# (U) Ask Raul: PDF Files

FROM: Raul, a DNI Analyst
Unknown
Run Date: 12/09/2004

Dear Raul,

(C) I get a fair number of PDF files in my DNI collection and when they are good, they're great. However, when they are even the least bit damaged there isn't a thing I can do. Sometimes, I'll try to load one and it will flash something at me from the Acrobat Reader and then it's gone. It looks like stuff should be coming out but all I end up with is an error message. What's up with these things?

Rachel

---

Dear Rachel,

(U) I love PDF. Portable Document Format from Adobe is one of the neatest document formats around. If it is an indication of anything, the current PDF specification document, for version 1.6, is over 1200 pages long. First, a bit of explanation on how it works.

(U) All PDF files are basically archives. The document is divided up into objects that are generally text, images, fonts, and encoding tables. Each of these objects can be encoded, compressed and/or encrypted. Generally speaking though, they are usually just compressed using a well-known method. Additionally, there is a structure within the file that tells where all the objects are. There's a whole lot of stuff going on inside these files but this is a reasonable enough basic explanation.

(U) So, here is a little example of an object:

Obviously, whatever happens to be inside this little blob of binary is not going to be something we can select against and especially not if it is underneath some other compression, encoding or both (i.e. zip, base64). Fortunately, Adobe is nice enough to tell us exactly which compression is being used (Flate) so decompressing this is a snap and here is what it looks like:

That's a bit better but as you can see, the text is a bit of a mess. That is because it is formatted for display. The real text is all of the material in parentheses. As you can see, some words are all in one piece while others have been separated in what appear to be some very strange ways. Notice how the word "love" was turned into (l)0(o)16(v)15(e). Therefore, the next thing we want to do is to take care of this formatting, which is a piece of cake, and turn this text back into a form we can search against it or read. If you'll notice, this is just the first paragraph of this article.

(U) So, now you see that to properly process a PDF document

**SERIES:**
**(U) Ask Raul - Answers to DNI Questions**

containing English text you'll have to:

1. Find the objects containing something of value (text, image, whatever).

2. Decompress, decode and/or decipher the object.

3. Remove the formatting from the text.

Seems simple enough but some PDF documents can have hundreds or thousands of objects. This means, potentially, a whole lot of decoding, decompressing and/or decryption. But it gets better!

(S) Take a peep at this:

Beautiful, isn't it? You could use everything available to you here at NSA. Put in all of these words into a dictionary in every possible character set and guess what? You'll never get a hit against this document. Why? Here you go.

(U) When we move outside of English with PDF, some very interesting and intelligent things get done. The designers of PDF looked at the problem like this:

*Hmmm. We'll, we could include whole fonts for particular character sets and deal with multiple character set encodings but man, even a font for an 8-bit character set is huge and if we take on languages like Chinese, Japanese and Korean, well, things just get much worse and our files will be humongous. Let's see. Wait a minute! I've got it. Instead of sending the entire font, let's see which characters we actually used and only send those. Brilliant! And better yet, we'll compress all this stuff and when we're through, the file will be much smaller. Great! We are good!*

(S) So, in order to perform this magical trick, they re-encode the text and create an encoding table that maps the new encoding to some known encoding. This being the case, in order to convert the underlying text back into something we can select against, we have to find these encoding tables and apply them to the encoded text in the appropriate object so as to render them in a form we can actually select against. Better still, each object might have had characters which came from multiple fonts so that we would then have to know about all the other encoding tables used by that object in order to re-encode it properly.

(C) Now, you say, "Great Raul! That's what we're doing, right?" Sadly enough, the answer is no. Worse still, no consideration has been given to the fact that the vast majority of the PDF files we run across are of such a form that we could, if we were so inclined, split them up and reassemble them as we saw fit. Yes, you see where I'm going here, don't you? If we can do this, it also means we should be able to deal with broken or damaged PDF files with great ease. Again, unfortunately, the sad truth is we can't do that either. It really isn't even a matter of our not being able to do it as much as one of simply not having done it. And PDF's aren't the only things we have this trouble with as you well know.

(C) So, there you go. Think of it like this:

If you were looking for Osama bin Laden, and you had entered every Arabic word known to mankind in every possible encoding and Osama were doing nothing more than using PDF and writing in

Arabic, you'd never get a hit. Quite reassuring, isn't it?

Raul



---