Will random data appended to a JPG make it unusable?

Asked 16 years, 3 months ago Modified 10 years, 3 months ago

Viewed 5k times

▲

25

▼

So, to simplify my life I want to be able to append from 1 to 7 additional characters on the end of some jpg images my program is processing*. These are dummy padding (fillers, etc - probably all 0x00) just to make the file size a multiple of 8 bytes for block encryption.

Having tried this out with a few programs, it appears they are fine with the additional characters, which occur after the FF D9 that specifies the end of the image - so it appears that the file format is well defined enough that the 'corruption' I'm adding at the end shouldn't matter.

I can always post process the files later if needed, but my preference is to do the simplest thing possible - which is to let them remain (I'm decrypting other file types and they won't mind, so having a special case is annoying).

I figure with all the talk of Steganography hullaballo years ago, someone has some input here...

(encryption processing by 8 byte blocks, I don't want to save pre-encrypted file size, so append 0x00 to input data, and leave them there after decoding) jpeg file-format steganography

Share

Improve this question

Follow

edited Aug 30, 2014 at 21:08

AstroCB
12.4k ● 20 ● 58 ● 74

asked Sep 8, 2008 at 23:39

Adam Davis
93.5k ● 60 ● 271 ● 333

4 Answers

Sorted by: Highest score (default)

25

✓

No, you can add bits to the end of a jpg file, without making it unusable. The heading of the jpg file tells how to read it, so the program reading it will stop at the end of the jpg data.

In fact, people have hidden zip files inside jpg files by appending the zip data to the end of the jpg data. Because of the way these formats are structured, the resulting file is valid in either format.

Share Improve this answer

Follow

edited Sep 12, 2008 at 17:50

answered Sep 8, 2008 at 23:43

pkaeding
37.6k ● 31 ● 106 ● 142

unusable (which was the question) is not the same as reducing it's usability. Bit of a nickpick, I'll grant. But, an important distinction because it may be a viable solution. – Stu Thompson Sep 9, 2008 at 9:17

Ok, I'm not sure that saying "without reducing its usability" could mean "make it unusable" but I will edit this just to make it more clear. – pkaeding Sep 12, 2008 at 17:50

7

You can .. but the results *may* be unpredictable.

Even though there is enough information in the format to tell the client to ignore the extra data it is likely not a case the programmer tested for.

A paranoid program might look at the size, notice the discrepancy and decide it won't process your file because clearly it doesn't fully understand it. This is particularly likely when reading data from the web when random bytes in a file could be considered a security risk.

Share Improve this answer Follow

answered Sep 8, 2008 at 23:53

Rob Walker
47.4k ● 15 ● 100 ● 137

3

You can embed your data in the XMP tag within a JPEG (or EXIF or IPTC fields for that matter). XMP is XML so you have a fair bit of flexibility there to do you own custom stuff.

It's probably not *the simplest thing possible* but putting your data here will maintain the integrity of the JPEG and require no "post processing".

You data will then show up in other imaging software such as PhotoShop, which may not be ideal.

Share Improve this answer Follow

answered Sep 9, 2008 at 0:59

Stuart Helwig

9,444 ● 8 ● 55 ● 69

0

As others have stated, you have no control how programs process image files and therefore some programs may find the images valid others may not.

However, there is a bigger issue here. Judging by your question, I'm deducing you're practicing "security through obscurity." It's widely considered a very bad practice. Use Google to find a plethora of articles about the topic.

Share Improve this answer Follow

answered Sep 9, 2008 at 3:44

xanadont

7,604 ● 6 ● 38 ● 51

5 What indicates this? I'm using the TEA algorithm to encrypt the data, but it encrypts 8 bytes at a time. Rather than including meta data inside the encrypted file to tell me how big the original data was, I'm padding the original out to an 8 byte boundary, and leaving the padding in place.
– Adam Davis Sep 9, 2008 at 6:06