La sicurezza nei documenti digitali: Il caso dei PDF (Parte 1)

Author : Davide Maiorca

Date : 15 ottobre 2018

Introduzione

Capita spesso di associare gli attacchi informatici effettuati attraverso l'uso di software malevolo ("malware", o più volgarmente "virus") a dei file eseguibili o alla navigazione in siti web infetti. In realtà, le tecniche di diffusione di software malevolo sono molteplici, e si basano spesso sull'utilizzo di strategie di social engineering. In altre parole, dal punto di vista dell'attaccante, è molto più semplice che la vittima apra una mail contenente un documento relativo a delle sue "debolezze" (ad esempio, una determinata bolletta da pagare) che direttamente un file eseguibile, il quale risulterebbe molto più sospetto. Per questo motivo, i documenti digitali sono dei vettori perfetti per veicolare attacchi informatici di vario tipo, in quanto sfruttano non solo le vulnerabilità informatiche del sistema target, ma anche quelle psicologiche della vittima.

Gli attaccanti possono nascondere attacchi sofisticati in documenti digitali: i formati più utilizzati sono certamente PDF e Microsoft Office (doc, xls...). Anche se, negli ultimi tempi, il secondo formato è preferito dagli attaccanti per la maggiore facilità di infezione, i file PDF sono ancora ampiamente usati per effettuare attacchi (ad esempio, [1]). In questa serie di articoli, composta da due parti, cercheremo di capire come un attaccante possa effettuare un attacco informatico attraverso l'uso di un file PDF.

In questo primo articolo, si fornirà una panoramica del funzionamento di un file PDF e della sua struttura. Nel secondo articolo, si andrà più in profondità nelle problematiche di sicurezza relative ai file PDF, mostrando un esempio reale di attacco informatico effettuato attraverso questo formato.

Struttura File PDF

PDF sta per Portable Document Format, ed è uno dei formati più popolari, assieme a Microsoft Office, per visualizzare documenti digitali.

La sua caratteristica principale è quella di supportare diverse funzionalità che rendono più interattiva l'esperienza dell'utente, come l'uso di form, la possibilità di visualizzare video, immagini e molto altro. Inoltre, supporta l'utilizzo di codice JavaScript per implementare funzionalità aggiuntive che migliorano l'esperienza complessiva dell'utente.

In linea generale, un file PDF può essere definito come un grafo di oggetti direttamente interconnessi fra di loro. Per poter capire come funziona, è possibile aprirlo direttamente con un programma come Notepad (o ancora meglio, Notepad++ [2]). Si può pertanto evincere che ogni file PDF è composto da quattro parti principali, mostrate nella Figura 1:

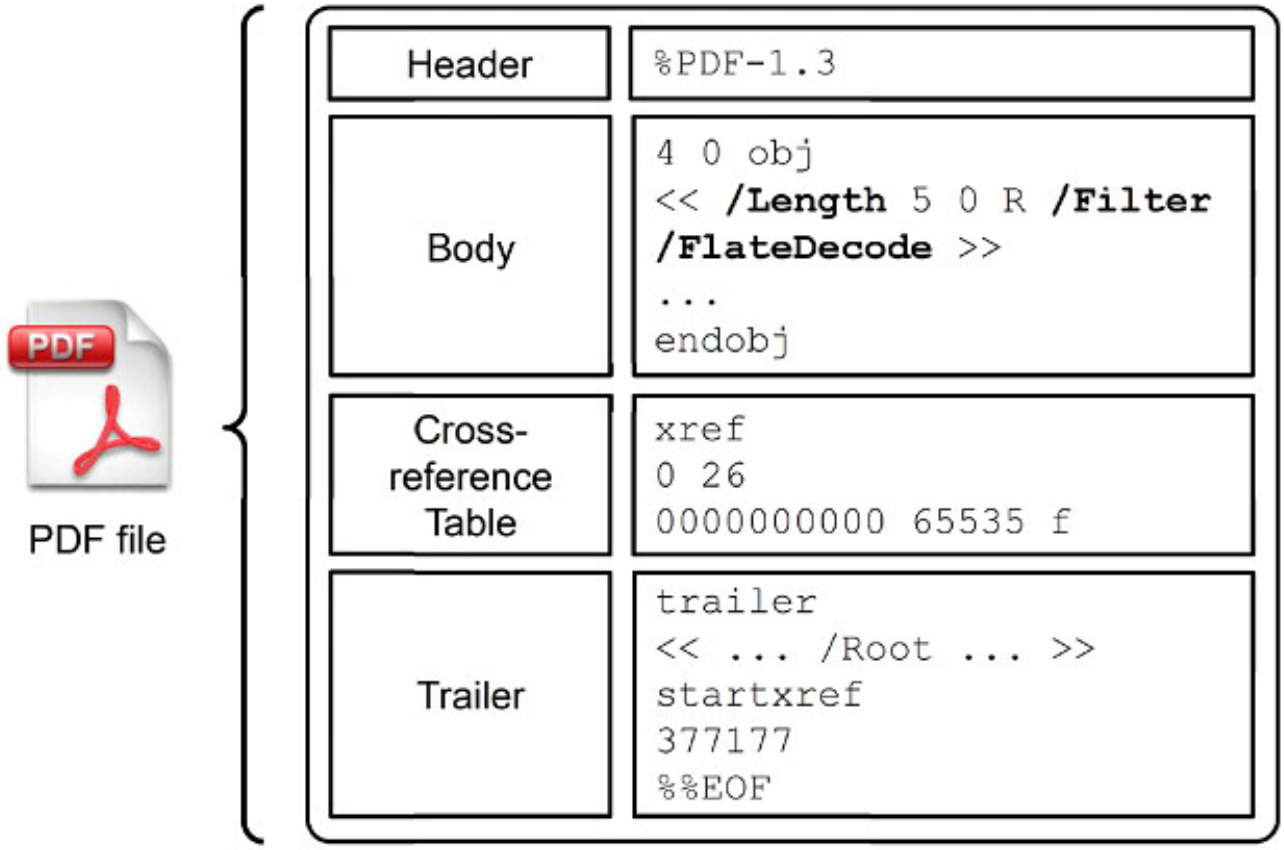

Figura 1 - Struttura generale di un file PDF

- **Header**. La prima riga del file, indicante la versione del formato PDF (attualmente, l'ultima versione del formato è la 1.7, rilasciata nel 2008). La versione del formato è importante perché ci sono state modifiche importanti fra un formato e l'altro, fra cui la possibilità di usare misure di sicurezza più avanzate. È possibile visionare le specifiche complete del formato qui: [3].
- **Body**. Una lista di oggetti che rappresentano il vero e proprio "cuore" del file. Ognuno di questi oggetti contiene delle informazioni sulla propria funzionalità (ad esempio, se si riferisce ad un font, a del codice di scripting, o ad una immagine). Per descrivere la

funzionalità si utilizzano dei dizionari (introdotti da "**>**") contenenti sequenze di keyword, introdotte da uno "/" (ad esempio, **/Length, /Filter, /FlateDecode**). Queste keyword sono generalmente espresse in coppia con un altro elemento del file, ad esempio un riferimento o un'altra keyword (ad esempio, **/Filter /FlateDecode** indica che si utilizza un filtro di compressione di tipo FlateDecode). Gli oggetti in un file PDF sono generalmente numerati seguendo la dicitura "NUM 0 obj", dove NUM rappresenta il numero dell'oggetto. Inoltre, è possibile che questi contengano dei dati (solitamente compressi attraverso un meccanismo di filtraggio, di cui parleremo nel prossimo articolo) che vengono elaborati direttamente dal Reader. Questi dati vengono chiamati stream. Infine, ognuno di questi oggetti può contenere dei riferimenti ad altri oggetti del file. Tali riferimenti sono indicati dalla notazione "NUM 0 R", dove NUM rappresenta, in questo caso, l'oggetto a cui riferirsi. È altresì importante sottolineare che gli oggetti non sono ordinati, ma la loro disposizione può anche essere casuale all'interno del body.

- **Cross-Reference Table (X-Ref Table)**. Tabella contenente le informazioni sulla posizione di determinati oggetti all'interno del file PDF. In altre parole, tale tabella può essere rappresentata come un elenco di indirizzi relativi alla posizione (in termini di byte) di ogni oggetto nel file. Questo è fondamentale, in quanto Adobe Reader deve sapere da dove iniziare a processare ogni oggetto del file. Se l'oggetto non è presente nella Cross-Reference Table, NON verrà analizzato da Adobe Reader. La Cross-Reference Table è introdotta dal tag xref.
- **Trailer**. Il trailer è un oggetto speciale che contiene alcune informazioni basilari del file PDF. In particolare, contiene un riferimento al primo oggetto della gerarchia e dei riferimenti ad altri oggetti contenenti dei metadati, ovvero informazioni riguardanti l'autore del file o indicanti lo strumento con cui il file è stato creato (ad es. salvataggio da Microsoft Office). È introdotto dal tag trailer.

Per poter visualizzare un file PDF, pertanto, il Reader dovrà eseguire le seguenti operazioni:

- Dovrà validare l'header per capire la versione del formato PDF.
- Cercherà l'oggetto trailer, e da lì inizierà a caricare i vari oggetti contenuti nel body seguendo le indicazioni fornite dalla Cross-Reference Table.

Si consideri ora il seguente esempio di file PDF, avente il trailer mostrato in Figura 2:

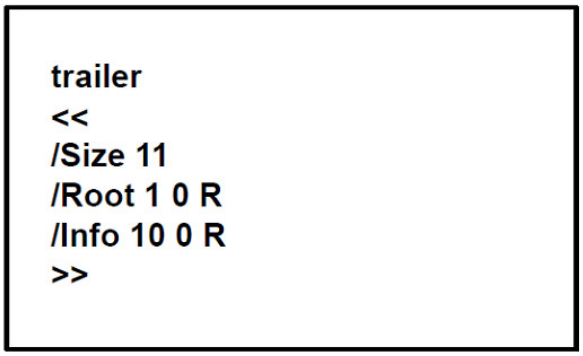

Figura 2 - Trailer di esempio di un file PDF

Adobe Reader inizierà ad analizzare il file a partire da questo trailer (indicante che sono presenti 11 oggetti – incluso il trailer stesso - nella gerarchia, si veda la keyword **/Size**). L'oggetto rappresentato dalla keyword **/Root** indica che il primo oggetto della gerarchia sarà il numero 1 (il fatto che sia esattamente il numero 1 è, tuttavia, una coincidenza. L'oggetto **/Root** può avere un qualunque numero). Il trailer ci dice anche l'oggetto numero 10 contiene informazioni sui metadati del file (keyword **/Info**).

Supponiamo che il body sia strutturato come in Figura 3. Di seguito, indichiamo solo una porzione degli oggetti in modo da semplificare la visualizzazione.

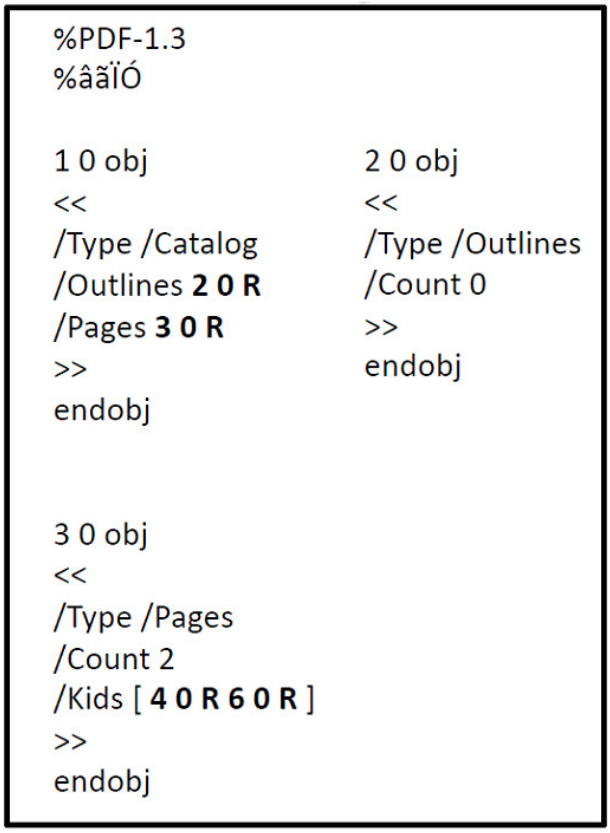

Figura 3 - Distribuzione di oggetti nel body di un file PDF

Questa immagine fornisce una serie di informazioni importanti su come gli oggetti vengono trattati da Adobe Reader. Dato che il primo oggetto della gerarchia è il numero 1 (informazione ottenuta dal trailer precedente), Reader inizierà ad analizzare l'oggetto 1 0 obj. Questo oggetto possiede due keyword fondamentali, **/Type** e **/Catalog**, che stanno ad indicare che quello è l'oggetto principale della gerarchia. Inoltre, sono presenti due riferimenti agli oggetti 2 e 3. L'oggetto numero 2 contiene informazioni relative alla presenza di sommari (non ve ne sono, infatti il **/Count** nell'oggetto numero 2 riferito alle **/Outlines** è accompagnato da uno zero). L'oggetto 3, invece, contiene informazioni sulle pagine del documento. È facile osservare come

l’oggetto si riferisca alla struttura, in termini di pagine, del documento (osservare le keyword **/Type** e **/Pages**). Il file avrà due pagine (**/Count 2**) e le informazioni relative alle due pagine sono contenute negli oggetti 4 e 6 (**/Kids**).

La struttura degli oggetti analizzati può essere pertanto riassunta come in Figura 4:

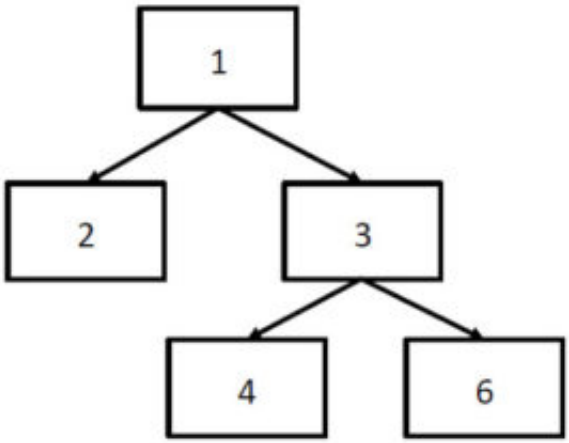

Figura 4 - Struttura degli oggetti nel body per il file PDF d'esempio

Come si può notare, il file PDF è organizzato in maniera precisa e solo gli oggetti presenti nel body (indicati nella Cross-Reference Table) sono effettivamente analizzati da Adobe Reader.

La Cross-Reference Table relativa al file è rappresentata in Figura 5:

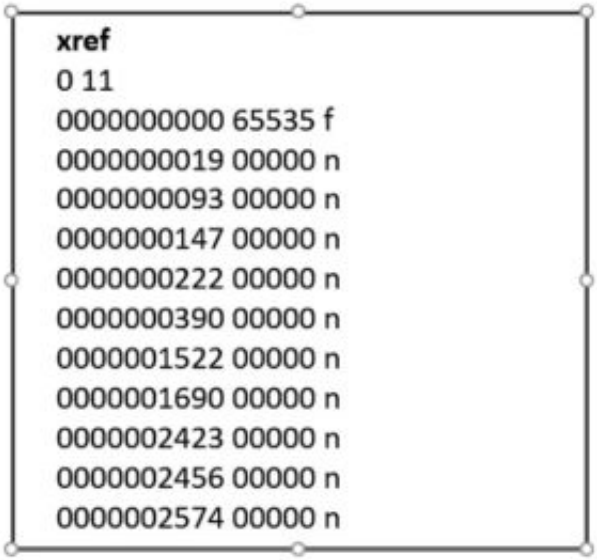

Figura 5 – Cross-Reference Table relativa al file PDF in esame

L’aspetto principale che si può evincere da questa tabella è il fatto che vi siano esattamente 11 oggetti, come indicato nel trailer (la posizione indicata è però quella dei 10 oggetti del body, il trailer non è incluso). Notare come il marker n, alla fine di ogni indirizzo del file, indichi che quell’oggetto verrà effettivamente analizzato e visualizzato da Reader.

Queste sono le basi per poter comprendere il funzionamento di un file PDF (e che saranno fondamentali per capire il funzionamento degli attacchi). Per maggiori informazioni, si rimanda alle specifiche ufficiali del formato [3]. Nel prossimo articolo, analizzeremo la struttura di un vero e proprio malware in PDF con le basi apprese in questa prima parte.

Riferimenti

- [1] MorphiSec Blog. [CRITICAL ALERT] CVE-2018-4990 Acrobat Reader DC Double-Free Vulnerability. http://blog.morphisec.com/critical-alert-cve-2018-4990-acrobat-reader-dc-double-free-vulnerability
- [2] Notepad++. https://notepad-plus-plus.org/download/v7.5.8.html
- [3] Adobe. PDF Reference 1.7. https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdfs/PDF32000_2008.pdf

Articolo a cura di **Davide Maiorca**