PA-193: SECURE CODING PRINCIPLES AND PRACTICES

TIFF IMAGE PARSER

ASHWIN A YAKKUNDI
RAJESH KUMAR
ANANYA CHATTERJEE

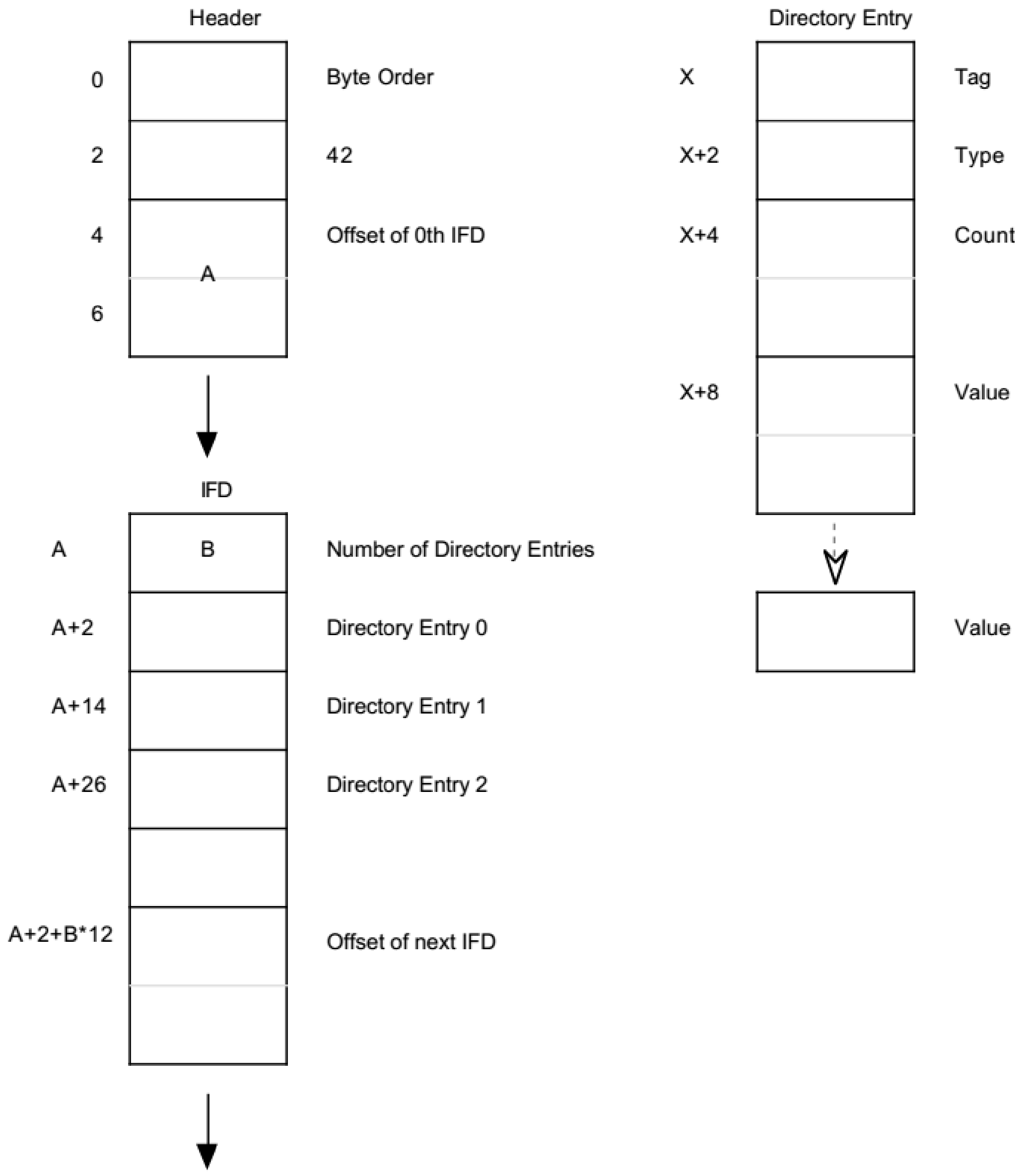
SCOPE

- To read TIFF 6.0 Image Header
- Parse the following:
  - Byte Order
  - TIFF Format ID
  - Image File Directory (IFD) Offset
  - Number of Directory Entries
  - Values in each Directory Entry

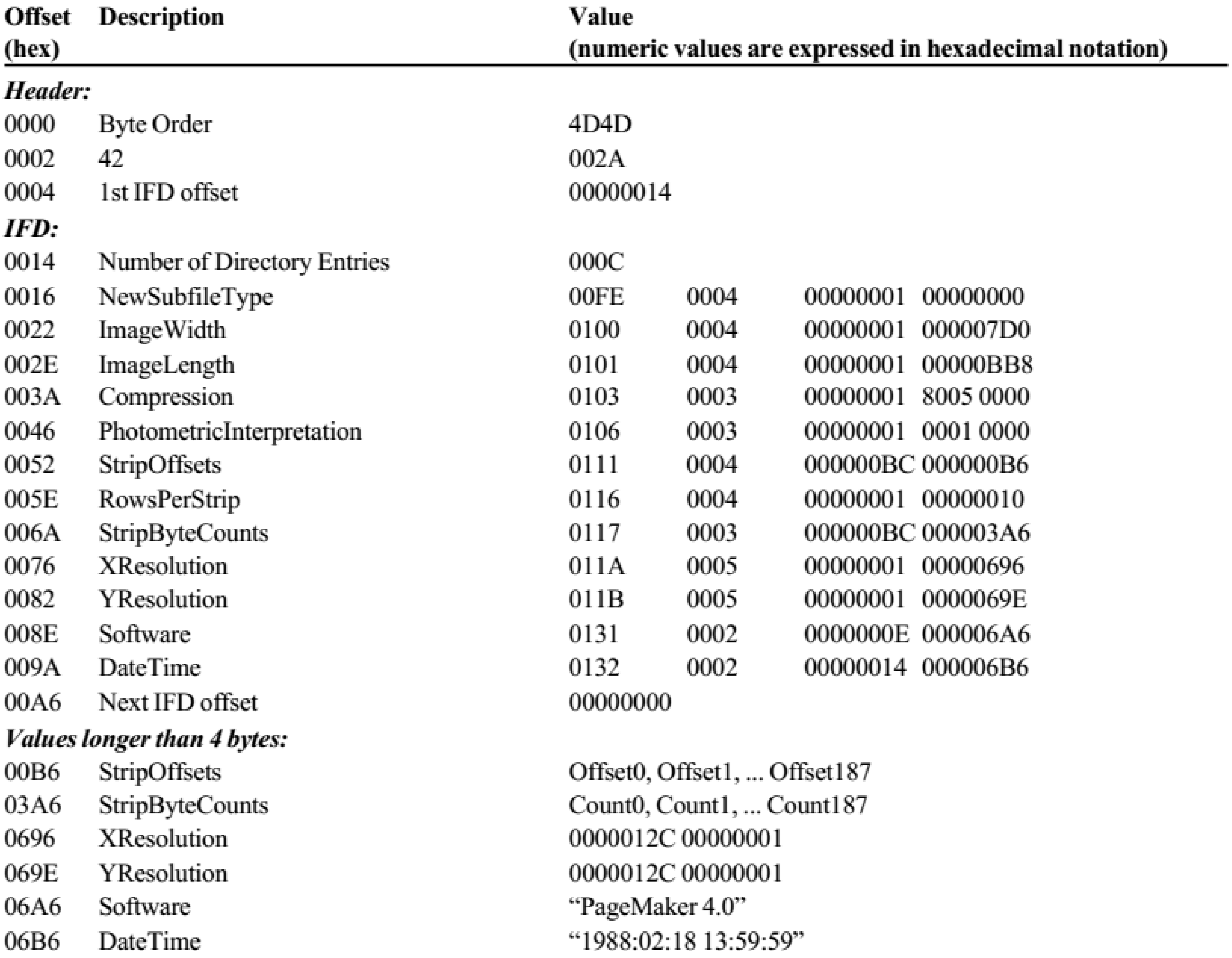
INTRODUCTION

- TIFF
  - Tagged Image File Format
  - is a file format for storing raster graphics images
    - Rectangular grid of pixels
  - popular among graphic artists, the publishing industry, and photographers
  - a *file* is defined to be a sequence of 8-bit bytes
    - bytes are numbered from 0 to N.
  - largest possible TIFF file is 2^32 bytes in length.

TIFF HEADER STRUCTURE



SAMPLE BILEVEL TIFF FILE



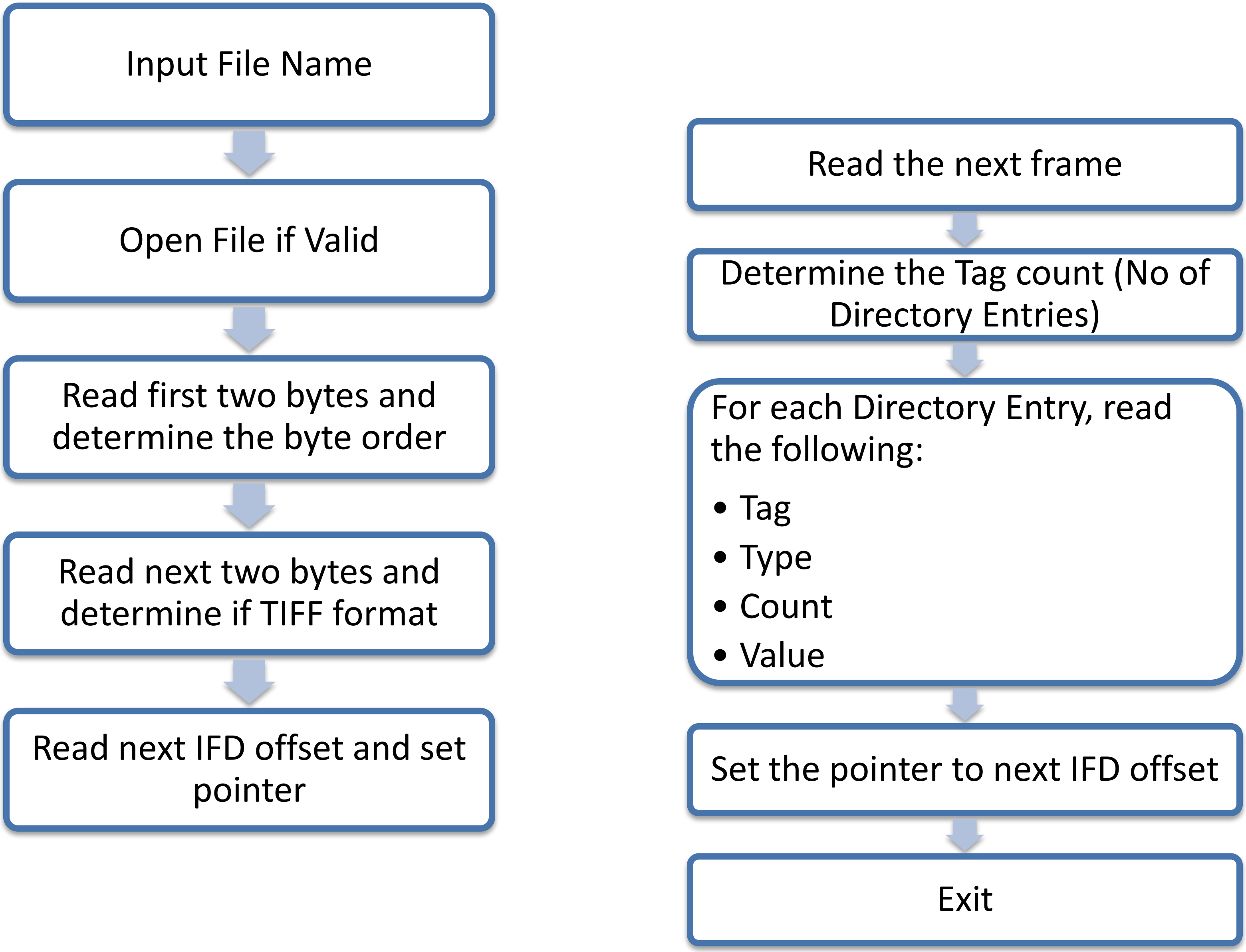
Offset (hex)	Description	Value (numeric values are expressed in hexadecimal notation)			
Header:					
0000	Byte Order	4D4D			
0002	42	002A			
0004	1st IFD offset	00000014			
IFD:					
0014	Number of Directory Entries	000C			
0016	NewSubfileType	00FE	0004	00000001	00000000
0022	ImageWidth	0100	0004	00000001	000007D0
002E	ImageLength	0101	0004	00000001	00000BB8
003A	Compression	0103	0003	00000001	8005 0000
0046	PhotometricInterpretation	0106	0003	00000001	0001 0000
0052	StripOffsets	0111	0004	000000BC	000000B6
005E	RowsPerStrip	0116	0004	00000001	00000010
006A	StripByteCounts	0117	0003	000000BC	000003A6
0076	XResolution	011A	0005	00000001	00000696
0082	YResolution	011B	0005	00000001	0000069E
008E	Software	0131	0002	0000000E	000006A6
009A	DateTime	0132	0002	00000014	000006B6
00A6	Next IFD offset	00000000			
Values longer than 4 bytes:					
00B6	StripOffsets	Offset0, Offset1, ... Offset187			
03A6	StripByteCounts	Count0, Count1, ... Count187			
0696	XResolution	0000012C 00000001			
069E	YResolution	0000012C 00000001			
06A6	Software	“PageMaker 4.0”			
06B6	DateTime	“1988:02:18 13:59:59”			

TIFF FRAME AND FILE STRUCTURE

```
struct TiffFrame {
	uint32_t width;
	uint32_t height;
	uint16_t compression;
	uint32_t rowsperstrip;
	uint32_t* stripoffsets;
	uint32_t* stripbytecounts;
	uint32_t stripcount;
	uint16_t samplesperpixel;
	uint16_t* bitspersample;
	uint16_t planarconfiguration;
	uint16_t sampleformat;
	uint32_t imagelength;
};
```

```
struct TiffFile {
	FILE* file;
	uint8_t systembyteorder;
	uint8_t filebyteorder;
	uint32_t firstrecord_offset;
	uint32_t nextifd_offset;
	uint64_t filesize;
	TiffFrame currentFrame;
};
```

STRUCTURE OF CODE



Input File Name

Open File if Valid

Read first two bytes and determine the byte order

Read next two bytes and determine if TIFF format

Read next IFD offset and set pointer

Read the next frame

Determine the Tag count (No of Directory Entries)

For each Directory Entry, read the following:

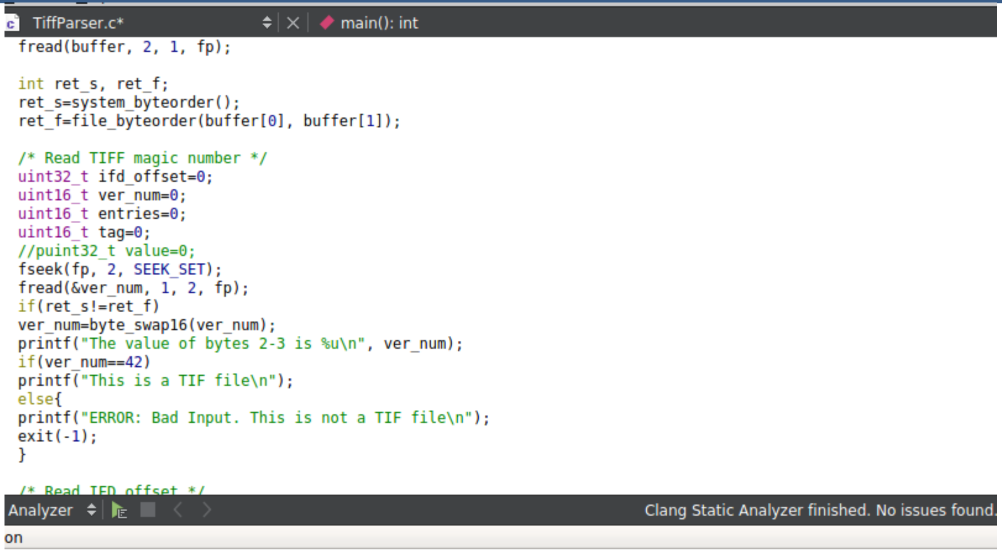
- Tag
- Type
- Count
- Value

Set the pointer to next IFD offset

Exit

TESTING OF CODE

- Use of Secure Lib Functions
- Input Validation
- Static Analysis
  - Clang
- Dynamic Analysis
  - Valgrind
- Testing with +ve and –ve inputs



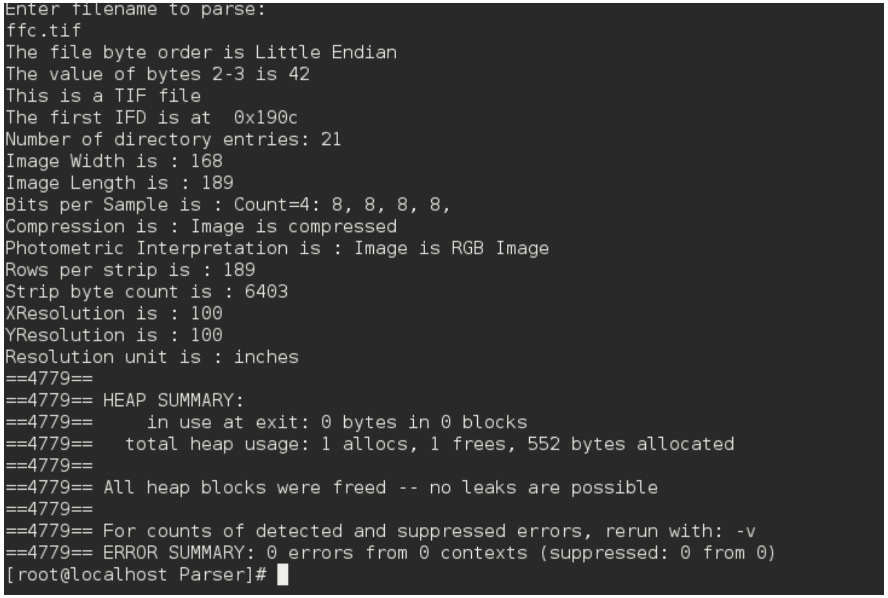
```c
fread(buffer, 2, 1, fp);

int ret_s, ret_f;
ret_s=system_byteorder();
ret_f=file_byteorder(buffer[0], buffer[1]);

/* Read TIFF magic number */
uint32_t ifd_offset=0;
uint16_t ver_num=0;
uint16_t entries=0;
uint16_t tag=0;
//puint32_t value=0;
fseek(fp, 2, SEEK_SET);
fread(&ver_num, 1, 2, fp);
if(ret_s!=ret_f)
ver_num=byte_swap16(ver_num);
printf("The value of bytes 2-3 is %u\n", ver_num);
if(ver_num==42)
printf("This is a TIF file\n");
else{
printf("ERROR: Bad Input. This is not a TIF file\n");
exit(-1);
}

/* Read IFD offset */
```

Clang Static Analyzer finished. No issues found.



```
Enter filename to parse:
ffc.tif
The file byte order is Little Endian
The value of bytes 2-3 is 42
This is a TIF file
The first IFD is at  0x190c
Number of directory entries: 21
Image Width is : 168
Image Length is : 189
Bits per Sample is : Count=4: 8, 8, 8, 8,
Compression is : Image is compressed
Photometric Interpretation is : Image is RGB Image
Rows per strip is : 189
Strip byte count is : 6403
XResolution is : 100
YResolution is : 100
Resolution unit is : inches
==4779==
==4779== HEAP SUMMARY:
==4779==     in use at exit: 0 bytes in 0 blocks
==4779==   total heap usage: 1 allocs, 1 frees, 552 bytes allocated
==4779==
==4779== All heap blocks were freed -- no leaks are possible
==4779==
==4779== For counts of detected and suppressed errors, rerun with: -v
==4779== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
[root@localhost Parser]#
```

DISTRIBUTION OF WORK

Rajesh Kumar Ashwin A Yakkundi	Conceptualization of flow. Setting the structure of the coding.
Ashwin A Yakkundi Ananya Chatterjee	Understanding the format. Implementation of code.
Rajesh Kumar Ananya Chatterjee	Code Verification and Testing.
Ananya Chatterjee Ashwin A Yakkundi	Sample Data Collection.
Rajesh Kumar Ananya Chatterjee	Setting up of Input Validation.
Ashwin A Yakkundi Rajesh Kumar	Preparation of presentation