Scalpel: A Frugal, High Performance File Carver

Golden G. Richard III, Vassil Roussev

Presented By:

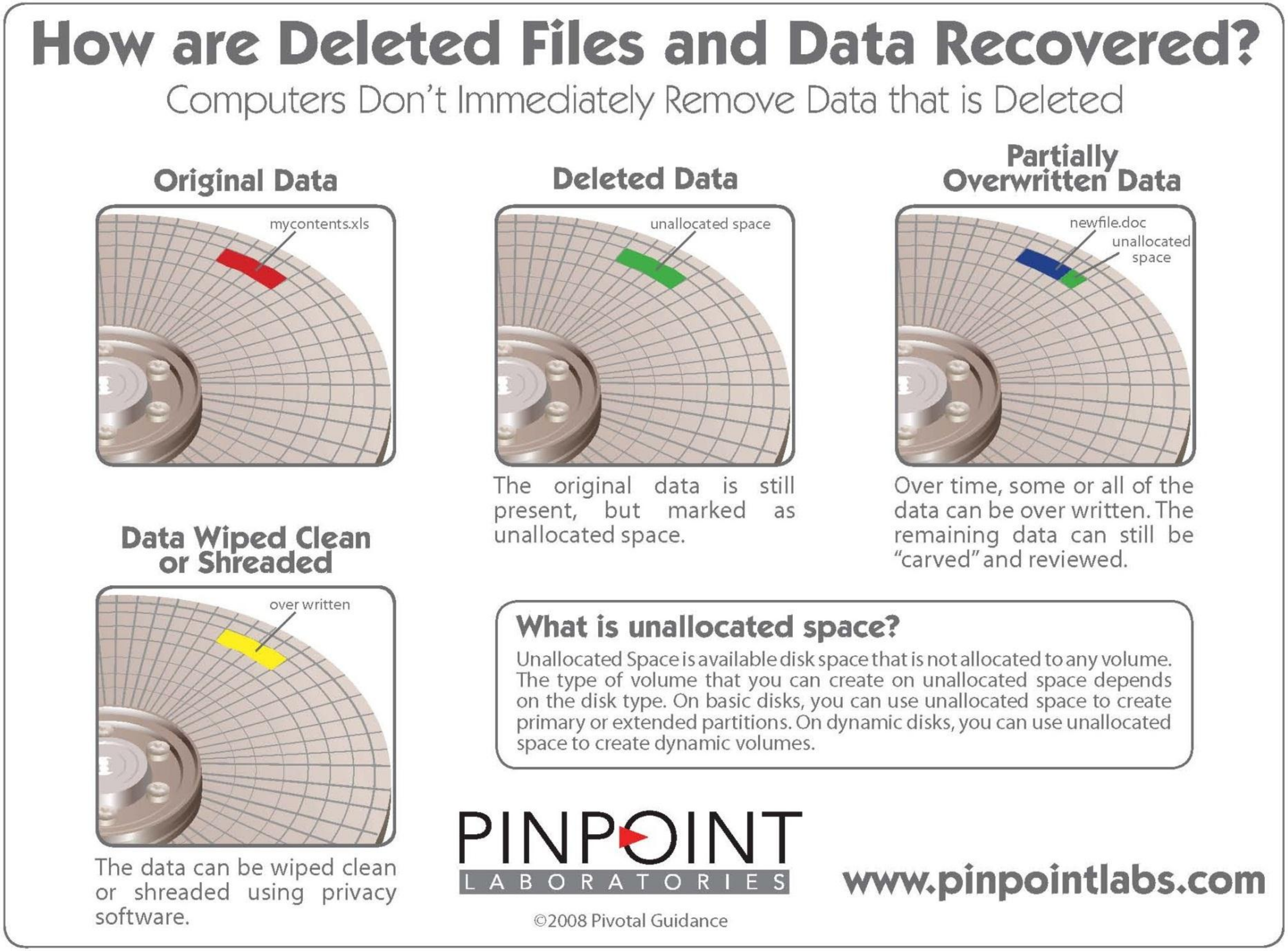
Muhammad Naseer Ali Bajwa

February 24, 2013

Curtain Raiser

1. Introduction
2. File Carving Strategies
3. Experimental Results
4. Conclusion
5. Future Work
6. Q/A

1. Introduction



How are Deleted Files and Data Recovered?

Computers Don't Immediately Remove Data that is Deleted

Original Data

Deleted Data

The original data is still present, but marked as unallocated space.

Partially Overwritten Data

Over time, some or all of the data can be over written. The remaining data can still be "carved" and reviewed.

Data Wiped Clean or Shreaded

The data can be wiped clean or shreaded using privacy software.

What is unallocated space?

Unallocated Space is available disk space that is not allocated to any volume. The type of volume that you can create on unallocated space depends on the disk type. On basic disks, you can use unallocated space to create primary or extended partitions. On dynamic disks, you can use unallocated space to create dynamic volumes.

PINPOINT
LABORATORIES

www.pinpointlabs.com

1. Introduction

(Cont.)

- **Header Footer Technique**
  - Strings of bytes at predictable offsets
  - Identify the beginning and ending of file of a certain type using a signature
    - 25 50 44 46 for PDF
    - 89 50 4E 47 0D 0A 1A 0A for MP3
  - Independent of file system
  - Works even if file metadata is destroyed

1. Introduction

(Cont.)

R

I

A

N

M

G

F

A T

E T

O

1. Introduction

(Cont.)

1. Introduction

(Cont.)

- **FRAGMENTATION**
  - Modern File Systems (NTFS, ext2/3) perform disc allocation that minimizes fragmentation
  - However, digitally important files (emails, jpeg, MS Word) have higher fragmentation

Outlook	58%
JPEG	17%
MS Word	16%

1. Introduction

(Cont.)

- **CONTRIBUTION**
  - Frugality

  - High Performance

  - Support for Distributed Implementations

2. File Carving Strategies

- **Guideline Principles**
  - Minimum time for searching headers and footers
  - Minimum Memory-to Memory copies
  - Minimize number of files to be carved

2. File Carving Strategies

- **SCALPEL INTERNALS**
  - Reads a configuration file defining file type to be carved
  - Configuration file also tells about specifications of headers and footers and the maximum file size for the file type

2. File Carving Strategies

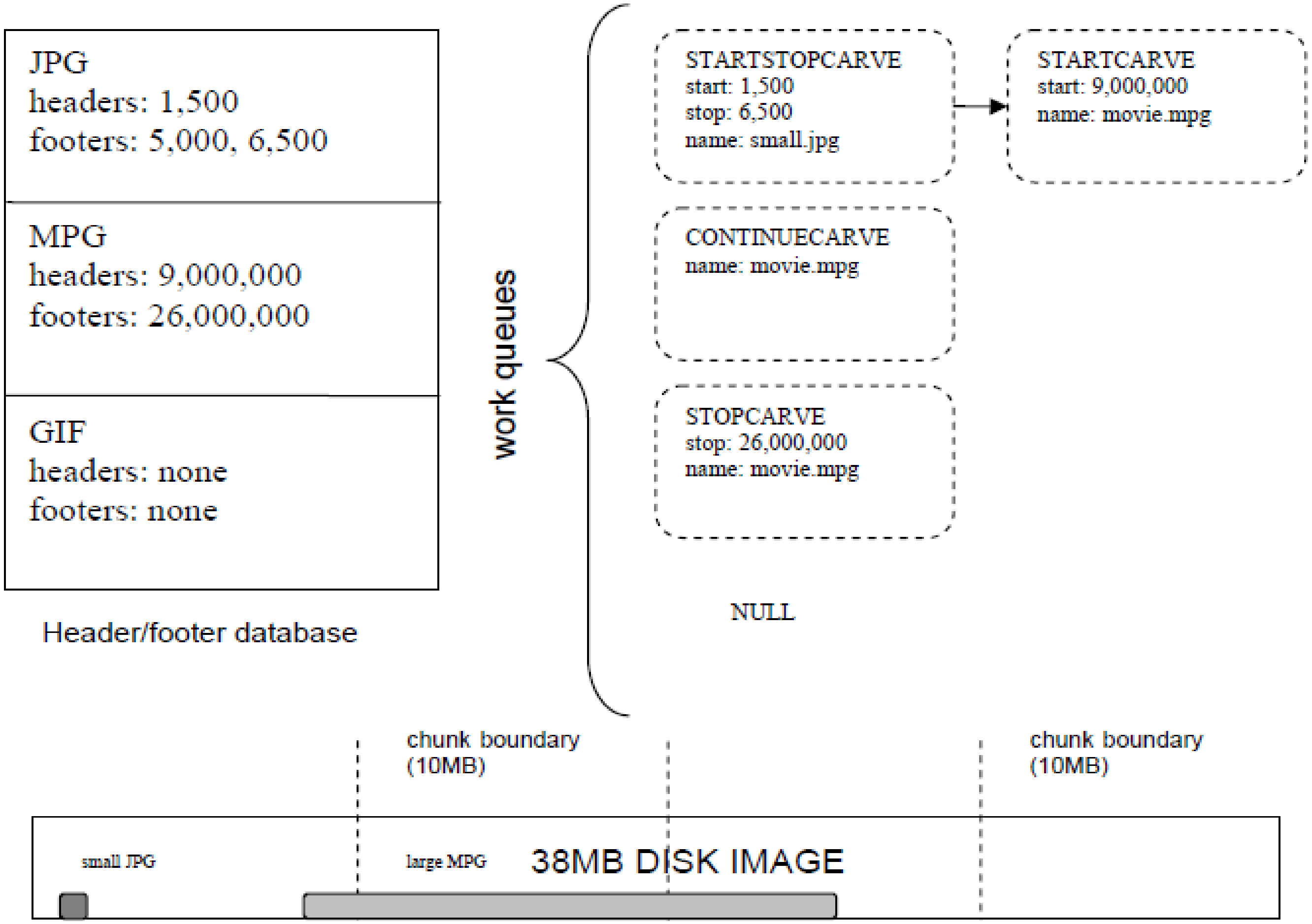
- **SCALPEL INTERNALS**
  - First Pass:
    - Reads entire disc image in chunks to search for file headers and maintains a database
    - Searches for footers, if footer is defined, that potentially match any header
      - Potentially matching header in the current chunk
      - Potentially matching header in previous chunk but close enough to the current position to meet maximum carve size requirements

2. File Carving Strategies

- **Scalpel Internals**
  - Interim Processing:
    - Populate a set of work queues
    - Each queue contains one of these records type
      - StartCarve
      - StartStopCarve
      - ContinueCarve
      - SropCarve

2. File Carving Strategies

- **Scalpel Internals**



JPG
headers: 1,500
footers: 5,000, 6,500

MPG
headers: 9,000,000
footers: 26,000,000

GIF
headers: none
footers: none

Header/footer database

work queues

STARTSTOPCARVE
start: 1,500
stop: 6,500
name: small.jpg

STARTCARVE
start: 9,000,000
name: movie.mpg

CONTINUECARVE
name: movie.mpg

STOPCARVE
stop: 26,000,000
name: movie.mpg

NULL

chunk boundary
(10MB)

chunk boundary
(10MB)

small JPG

large MPG

38MB DISK IMAGE

2. File Carving Strategies

- **Scalpel Internals**
  - Second Pass:
    - Processes the entire image again in chunks
    - Write the carved data to files directly from the buffer that holds disc image

2. File Carving Strategies

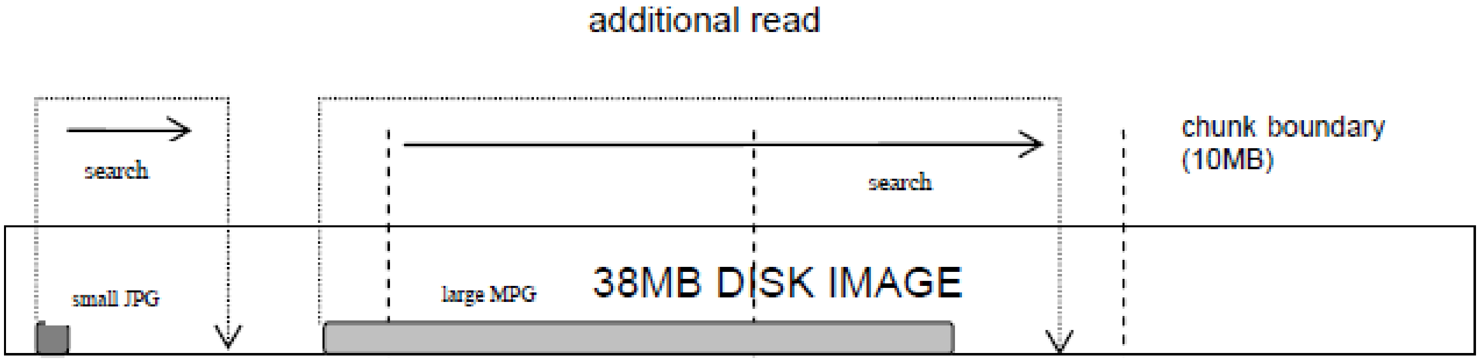
- **Analysis of Scalpel**
  - No extraneous memory-to-memory copies
  - Use of *seek* operation to skip consecutive chunks
  - Lower bound on number of bytes read is T_{read} (No header found)
  - Worst case performance in $2 \times T_{read}$
  - Minimize T_{write} by never writing carved file unless associated data meet all the requirements imposed by configuration file

2. File Carving Strategies

- **FOREMOST 0.69**
  - Performs all carving operations in *single* pass
  - Find a header in a chunk
  - If *enough* data is available in the chunk, carve it
  - If not, build an in-memory buffer and keep reading
    - If file has a footer, carve in between
    - Otherwise, write everything upto limit
  - Start over

2. File Carving Strategies

- **Foremost 0.69**



2. File Carving Strategies

- **Analysis of Foremost 0.69**
  - Reads $k \times T_{read}$ to process a disc image
  - May perform additional reads (How Many?)
  - Experiments show $1 < k < 45$
  - Requires substantial memory to build buffers
  - Extraneous memory-to-memory writes

2. File Carving Strategies

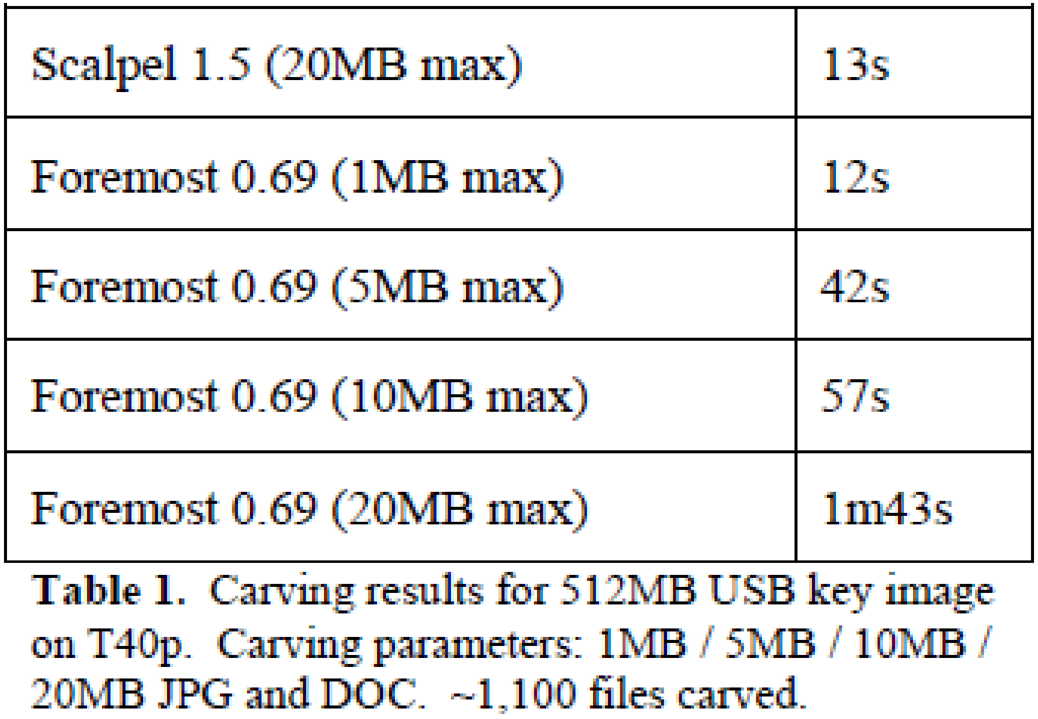
- **Difference between Scalpel and Foremost**
  - Foremost always carves, Scalpel does not
    - Use –b switch to emulate Foremost
  - Foremost misses overlapping headers, Scalpel does not
    - Use –r switch to emulate Foremost

3. Experimental Results

- **EXPERIMENTAL SETUP**
  - Used same configuration files for both Scalpel and Foremost
  - Two machines used
    - *350MHz Pentium 2 with 512MB of RAM and no swap space. 4 port ATA-133 IDE controller, 7200rpm 80GB drive for holding carve results. Operating System: Knoppix 3.7.*
    - *Thinkpad T40p, 1.7GHz Pentium M, with 2GBof RAM and 4GB of swap space. 7200rpm 60GB drive. Operating System: RedHat 9 with upgraded 2.40.20 kernel.*
  - Both tools carved exactly the same files

3. Experimental Results

Linux

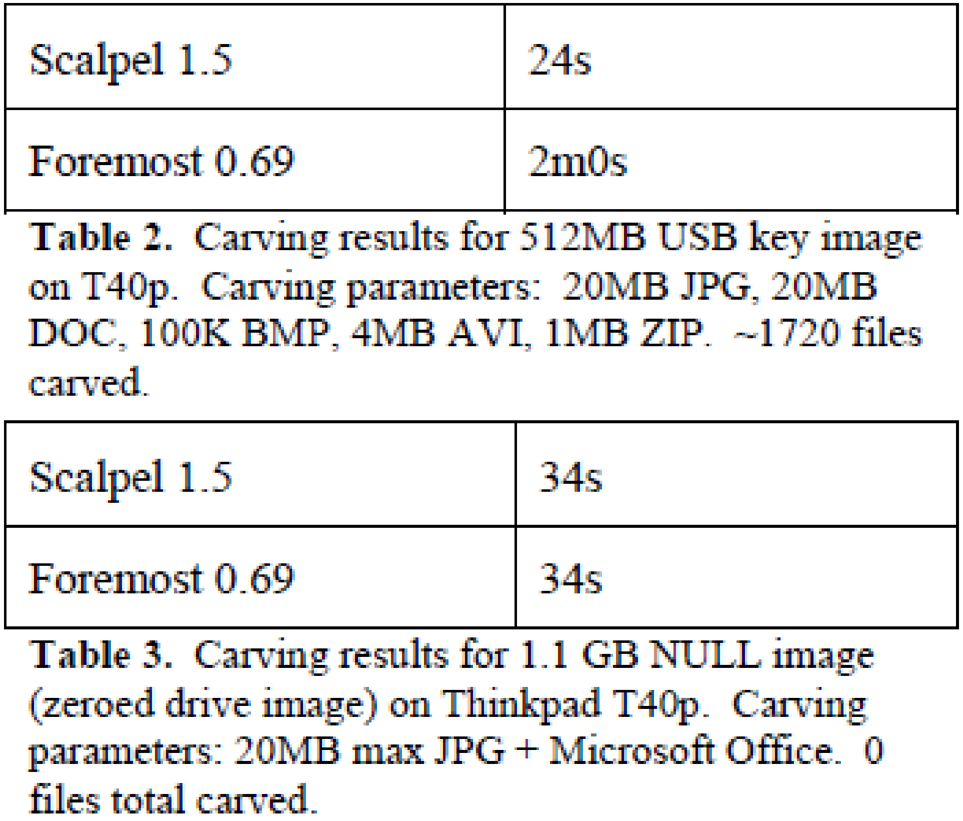


Scalpel 1.5 (20MB max)	13s
Foremost 0.69 (1MB max)	12s
Foremost 0.69 (5MB max)	42s
Foremost 0.69 (10MB max)	57s
Foremost 0.69 (20MB max)	1m43s

Table 1. Carving results for 512MB USB key image on T40p. Carving parameters: 1MB / 5MB / 10MB / 20MB JPG and DOC. ~1,100 files carved.

3. Experimental Results

Linux



Scalpel 1.5	24s
Foremost 0.69	2m0s

Table 2. Carving results for 512MB USB key image on T40p. Carving parameters: 20MB JPG, 20MB DOC, 100K BMP, 4MB AVI, 1MB ZIP. ~1720 files carved.

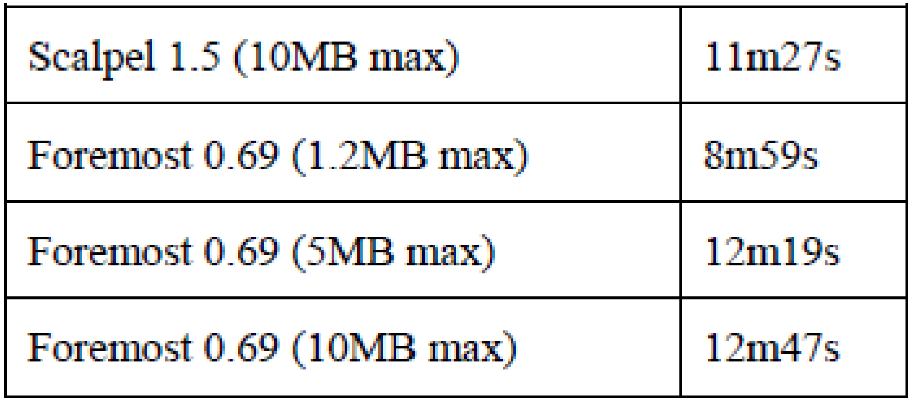
Scalpel 1.5	34s
Foremost 0.69	34s

Table 3. Carving results for 1.1 GB NULL image (zeroed drive image) on Thinkpad T40p. Carving parameters: 20MB max JPG + Microsoft Office. 0 files total carved.

3. Experimental Results

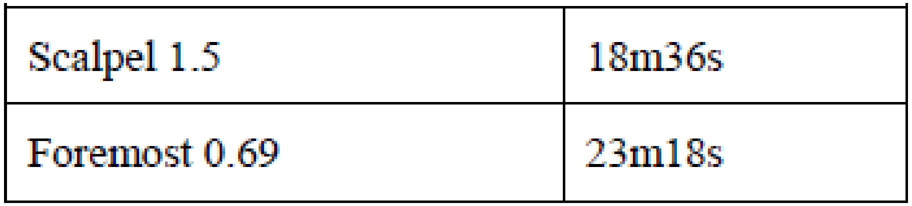
Linux

- Foremost requested 263 MB, 4.9GB and 21 GB space for additional reads for 1.2 MB, 5 MB and 10 MB carve sizes respectively.
- Foremost requested 48 GB additional space



Scalpel 1.5 (10MB max)	11m27s
Foremost 0.69 (1.2MB max)	8m59s
Foremost 0.69 (5MB max)	12m19s
Foremost 0.69 (10MB max)	12m47s

Table 4. Carving results for 1.2 GB FAT32 (from e-bay) on P2-350. Carving parameters: 1.2/5/10MB JPG. ~2,200 files carved.



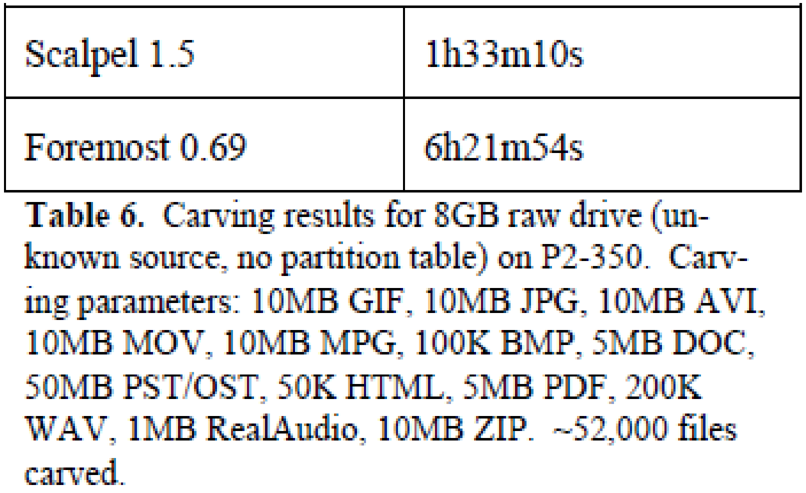
Scalpel 1.5	18m36s
Foremost 0.69	23m18s

Table 5. Carving results for 1.2 GB FAT32 (from e-bay) on P2-350. Carving parameters: 10MB GIF, 10MB JPG, 10MB AVI, 10MB MPG, 10MB DOC, 50K HTML, ~5,000 files carved.

3. Experimental Results

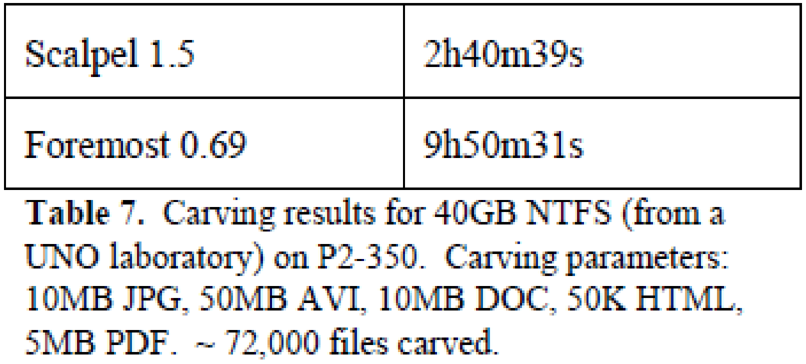
Linux

- Foremost 0.69 performs 238,270,750,000 bytes of reads in addition to its single pass over the 8GB image.
- Foremost performs 117,622,357,936 bytes of additional reads in addition to a single pass over the 40GB image.
- As the number of types and maximum sizes for carved files increases, the performance of Fore-most falls farther behind that of Scalpel.



Scalpel 1.5	1h33m10s
Foremost 0.69	6h21m54s

Table 6. Carving results for 8GB raw drive (unknown source, no partition table) on P2-350. Carving parameters: 10MB GIF, 10MB JPG, 10MB AVI, 10MB MOV, 10MB MPG, 100K BMP, 5MB DOC, 50MB PST/OST, 50K HTML, 5MB PDF, 200K WAV, 1MB RealAudio, 10MB ZIP. ~52,000 files carved.



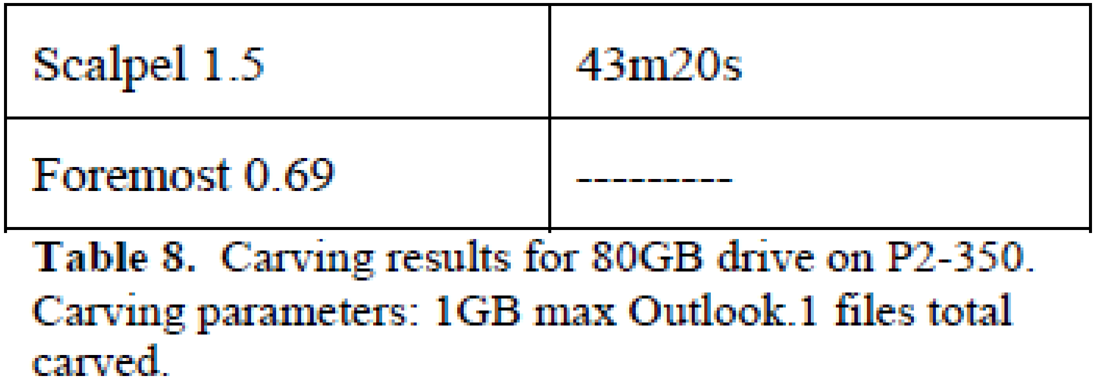
Scalpel 1.5	2h40m39s
Foremost 0.69	9h50m31s

Table 7. Carving results for 40GB NTFS (from a UNO laboratory) on P2-350. Carving parameters: 10MB JPG, 50MB AVI, 10MB DOC, 50K HTML, 5MB PDF. ~ 72,000 files carved.

3. Experimental Results

Linux

- Al last Foremost Crashed!

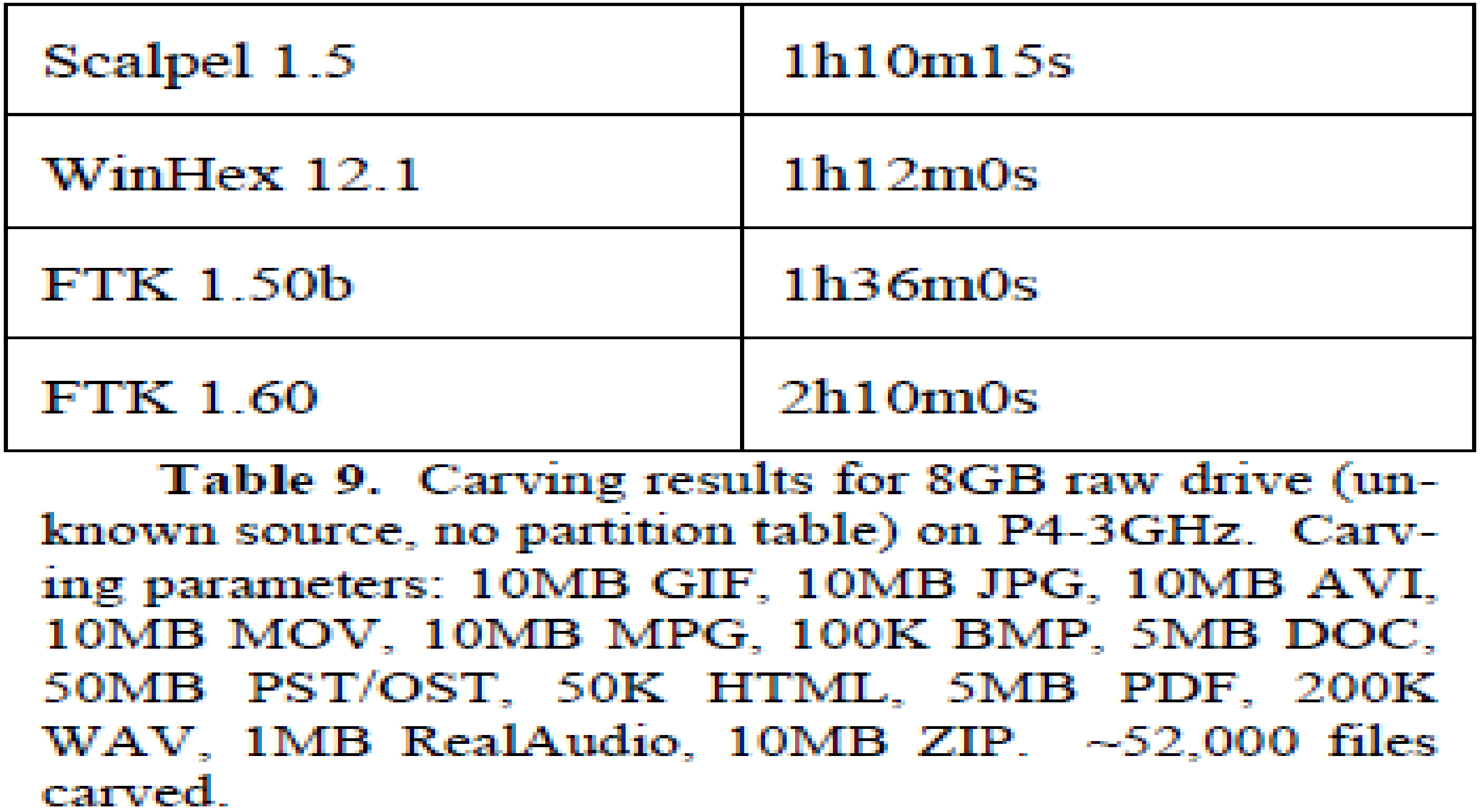


Scalpel 1.5	43m20s
Foremost 0.69	---------

Table 8. Carving results for 80GB drive on P2-350. Carving parameters: 1GB max Outlook.1 files total carved.

3. Experimental Results

Windows XP



Scalpel 1.5	1h10m15s
WinHex 12.1	1h12m0s
FTK 1.50b	1h36m0s
FTK 1.60	2h10m0s

Table 9. Carving results for 8GB raw drive (unknown source, no partition table) on P4-3GHz. Carving parameters: 10MB GIF, 10MB JPG, 10MB AVI, 10MB MOV, 10MB MPG, 100K BMP, 5MB DOC, 50MB PST/OST, 50K HTML, 5MB PDF, 200K WAV, 1MB RealAudio, 10MB ZIP. ~52,000 files carved.

3. Experimental Results

Windows XP

Tool	Platform	Number of Carved GIF Files	Number of Corrupt Files
Scalpel	Windows/Linux	4817	~ 400
Foremost 0.69	Linux	4817	~400
WinHex 12.1	Windows	4817	~400
FTK 1.50b	Windows	3463	2442
FTK 1.60	Windows	4194	~100

4. Scalpel Performance Summary

- Carves exactly the same set of files, for given configurations, on both Linux and Windows
- Performance difference between P2 and P4 machines is insignificant
- It is not optimized for Windows yet

5. Conclusion

- The tool presented in this paper is able to carve files
  - Quickly
  - Accurately
  - Frugally
- It
  - Is open source
  - Avoids unnecessary memory-to-memory copies
  - Performs exactly two sequential passes over a disc image to perform carving operation

5. Future Work

- More accurate header analysis
- Incorporating the tool into framework of distributed digital forensics
- Optimization on Windows platform
- Compilation and Testing for other Linux flavors like Mac OS X

Any Question?