AFEIC: Advanced Forensic Ext4 Inode Carving

Andreas Dewald [1] Sabine Seufert [2]

[1]ERNW Research GmbH

[2]Basys GmbH

Date: 23.03.2017

In cooperation with Friedrich-Alexander University Erlangen (FAU)

Outline

Introduction

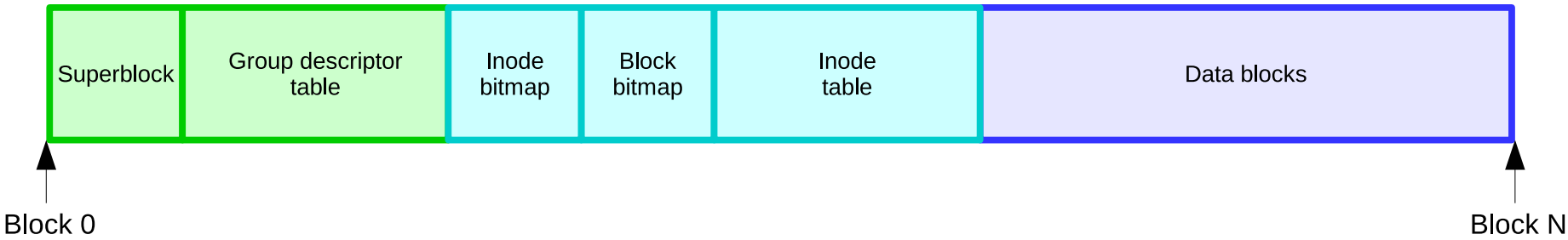
Motivation

- Postmortem file system forensics often requires reconstruction of lost or deleted files
- File carving on Ext4 has its limits
- Metadata analysis might not be possible if essential metadata is overwritten or modified

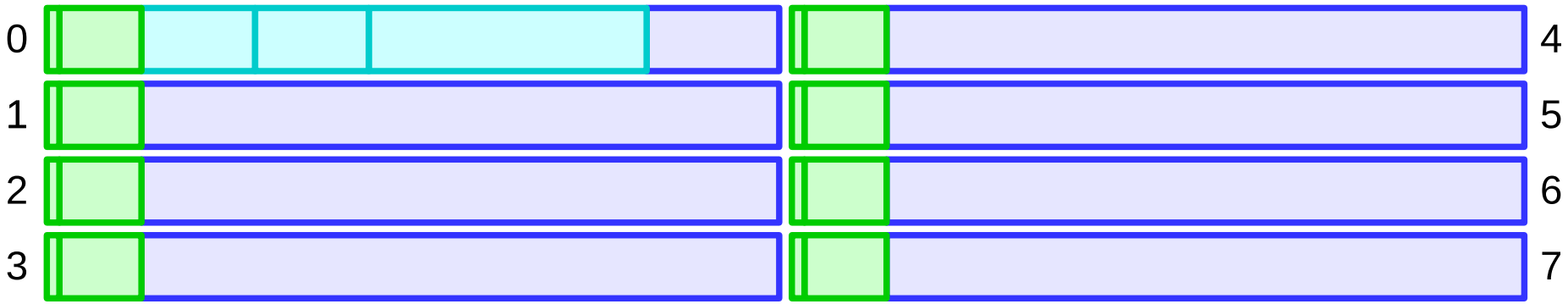
$\Rightarrow$ Attempt combination of carving and metadata analysis

- Use of search patterns to carve inode structures
- Interpret carved metadata for file recovery

- Ext4 file system stores metadata in the superblock and group descriptor table
- Layout is based on sequential blocks of 1024, 2048 or 4096 bytes
- Blocks are numbered and grouped into block groups



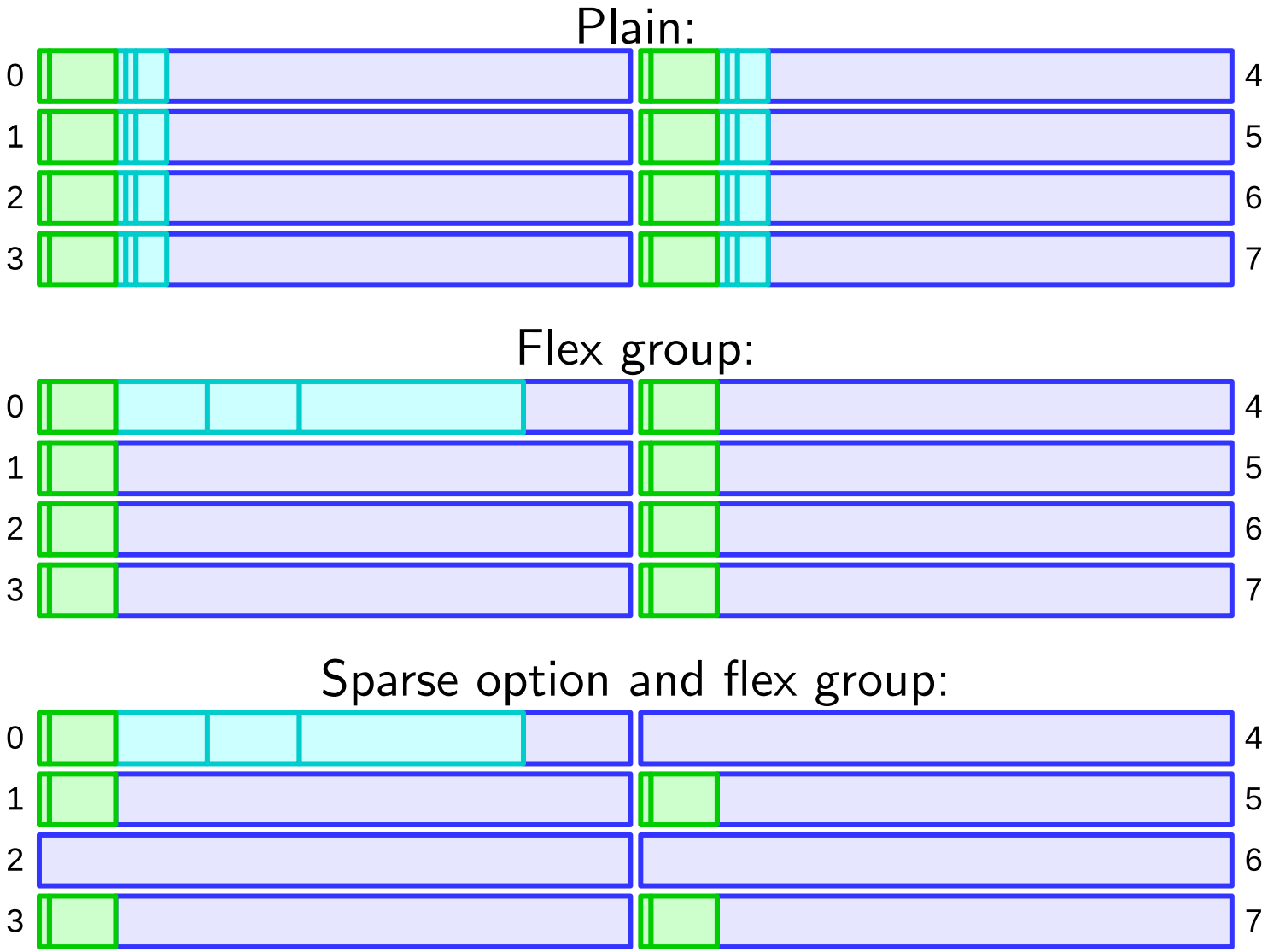
- Flex groups: Inode tables, block bitmaps and inode bitmaps of multiple block groups are merged into one block group



- Sparse option
  - Conceptually, superblock and group descriptor table are copied in every block group
  - With the sparse option enabled, block groups contain these copies only if their number ...
    - ... is 0 or 1
    - ... is a power of 3 $\Rightarrow$ 3, 9, 27, ...
    - ... is a power of 5 $\Rightarrow$ 5, 25, 125, ...
    - ... is a power of 7 $\Rightarrow$ 7, 49, 343, ...

Introduction

Ext4 file system - Sparse option



- An inode table spans a certain number of inodes representing files
- Inodes contain information essential for file recovery
  - File type
  - Time stamps
  - File size
  - Extents
  - etc.
- Extents reference a file's content data blocks
  - In contrast, the predecessor Ext3 uses indirect block pointers

Inode Recovery

Outline

- Since inodes do not have magic bytes, a combination of heuristic search patterns is used to find inode structures
- Found inodes can be interpreted:
  - Inodes of regular files are used to recover file content
  - Inodes of directories are used to recover file names (optional)
- Inode number: Index of inode within inode table
  - Can be determined from physical address
  - Is needed to lookup the file name

Inode Recovery

Phases

- Recovery process is performed in five phases:
  1. **Initialization:**
     Gather all required/available Ext4 parameters
  2. **Inode carving:**
     Find potential inodes using search patterns
  3. **Directory tree** (optional)
     Inodes representing directories are recovered and interpreted to acquire the directory tree
  4. **Regular files:**
     Inodes representing regular files are recovered
  5. **Files without content:** (optional)
     Remaining directory tree is built and files are created without file content

Inode Recovery

Inode attributes

Offset	Length	Description
0	2	mode (file type and access)
2	2	lower 16 bit user-ID
4	4	lower 32 bit file size
8	4	atime
12	4	ctime
16	4	mtime
20	4	dtime
24	2	lower 16 bit group ID
26	2	link count
28	4	sector count
32	4	flags
36	4	unused
40	60	block pointers or extent data structure
		...
132	16	additional space for time stamps
		...

Inode Recovery

Phases

- Recovery process is performed in five phases:
  1. **Initialization:**
     Gather all required/available Ext4 parameters
  2. **Inode carving:**
     Find potential inodes using search patterns
  3. **Directory tree** (optional)
     Inodes representing directories are recovered and interpreted to acquire the directory tree
  4. **Regular files:**
     Inodes representing regular files are recovered
  5. **Files without content:** (optional)
     Remaining directory tree is built and files are created without file content

Inode Recovery

Modes

- Content data mode:
  - Regular files are recovered and named after their inodes' physical address
  - Inodes from data blocks are recovered as well (e.g. Ext4 journal)
  - Necessary Ext4 parameters: block size
- Metadata mode:
  - Inode numbers are calculated to use the directory tree
    physical address $\mapsto$ inode number
  - Directory tree structure is built from inodes of directories
    inode number $\mapsto$ file name
  - Necessary Ext4 parameters: block and inode size, inode ratio, flex group size, sparse and 64 bit option

Inode Recovery

Calculation of inode number

$$f(a) = \left(\frac{a - (bg_a + o_s + o_i + o_r)}{i} + n_i + 1 \right)$$

a = physical address [*Byte*]

bg_a = offset to block group for address a [*Byte*]

i = inode size [*Byte*]

n_i = number of inodes per block group

o_s = size of superblock and GDT [*Byte*]

o_i = size of bitmaps [*Byte*]

$$o_r = \begin{cases} 1024\ Byte & \text{if block size} = 1024\ Byte \\ 0 & \text{else} \end{cases}$$

Inode Recovery

Proof of concept implementation

- Usable as a module for the Sleuthkit Framework Version 4.1.3
- Inode carving patterns and Ext4 parameters can be configured
- Inodes found in the Ext4 journal can be recovered without interpreting it
- Duplicate files are ignored (Content data mode)
- Full directory structure can be recovered independently of contents

Evaluation

Correctness

Image name and mode	all inodes	regular files	folders
small.img	**128**	**121**	**7**
metadata mode		121	7
content data mode		124	0
small_newFiles.img	**349**	**340**	**9**
metadata mode		340	9
content data mode		355	0
small_fastExt4.img	**128**	**121**	**7**
metadata mode		118	5
content data mode		119	0
small_fastdiffExt4.img	**128**	**121**	**7**
metadata mode		0	0
content data mode		125	0
small_fastNTFS.img	**128**	**121**	**7**
metadata mode		121	7
content data mode		124	0

Evaluation

Selectivity

- Testcase: Ubuntu 12.04 (16 GB)
- Selectivity of search patterns:

pattern	hits	t-miss	a-miss	select.
access rights	151k	1.02M	99.9M	0.6%
time interval	72.3k	207k	238k	0.003%
time consist.	209k	5.71M	1.42G	8.89%
link count	201k	29.8M	7.59G	47.6%
extent flag	166k	13.4M	3.28G	20.6%
extent header	166k	515k	53.2k	0.004%
file type	151k	4.51M	905M	5.7%

Conclusion

- Inodes can be found using search patterns without using the superblock
- With some parameters of the Ext4 file system, additional metadata or files from the journal can be recovered
- Reformatted Ext4 file systems contain recoverable data
- Combination of file carving and metadata analysis is suitable on Ext4

End

Thank you for your attention.
Questions?

Slides and open-source tool:
https://www1.cs.fau.de/content/ext4-file-recovery