Reproducibility of Digital Evidence in Forensic Investigations

Lei Pan and Lynn M. Batten

{ln,lmbatten}@deakin.edu.au

School of IT, Deakin University,
Melbourne, Australia

August 19, 2005

Content of this Talk

- Introduction
- Reproducibility
- Existing forensic system models
- Our new model
- A forensic investigation example
- Analysis of example cases
- Conclusions

Introduction

Digital investigation is to use “scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence... ” (Palmer, 2001)

Reproducibility

- Reproducibility is the ability to achieve a consistent level of quality throughout the process, no matter how many times it is repeated under the same conditions.
- “Reproducibility of experimental notions allows to express and to test experimentally falsifiable hypotheses concerning the equality and repetition of the corresponding result values.” (`http://en.wikipedia.org/`)

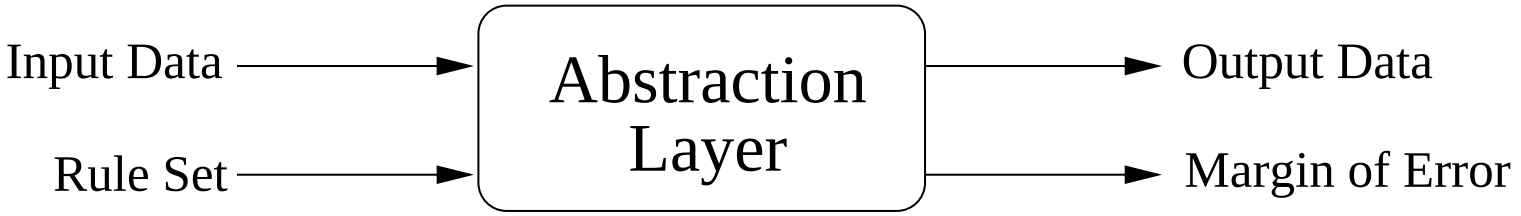
About Abstraction Layers

- Abstraction layers can be introduced to cluster similar things together.
- Abstraction layers can be introduced in all forms of digital data, and also in the tools and systems used to store or transmit them.

The Hadley Model

- The Hadley model defines input as writing data and output as reading data.
- “All input and output is the translation and transport of data.” (Gerber and Leeson, 2004)
- Each layer represents a physical location of data.

Carrier's Model (Carrier, 2003)



- Each layer represents a set of formats of digital sequences.
- Abstraction layers are chosen according to the different situations and requirements.

Time-stamps

- Hosmer (Hosmer, 2002) recognizes the importance of the time synchronization between digital entities in a digital investigation.
- Ideally, secure auditable time stamps assure accuracy, authentication, integrity, non-repudiation and accountability.

A New Model

1. Setting up abstraction layers
2. Assigning ‘read’ and ‘write’ operations
3. Time-stamping all operations

Setting up Abstraction Layers

We use:

- the same structure of single layer as Carrier does.
- four key points to establish abstraction layers in a system (similar to Gerber and Leeson's).
  a) Create a layer only when necessary.
  b) Each layer represents a location or function.
  c) Each layer represents existing components based on input-output.
  d) The number of layers should be carefully chosen.

Assigning Operations

- A ‘read’ operation is denoted pictorially as an arrow starting from the layer in which the digital sequence resides and pointing towards the layer to which the content of the sequence is extracted.
- A ‘write’ operation is represented as an arrow starting from the layer upon which the digital sequence resides and pointing to the layer on which the sequence is finally stored.

Time-stamping

- Trusted time-stamps are obtained, recorded and applied during the entire investigative process.
- A ‘read’ operation is time-stamped at the beginning of a command to read information; a ‘write’ operation is time-stamped at the conclusion of the ‘write’ operation.
- If a ‘read’ fails to start due to data synchronization, then we delay it until the ‘write’ finishes.

Situations Affecting Reproducibility

- Write-After-Read (WAR) means a ‘write’ operation changes the content in what we ‘read’ before.
- Write-After-Write (WAW) operation happens after a previous ‘write’ operation. So the content of the previous write is erased.

A Digital Investigative Example

- Target: Linux 2.6 host named `pie` with an IDE drive (with two `ext2` partitions) and a CD-RW
- Software Tools: `cfdisk`, `lsof`, `dd`, `nc`, `mkisofs`, `cdrecord`
- Environment: LAN connection and a file server `bee`
- Task: Collecting evidence from the hard disk of `pie`

Disk Usage Information

Suspicious files are identified at `/home/tunna/.pic/hidden/`. The disk usage information is displayed by executing `cfdisk`.

```
Filesystem      Used Available Use% Mounted
/dev/hda1    3810876   5314868  42% /
/dev/hda2   26041508  36914748  42% /home
```

An additional requirement is that we are not allowed to power off the target host.

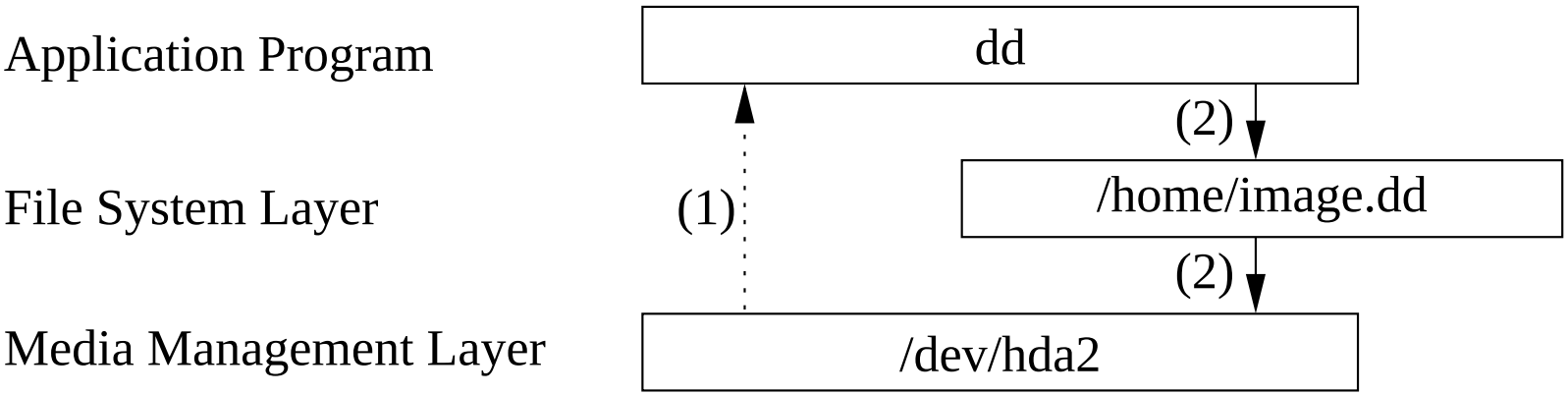
Case 1

```
pie #> dd if=/dev/hda2 of=/home/image.dd bs=1024 \
        count=26041508
```

This naïve approach attempts to image the first 26 million blocks (of size 1KB) of the 2nd partition of the target disk.

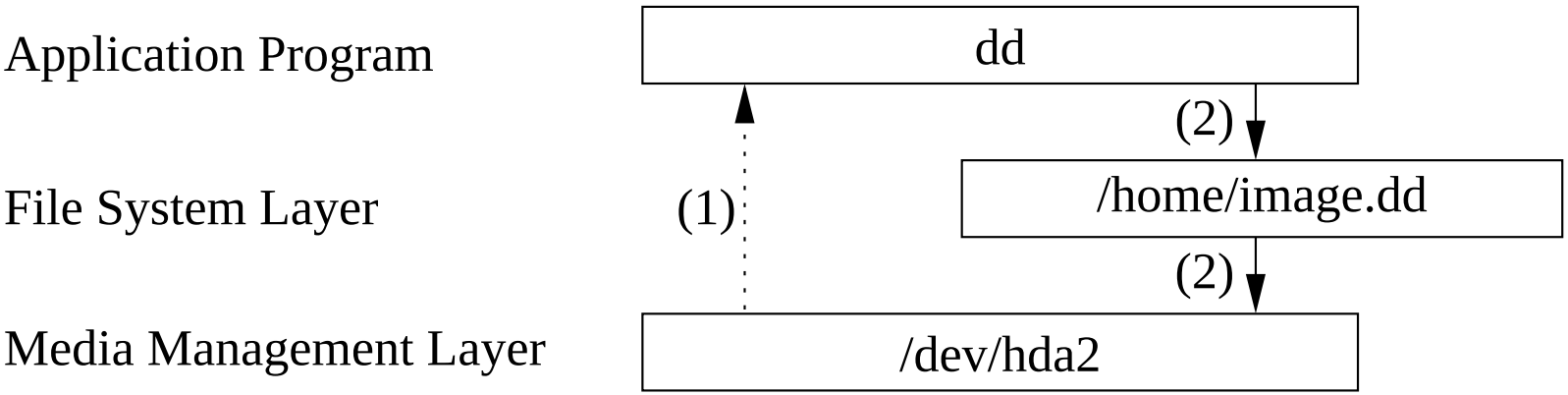
Case 1

```
pie #> dd if=/dev/hda2 of=/home/image.dd bs=1024 \
       count=26041508
```



Case 1

```
pie #> dd if=/dev/hda2 of=/home/image.dd bs=1024 \
       count=26041508
```



WAR on the 2nd partition.

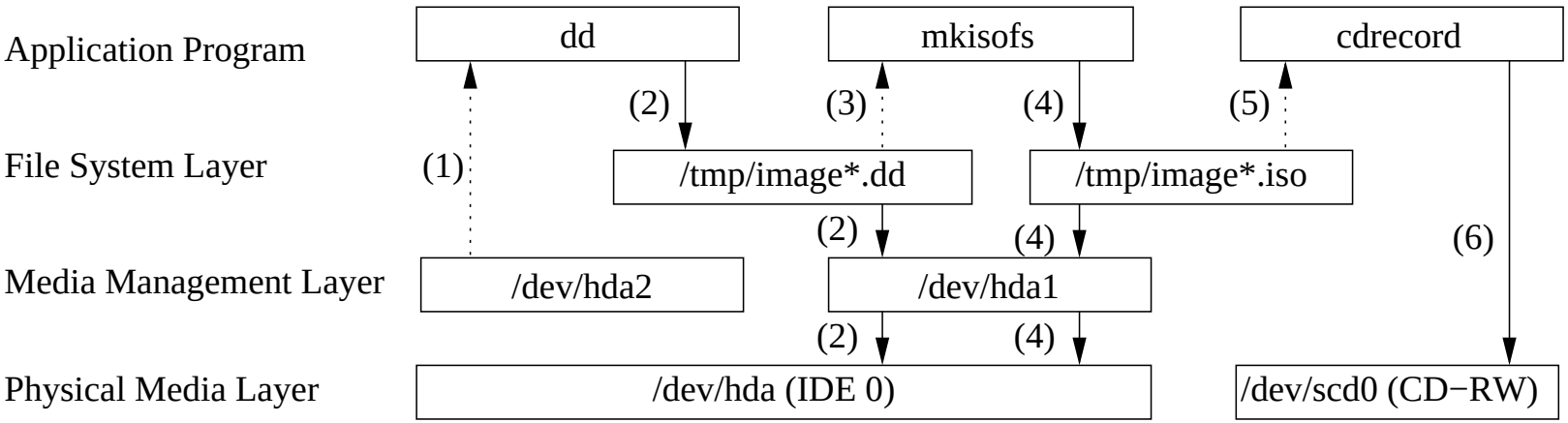
Case 2

```
pie #> dd if=/dev/hda2 of=/tmp/image1.dd bs=1024 \
 count=700000; mkisofs -o /tmp/image1.iso /tmp/image.dd; \
 cdrecord -v speed=4 dev=0,0,0 -data /tmp/image1.iso
```

This is to convert disk image file into `iso` image and then to burn it on the CD.

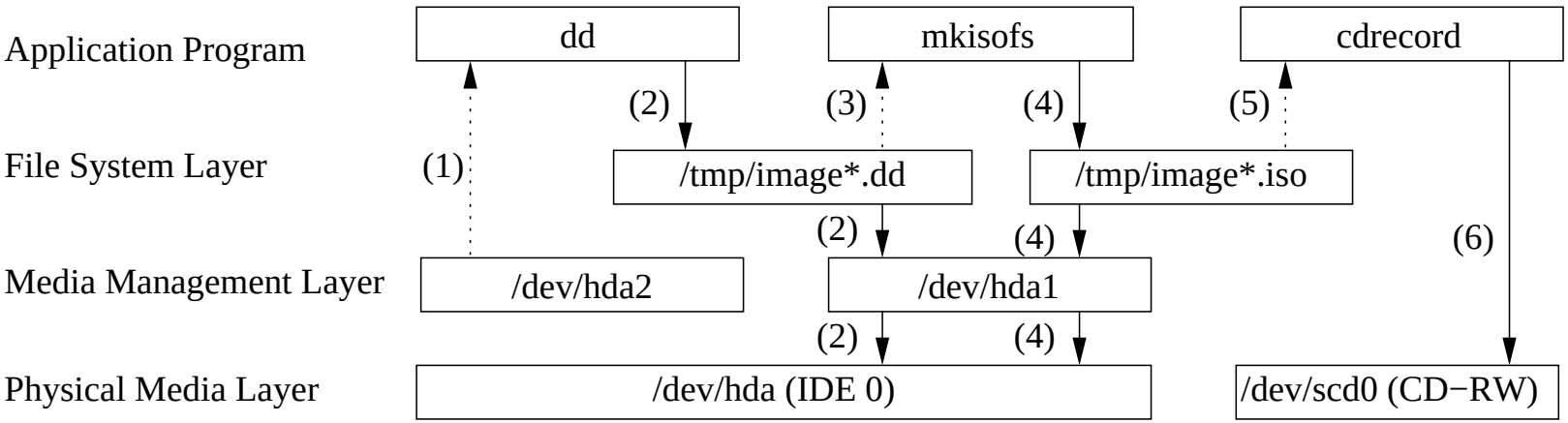
Case 2

```
pie #> dd if=/dev/hda2 of=/tmp/image1.dd bs=1024 \
 count=700000; mkisofs -o /tmp/image1.iso /tmp/image.dd; \
 cdrecord -v speed=4 dev=0,0,0 -data /tmp/image1.iso
```



Case 2

```
pie #> dd if=/dev/hda2 of=/tmp/image1.dd bs=1024 \
 count=700000; mkisofs -o /tmp/image1.iso /tmp/image.dd; \
 cdrecord -v speed=4 dev=0,0,0 -data /tmp/image1.iso
```



WAW on the hard drive and its 1st partition.

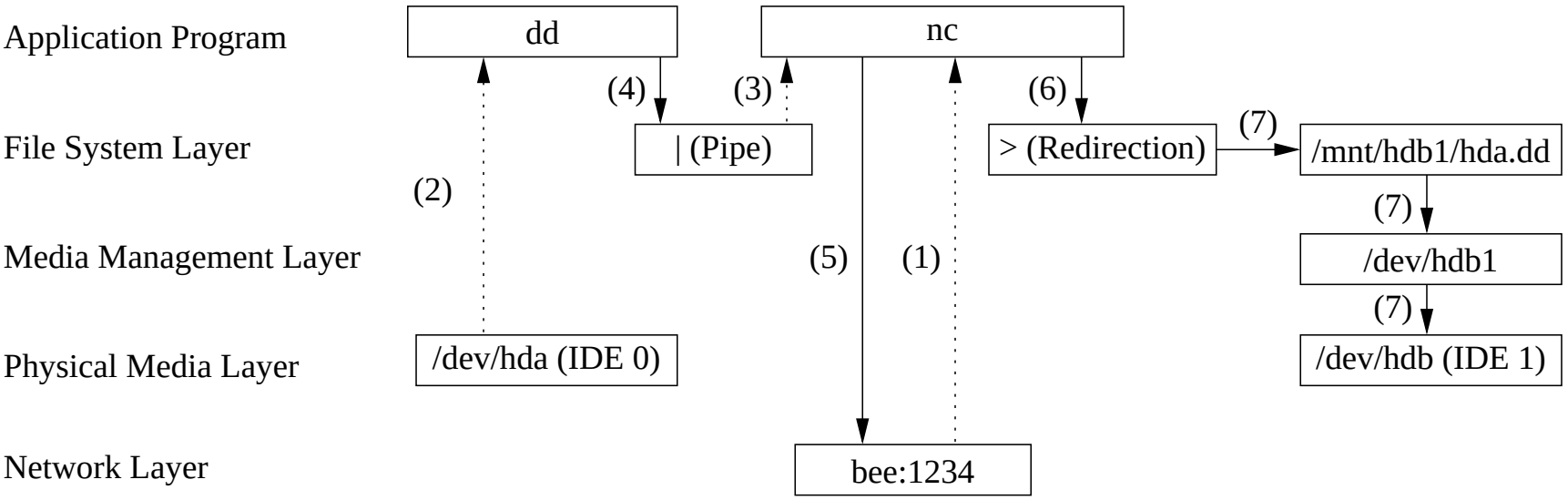
Case 3

```
jo@bee $> nc -l -p 1234 > /mnt/hdb1/hda.dd
pie #> dd if=/dev/hda | nc bee 1234
```

This uses the `netcat` program to transfer the disk image across the LAN, where `bee` is a trusted file server.

Case 3

```
jo@bee $> nc -l -p 1234 > /mnt/hdb1/hda.dd
pie #> dd if=/dev/hda | nc bee 1234
```



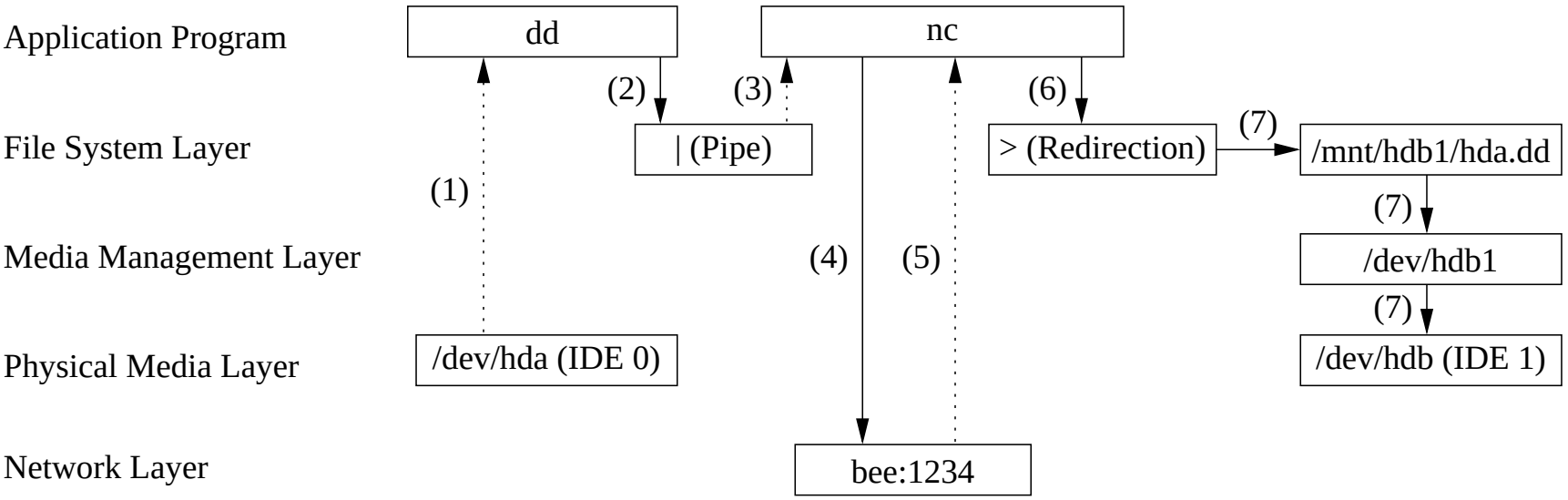
Case 3

```
jo@bee $> nc -l -p 1234 > /mnt/hdb1/hda.dd
pie #> dd if=/dev/hda | nc bee 1234
```

The Linux kernel blocks ‘read’ operations on pipes and nc blocks ‘read’ operations on port 1234 until ‘writes’ are done. Therefore, ‘reads’ finish after ‘writes’.

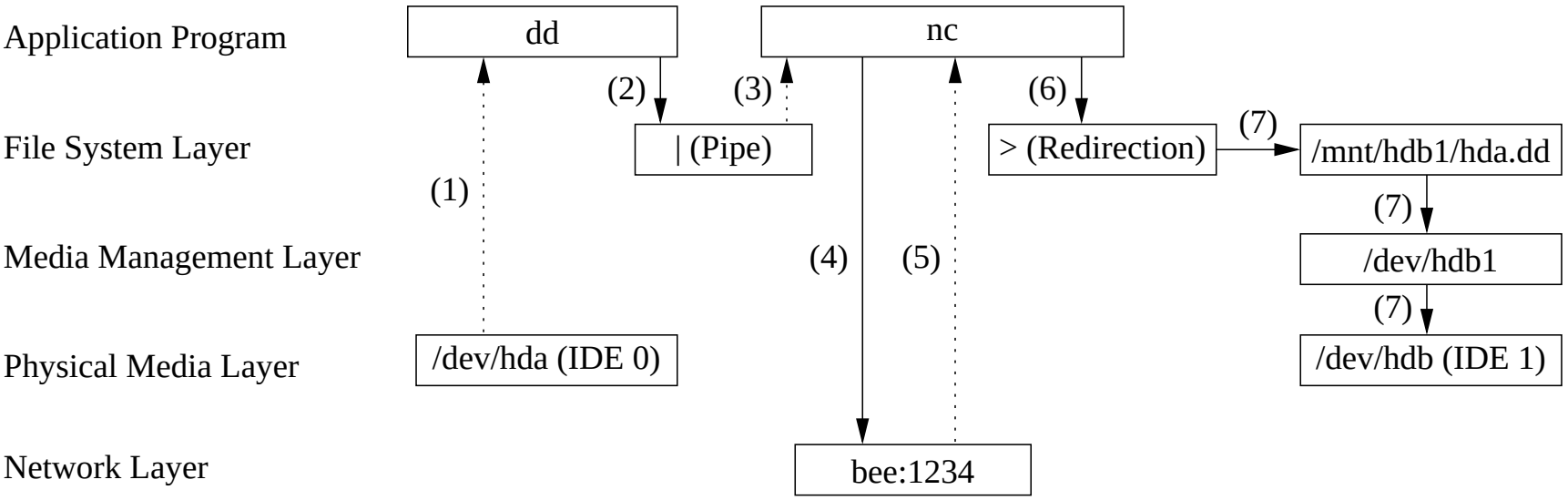
Case 3

```
jo@bee $> nc -l -p 1234 > /mnt/hdb1/hda.dd
pie #> dd if=/dev/hda | nc bee 1234
```



Case 3

```
jo@bee $> nc -l -p 1234 > /mnt/hdb1/hda.dd
pie #> dd if=/dev/hda | nc bee 1234
```



No WAR, nor WAW.

Analysis of the Example

- Carrier's model only supports a single program's behavior.
- The Hadley model couldn't distinguish the legitimate actions from the inappropriate ones with respect to reproducibility.

Conclusions

- Our model is generic, well-scaled and highly compatible with all functions what current digital systems provide.
- The results of the example cases highlight the importance of the order of operations to the reproducibility of the investigation process.

References

Carrier, B. (2003). Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers. *International Journal of Digital Evidence*, 1.

Gerber, M. and Leeson, J. (2004). Formalization of computer input and output: the Hadley model. *Digital Investigation*, 1:214 – 224.

Hosmer, C. (2002). Proving the Integrity of Digital Evidence with Time. *International Journal of Digital Evidence*, 1.

Palmer, G. (2001). A Road Map for Digital Forensic Research. Technical Report DTR - T001o -01, Digital Forensic Research Workshop.