Preflight DNS

By Marcus Wengelin, Bitsec AB

Sample1 (embedded1.pdf)

The PDF-file was generated by LibreOffice

Contains the following object references:

- JavaScript
- Image
- OLE(Object Linking and Embedding)-Object
- Plugin

The table below shows which pdf-readers made DNS-request for which embedded object.

	Adobe	Foxit	Firefox	Chrome	Office	Evince	Edge
JavaScript	N		Y	N		N	N
Image	N		Y	N		N	N
OLE-Obj	N		N	N		N	N
Plugin	N		Y	N		N	N

Notes: As shown by the table above, it is quite difficult to force the client to perform DNS queries. These are the normal items you would expect to be embedded in a pdf-file, maybe we will have some success. To achieve more DNS name resolutions we need to explore the less common embedded objects.

Sample2 (embedded2.pdf)

This PDF-file was generated using Scribus, which has more features than LibreOffice.

This pdf contains the following object references :

- Goto-button
- Submit-button
- Import-button

	Adobe	Foxit	Firefox	Chrome	Office	Evince	Edge
Goto	N/N		N/N	N/N		N/N	N/N
Submit	N/Y		N/N	N/Y		N/N	N/N
Import	N/N		N/N	N/N		N/N	N/N

Obs: Y – DNS queries detected, N – No DNS Queries detected (Without interaction/Without interaction)

Notes:

Adobe, Submit, User-interaction : Presents the user with a warning before sending the query
Edge : Does not load the buttons
Chrome, Submit, User-interaction : Sends the query as soon as the user clicks the button
Firefox : Does not load the buttons

Sample3 (embedded3.pdf)

This PDF-file was generated using Scribus, and edited using a hex editor. This sample is more focused on file metadata than the previous samples, which focused more on objects.

This pdf contains the following references :

- Creator
- Producer
- Title
- Author
- Subject

	Adobe	Foxit	Firefox	Chrome	Office	Evince	Edge
Creator	N		N	N		N	N
Producer	N		N	N		N	N
Title	N		N	N		N	N
Author	N		N	N		N	N
Subject	N		N	N		N	N

Notes : As shown by the table above, the metadata is consistently treated as just plain text across all platforms.

Sample4 (embedded4.pdf)

Here we try different font manipulations. The first manipulation is in the DroidSans font declaration, and we inject a URI-object instead of the font-name. The second manipulation is of the FreeMono font, where we inject the URI as a subtype instead of the name declaration. The third edit is a manipulation of the font-file declaration of the Cantarell font, and we instead insert a URI-object. The fourth manipulation is also a manipulation of the font-file declaration in the Deja Vu Sans font. Here we instead append the URI-object to the file-declaration The fifth manipulation is of the Liberation Serif font, and here we just inject the URI-object right before the FontFile object.

	Adobe	Foxit	Firefox	Chrome	Office	Evince	Edge
DroidSans	N		Y	N		N	N
FreeMono	N		Y	N		N	N
Cantarell	N		Y	N		N	N
DejaVu	N		Y	N		N	N
LibSerif	N		Y	N		N	N

Notes : Some software shows a warning that the pdf may be corrupted, the other programs do not properly display the font. It would seem like Firefox does premature DNS-lookups on all URI-objects found in the document.

Conclusion

Current PDF-readers seem to be decently protected against the preflight-DNS phenomenon. The most vulnerable PDF-reader is Firefox, which is consistently vulnerable to most of the attempted attacks, and most importantly, those that do not require any user interaction.

To move forward with this project, I would suggest writing a fuzzer which injects urls and URI-objects at multiple places in the PDF-file, and also implement automated tests.

Software used :

Adobe Acrobat Reader DC v 2015.017.20050
Microsoft Edge v 25.10586.0.0
Google Chrome v 51.0.2704.106
Mozilla Firefox ESR 45.2.0
Evince 3.14.1