0	E2	E0	E3	AC	AF	AE	A1	68	DE	E3	B0	80	93	62	75	38	.....h	...bu8	
00000bc0	87	02	0B	C3	09	81	FC	72	BC	E7	77	2D	FD	F8	FF	76	.....r	w-...v	
00000bd0	16	88	CE	06	49	17	59	F6	7D	07	87	C7	0D	40	18	40	...I	Y}...@	@
00000be0	9F	5C	0C	44	E9	61	04	92	E6	E4	FC	2E	D7	94	74	05	.\D	a	.....t
00000bf0	2A	A8	78	01	22	76	7A	43	2A	7A	D3	FE	74	76	0B	F3	*.x	"vzC*z	..tv..
00000c00	46	36	B7	F4	32	DC	D6	B2	FB	F9	34	3D	6B	69	24	E0	F6	..2	....4=ki\$
00000c10	72	43	44	45	C3	87	30	5A	74	C0	0C	41	70	C4	20	4D	rCDE	..0Zt	..Ap..M
00000c20	6C	D8	0A	5D	FB	69	D3	31	52	B0	51	63	D5	1F	54	5F	l	..]	i.1R.Qc..T
00000c30	E9	CB	DC	65	66	67	5B	B2	E1	AE	32	30	AD	3C	25	27	...	efg[	...20.<%'
00000c40	25	45	FF	19	52	6F	4E	F2	3F	47	4A	F6	2A	7A	F8	82	%E	..RoN	?GJ.*z..
00000c50	57	BD	0F	CF	9E	B9	03	D3	AA	88	51	6E	B6	C6	AE	1A	W	.....Qn	....
00000c60	A6	18	97	60	A5	68	64	AA	5A	37	A6	59	EA	98	61	6E	...	hd.Z7.Y	..an
00000c70	AF	A0	5C	4E	54	91	93	D5	8E	BF	D9	72	90	24	EA	95	..\NT	.....r	.\$..
00000c80	B1	6E	D2	8B	3D	BB	F3	87	31	E1	A0	BE	63	81	4B	C5	.n	..=	...1...c.K
00000c90	49	C0	01	2E	C4	F4	08	42	1F	94	92	90	95	0D	D8	D1	I	.....B	.....
00000ca0	3A	D3	D4	D5	D6	8F	5B	19	DF	18	DC	BB	DE	DF	F7	6C	.....[	.....l	
00000cb0	E3	E3	F3	8F	E6	E7	E8	24	EA	EB	FB	DA	EF	EF	F0	A7	.....\$		

F

Wh

```

00000a00  E0 07 00 00 53 4D 54 4A 00 00 00 10 00 D0 07  ....SMTJ.....
00000a10  48 00 50 00 20 00 4C 00 61 00 73 00 65 00 72 00  H.P. .L.a.s.e.r.
00000a20  4A 00 65 00 74 00 20 00 50 00 32 00 30 00 31 00  J.e.t. .P.2.0.1.
00000a30  35 00 20 00 53 00 65 00 72 00 69 00 65 00 73 00  5. .S.e.r.i.e.s.
00000a31  *  seg000:00000B0E 50          db  50h ; P
00000a32  *  seg000:00000B0F 4F          db  4Fh ; 0
00000a33  *  seg000:00000B10          ; -----
00000a34  *  seg000:00000B10 33 C0      xor  eax, eax
00000a35  *  seg000:00000B12 66 B8 9D 01  mov  ax, 19Dh
00000a36  *  seg000:00000B16 91          xchg eax, ecx
00000a37  *  seg000:00000B17 EB 12      jmp  short loc_B2B
00000a38  *  seg000:00000B17          ; -----
00000a39  *  seg000:00000B19 00          db   0
00000a3a  *  seg000:00000B1A          ; -----
00000b00  *  seg000:00000B1A          loc_B1A:
00000b01  *  seg000:00000B1A 33 C0      xor  eax, eax
00000b02  *  seg000:00000B1C 5A          pop  edx
00000b03  *  seg000:00000B1D B0 FF      mov  al, 0FFh
00000b04  *  seg000:00000B1F          loc_B1F:
00000b05  *  seg000:00000B1F 49          dec  ecx
00000b06  *  seg000:00000B20 30 04 0A   xor  [edx+ecx], al
00000b07  *  seg000:00000B23 FE C8      dec  al
00000b08  *  seg000:00000B25 85 C9      test ecx, ecx
00000b09  *  seg000:00000B27 75 F6      jnz  short loc_B1F
00000b0a  *  seg000:00000B29 EB 06      jmp  short loc_B31
00000c00  *  seg000:00000B2B          ; -----
00000c01  *  seg000:00000B2B          loc_B2B:
00000c02  *  seg000:00000B2B FC          cld
00000c03  *  seg000:00000B2C E8 E9 FF FF  call loc_B1A
00000c04  *  seg000:00000B31          loc_B31:
00000c05  *  seg000:00000B31          ; -----
00000c90  49 C0 01 2E C4 F4 08 42 1F 94 92 90 95 0D D8 D1  I.....B.....
00000ca0  3A D3 D4 D5 D6 8F 5B 19 DF 18 DC BB DE DF F7 6C  :.....[.....1
00000cb0  E3 E3 F3 8F E6 E7 E8 24 EA EB FB DA EF EF F0 A7  .....$.

```

# Fast fingerprinting of OLE2 files

What we found? Troj/DocDrop-S

Bit index	Property name	Spec notes	Probable meaning
3	rupYear	“The value MUST be 0x07cc or 0x07cd” “Excel97 writes 0x07cc for rupYear” (1996)	Maps to Excel spec version.  Is set to 0x0700 in this instance; 1792!
5	fRisc	“MUST be 0” “last edited on RISC platform”	Unknown. This bit is so commonly set that the given meaning in the spec seems unlikely.
7	fWinAny	“SHOULD be 1”	Flag denoting if the file has ever been edited on Windows.
8	fMacAny	“MUST be 0”	Flag denoting if the file has ever been edited on Mac. Again, this violation is very common, and it would make little sense to have this bit, have Mac products, yet never set it!
11	2 <sup>nd</sup> bit of unused1 field	“Undefined and MUST be ignored”	Unknown.
35	reserved2	20 bits, “MUST be zero”	Unknown. The 20 following bits often have a consistent pattern within a group it is likely these bits are in use and have an undocumented meaning.
38-39	Bits 2-3 of the reserved2 bits	As above	As above



# Fast fingerprinting of OLE2 files

What we did next?

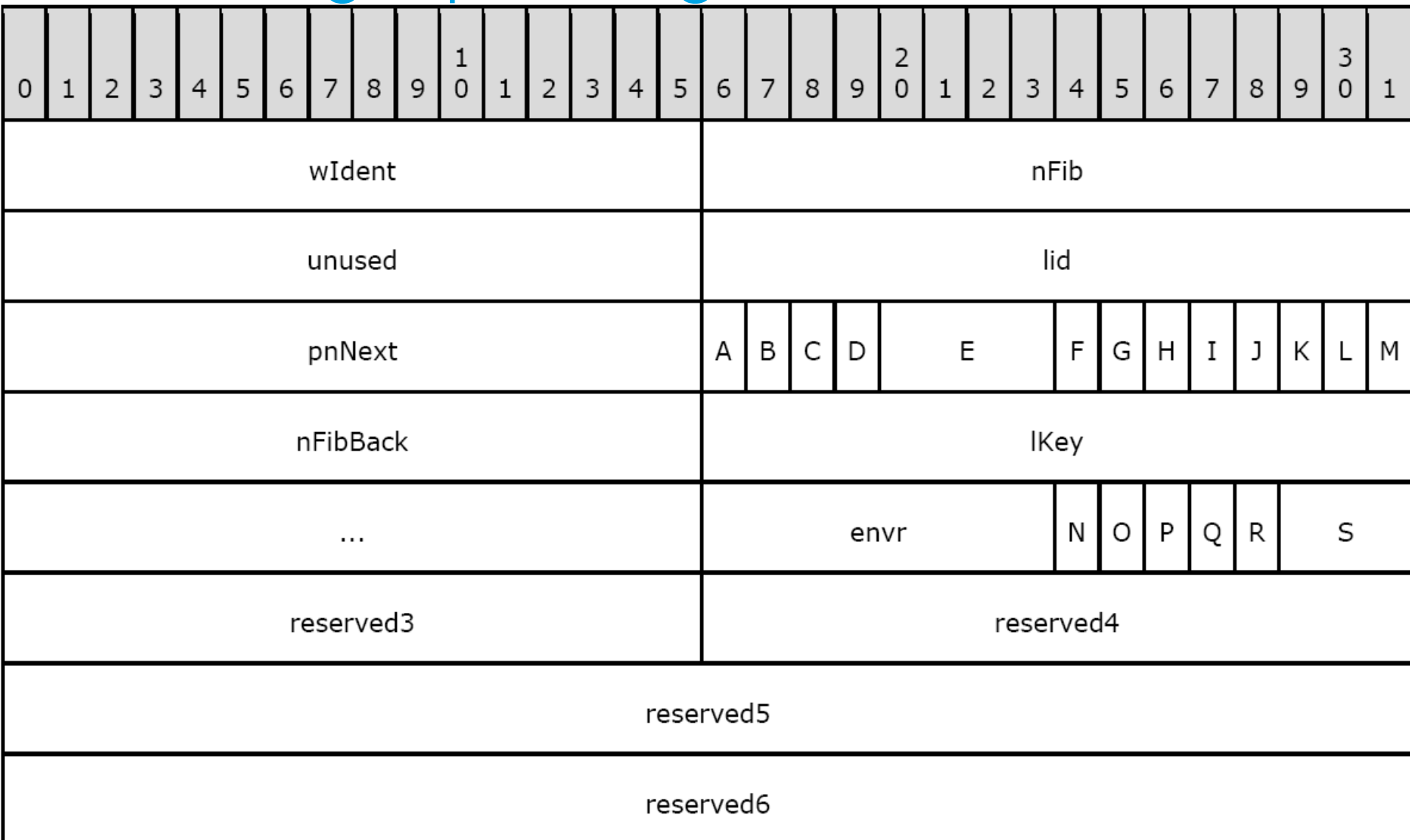
- Looked at a more robust directory chain parser
- Extended to WordDocument
- ...

# Fast fingerprinting of OLE2 files

What we did next? Extended to WordDocument

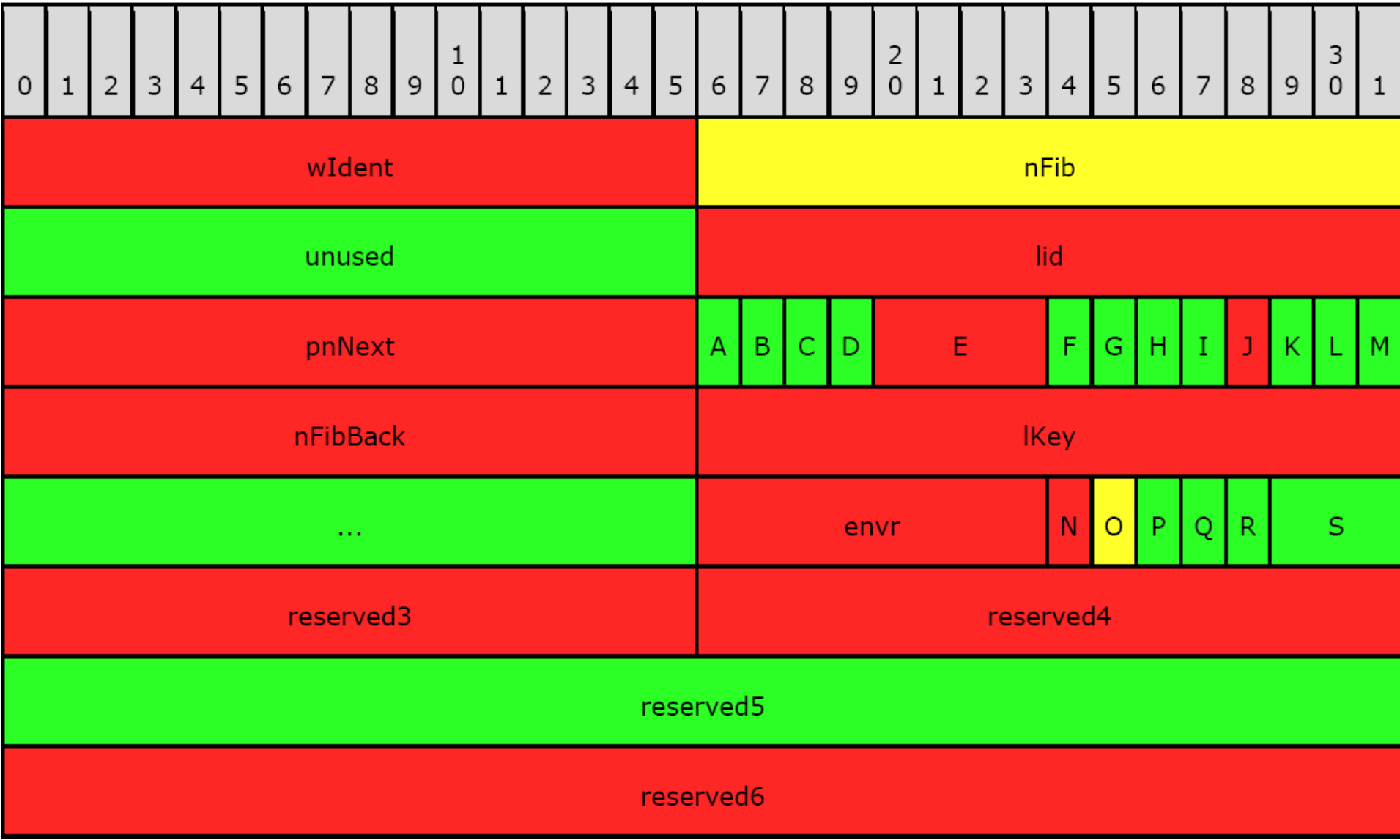
- File Information Block (Fib)
- Starts with FibBase
- 32 bytes 12 possible Must and 2 possible Should
- Going further in the docs looks promising

# Fast fingerprinting of OLE2 files



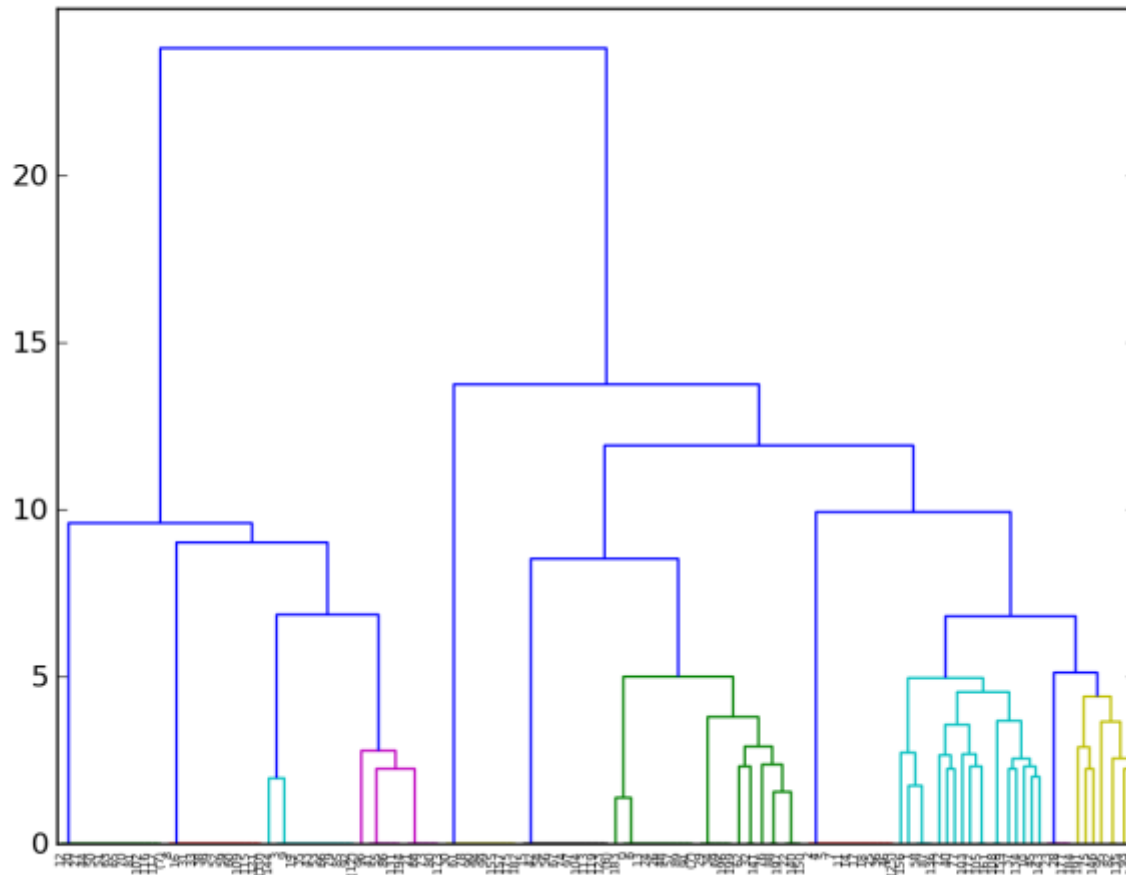
# Fast fingerprinting of OLE2 files

What we did next? Extended to WordDocument



# Fast fingerprinting of OLE2 files

What we did next? WordDocument



# Fast fingerprinting of OLE2 files

What we did next? WordDocument

- Group 7
- 36 files all same violation
- ...

# Fast fingerprinting of OLE2 files

## Conclusion

- Spec violations can provide good grouping.
- Quick/cheap less than 4 loads
- The fingerprints are common over diverse campaigns

# Fast fingerprinting of OLE2 files

Questions?

• ...