UNIVERSITY OF MANNHEIM

Forensig2
ForensicImageGeneratorGenerator

The Forensic Image Generator Generator

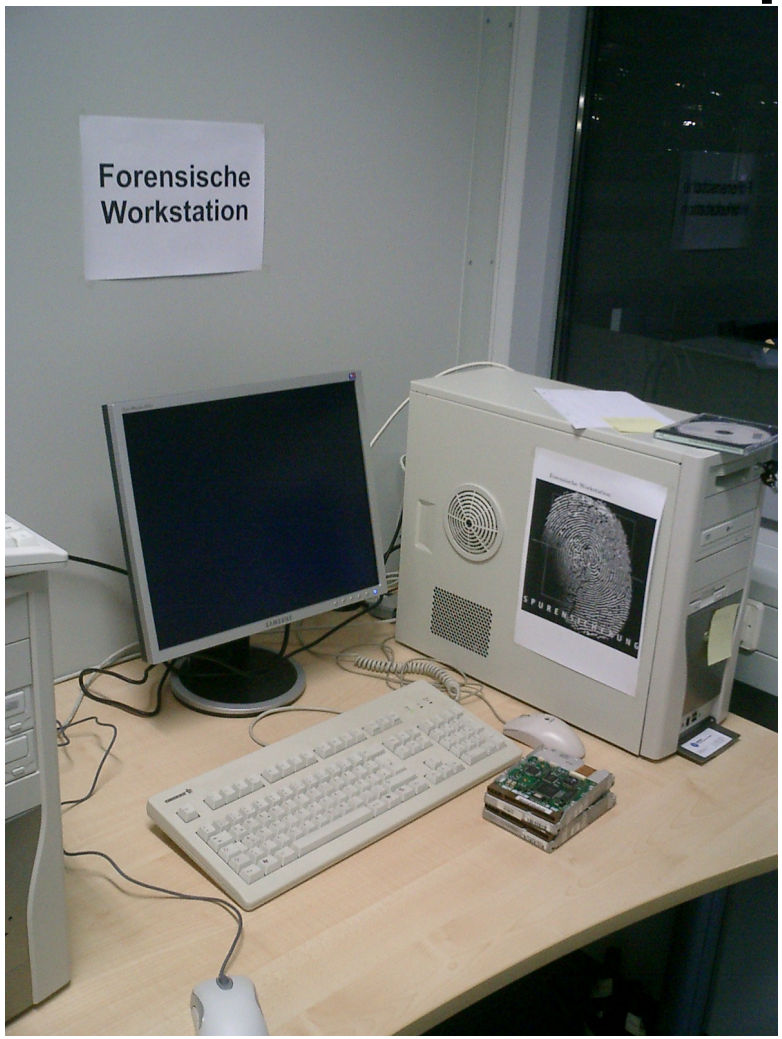
Felix C. Freiling
Christian Moch

Laboratory for Dependable Distributed Systems
University of Mannheim, Germany

UNIVERSITY OF MANNHEIM

Forensig2 ForensicImageGeneratorGenerator

Forensic Computing Course



- Research oriented graduate course to handling and analysis of digital evidence
- Exercises with “real” evidence
- Fokus on low level tools
- 20 – 30 students every year (summer term)

UNIVERSITY OF MANNHEIM

Forensig2
ForensicImageGeneratorGenerator

“Asservatenkammer”

“Safe” with digital evidence

Outline

- Background
  - Current educational methods
  - Motivation
- Structure of the generator
- Experiences
- Summary/Outlook

Current educational methods 1/2

- Construct cases by hand (manual approach)
  - \+ Full control
  - \+ No privacy problems
  - \+ Nearly no technical restrictions
  - \+ Knowledge of the ground truth
  - – Time consuming: Not practical for a large number of cases
- Honeypots
  - \+ Real cases
  - ± Amost no privacy restrictions
  - – No control
    - Mostly interesting cases, but no guarantee
  - – Results of interesting cases can be found online

Current educational methods 2/2

- Second hand harddisks
  - \+ Easy to acquire
  - \+ Analysis raise awareness about privacy of deleted data
  - – Not always interesting
  - – Not typical for real digital investigations
  - – Privacy concerns

This is how we did it up to now!

What we want

- Individual disk images
- Easy and fast to produce
- Knowledge of ground truth
- No privacy concerns

UNIVERSITY OF MANNHEIM

Forensig2 ForensicImageGeneratorGenerator

The Answer

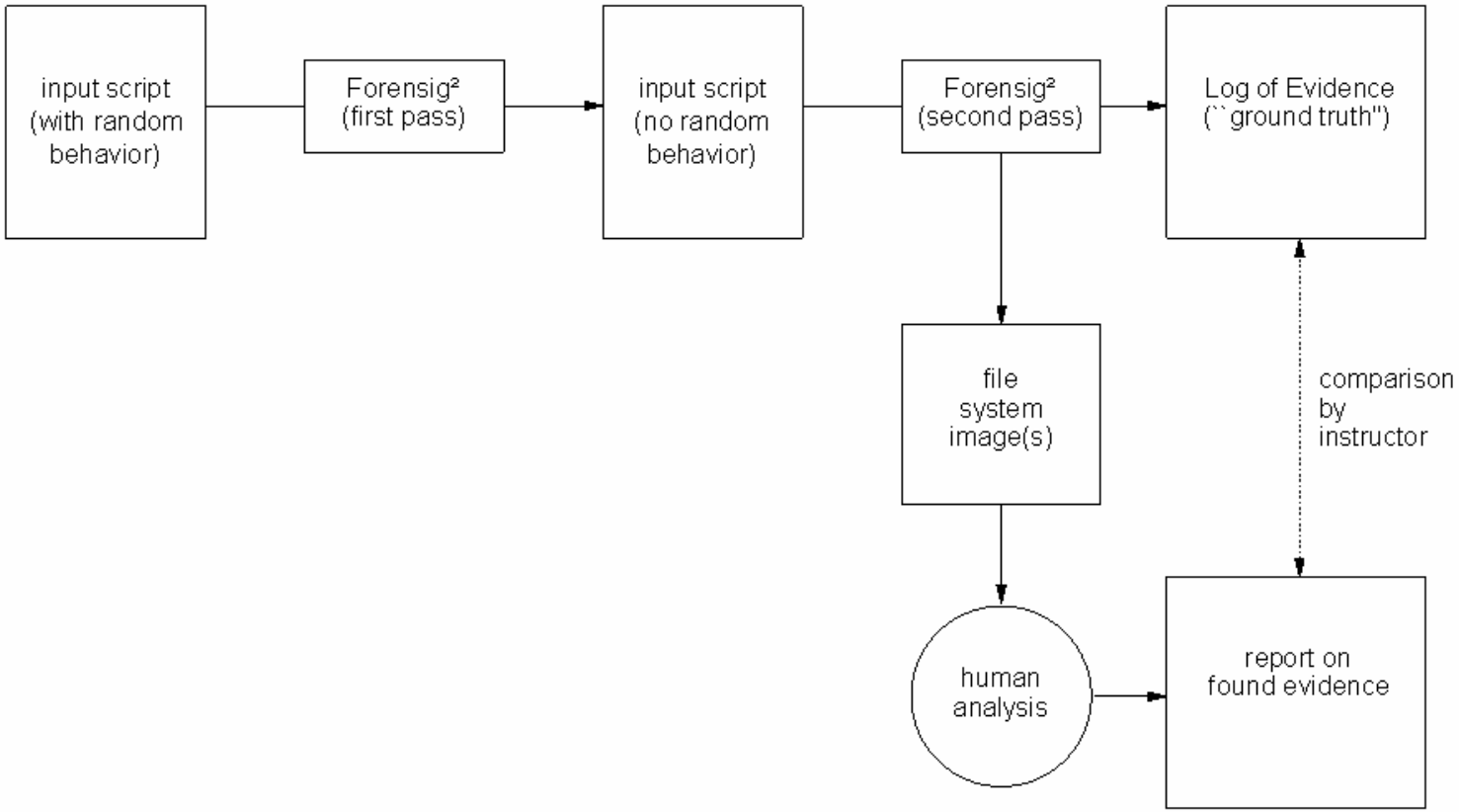
Forensig2

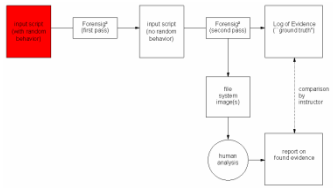
ForensicImageGeneratorGenerator

Outline

- Background
- Structure of the generator
  - Overview
  - Modules
  - Implementation
  - Expandability
- Experiences
- Summary/Outlook

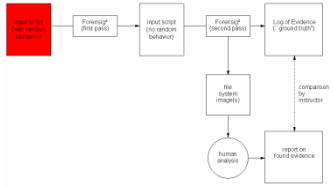
Structural overview





Input Script 1/2

- Defines the ground truth
  - Including timeline
- Possibility to use random elements
- Input script language is Python
  - Full programming language
  - Including classes, modules etc.
  - Plus some additional commands for the generation process



Input script 2/2

```
timeline.setTrueTime(673145110)
dd = DataDisk()
dd.setSize(100000)
dd.create()
dd.writeDOSPartition((
    (48000,83,True),
    (
          (24000,41,False),(24000,41,False)
    )
    ))
dd.mkfs('0.2.1','ext2')
dd.mkfs('0.2.2','minix')

extension = ['pdf', 'jpg', 'png']
for i in range(1,50):
    currentExtension = random.choice(extension)
    dd.copyFromHost('0.2.2',
    '/home/forensig/cases/copyrightviolation/pictures/pic%s.%s' %
          (i, currentExtension), '%s.%s' % (i, currentExtension))
    dd.rm('0.2.2', '%s.%s' % (i, currentExtension)
dd.deleteDOSPartitionSignature()
```

2.5.1991, 2:45

100.000 kB Disk

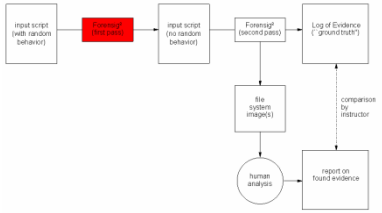
"partion table"

create file system

Python!

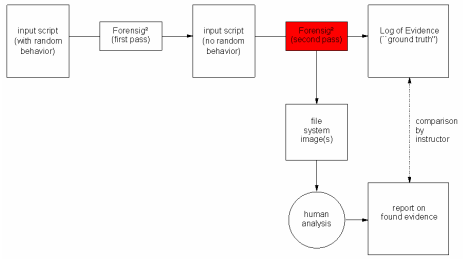
filename on host

filename on "evidence"



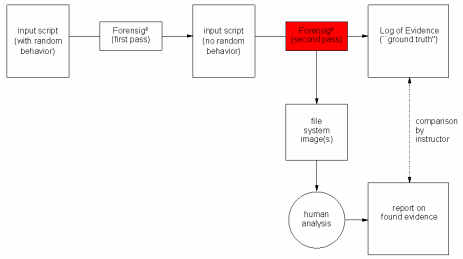
Generator first pass

- Translates the input script to another Python program
- Eliminates randomness
  - Fixed value of the random seed
  - Produces deterministic script (makes generation reproducible)
- Command: `./Generator.py inputscript.3fg`
  - Output: Python program `inputscript.3fgg`



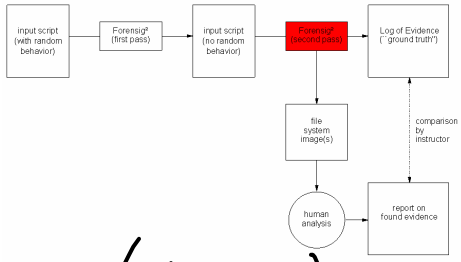
Generator second pass

- Generator of the disk images
  - Output of generator's first pass
  - No randomness
- Uses modules of Forensig²
  - Building the system
    - System
    - SystemDisk, DataDisk
    - CPU, Memory and other components
  - Time control
  - Executing operations
  - Documentation
  - Auxiliary modules
    - Qemu
    - Hexer
    - Partitioner



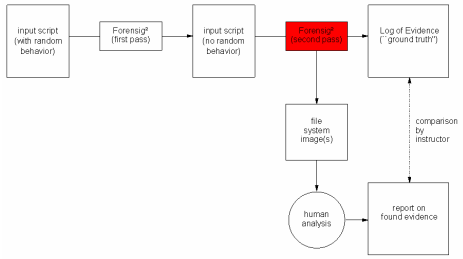
Design – Disk image

- RAW format
- Alternatives
  - VMWare (vmdk)
  - Qemu (qcow2)
  - Virtual PC (vhd)
- Advantage
  - RAW format is used by many tools
- Disadvantage
  - High disk space usage



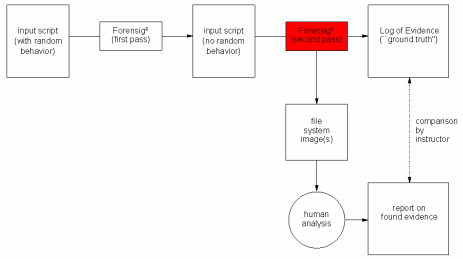
Design – User interaction

- User interaction without guest additions (VMware)
  - More realistic traces
  - Guest additions have to be installed
- All processes are invoked using **`ssh`**
  - Widespread use for remote administration
    - Realistic traces
  - Processes are invoked directly (without a shell)
    - No pipes
    - No **`.bash_history`**
    - Interactive shell can be initiated with **`bash -i`**
      - Pipes are possible
      - **`.bash_history`** is written



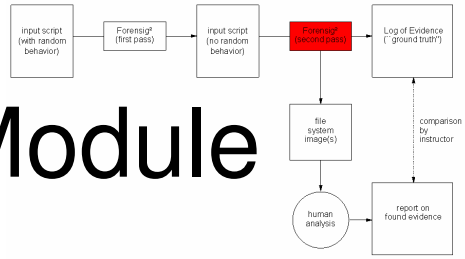
Design – OS installation

- Ubuntu-vm-builder
  - Can install different Ubuntu Linux Versions
  - Configuration of all important installation options
- Alternatives
  - Pre-configured environment
  - Prepared disk image



Extensibility 1/3

- Additional modules
  - Every module has to contain at least one class
  - Mandatory
    - Getname method,
    - including the config file
    - Some commands to import the logging facility



Extensibility 2/3: A Minimal Module

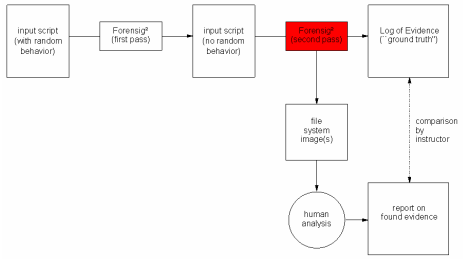
```
[…]
logging.config.fileConfig("config/logging.cfg")
cfg = Config(file('config/base.cfg'))
[...]
class Empty():
  def __init__(self):
    self.__name = uuid.uuid1()
    logging.info('New instance Empty. Instance name: %s' % self.getName())

  def getName(self):
    return '%s' % self.__name

  def newMethod(self):
    return 'Empty.py module'

  def report(self, xmlobject=None, parent=None):
    if not xmlobject:
       xmlobject = Reporter()
    e = xmlobject.newChild(parent, 'Empty', self.getName())

    if not parent:
       return xmlobject.writeXML('%sEmpty_%s_%s.xml' % (cfg.reporter.path,
                 self.getName(), time.time()))
```



Extensibility 3/3

- Usage

```
e = new Empty()
print  e.newMethod()
e.report()
```

- Output

```
Empty.py module
```

Outline

- Background
- Structure of the generator
- Experiences
- Summary/Outlook

Copyright Violation Script 1/2

1. Linux System with Gnome
2. Instructor uses Firefox with X11-Forwarding manually and downloads some MP3 files
3. The generator archives the files to a tar-archive
4. The tar archive is encrypted with gpg
5. The gpg file is copied to the Apache2 htdocs folder
6. The file is downloaded by another host
7. Simulated hard disk crash (overwriting random sectors)

Copyright Violation Script 2/2

- Evaluation
  - Most evidence found
    - gpg password in `.bash_history`
    - Disk crash suspected
    - Donwload entry in the Apache log file
    - Filenames of the MP3s
  - Not found
    - Origin of the MP3s

Lessons Learnt: `mkfs`

- Used internally by Forensig²
  - To create file systems in a partition
- Undocumented behavior
  - Working on files, `mkfs` replace the file
  - The created file includes only the filesystem
    - No data of the previous filesystem is left
    - Results in unrealistic images

Outline

- Background
- Structure of the generator
- Experiences
- Summary/Outlook

Summary

- Input script produces disk images
- Two generator stages
  - First pass containing random elements
  - Second pass reproducible
- Extensibility
  - Python Syntax
- OS installation, User interaction

Outlook

- Forensig² development
  - More modules to extend the possibilities of Forensig²
  - GUI Interaction
  - More tests with students (next summer term)
- Related research plans
  - Evaluate the analysis process itself
    - Are the analysis methods good?
    - Do they meet court standards?
    - Could it get better?
  - Utilize methods of social psychology to understand how the investigator come to his results