Osservatorio Nazionale di Informatica Forense
Seminario formativo ONIF 2024
"Il senso di Amelia per la Digital Forensics"
18 ottobre 2024

Advanced PDF forensics:

Lo strano caso dello scanner intelligente

Marco "*Darth Adobe*" Calamari
marco.calamari@ordineingegneripisa.it
Osservatorio Nazionale Informatica Forense - Ordine degli Ingegneri della provincia di Pisa

Il vostro anfitrione

https://www.linkedin.com/in/marcocalamari/

- **Marco Calamari, classe 1955, ingegnere nucleare, nell'ICT ha seguito un lungo cammino da umile sviluppatore ad architetto di applicazioni, ed è specializzato in gestione di software legacy. Opera come consulente in ambito informatico e di Computer Forensics dal 1990, ed ha maturato 15 anni di esperienza nella formazione in Olivetti ed Elea.**
- **Affiliazioni: ONIF, AIP, Opsi, PWS**
- **Appassionato di privacy e crittografia, ha contribuito ai progetti FOSS Freenet, Mixmaster, Mixminion, Tor e Globaleaks.**
- **Fondatore del *Progetto Winston Smith* e del convegno e-privacy, che quest'anno è giunto alla trentacinquesima edizione.**
- **Dal 2003 scrive su Punto Informatico, ZeusNews.it, Medium, Giano.news, Galileo ed altre testate la rubrica "Cassandra Crossing", che è arrivata a circa 600 puntate (www.cassandracrossing.org).**
- **Membro della Commissione Informazione dell'Ordine di Pisa, ha tenuto numerosi corsi per il CNI e gli ordini provinciali della Toscana.**

Il racconto di oggi

Questa presentazione è il seguito ideale di quella presentata ad Amelia nel 2022, che potrebbe essere utile consultare, ma che comunque riassumeremo.

Il racconto di oggi

- **Riassunto delle puntate precedenti: PDF e PostScript**
- **Casi “classici” di analisi forense di PDF**
- **Lo strano caso**
- **Il dubbio ed il momento** *mmmhhh...*
- **Di nuovo in caccia**
- **La confessione dello scanner intelligente**
- **Una morale della storia**

Riassunto delle puntate precedenti

PDF ed analisi forense

Parlare di "analisi forense" per un determinato formato di file può sembrare a prima vista qualcosa di eccessivo. Le cose non stanno così per quattro motivi.

Il primo motivo è che il formato pdf è ormai da anni uno standard de facto per lo scambio di documenti testuali od illustrati. Oltre il 90% dei documenti testuali trasmessi in formato elettronico sono in formato PDF. E questo ha fatto si che il formato pdf divenisse anche uno standard internazionale quasi "de iure", secondo ISO 32000-1:2008 e ISO 32000-2:2020.

Il secondo è che i contenuti di un file pdf sono scritti in un vero linguaggio di programmazione, il PostScript, che pur essendo orientato alla grafica è un linguaggio ricco e complesso. Utilizza blandamente il concetto di oggetti, la RPN (notazione polacca inversa) ed eredita i concetti del Forth, facendo un esteso uso degli stack.

Per fortuna l'analista forense non deve impararlo. [7] [8] [9]

PDF ed analisi forense

Il terzo è che i file PDF sono, nella quasi totalità dei casi, scritti automaticamente da dei driver di stampa, sotto il controllo di applicativi delle tipologie più svariate.

Il particolare modo, tra i tanti possibili, con cui un file pdf è scritto, consente di attribuirne la creazione in maniera spesso molto precisa a particolari driver, sistemi operativi ed applicazioni.

Il quarto motivo è che ormai da diverso tempo accade che i contenuti di file PDF vengano modificati con vari programmi, oppure direttamente assemblati, in modo da costituire un documento oggettivamente alterato.

Infine, avendo ancora i file PDF una ingiustificata ed immeritata reputazione di inalterabilità, si tratta senz'altro di un terreno di lavoro adatto per l'informatico forense.

PDF e le stampanti

Le moderne stampanti non funzionano più con gli aghi, i martelletti o le palline ad impatto delle macchine da scrivere elettromeccaniche. Questi tipi potevano pprodurre solo documenti testuali o poco più.

Le stampanti moderne (o meglio, il loro driver di stampa) colloquiano col computer (o meglio con l'applicazione che sta stampando) con un vero e proprio linguaggio di programmazione, linguaggio che permette di descrivere e stampare testo e grafica in maniera più o meno sofisticata.

Dopo diversi linguaggi di stampa primitivi, nel 1982 Adobe System realizza il linguaggio PostScript e le prime famiglie di font vettoriali. Un successo travolgente. Apple inserisce il PostScript nella sua innovativa stampante laser Apple LaserWriter; il linguaggio viene licenziato sulle stampanti di fascia alta di tutti I produttori.

Jobs ne fa un fulcro della potenza dei Macintosh, e poi dei NeXT.

La rivoluzione del publishing che ne seguirà permetterà a chiunque di realizzare una pubblicazione a casa, ma manderà ugualmente a casa legioni di linotipisti.

PDF e PostScript

Oggi il PostScript è dappertutto, perché ha figliato due parenti strettissimi, EPS (Encapsulated PostScript) e l'arcinoto PDF.

Il PDF (Portable Document Format) è un formato aperto, standard ISO 32000-2. E' al 99% PostScript, perché ne eredita tutti i comandi di stampa, la struttura del linguaggio e la gestione dei font.

Un file PDF contiene dati e metadati, ed ha un meccanismo per includere file binari come ad esempio le immagini TIFF o JPG, codificandole ROT64, e comprimendoli come ZIP od altri formati. [3]

Un file PDF è costituito principalmente da oggetti entrocontenuti, che vengono "creati" dalla stampante "interpretando" il file PDF, poi posti su uno stack, ed infine "renderizzati" singolarmente sulla pagina, che alla fine del processo viene fisicamente stampata. Gli oggetti sono testuali, vettoriali (font, grafica) e bitmap (foto, immagini)

Un file PDF può consistere anche interamente di caratteri stampabili, ma normalmente non è così per occupare meno spazio. L'inizio di ogni file è tuttavia sempre in ASCII, come nella slide seguente.

PDF e PostScript

Adobe® PostScript® 3™

```
%PDF-1.4
%\E2\E9\CD\D3
333 0 obj
<</Linearized 1/L 47721/O 783/E 5747/N 17/T 37053/H [ 576 728]>>
endobj
xref
333 15
0000000033 00000 n
0000001133 00000 n
0000000533 00000 n
trailer
<</Size 396/Prev 37041/XRefStm 924/Root 382 0 R/Info 43 0
R/ID[<0C215302E743484600E00D365D><B7881071E4687F2A90A2DF56D26>]>>
startxref
0
%%EOF
355 0 obj
<</Length 237/C 311/Filter/FlateDecode/I 373/L 275/S 752 >> stream
```

Casi "classici" di analisi forense

Casi "Classici"

Un file PDF può essere costituito da un'unica bitmap, come ad esempio i file PDF generati da uno scanner.

Se generati da una normale applicazione, invece, i file PDF sono costituiti da molte entità grafiche, sovrapposte e posizionate opportunamente. Ogni singola stringa, talvolta ogni singolo carattere, è un'entità perfettamente riconoscibile anche nel file di stampa PDF.

In generale, ogni programma produce i file PDF in maniera tipica, con una struttura interna caratteristica come un'impronta digitale. Ad esempio, tutti i programmi che manipolano e sovrappongono bitmap, rettangoli od entità grafiche, le inseriscono singolarmente, come "oggetti" separati, nel file PDF.

Quindi, ad esempio, se una pagina di un documento Word viene parzialmente "censurata" sovrapponendovi un rettangolo bianco, e poi stampata in formato PDF, l'analisi del file PDF risultante permette di ricostruirla, ad esempio eliminando o rendendo trasparente il rettangolo stesso.

Casi "Classici"

Il metodo più elementare per eseguire una veloce preanalisi della struttura ad oggetti di un file PDF è quella di convertirlo in un formato editabile, e poi esaminarlo con un programma in grado di aprire quel tipo di file.

L'operazione si può fare in maniera semplice, utilizzando tool liberi a linea comando reperibili su Github. [6]

Tuttavia la cosa più veloce è convertire il PDF in formato Word utilizzando uno dei siti specializzati, come ILovePDF, in grado di compiere l'operazione in pochi secondi.

Aprendo poi il file "censurato" così convertito, utilizzando un elaboratore di testi, è immediato selezionare e cancellare il rettangolo bianco, rivelando il testo sottostante in pochi secondi.

E' anche possibile "spostare" con il mouse e selezionare le varie bitmap e parti di testo per rendersi in prima approssimazione conto della struttura del file.

Casi "Classici"

Dal punto di vista forense è interessante da una parte recuperare informazioni non visibili dall'esterno, quali commenti interni, metadati interni, metadati contenuti negli oggetti bitmap (TIFF, JPG, etc.

E' sempre importante recuperare struttura interna e suddivisione in oggetti di un file PDF.

Con opportune librerie di analisi, disponibili spesso come software libero, è possibile ad esempio produrre rapidamente un elenco delle bitmap presenti in un file PDF, ed estrarle come file separati. I normali programmi di grafica possono poi essere impiegati per esaminare le singole immagini, estrarne i metadati, verificarne la genuinità, etc.

```
page num  type   width height color comp bpc  enc interp  object ID x-ppi y-ppi size ratio
--------------------------------------------------------------------------------
   1   0 image    1240  1753  rgb     3   8  jpeg   no         18  0   150   150 41.5K 0.7%
   1   1 stencil  1680  3180  -       1   1  ccitt  no         19  0   300   300 9192B 1.4%
   2   2 image    1240  1753  rgb     3   8  jpeg   no          4  0   150   150 55.3K 0.9%
   2   3 stencil  2136  3188  -       1   1  ccitt  no          5  0   300   300 6068B 0.7%
```

Lo strano caso

Lo strano caso

Viene richiesta l'analisi di un file PDF, che appare essere la scansione di un documento multipagina con pagine eterogenee; si ipotizza infatti che la firma autografa presente in una pagina sia stata inserita "copiandola" da un altro documento.

Si eseguono i classici comandi di analisi, e la firma appare essere un oggetto separato dalla bitmap che contiene la scansione di base del documento. La firma è anche ulteriormente divisa in due bitmap separate, una contenente il nome, l'altra contenente il cognome

```
$ pdfimages -list documentofirmato.pdf
page   num  type   width height color comp bpc  enc interp  object ID x-ppi y-ppi size ratio
--------------------------------------------------------------------------------------------
 ...
   2     2 image     240  753  rgb     3   8  jpeg   no         4  0   150   150  5.3K 0.9%
   2     3 stencil   236  188  -       1   1  ccitt  no         5  0   300   300 6068B 0.7%
 ...
```

"Documento falsificato in maniera elementare, Watson, e caso risolto?"

Il dubbio ed il momento mmmhhh

La suddivisione in due o più parti di una firma digitalmente artefatta non è cosa inutile.

Un abile manipolatore digitale la può dividere per renderla diversa rispetto all'originale, utilizzando parti di due firme diverse della stessa persona, posizionando nome e cognome in maniera differente, ed eseguendo anche piccole distorsioni.

Ma ... perché le due immagini del nostro caso sono di tipo diverso; una è un jpeg a colori con 24 bit di profondità, mentre l'altra è addirittura in bianco/nero, e compressa con formato CCITT, quello dei telefax?

Per quale motivo il presunto manipolatore avrebbe dovuto farlo?

E perché ci sono diverse altre bitmap contenenti altre parti del documento, senza che queste sembrino necessarie?

Qualcosa non torna.

E se si trattasse di tutt'altro ... mmmmhhhh ...

Di nuovo in caccia

La nuova indagine

Una rilettura della struttura del file sotto esame rivela una cospicua e apparentemente inutile "frammentazione" dell'immagine della pagina in piccole bitmap, la maggior parte non a colori o scala di grigio ma addirittura monocromatiche in formato CCITT.

Non si tratta di oggetti generati "normalmente" durante la manipolazione di un file grafico.

Ripartiamo quindi dall'inizio, cioè dall'apparecchiatura che risulta aver compiuto la scansione.

Dai metadati interni del file PDF, la scansione risulta effettuata:

```
$ pdfinfo documento.pdf
Creator:         Canon iR-ADV C5235  PDF
Producer:        Adobe PSL 1.2e for Canon
```

Cioè su una fotocopiatrice Canon imageRUNNER Advance C5235

Il sospettato

Si tratta di una fotocopiatrice/stampante/scanner multifunzione di concezione abbastanza recente.

Per prima cosa controlliamo, sul manuale utente, il capitolo sulla generazione dei file PDF.

La confessione dello scanner intelligente

L'indizio

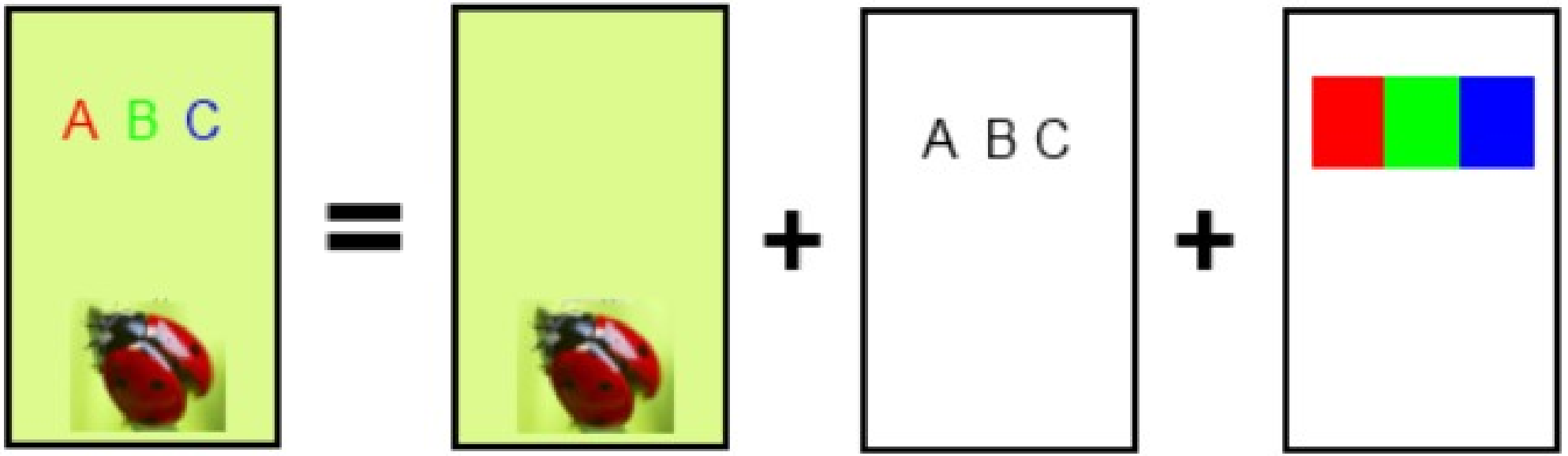
La documentazione dello scanner rivela che oltre alle ordinarie modalità di salvataggio “compresso” [1], basato sulla diminuzione della risoluzione delle bitmap, e la loro memorizzazione in formato binario, è dotato di una modalità di compressione estrema, basata sulla analisi e decomposizione della scansione in più bitmap.

Questo algoritmo MRC - Mixed raster content [4] è un metodo per comprimere le immagini che contengono sia testo comprimibile in formato binario sia componenti a tono continuo, utilizzando metodi di segmentazione delle immagini per migliorare il livello di compressione e la qualità dell'immagine renderizzata. [5].

Separando l'immagine in componenti con diverse caratteristiche di comprimibilità, è possibile applicare l'algoritmo di compressione più efficiente e accurato per ciascun componente.

La spiegazione

Nel caso dell'implementazione effettuata dal software dello scanner, dopo la scansione e prima della scrittura del pdf, analizza l'immagine alla ricerca di zone dell'immagine che possano essere descritte con una profondità di colore minore, o comunque per caratteristiche omogenee.



Il diabolico piano … che non c'era

Identificate queste parti, le "taglia" dall'immagine originale, sostituendole con rettangoli perfettamente uniformi. L'immagine di base residua si comprime molto di più, avendo zone vuote perfettamente omogenee.

Le bitmap "ritagliate" vengono invece ridotte ad immagini con piccola profondità di colore, possibilmente solo bianco e nero, e memorizzate con un algoritmo di compressione specializzato per questo tipo di immagini, comprimendo moltissimo anche le bitmap ritagliate.

Il risultato è un file PDF di dimensioni minori, che mantiene la risoluzione di quello originale, a prezzo di una piccola perdita di informazione nelle zone a bassa profondità di colore, e della possibile creazione di artefatti di compressione; su questi ultimi non possiamo addentrarci in questa sede.

Il file PDF esaminato non risulta quindi manipolato.

La morale della storia, anzi tre

La morale

Tutti i problemi complessi possiedono una soluzione semplice, che spesso però è sbagliata.

Siamo solo piccoli Watson, ma il nostro lavoro può fare la differenza; non possiamo permetterci errori.

Non c’è problema forense che non meriti, anzi non richieda, una seconda occhiata.

Per approfondire

[1] Adobe reference - Optimizing PDF

https://helpx.adobe.com/acrobat/using/optimizing-pdfs-acrobat-pro.html

[2] Entropy Based Estimation Algorithm Using Split Images

https://www.researchgate.net/profile/Altan-Mesut/publication/320009161_Entropy_Based_Estimation_Algorithm_Using_Split_Images_to_Increase_Compression_Ratio

[3] Image Compression Techniques -An Overview

https://www.researchgate.net/publication/339551866_Image_Compression_Techniques_-An_Overview

[4] MRC (Mixed Raster Content) compression Algorithm

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

[5] MRC to encode document images to PDF format

https://www.vintasoft.com/docs/vsimaging-dotnet/Programming-Pdf-Optimize_And_Compress_Pdf_Document-Mrc.html

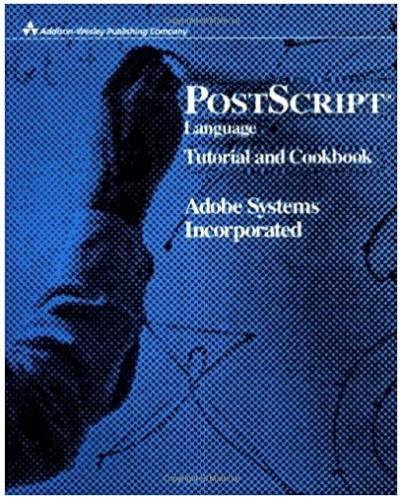
[6] Elenco ragionato di strumenti software per l'analisi di file PDF

https://github.com/zbetcheckin/PDF_analysis

[7] Corso pratico di Postscript

(1985, opera dell'autore, vero esempio di archeologia informatica!)

https://www.marcoc.it/corsops/indice.htm



[8] PostScript ® LANGUAGE REFERENCE - third edition

https://www.adobe.com/jp/print/postscript/pdfs/PLRM.pdf

[9] PDF Association resources index

https://www.pdfa.org/resource/pdf-specification-index/

Grazie per l'attenzione.

Domande?

+ Marco A. Calamari marco.calamari@ordineingegneripisa.it --+
| |
| DSS/DH: 8F3E 5BAE 906F B416 9242 1C10 8661 24A9 BFCE 822B |
| Cell: (+39) 347 8530279   Tel: (+39) 050 576031 |
| Skype-Twitter: calamarim |
| |
+ P.E.C.: marcoanselmoluca.calamari@ingpec.eu -------------+