

PhoneyPDF

A virtual PDF analysis framework

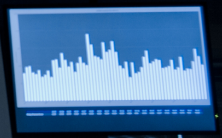
Kiran A. Bandla

Feb 2014

PhoneyPDF



14 272 02 44 00



Lightning version

- Virtual PDF honeyclient framework
- Its `dynamic`
- Python + OS X|Linux
- JavaScript [PyV8]
- Adobe DOM, Adobe XFA [lxml]
- Yara signatures
- ML Ready
- Standalone tool or Library

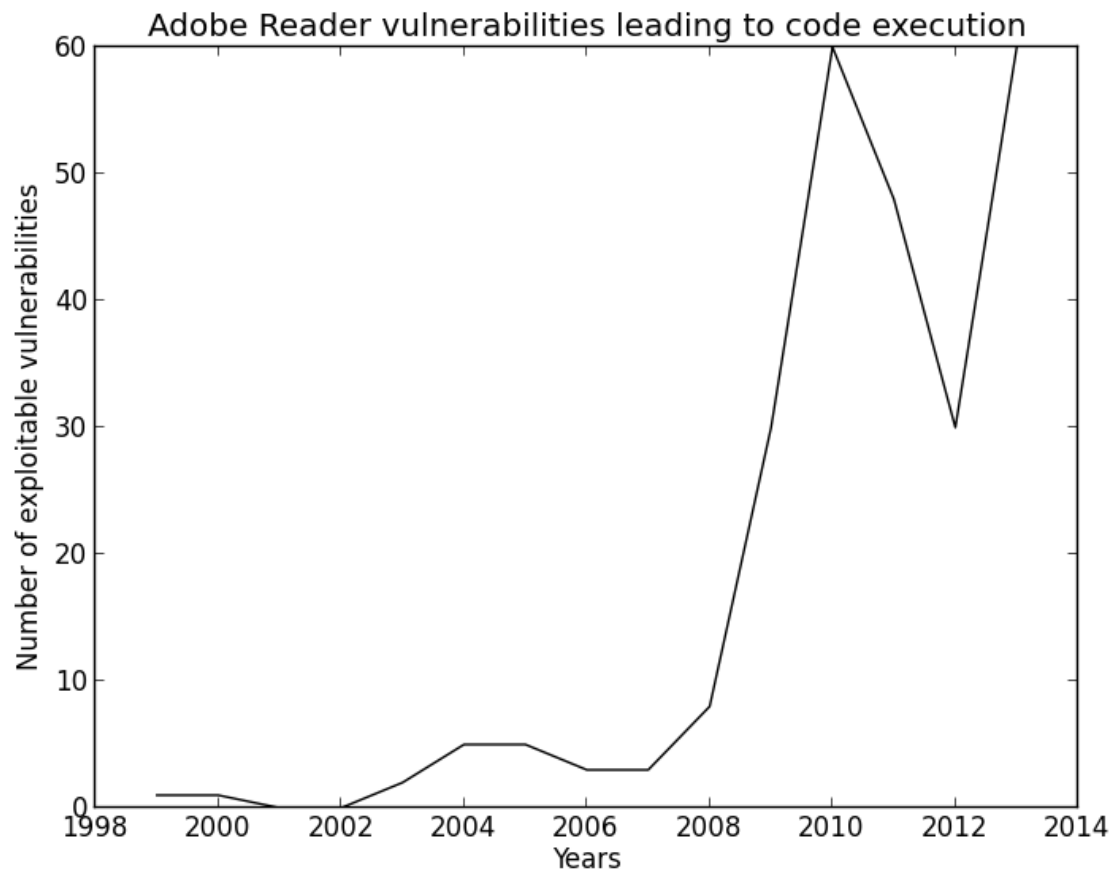
PhoneyPDF

Adobe Reader

- Version 9
- Specification
- Based on SpiderMonkey
 - Used to fingerprint
- Implementation differences across versions

PhoneyPDF

Code Exec Vulns in Reader



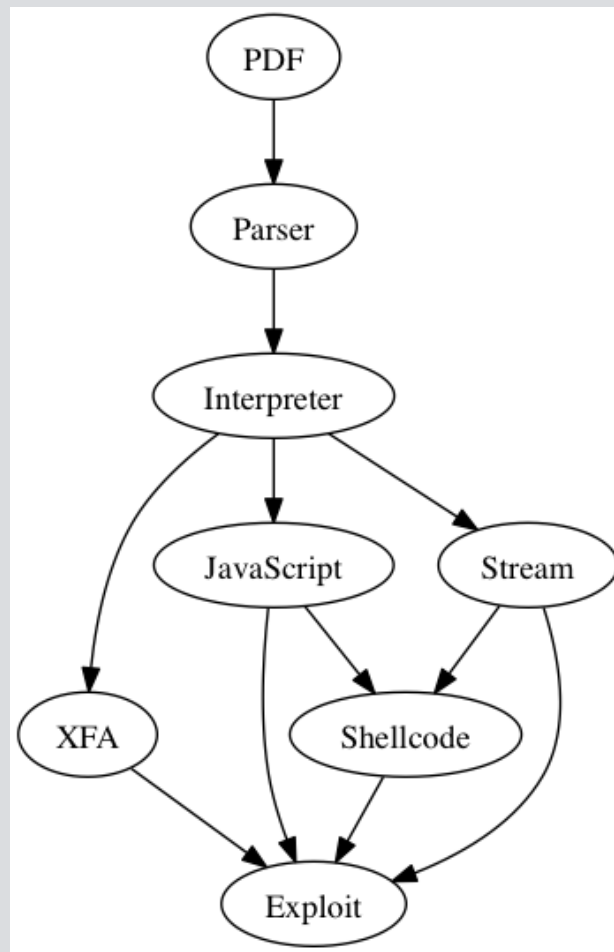
PhoneyPDF

Previous Work

- Mostly static
 - pdf-parser
 - peepdf
 - pdf x-ray, etc
- Lectures
 - omg-wtf-pdf
 - Malicious PDF analysis, etc
- A few published papers
- PhoneyPDF is `dynamic`

PhoneyPDF

Design



PhoneyPDF

The Parser

- Based on Didier's pdf-parser
- Character-based state machine
- Heavily modified, improved
- Pythonic representation
 - id, md5, stream, dict, etc
- De-obfuscation
- Filters
- Features extraction for Machine Learning

Parsing is not fun

```

1 %PDF-1.6^M%aaT0^M
2 125 0 obj^M<</Filter/FlateDecode/First 12/Length 336/N 2/Type/ObjStm>>stream^M
3 hP L P 8 <83>0^P w h ^ A 3 ^ A O ^ _ ^ ] E p s ^ 2 6 D ^ ] ^ [ H ^ _ <9c> F ^ A Q 4 ^ A Q <9b> <8e> o 2 B ^ Y [ e ^ A % ^ 1 U ^ I ^ + Y ^ 1 ; ^ A <88>
> ^ V ^ A ^ T Z D <8d> I <9b> J W ^ ' ; O H ^ A ^ B <83> ^ o ^ A ^ B S ^ x ^ e ^ 3 ^ A W 5 U D ^ e ^ a ^ 2 N ^ e d ^ A U 5 ^ - ^ V ^ I ^ # ^ ^ A ^ O ^ a ^ e ^ ] " ^ 5 <96> ^ A ^ F
4 U ^ B R ^ A ^ a ^ 96> ^ 90> ^ A ^ s ^ Y <8d> ^ A ^ G ^ _ ^ i ^ o ^ y ^ 3 ^ A ^ O ^ I ^ A ^ N ^ A ^ N ^ E ^ Y 2 <9b> ^ A ^ Q ^ h ^ 3 ^ i ^ A ^ E ^ J ^ 8 ^ A ^ ) U ^ A ^ Z ^ S <90> ^ V ^ a ^ 90> ^ 9a>
5 endstream^Mendobj^M126 0 obj^M<</Filter/FlateDecode/First 120/Length 797/N 17/Type/ObjStm>>stream^M
6 h P A T U h U 3 ^ A L Y s p o E c a ] , E . <82> ^ A ^ 1 4 0 ^ 8a> ^ A H <96> ^ A ^ V ^ A D <85> ^ 9b> ^ " <89> Q ^ A N l u [ y ^ m ^ ] n <96> b {
^ <8a> ^ <98> y [ p ^ t ^ ] 0 ^ <95> Y M ^ z U T <90> p s y ^ A ^ ? ^ a ^ e ^ 99> i ^ n ^ C M ^ z <8f> ^ A W ^ e O T o i C A ^ x x q ^ ~ I B c ^ + 7 ^ I ^ - F <9b> ^ s
7 , : ^ a ^ 7 ^ A R B q ^ - ^ a ^ 94> ^ B ^ a ^ 9d> ^ t ^ y ^ L <9f> ^ s ; ^ E <8b> ^ <9f> ^ ] <84> ^ A ^ <95> ^ u ^ t ^ A W ^ i ^ K Z U ^ - ^ A ^ * { ^ i ^ i ^ <8a> ^ U H I ^ l e ^ o ^ # ^ u ^ <86>
99> 5 U ^ A ^ [ <88> ^ a ^ d ^ u ^ e ^ o ^ <9c> ^ + ^ N ^ t ^ o ^ A ^ U e p ^ q U ^ * ^ <9e> , ^ A L ^ O ^ E ^ <90> ^ <8b> ^ I ^ u ^ + [ b ^ a ^ v ^ e ^ a ^ 6 ^ A ^ M ^ v ^ U ^ O ^ <9c> ^ ' Y Y ^ U ^ N ^ - ^ 1 ^ A ^ j
9c> ^ B ^ o ^ d ^ j ^ l ^ O ^ $ ^ H B ^ | ^ A ^ <84> ^ E <97> ^ i <88> ^ b <8e> ^ 6 ^ a ^ f ^ 0 ^ A ^ \ ^ P ^ E ^ n ^ k ^ U ^ <8e> ^ n ^ a ^ p ^ A ^ X ^ { <8d> ^ A ^ z ^ o ^ 1 ^ P H I ^ _ ^ a ^ v ^ A ^ B H u ^ E ^ c
8 ^ e ^ s ^ A ^ P ^ o ^ A ^ R ^ - ^ u ^ 8 I ^ O ^ / ^ A ^ 0 ^ } ^ p ^ n ^ 0 ^ y ^ <81>
9 a << l Y O ^ ? ^ A ^ V ^ + ^ <86> ^ <83> ^ <96> ^ <98> ^ A ^ { ^ A ^ D ^ p ^ e ^ <94> ^ A ^ R ^ o ^ u ^ 3 ^ <85> ^ E ^ k ^ v ^ A ^ A ^ A ^ 0 ^ A ^ E ^ M ^ l ^ ' ^ <91> ^ o ^ A ^ M ^ A ^ L ^ A ^ [ ^ i ^ [ 4 M ^ : ^ A ^ R ^ G ^ o ^ e
10 endstream^Mendobj^M127 0 obj^M<</Filter/FlateDecode/First 67/Length 308/N 10/Type/ObjStm>>stream^M
11 h P <8c> ^ <91> O K ^ A ^ e ^ A ^ P ^ A ^ c ^ e ^ i ^ + ^ e ^ d ^ y ^ C M K ^ E ^ v ^ 5 ^ A ^ W ^ m ^ A ^ M ^ W ^ ^ o ^ A ^ P ^ i ^ z ^ A ^ B ^ 1 ^ A ^ [ ^ <92> ^ T ^ n ^ a ^ G ^ W ^ g ^ <92>
12 A ^ $ ^ A ^ K ^ % ^ y ^ o ^ v ^ A ^ W ^ P ^ q ^ n ^ B ^ 4 ^ A ^ F ^ O ^ <94> ^ N ^ A ^ V ^ u ^ <9c> ^ A ^ P ^ O ^ <88> ^ u ^ A ^ ] ^ N ^ D ^ D ^ R ^ d ^ ! ^ A ^ * ^ 4 ^ T ^ q ^ A ^ Z ^ J ^ F ^ A ^ R ^ c ^ s ^ + ^ d ^ S ^ o ^ C ^ % ^ <8b>
13 . <85> ^ A ^ L ^ j ^ p ^ D ^ <90> ^ R <92> ^ ^ <93> ^ ! ^ ! ^ o ^ d ^ k ^ z ^ <84> ^ + ^ X ^ A ^ ] ^ i ^ r ^ \ ^ <9d> ^ E ^ o ^ ? ^ <9d> ^ B ^ o ^ <94> ^ <8f> ^ p ^ e ^ r ^ e ^ g ^ K ^ <97> ^ E ^ j ^ E ^ /
P ^ A ^ N ^ A ^ S ^ a ^ d ^ / ^ <86> ^ A ^ A ^ W ^ <9d> ^ ^ <8d> ^ ; ^ o ^ y ^ b ^ w ^ v ^ C ^ n ^ <97> ^ A ^ ^ A ^ C ^ A ^ ^ u ^ / ^ i ^ A ^ M
14 endstream^Mendobj^M128 0 obj^M<</Filter/FlateDecode/First 5/Length 135/N 1/Type/ObjStm>>stream^M
15 h P 2 1 T 0 P ^ o ^ t ^ N ^ w ^ . J M , E i i ^ s I , I O p ^ + 2 2 0 4 0 0 2 2 5 ^ o ^ 4 2 6 6 O 5 ^ P 7 0 P x ^ <84> ^ " E ^ / O ^ p ^ L ^ E ^ D ^ J U ^ O ^ E ^ , K u ^ @ ^ I ^ i ^ U ^ p ^ I ^ - ^ I ^ l ^ I ^ k ^ -
16 4 l ^ A ^ M
17 endstream^Mendobj^M129 0 obj^M<</Filter/FlateDecode/First 41/Length 873/N 6/Type/ObjStm>>stream^M
18 h P i V I r U 3 ^ A L ^ ~ ^ A ^ U ^ A ^ A ^ <93> ^ C ^ A ^ W ^ <92> ^ l ^ U ^ l ^ ' 3 J ^ q ^ L ^ = ^ U ^ 0 ; ^ u ^ o ^ n ^ 7 ^ a ^ C ^ - ^ A ^ A ^ R ^ ' ^ <94> ^ e ^ o ^ ' ^ <8d> ^ + ^ e ^ A ^ W ^ $ ^ m ^ E ^ { ^ e ^ A ^ S ^ I ^ A ^ N ^ <94>
> ^ j ^ <85> ^ i ^ b ^ ( ^ A ^ F ^ # ^ F ^ U ^ n ^ h ^ c ^ o ^ ; ^ o ^ / ^ A ^ B ^ <9e> ^ o ^ A ^ M ^ V ^ A ^ A ^ * ^ 1 6 5 i ^ ? ^ A ^ S ^ E ^ s ^ O ^ i ^ n ^ <80> ^ p ^ 0 ^ <97> ^ 4 ^ \ ^ A ^ C ^ <8b> ^ U ^ O ^ j ^ e ^ A ^ D ^ <93> ^

```

Example #1: Raw to Python

```
trailer
<</Size 22 /Root 23 0 R>>
```

```
>> dir(pdfObj)
['content', 'dict', 'id', 'type']
```

```
>> pdfObj
<PDFElement id:None type:PDF_ELEMENT_TRAILER>
```

```
>> pdfObj.dict
{'/Size': 22.0, '/Root': <PDFIndObjRef id=23 gen=0>}
```

Example #2

RAW dict Object

```
<<
  /DA (/CourierStd 10 Tf 0 g)
  /F 4
  /FT/Btn
  /Ff 65536
  /MK <</TP 1>>
  /P 1 0 R
  /Parent 19 0 R
  /Rect [12.5 5.1 18.3 7.7]
  /Subtype /Widget
  /T (favwwbw[0])
  /TU (favwwbw)
  /Type/Annot
>>
```

Pythonic equivalent

```
{
  '/DA' : '/CourierStd 10 Tf 0 g',
  '/F' : 4.0,
  '/FT' : '/Btn',
  '/Ff' : 65536.0,
  '/MK' : {'/TP': 1.0},
  '/P' : <PDFIndObjRef id=1 gen=0>,
  '/Parent': <PDFIndObjRef id=19 gen=0>,
  '/Rect': [12.5, 5.1, 18.3, 7.7],
  '/Subtype': '/Widget',
  '/T' : 'favwwbw[0]',
  '/TU' : 'favwwbw',
  '/Type': '/Annot'
}
```

PhoneyPDF

The Analysis Engine

- Consumes parsed objects
- Analysis logic
- ‘Renders’ the PDF
 - Walk objects
 - JavaScript, Adobe DOM Emulation
 - Adobe XFA
- Yara Signatures
- Revisit this in a few slides

PhoneyPDF

JavaScript

- PyV8
- One context
- Debug mode
- Adobe DOM Emulation!
- Heapspray detection
- Yara

JavaScript #2

- Bootstrap
 - SpiderMonkey hacks
 - `this` object
 - Ex: `Object.prototype.eval = this.eval;`
 - return -1 for some functions on undefined objects
- Hooks
 - eval, timeout functions, etc
- Emulate Adobe DOM

Adobe DOM Emulation

- Console object for logging
- Collab,FullScreen,Thermometer,util,app,doc..
- Object hierarchy; initialization order
 - app object needs FullScreen, Thermometer, etc
- Generic handlers – log
- Custom handlers – detection
 - spell.customDictionaryOpen : CVE-2009-1493
- Argument introspection
- PDF to JS bridge

Adobe XML Forms Architecture / XFA

- 2 method for forms in PDF
 - AcroForms
 - Adobe XML Forms Architecture (XFA) forms
- lxml
- JavaScript in XFA objects
- XML to XFA DOM
- JavaScript bridge

‘Render’ the PDF

- Based on the PDF Specification document
- Stay close to Adobe’s implementation
- Start emulation
 - Create JavaScript context
 - Walk PDF objects, avoiding cycles
 - Dynamic handlers based on object type
 - Monitored fields from XFA and JavaScript

PhoneyPDF

Handlers

- 6 types; based on object type
 - Comment, XREF, Trailer, StartXREF, Indirect, Malformed
- Walk object list based on object's `/Type`
- About 35 types; extensible
- Object relations

/Action type handler

```
# Specific handlers for various types
if pdfElement.dict[KEY_TYPE] == KEY_ACTION:
    #/Action object
    keys = pdfElement.dict.keys()
    if KEY_S in keys:
        if pdfElement.dict[KEY_S] == KEY_JAVASCRIPT:
            js = None
            # Handle JS here
            # This has to be either 1) text string or 2) Stream
            if hasattr(pdfElement, 'stream'):
                # Stream!
                #TODO: JS is served. Pick it up
                js = pdfElement.stream
            elif isinstance(pdfElement.dict[KEY_JS], str):
                js = pdfElement.dict[KEY_JS]
                #TODO: JS is served. Pick it up
            elif isinstance(pdfElement.dict[KEY_JS], PDFIndObjRef):
                js = self.__getIndirectObjectJavaScript(pdfElement.dict[KEY_JS])
                #TODO: JS is served. Pick it up
            self.logger.debug("[%s ID:%s] Executing Javascript" % (pdfElement.dict[KEY_TYPE], pdfElement.dict[KEY_S]))
            if js:
                self.featureCollection.incFeatureValue("F_C_JS_SCRIPTS_FOUND")
                self.__executeJavaScript(js)
        elif pdfElement.dict[KEY_S] == KEY_LAUNCH:
```

PhoneyPDF

A few /Types

```
KEY_ROOT      = '/Root'  
KEY_CATALOG   = '/Catalog'  
KEY_ACTION    = '/Action'  
KEY_OPENACTION = '/OpenAction'  
KEY_AA        = '/AA'  
KEY_JAVASCRIPT = '/JavaScript'  
KEY_RENDITION = '/Rendition'  
KEY_OUTLINES  = '/Outlines'  
KEY_JS        = '/JS'  
KEY_PAGES     = '/Pages'  
KEY_PAGE      = '/Page'  
KEY_METADATA  = '/Metadata'  
KEY_SUBTYPE   = '/Subtype'
```

CVE-2010-0188

Adobe Reader TIFF vulnerability

- Most popular in Exploit Kits, still
- Complex flow to trigger the vulnerability
- PhoneyPDF detects it; extracts payload
- Exploit Kits using this vulnerability
 - Blackhole
 - Whitehole
 - Fiesta/Neisexploit
 - Cool Styxy
 - Many more!

PhoneyPDF

Open Source

- github.com/verisign/phoneypdf
- 3-BSD
- Dependencies
 - PyV8
 - V8
 - lxml
 - Yara
- Selective loading

PhoneyPDF

Questions

PhoneyPDF