

CMPUT 333

Security in a Networked World

Lab Assignment 1
Part 2 – File Signatures

Overview

- This presentation aims to provide additional information that may be needed to complete part 2 of assignment 1.
- We will be discussing file types and file signatures, along with a couple of examples.

File Types?

- Why is it that when you open a .pdf or .docx file in a text editor like Vim you see a bunch of gibberish?
- Why is it that when you open a .pdf file using Acrobat Reader or a .docx file using Microsoft Word you see meaningful information?
- What is the difference between an ASCII text file and any other file type that exists out there?
- We will explore these issues during this presentation.

File Types

- Here's the deal: programs like Microsoft Word allow you to do some pretty fancy stuff, visually (at least compared to plain ASCII text files.)
- They do this by encoding a bunch of control information into a file, allowing for more graphical versatility.
- This information may include spacing, font size, font type, color, graphic encoding, etc.

File Types

- On the flip side of things, plain text (ASCII) files are relatively straight forward, i.e. each byte of the binary data contained within a file corresponds to one ASCII character that can be displayed to the screen (as long as it is a printable character).

File Types

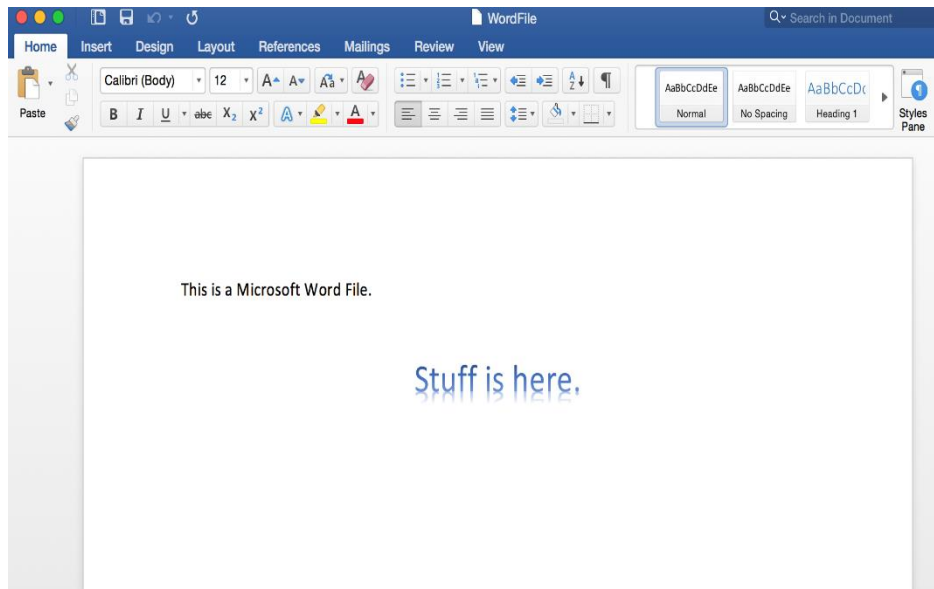
- Next we will take a look at an example involving a Microsoft Word file.
- As you may or may not know, these files are often called docx files because of their extension .docx.

Word Example

- Microsoft Word will encode the following file on the left

Word Example

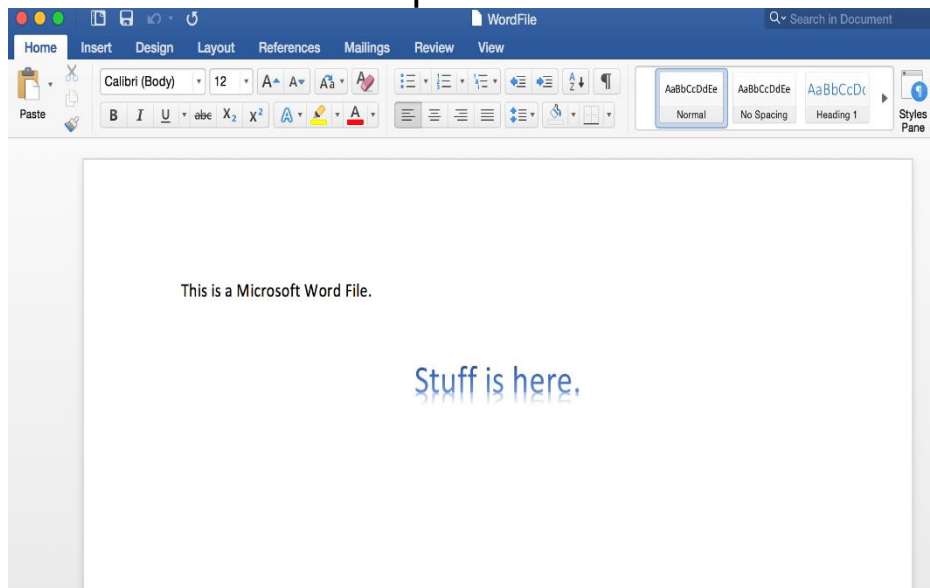
- Microsoft Word will encode the following file on the left



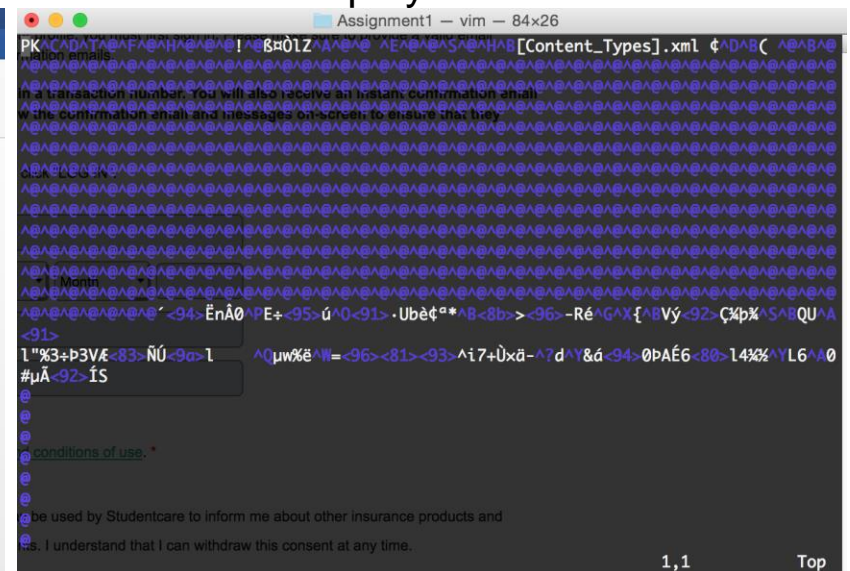
Word Example

- Microsoft Word will encode the following file on the left
- And store it as a bunch of seemingly random characters that you see on the right.

.docx file opened in word



.docx file displayed as ASCII file



Word Example

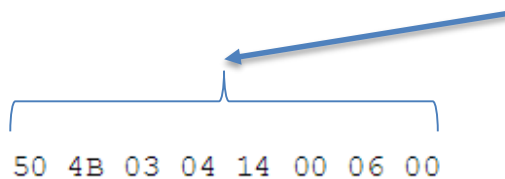
- Why does this seemingly basic .docx file (when viewed using Microsoft Word) look so incredibly complicated and bloated when viewing its actual binary contents?
- Again this goes back to the encoding that must take place in order to allow for visuals that go beyond simple ASCII characters.
- Encoding occurs for many other files type too, such as PDFs, movies, pictures, spreadsheets, etc.

File Signatures

- Now the next question you may be asking yourself is “How can a program tell if a file is of a specific file type, and thus encoded in a way that it is expecting it to be?”
- The answer is in something called *file signatures*.
- Much like network packets have headers and trailers, so do computer files.
- These headers and trailers encompass a file’s signature.
- This is not to be confused with file checksums which are usually used for integrity purposes.
- For a nice list of file signatures pertaining to most common file types in existence nowadays, you can check out this site: www.garykessler.net/library/file_sigs.html
- Let us now revisit the previous Microsoft Word file and see if it indeed has the correct header.

File Signatures

- First, if we visit the file signature site we can see that the header for .docx files should be:



50 4B 03 04 14 00 06 00

DOCX, PPTX, XLSX

PK.....

Microsoft Office Open XML Format (OOXML) Document

NOTE: There is no subheader for MS OOXML files as there is with DOC, PPT, and XLS files. To better understand the format of these files, rename any OOXML file to have a .ZIP extension and then unZIP the file; look at the resultant file named *[Content_Types].xml* to see the content types. In particular, look for the *<Override PartName=* tag, where you will find *word*, *ppt*, or *xl*, respectively.

Trailer: Look for 50 4B 05 06 (PK. .) followed by 18 additional bytes at the end of the file.

- This means the first byte will be 0x50, the second 0x4B, and so on.

File Signature Example

- Now if we revisit the binary data of our .docx file we can see that the first two characters are 'PK', which corresponds to 0x50 and 0x4B on the ASCII chart, respectively.

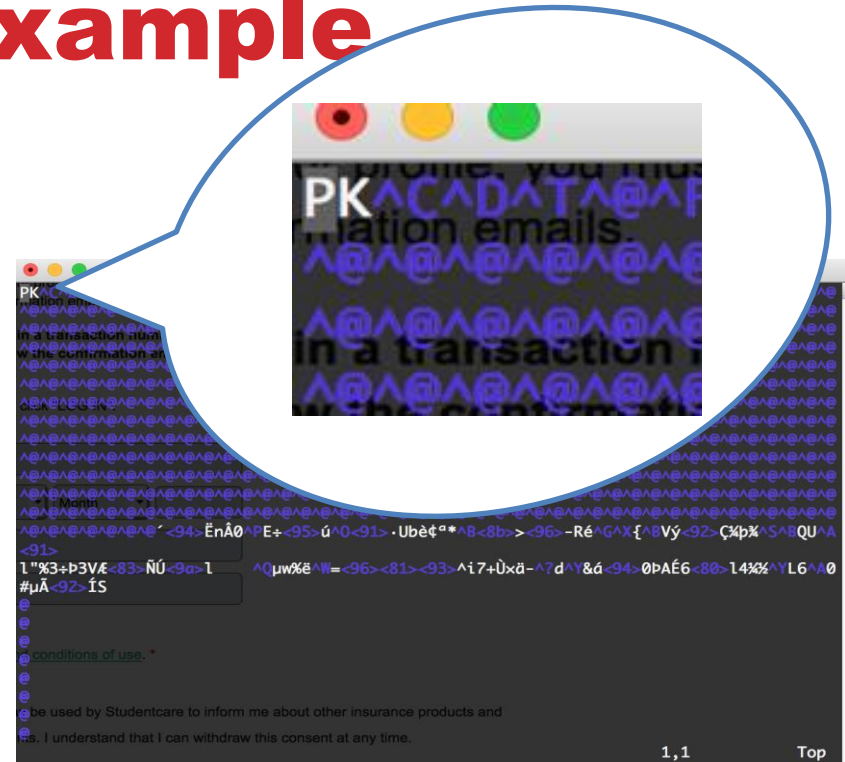
14

- 

[illegible]

File Signature Example

- Now if we revisit the binary data of our .docx file we can see that the first two characters are 'PK', which corresponds to 0x50 and 0x4B on the ASCII chart, respectively.



File Signature Example

- Now if we revisit the binary data of our .docx file we can see that the first two characters are 'PK', which corresponds to 0x50 and 0x4B on the ASCII chart, respectively.



- Looking at the hexadecimal representation of this .docx file as opposed to the ASCII values we can see that the rest of the header is indeed there.

[illegible]

File Signature Example

- Now if we revisit the binary data of our .docx file we can see that the first two characters are 'PK', which corresponds to 0x50 and 0x4B on the ASCII chart, respectively.



- Looking at the hexadecimal representation of this .docx file as opposed to the ASCII values we can see that the rest of the header is indeed there.

[illegible]

```
0000000: 504b 0304 1400 0600 0800
0000010: d26c 5a01 0000 2005 0000
0000020: 6f6e 7465 6e74 5f54 7970
0000030: 6c20 a204 0228 a000 0200
```

Food for thought

- **Here are some questions you can explore on your own time:**
 1. What happens when I change a file's header and/or trailer?
 2. What happens when I change some data that is not in a file's header?
 3. What is the point of a file extension (.pdf, .docx, etc.)
 4. How can file signatures help us complete part 2 of this assignment?

QUESTIONS ?