Model-Driven Engineering in Digital Forensics

Jeroen van den Bos
joint work with Tijs van der Storm
What is “digital forensics”? 

From Wikipedia: “Digital forensics is a branch of forensic science encompassing the recovery and investigation of material found in digital devices, often in relation to computer crime.”
Digital forensics in practice

As part of some criminal investigation, hardware is seized by police and suspects are arrested and held in custody.

The judge wants to know: where is the evidence of a crime?
Typical process

Acquisition
Securing the data

Recovery
Turning data into information

Analysis
Finding relevant information
Hard drive recovery

Data acquired: 1TB.
Hard drive recovery

264GB allocated to main file system.
Hard drive recovery

40GB found analyzing metadata, additional 240GB file system.
Hard drive recovery

That leaves 456GB unaccounted for.
Hard drive recovery

96GB probably contains files. What about the other 360GB?
File carving

- File carving is the process of recovering files without the help of (file system) storage metadata.
- A file carver deals with multiple concerns
  - Selecting and/or combining blocks of data from the input as candidate files.
  - Deciding whether a given block of data is of some interesting format.
  - Handle the scale required by terabytes of input data and hundreds of file formats and variants.
File carving

- File carving is the process of recovering files without the help of (file system) storage metadata.

- A file carver deals with multiple concerns
  - Selecting and/or combining blocks of data from the input as candidate files.
  - Deciding whether a given block of data is of some interesting format.
  - Handle the scale required by terabytes of input data and hundreds of file formats and variants.
File carving

Clusters acquired from a hard drive:
File formats and file carving

- Constant need for addition/modification
- New formats emerge continuously
- Lots of vendor-specific variants
- Tools must be high performance
- Typically hand-optimized
A DERRIC description

1. Header
   Name and encoding/type defaults
   
   format PNG

   strings ascii
   size 1
   unit byte
   sign false
   type integer
   order lsb0
   endian little

2. Sequence
   Data structure ordering
   
   sequence

3. Structures
   Layout of individual data structures
   
   structures

   IHDR {
       l: lengthOf(d)
       size 4;
       n: “IHDR”;
       d: { ... };
       c: checksum
       (...) size 4;
   }
structures

Chunk {
  length: lengthOf(chunkdata) size 4;
  chunktype: type string size 4;
  chunkdata: size length;
  crc: checksum(algorithm="crc32-ieee",
                 fields=chunktype+chunkdata) size 4;
}

IHDR = Chunk {
  chunktype: "IHDR";
  chunkdata: {
    width: !0 size 4;
    height: !0 size 4;
    bitdepth: 1|2|4|8|16;
    imagesize: (width*height*bitdepth)/8 size 4;
    colourtype: 0|2|3|4|6;
    interlace: 0|1;
  }
}
Applying Derric

- Each format has one/several descriptions
- Code generator transforms descriptions into validators (implemented in Rascal)
- File carver uses validators to locate files
File carving architecture
Conclusion

- Derric allows separation of concerns
- Experimental results show:
  - Performance (speed and accuracy) is excellent
  - Model-driven approach allows for increased scalability through additional transformations
Questions?