XIRAF
Ultimate Forensic Querying

DFRWS - August 15, 2006

Wouter Alink, Raoul Bhoedjang
Netherlands Forensic Institute

Peter Boncz, Arjen de Vries
Centrum voor Wiskunde en Informatica
Introduction

XIRAF

“An XML Information Retrieval Approach to Digital Forensics”

Collect, manage, and query information extracted from digital evidence
Outline

• Problem statement
• XIRAF approach
• XIRAF architecture
• Forensic application areas
• Initial experiments
• Conclusion
Typical investigation steps

1. Media capture
2. Feature extraction
3. Analysis
4. Reporting
Problem identification

• Large amounts of data
  • Investigation restricted by deadlines
  • Too much information to track manually

• Diversity of data and tools
  • Many different formats
  • Many stand-alone forensic tools
Approach

- Clean separation between feature extraction and analysis
- A single, XML-based output format for tools
- XML database technology to analyze extracted features
- Use of existing forensic analysis tools
XIRAF architecture
Tool wrapper

- data from evidence files `Photo03.jpg`
- Optional: additional metadata

metadata (features/traces)
- new view of the original data
Tool repository

- Feature extraction tools
- Gain knowledge about an ‘object’:
  - volume
  - file-system
  - image
  - email
- Some of the wrapped tools:
  - file-system dissector
  - windows registry analyzer
  - EXIF-data parser
  - carving tool
  - IE-history parser
  - Hashing tool
XIRAF architecture
Feature extraction framework

- Tool Repository
  - tool A
  - tool B
  - tool C

- case initialization

- is there input for the tool?

- run tool

- storage subsystem
Feature extraction framework

Tool Invocation
- fetch data for tool
- for each item of data: call wrapper
- collect and check output
- merge with current data

Storage subsystem
- Case Data (BLOB)
- Annotations (XML)

Tool-execution wrapper
- pre-process input
- post-process output

Forensic Analysis Tool

Digital Forensic Research Workshop - August 15, 2006
Feature extraction
XIRAF architecture

Feature Extraction Framework

Tool Repository
  - tool A
  - tool B
  - tool C

Tool Invocation Process

Storage Subsystem
  - Annotations
  - XML document

Case Data
  - Binary Large Object (BLOB)

Query Interface
  - MonetDB/XQuery DBMS
  - StandOff extensions

To tool A
To tool B
To tool C

Tool Repository

Query Interface

To Tool Repository
To Feature Extraction Framework
To Tool Invocation Process
To Storage Subsystem
To Query Interface
Virtual BLOB and XML

<case name="testcase">
  <filename>Phonedatapath</filename>
  <image path="casedata/HD-B.jpg" start="7600" end="19999"/>
  <image path="casedata/HD-C.e01" start="20000" end="29999"/>
  <image path="casedata/HD-D.c01" start="30000" end="59999"/>
</case>

<modified><date>2006-08-15T09:10:00</date></modified>

...
Storage subsystem

- Virtual BLOB mapping
  - evidence files
  - alternative representations
- Single XML document
  - extracted features
  - references to layout
XIRAF architecture
XQuery language

- Database language:
  - large XML documents
  - sorting/grouping/selecting/(updating)

- Example: timeline
  - different tools produce date-elements

```xquery
for $i in doc("case.xml")//date
order by $i
where $i > $lowerbound
    and $i < $upperbound
return $i
```
Forensic application areas

- search for keywords, MD5s, URLs

```xml
for $i in doc("case.xml")//file
for $j in doc("CP-hashes.xml")//md5
where $i/md5 = $j
return <file> { $i/@name } </file>

let $word_list :=
  doc("terrorism-words.xml")//word
for $i in doc("case.xml")//*
where some $i in $word_list
  satisfies blob-contains($i,$j)
return element { name($i) } { $i/@* }
```
Benefits

- Exploit exhaustive runs of tools
- Use knowledge from previous investigations
- Integrated data schema

Added functionality:
- XQuery extensions to relate XML to Virtual BLOB content
let $d := doc("case.xml")$
for $i$ in $d//object_of_interest$
where $i/descendant::contains[so-contains(keyword_1)]$
and $i/ancestor::contains[so-contains(keyword_2)]$
and (some $j$ in $i//date//date$
satisfies $j \geq \text{lowerbound}$ and $j < \text{upperbound}$)
return element { name($i$) } { $i/@*$ }
XIRAF architecture
Initial Experiments

- Evidence: 2 hard disks
  - (2 x 120GB)
- ~200MB XML
  - ~2.5M elements
- Recognized ~90000 files
  - file-systems / unallocated space
- ~500000 timestamps
  - file-system, registry, EXIF, .LNK, log-entry, cookie, etc
Conclusion

- Separation of feature extraction and analysis seems a viable approach
- Integrated querying of multiple tools becomes possible
Status & Future Work

- Prototype implementation (Java/Python)
- Make system production-ready
- More tools, query patterns
- Connect XIRAF to existing knowledge-bases
More information

- xiraf-info@holmes.nl
- http://www.forensischinstituut.nl/
- http://monetdb.cwi.nl/