File uploads

The Jack in the AppSec Box

Gen AI Picture

Dirk Wetter

Who’s talking?

Selfemployed Security Consultant

- Pentests: apps, networks, any systems.
- Help doing/fixing stuff properly
- Infosec management

Open Source engagements

- **Long time involved in OWASP activities**
  - national (German)International
    - OWASP AppSec Global in Hamburg 2013,
    - several international program committees
  - OWASP Docker Top 10 (not really alive anymore)
  - Some smaller contributions ...
- **Side "hobby": testssl.sh**

Threats (TL;DR)

- Server side
- Client side ("download" again from server)

Threats (TL;DR)

- **Server side**
  - DoS (disk space) or DoW (w=wallet)
  - Code injection / execution
    - in application
    - while backend processing
  - Overwrite user/ server files
  - Lack of access controls

Threats (TL;DR)

- **Client side**
  - First: do we care?
    - Depends!
      - Application
      - Business
    - No one size fits all

File uploads 101

- **Important points first — applies not only for this talk:**

Server side controls!

File content validation 101

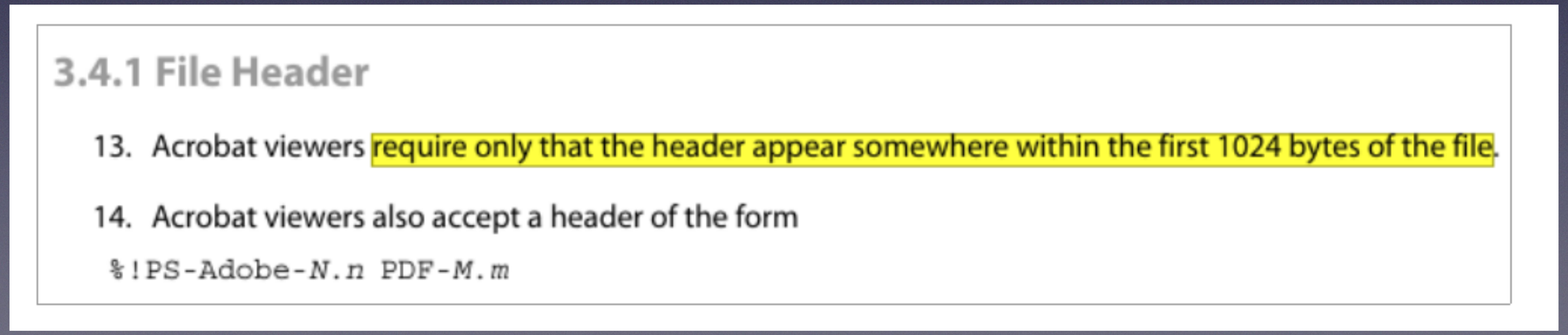
- **Never judge by the extension (or content type)!**
  - `file.jpg` or `file.png` could be
    - HTML file with embedded JS
    - executable script with a reverse shell using a netcat listener
    - not only a JPEG/PNG
  - files can be renamed: `file.jpg` → `file.php`
  - double extension: `file.asp.png`
  - `file.jpg%00.asp`, `file.asp\.png` `file.jpg%0d%0a.php`, `file.jsp/.jpg`

File type validation 101

- **MIME type rules! (a.k.a. magic/file signatures)**
  - Look at the ~first couple of bytes in file
    - `GIF89a`, `PNG`, `%PDF-1.`, `0xFFD8` (JPEG)
  - Libraries take ~good care — but …

File content validation: polyglots

- **Problem:**
  - Some file formats are not strict
    - PDF marker can be in first 1024 bytes (definition **2008**):



3.4.1 File Header

13. Acrobat viewers require only that the header appear somewhere within the first 1024 bytes of the file.

14. Acrobat viewers also accept a header of the form

`%!PS-Adobe-N.n PDF-M.m`

File content validation: polyglots

- **Problem:**
  - Some file formats are not strict
    - PDF marker in ISO Standard 32000 (**2020**)

7.5.2 File header

The PDF file begins with the 5 characters “%PDF–” and byte offsets shall be calculated from the PERCENT SIGN (25h).

NOTE 1 This provision allows for arbitrary bytes preceding the %PDF- without impacting the viability of the PDF file and its byte offsets.

File content validation: polyglots

- **Problem:**
  - Some file formats are not strict
    - PDF marker can be in first 1024 bytes (definition)
    - Or start with an offset, but…

FILE	JPEG	PDF
00000: ff d8	"START OF IMAGE" MARKER	
00002: (ff e0)*<size.16> <content>	"APP0" MARKER (REQUIRED HEADER)	
00014: ff fe <size.16>	"COMMENT" MARKER	
+4: %PDF-1.5	COMMENT CONTENT	PDF SIGNATURE
999 0 obj <<>> stream		STARTING A DUMMY BINARY OBJECT
00039: ...	(OTHER MARKERS, ORIGINAL JPEG DATA)...	
xx : ff d9	"END OF IMAGE" MARKER	
xx+2 : endstream endobj		CLOSING THE DUMMY OBJECT
xx+14: %PDF-1.5 ...		ORIGINAL PDF CONTENTS (MULTIPLE SIGNATURES ARE IGNORED)

*REPLACED WITH 00 00 TO BYPASS ADOBE FILTER

https://github.com/angea/pocorgtfo/blob/master/releases/pocorgtfo03.pdf

challenge and begun dissecting the format. The first four bytes are a valid non-ASCII JavaScript variable 0xFF 0xD8 0xFF 0xE0. Then the next two bytes specify the length of the JPEG header. If we make that length of the header 0x2F2A using the bytes 0x2F 0x2A as you might guess we have a non-ASCII variable followed by a multi-line JavaScript comment. We then have to pad out the JPEG header to the length of 0x2F2A with nulls. Here's what it looks like:

```
~/Downloads xxd -c 32 xss.jpg | head -2
00000000: ffd8 ffe0 2f2a 4a46 4946 0001 0101 0048 0048 0000 0000 0000 0000 0000 0000 0000  ..../*JFIF.....H.H.............
00000020: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000  ................................
~/Downloads xxd -c 32 xss.jpg | head -379 | tail -2
00002f20: 0000 0000 0000 0000 0000 0000 0000 fffe 001c 2a2f 3d61 6c65 7274 2822 4275 7270  ..................*/=alert("Burp
00002f40: 2072 6f63 6b73 2e22 293b 2f2a ffdb 0043 0003 0202 0302 0203 0303 0304 0303 0405   rocks.");/*...C................
~/Downloads
```

https://portswigger.net/research/bypassing-csp-using-polyglot-jpegs

File content validation: polyglots

- **Problem:**
  - Some file formats are not strict
    - PDF marker can be in first 1024 bytes (definition)
    - Or start with an (kind of an) offset
    - Or is not unique: PK
      - zip apk epub jar xpi docx odp ods odt pptx xlsx ...
    - ZIP, 7z, RAR, HTML do also not enforce a zero offset

```
~/somedir for f in *; do
for> echo -n "$f   "; xxd -c4 $f|head -1
for> done
file1.docx   00000000: 504b 0304  PK..
file2.zip    00000000: 504b 0304  PK..
file3.jar    00000000: 504b 0304  PK..
~/somedir
```

Demo

Even worse...

- **PDF**
  - JavaScript
  - actions; links —> can download stuff / SSRF
  - embedding files like malware (e.g. EXE)
- see doom.pdf before my talk
- **Office docs**
  - Can contain viruses, macros etc.
  - Appear as ZIP
  - ZIP bombs
  - XXE
  - XML and other Injection

File content validation

- **Defense (server side)**
  - Developer: use library for validation
    - Test it: polyglots, e.g. github.com/Polydet/polyglot-database
    - Refine as needed (strict start at @0x00)
  - Allow-list your file types, deny all others
  - Re-downloads: as attachment
  - Embedded:
    - X-Content-Type-Options: nosniff,
    - e.g. CSP as XSS (further) mitigation
  - Resize image
    - Careful with tools like ImageMagick

Imagemagick : Vulnerability Statistics

Products (3) Vulnerabilities (652) Search products CVSS Report Metasploit Modules

Vulnerability Trends Over Time

Year	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	File Inclusion	CSRF	XXE	SSRF	Open Redirect	Input Validation
2016	7	1	0	0	0	0	0	0	1	0	0
2017	58	41	0	0	1	0	0	0	0	0	13
2018	3	15	0	0	0	0	0	0	0	0	0
2019	11	16	0	0	0	0	0	0	0	0	0
2020	31	6	0	0	0	0	0	0	0	0	0
2021	4	1	0	0	0	0	0	0	0	0	0
2022	7	5	0	0	0	0	0	0	0	0	1
2023	6	7	0	0	0	0	0	0	0	0	1
2024	0	0	0	0	0	0	0	0	0	0	0
2025	1	0	0	0	0	0	0	0	0	0	0
Total	128	92			1				1		15

https://www.cvedetails.com/vendor/1749/Imagemagick.html

Be careful with BE processing

- **Bad ol' times: ImageTragick (https://imagetragick.com)**

CVE-2016-3714 - Insufficient shell characters filtering leads to (potentially remote) code execution

```
$ convert 'https://example.com"|ls "-la' out.png
total 32
drwxr-xr-x 6 user group 204 Apr 29 23:08 .
drwxr-xr-x+ 232 user group 7888 Apr 30 10:37 ..
```

CVE-2016-3718 - SSRF

```
ssrf.mvg
-=-=-=-=-=-=-=-
push graphic-context
viewbox 0 0 640 480
fill 'url(http://example.com/)'
pop graphic-context

$ convert ssrf.mvg out.png # makes http request to example.com
```

File content ~~validation~~ disarming

- **Defense**
  - If possible use library for validation
    - Test it: polyglotts
  - Allow list your file types, deny all others
    - Re-downloads as attachment
    - Embedded:
      - X-Content-Type-Options: nosniff,
      - e.g. CSP as XSS (further) mitigation
  - Resize image
    - Careful with tools like ImageMagick
  - Other tools: CDR = Content Disarm & Reconstruction

File content ~~validation~~ disarming

- **CDR tools**
  - Work also on other file types
  - DocBleach (Java) https://github.com/docbleach/DocBleach/wiki/
    - Office files, PDF , RTF
    - Removes Javascript, embedded files, macros
  - Several closed source software (MetaDefender/DeepCDR, resec, votiro, gatescanner,...)

File storage

- **Threat: exec uploaded pics**
  - `file.php.png`, `file.jar%09.png`, `backdoor.jsp%0a.png`
  - `phar_polyglot.jpg`
  - `reverseshell.asp`
- **Defense:**
  - Treat it like data, not code!
  - store files "outside" application
    - minimum: directory
    - better: separate host (HTTP Header: CSP , `Content-Type-Options: nosniff`)

File names

- **Payload all the file names!**
- **Hackers best payload friends:**
  - Server:
    - XSS `file"><img onload=prompt(1)>.tiff`
    - file traversal `file.jsp/../../../web.xml%09.pdf`
    - Command injection `; sleep 43;.png`
    - SQLi `file%00'OR 1=1.png` `'sleep(42)-- .jpg`
  - Client:
    - Inject shell commands in scripts
    - Inject cmd line options into commands
      - Wild card expansion ;-)

Wild card expansion ;-)

```
/tmp/testdir find . -type f | xargs ls -l
-rw-r-----@ 1 dirkw  wheel  0 Aug 25 00:51 ./a/aa/dee
-rw-r-----@ 1 dirkw  wheel  0 Aug 25 00:43 ./a/file1
-rw-r-----@ 1 dirkw  wheel  0 Aug 25 00:43 ./a/file2
-rw-r-----@ 1 dirkw  wheel  0 Aug 25 00:43 ./file1
-rw-r-----@ 1 dirkw  wheel  0 Aug 25 00:43 ./file2
/tmp/testdir touch ./-lR
/tmp/testdir ls *
-rw-r-----@ 1 dirkw  wheel  0 Aug 25 00:43 file1
-rw-r-----@ 1 dirkw  wheel  0 Aug 25 00:43 file2

a:
total 0
drwxr-x---@ 3 dirkw  wheel  96 Aug 25 00:51 aa
-rw-r-----@ 1 dirkw  wheel   0 Aug 25 00:43 file1
-rw-r-----@ 1 dirkw  wheel   0 Aug 25 00:43 file2

a/aa:
total 0
-rw-r-----@ 1 dirkw  wheel  0 Aug 25 00:51 deep
/tmp/testdir
```

```
/tmp/testdir find . -type f  | sort
./-rf
./a/aa/bbb
./file1
./file2
/tmp/testdir rm *
zsh: sure you want to delete all 4 files in /tmp/testdir?
(waiting ten seconds)
 [yn]? y
/tmp/testdir find . -type f  | sort
./-rf
/tmp/testdir
```

File names

- **Defense**
  - Rename files
    - best before content validation
    - or at least immediately afterwards
  - If not possible
    - reject / sanitise

Tusen takk

Gen AI Picture

- mail@drwetter.eu
- dirk@owasp.org
- linkedin.com/in/drdirkwetter/

Slides : https://drwetter.eu/fileuploads.pdf