Osservatorio Nazionale di Informatica Forense
Seminario formativo ONIF 2025
"La Scienza del Digitale: evoluzione e riconoscimento della figura dell'Informatico Forense"
Brescia - 10 giugno 2025

Advanced PDF forensics:

Coccardine posticce ed altre storie

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, giunto quest'anno alla trentaseisima edizione.**
- **Dal 2003 scrive su Agenda Digitale, ZeusNews.it, Medium, Giano.news, Galileo ed altre testate la rubrica "Cassandra Crossing", giunta ad oltre 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.**

La *puntata* di oggi

Questa presentazione è il seguito di quelle che ho avuto l'onore di presentare ad Amelia nel 2022 e 2024, che potrebbe essere interessante consultare, ma che comunque riassumeremo. La bibliografia al termine copre comunque interamente questa piccola "*saga*".

Di cosa parleremo

- **Riassunto delle puntate precedenti: PDF e PostScript**
- **Struttura di un file PDF**
- **Modalità di firma digitale di un file PDF**
- **Malware PDF e file polimorfi**
- **Un esempio semiserio di PDF eseguibile**

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 impare il Forth. [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 stato 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.

Il 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 John Warnock (Adobe System) realizza il linguaggio PostScript e le 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. E senza bisogno dell'IA!

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 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 , commenti e metadati, e può includere file binari, come ad esempio le immagini TIFF o JPG, nativamente, codificandole ROT64, o comprimendole come ZIP od altri formati. [3]

Un file PDF è costituito da oggetti entrocontenuti, che vengono "creati" dall'interprete PostScript della 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 possono essere testuali, vettoriali (font, grafica) e bitmap (foto, immagini)

Un file PDF può consistere 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.

L’inizio di un file PDF


```
%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
```

Cenni di struttura di file PostScript

Cenni della struttura di un PDF

Il testo o le immagini che un PDF permette di visualizzare e stampare non sono solo bitmap, ma anche vettoriali.

Un file pdf può essere costituito anche solo da un'unica bitmap, come i file PDF generati da uno scanner.

Se generati da una normale applicazione, invece, i file PDF sono costituiti da singole 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.

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 viene "censurata" sovrapponendovi un rettangolo bianco, l'analisi del file PDF permette di ritrovarla.

Compressione e scansione

Un file PDF è internamente composto da “oggetti”, che vengono creati e posti sugli stack durante l’interpretazione delle varie sezioni dei file PDF.

La sezione del file PDF contenente i dati dell’oggetto, detta “stream”, è racchiusa tra le parole chiave stream ed endstream

La creazione dell’oggetto avviene ponendo il numero dell’oggetto da creare ed il comando “obj” sullo stack, aggiungendo i parametri necessari per il tipo di oggetto, il nome del filtro da usare per interpretare lo stream, ed infine lo stream stesso.

Si termina la creazione con il comando “endobj”, che finalmente genera l’oggetto. Esso, appena creato, viene a sua volta posto sullo stack, per essere poi “consumato” da altri comandi (ad esempio un oggetto bitmap viene consumato dal comando che lo imprime sulla pagina).

Un oggetto obbligatorio in ogni file è il dizionario XREF; è un indice che fornisce posizione e lunghezza di ogni oggetto contenuto nel PDF

Modalità di firma di file PDF

Firma CAdES e PAdES

Un file PDF può essere firmato digitalmente in diversi modi.

Trattandosi di un file, come qualunque altro file può essere firmato in formato CAdES .P7M, che incapsula il file di origine e richiede di estrarlo per poterlo utilizzare; nel caso di un PDF, per poterlo visualizzare.

Lo standard di firma PAdES non ha invece questo problema.

La struttura standard di un file PAdES (PDF Advanced Electronic Signature) è quella di un documento PDF con una firma digitale apposta all'interno, come nuovo "oggetto" PostScript.

Il file firmato mantiene l'estensione .pdf e può essere aperto con qualsiasi lettore PDF, anche se non in grado di interpretare la firma.

Al suo interno, oltre al contenuto del documento, sono presenti informazioni sulla firma, il certificato del firmatario, la data della firma e altri dati crittografici che garantiscono l'integrità e la validità del documento.

E' di solito possibile "visualizzare" la firma come elemento grafico.

Firma PAdES

```
52 0 obj
<</Type/XObject/Resources<</XObject<</FRM 50 0 R>>/ProcSet[/PDF/Text/ImageB/ImageC/Ir
>>/Subtype/Form/BBox[0 0 197 60]/Filter/FlateDecode/Length 29>>
stream
x•+T0T0•B••••••••••••••••••••M•••
endstream
endobj

51 0 obj
<</Contents<3082090B06092A864886F70D010702A08208FC308208F8020101310D300B0609608648016
0201300B06092A864886F70D010701A08206083082060430820 4ECA003020102102AE2145471B7FF68F
.......
0000000000000000000000000000000000000000000000000000>
/SubFilter/ETSI.CAdES.detached/Name<FEFF00430061006C0061006D00610072006900200 4D0061C
20004C007500630061>
/Filter/Adobe.PPKLite/M(D:20250530070959+00'00')/ByteRange [0 2510180 2551142 842 ]
•
/Type/Sig>>
endobj

xref
0 1
0000000000 65535 f
```

Firma XAdES

Un file PDF potrebbe essere firmato digitalmente anche in formato XAdES; questo formato usa l'xml per creare un "record di firma".

Il formato XAdES viene normalmente usato per firmare file in formato XML.

Potrebbe tuttavia essere utilizzato anche per file di altro tipo, incluso i PDF, ma i programmi commerciali esistenti per l'apposizione di firme digitali non lo supportano.

Il formato XAdES viene invece utilizzato nel PCT e nella fatturazione elettronica per firmare i file XML necessari per il funzionamento del sistema.

Firme multiple

Sia il formato PAdES che XAdES permettono l'apposizione di firme multiple.

Anche il formato CAdES permette l'apposizione di firme multiple, ma produce il fastidiosissimo "effetto matrioska".

Infatti per verificare e visualizzare un file in formato CAdES con firme multiple è necessario verificare singolarmente la firma più "esterna", estrarre il file contenuto, e ripetere le due operazioni per ogni ulteriore firma, fino ad arrivare a quella più interna.

La cosa è particolarmente deleteria quando in realtà non interessa verificare le firme, ma solo visualizzare o stampare il file originale!

E magari, come nel PCT, questo va ripetuto per una dozzina di allegati di un atto.

Firma PAdES invalida

Può accadere che all'apertura di un file pdf firmato, ad esempio utilizzando Acrobat Reader, il programma segnali che almeno una delle firme non è valida.

A parte gli ovvi casi di corruzione o manipolazione del documento, questo può accadere, anche se con frequenza decrescente, nel PCT per alcuni provvedimenti del Giudice.

Il fenomeno è dovuto alla "interferenza" tra la firma PAdES del Giudice (coccardina e linea di caratteri verticale sul margine destro) e quella PAdES del provvedimento (messaggio blu sul margine superiore), che nelle prime versioni del PCT non erano correttamente gestite.

Man mano che i vecchi procedimenti terminano, il fenomeno è naturalmente destinato ad esaurirsi.

E' comunque possibile, anzi doveroso, verificare, utilizzando il pannello di firma di Acrobat Reader, i motivi della segnalazione e determinarne la causa precisa (qui le istruzioni).

“Falsificazioni” della firma di file PDF

Falsificazioni di firme PDF

La “falsificazione” di un file PDF correttamente firmato e correttamente verificato, utilizzando software privi di falle, è matematicamente impossibile; un controllo con un apposito software la rileva facilmente..

Solo in ambito malware, sono possibili falsificazioni (più esattamente “alterazioni”) perfette, inserendosi in modalità MITM nel processo di firma tramite malware. Ma in questo caso si appone la firma di un altro.

E’ stato però riscontrato anche l’utilizzo di alterazioni di un pdf firmato in formato PAdES, che riguardano solo l’aspetto grafico della firma, oppure il nome di chi l’ha apposta. Una firma PAdES è infatti costituita dall’immagine di una coccarda, di un’icona del produttore del programma di firma (completata con il nome del firmatario), o dalla riproduzione di una firma autografa del firmatario.

Tutti questi oggetti grafici possono ovviamente essere modificati, utilizzando programmi appositi per alterare il file PDF.

Più rozzamente, ma molto più semplicemente, possono essere manipolati importando il file pdf in un word processor, modificandone il contenuto, ed infine esportandolo nuovamente in formato PDF.

Coccardine posticce

Questo processo genera ovviamente file falsificati solo nell'aspetto grafico.

Infatti una verifica crittografica della firma interna, anche se ancora presente nel file, permette di verificare infallibilmente la falsificazione.

Tuttavia è purtroppo abbastanza frequente che documenti digitali, che appaiono graficamente firmati non vengano verificati crittograficamente, ma "accettati" acriticamente come validi; in questo caso anche una alterazione solo della parte grafica potrebbe raggiungere il suo scopo.

Per quanto esposto in precedenza, in ambito PCT, in presenza di problemi di firme non valide, può accadere che il documento venga utilizzato senza nessun ulteriore accertamento.

Un consulente tecnico, od un utente accorto, può facilmente individuare queste situazioni, ma solo una corretta "educazione digitale" di tutti gli attori interessati può rendere queste grossolane falsificazioni certamente inefficaci.

Malware PDF

Negli stream binari, che sono contenuti in un file PDF e che vengono utilizzati per creare gli oggetti, può in realtà essere inserito qualsiasi contenuto [10] [11].

In particolare possono essere inseriti interi file di qualsiasi tipo, che possono o meno essere visualizzati, e che possono essere "invisibili" agli utenti.

Questa potente funzionalità del formato PDF (in realtà del linguaggio PostScript stesso) apre tuttavia la strada alla scrittura di malware di vario tipo.

E' infatti possibile inserire in un PDF dei contenuti eseguibili, ad esempio un file .doc contenente macro VB, oppure un file .exe eseguibile direttamente.

In unione con altre vulnerabilità del sistema operativo e dell'applicazione bersaglio, questo può permettere l'esecuzione di software arbitrario con i privilegi dell'utente locale.

PDF eseguibili

Una dimostrazione tanto efficace quanto innocua della possibilità di inserire contenuti attivi in un file PDF è stata la produzione di una versione PDF dello storico e popolarissimo arcade "Doom". Esiste infatti una tradizione hacker di far girare questo antico videogioco su qualsiasi cosa. Ne sono state prodotte versioni per cellulari, gadget, display di stampanti, ed addirittura per test di gravidanza elettronici.

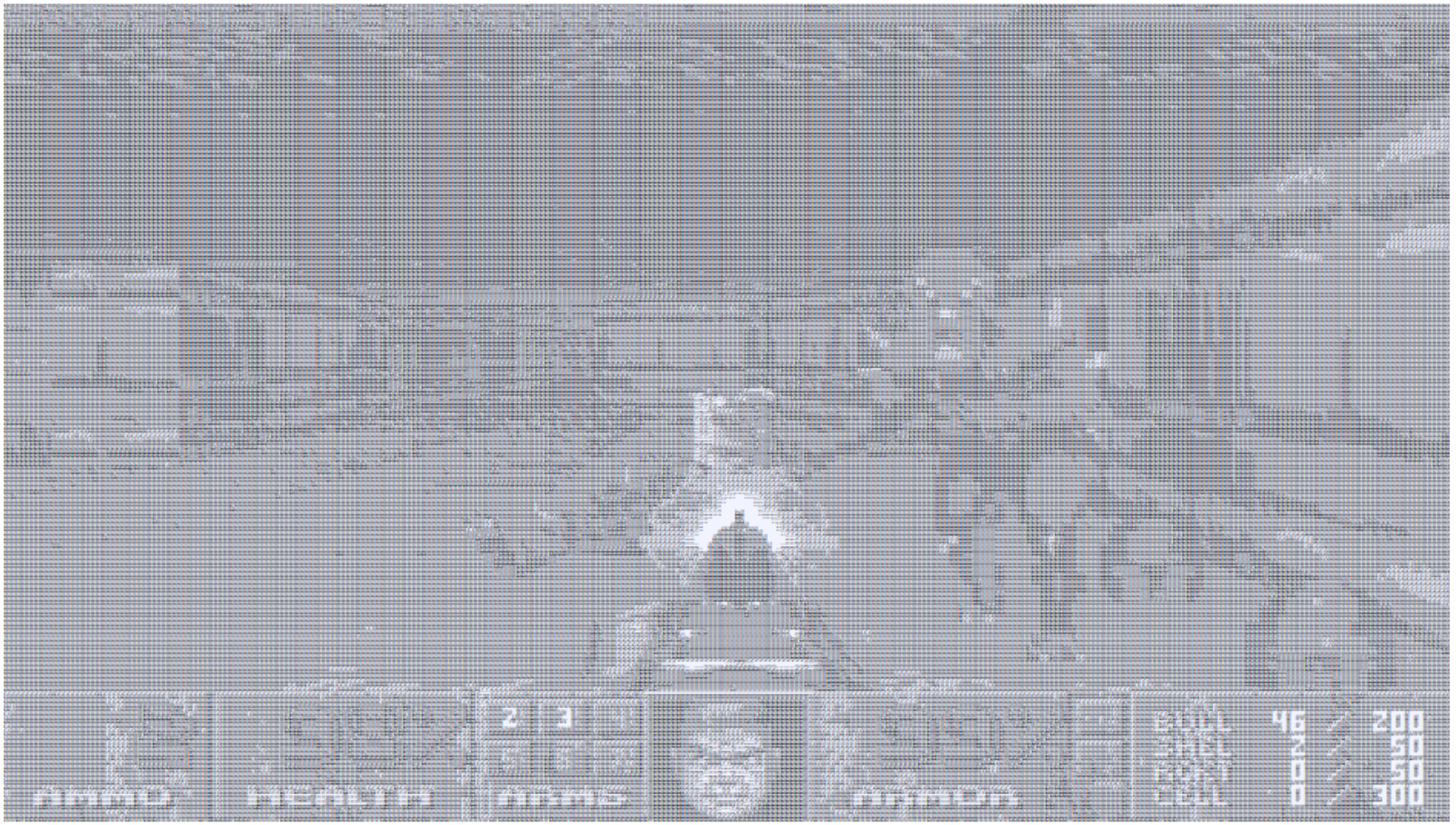
Aprendo il file PDF doompdf.pdf con qualsiasi browser basato sul motore Chromium, come Chrome od Edge (non funziona aprendolo con Acrobat Reader), è possibile giocare ad una versione interattiva e funzionante del gioco. Potete anche semplicemente navigare, a vostro rischio e pericolo, a questo link:

https://doompdf.pages.dev/doom.pdf

il progetto su Github:

https://github.com/ading2210/doompdf

doompdf.pdf



FOR A PARTICULAR PURPOSE. You are welcome to change and distribute
copies under certain conditions. See the source for more information.

I_Init: Setting up machine state.
M_Init: Init miscellaneous info.
R_Init: Init DOOM refresh daemon - .....
P_Init: Init Playloop state.
S_Init: Setting up sound.
D_CheckNetGame: Checking network game status.
startskill 2 deathmatch: 0 startmap: 1 startepisode: 1
player 1 of 1 (1 nodes)
Emulating the behavior of the 'Doom 1.9' executable.
HU_Init: Setting up heads up display.
ST_Init: Init status bar.
I_InitGraphics: framebuffer: x_res: 320, y_res: 200, x_virtual: 320, y_virtual: 200, bpp: 32
I_InitGraphics: framebuffer: RGBA: 8888, red_off: 16, green_off: 8, blue_off: 0, transp_off: 24
I_InitGraphics: DOOM screen size: w x h: 320 x 200
I_InitGraphics: Auto-scaling factor: 1
frame time: 43 ms (23 fps)
frame time: 42 ms (23 fps)
frame time: 44 ms (22 fps)
frame time: 108 ms (9 fps)
frame time: 147 ms (6 fps)

DoomPDF

esc	^	use
<	v	>
enter	fire	

Controls:

```
WASD, q = esc, z = enter, e = use, space = fire
shift+WASD = sprint, m = map, 1-7 = weapons
```

Type here for keyboard controls.

```
Upload custom WAD files at: https://doompdf.pages.dev/
Source code: https://github.com/ading2210/doompdf
Note: This PDF only works in Chromium-based browsers.
```

Grazie per l'attenzione.

(e magari buon divertimento!)

Ci sono 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 -------------+

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 software per l'analisi di file PDF

https://github.com/zbetcheckin/PDF_analysis

[7] Corso pratico di Postscript

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

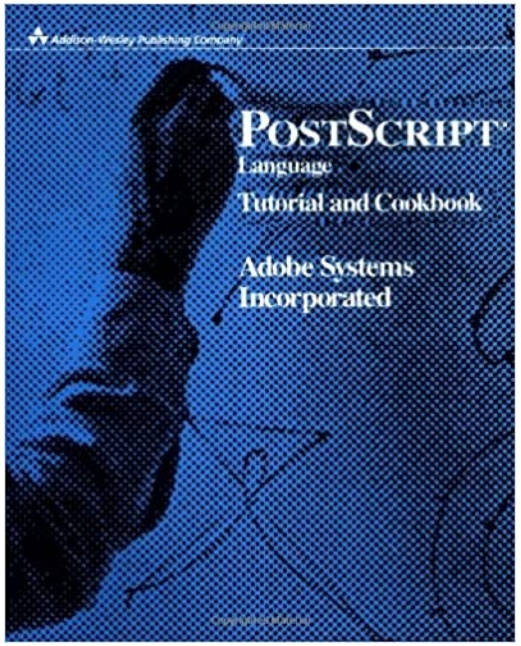
https://www.marcoc.it/corsops/indice.htm

[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/



Per approfondire

[10] Hiding Malicious Content in PDF Documents

https://arxiv.org/pdf/1201.0397

[11] JPCERT/CC blog: MalDoc in PDF - Detection bypass by embedding a malicious Word file into a PDF file

https://blogs.jpcert.or.jp/en/2023/08/maldocinpdf.html