available at www.sciencedirect.com

SCIENCE DIRECT®

journal homepage: www.elsevier.com/locate/diin

Digital Investigation

XIRAF – XML-based indexing and querying for digital forensics

W. Alink[a,*], R.A.F. Bhoedjang[a], P.A. Boncz[b], A.P. de Vries[b]

[a]Netherlands Forensic Institute (NFI), Laan van Ypenburg 6, The Hague, The Netherlands
[b]Centrum voor Wiskunde en Informatica (CWI), Amsterdam, The Netherlands

Keywords:
XIRAF
Forensic digital investigation
XML database
Tool-integration
XQuery
Standoff annotation

ABSTRACT

This paper describes a novel, XML-based approach towards managing and querying forensic traces extracted from digital evidence. This approach has been implemented in XIRAF, a prototype system for forensic analysis. XIRAF systematically applies forensic analysis tools to evidence files (e.g., hard disk images). Each tool produces structured XML annotations that can refer to regions (byte ranges) in an evidence file. XIRAF stores such annotations in an XML database, which allows us to *query* the annotations using a single, powerful query language (XQuery). XIRAF provides the forensic investigator with a rich query environment in which browsing, searching, and predefined query templates are all expressed in terms of XML database queries.

1. Introduction

A typical digital forensic investigation involves these four phases:

1. media capture (e.g., forensic disk duplication);
2. feature extraction (e.g., parsing file systems, mailboxes, chat logs, etc.);
3. analysis (browsing, querying, correlating);
4. reporting (writing down findings for court).

This paper addresses two key problems that occur in the feature extraction and analysis phases of a computer system investigation. First, the amount of data to process in a typical investigation is huge. Modern computer systems are routinely equipped with hundreds of gigabytes of storage and a large investigation will often involve multiple systems, so the amount of data to process can run into terabytes. The amount of time available for processing this data is often limited (e.g., because of legal limitations). Also, the probability that a forensic investigator will miss important traces increases every day, because there are simply too many objects to keep track of.

Second, the diversity of the data present on a typical hard disk is overwhelming. A disk image contains a plethora of programs and file formats. This complicates processing and analysis and has led to a large number of special-purpose forensic analysis tools (browser history analyzers, file carvers, file-system analyzers, etc.). While it is clear that the output of different tools can and should be combined in meaningful ways, it is difficult today to obtain an integrated view on the output of different tools. And again, it is quite unlikely that a forensic investigator has both the time and the knowledge to apply all appropriate tools to the evidence at hand.

Our approach to solving these problems involves these key elements:

- a clean separation between feature extraction and analysis;
- a single, XML-based output format for forensic analysis tools;
- the use of XML database technology for storing and querying the XML output of analysis tools.

* *Corresponding author.*
E-mail address: wouter@holmes.nl (W. Alink).
1742-2876/$ – see front matter © 2006 DFRWS. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.diin.2006.06.016

Feature extraction and analysis are often interleaved and are sometimes seen as a single step. By separating feature extraction from analysis, we can, to a large extent, automate the feature extraction phase. This is essential for dealing with the ever-increasing amounts of input data. The use of XML as an intermediate format allows us to manage the heterogeneity of both the input data and of forensic feature extraction tools. Different tools with a similar function can be wrapped so that they produce similarly structured (XML) output. That output can then be processed by a single analysis tool that no longer has to deal with the idiosyncrasies of various input formats. Finally, by storing the XML annotations in a database system, we obtain all the benefits of declarative, general-purpose query languages.

To test this approach, we have implemented a prototype system called XIRAF (an XML Information Retrieval Approach to digital Forensics). XIRAF automatically extracts features from disk images and stores those features in a high-performance XML database system. The XML database *and* the disk-image data that is referenced by the XML annotations can be accessed through XQuery (Boag et al.), an XML query language. Since we do not expect all forensic analysts to be XQuery experts, we provide, through a web interface, a number of predefined query templates and standard analysis (e.g., a timeline).

The remainder of the paper is structured as follows. Section 2 discusses related work. Section 3 gives an architectural overview of XIRAF. Section 4 describes application areas in which XIRAF can be useful. Section 5 gives an overview of our initial experiences with the prototype. Finally, Section 6 presents our conclusions and our plans for future work on XIRAF.

2. Related work

Our work on XIRAF is related to several other fields and efforts. First, and perhaps foremost, we are aware of several ongoing projects in the law enforcement community that aim to automate feature extraction for large evidence sets. The need for such automation has been expressed by various authors (Buchholz and Spafford, 2004; Carrier and Spafford, 2003; Mohay et al., 2003; Sheldon, 2005). Unfortunately, very little is published about these projects. One such project is the Computer Forensic Investigative Toolkit (CFIT) (Mohay et al., 2003), a system developed by Australia's Defence and Science Technology Organization. To the best of our knowledge, CFIT focuses on automatic feature extraction and data visualization rather than the querying of extracted features.

XIRAF builds on recent advances in information retrieval and on XML-based information retrieval in particular. XML database systems are relatively new and large forensic data sets pose significant challenges to them.

Mainstream commercial toolkits such as Encase and FTK provide a user-friendly interface to a built-in set of forensic analysis tools. EnCase also provides its own scripting language, but no API that allows one to plug in existing, external tools written in a common programming language. XIRAF differs principally from these tools by its use of a query-able, intermediate data store that isolates feature extraction from analysis. As we will argue in this paper, this offers important benefits.

3. XIRAF

The XIRAF framework consists of three components (see Fig. 1). The *tool repository* houses a collection of feature extraction tools. The *feature extraction manager* orchestrates the invocation of these tools, merges their XML outputs, and stores the result in the *storage subsystem*. The storage subsystem consists of binary large objects that hold raw evidence data and an XML database that holds all extracted features.

3.1. The feature extraction manager

From XIRAF's perspective, an investigation starts when one or more raw digital evidence items, usually disk images, are fed to the system. Initially nothing is known about the content of these evidence items. The content is simply a single piece of binary data that we will refer to as a Binary Large OBject (BLOB).

The feature extraction manager is responsible for extracting from the input BLOBs as many useful features as possible. It does this by running tools from the tool repository in the correct order and by applying them to the correct inputs. It also tracks which objects have already been annotated by other tools and prevents duplicate annotations.

It is the tasks of individual tools to extract specific features from the BLOBs. A tool will normally operate on one or more byte ranges in the current BLOB set. Such a byte range is called

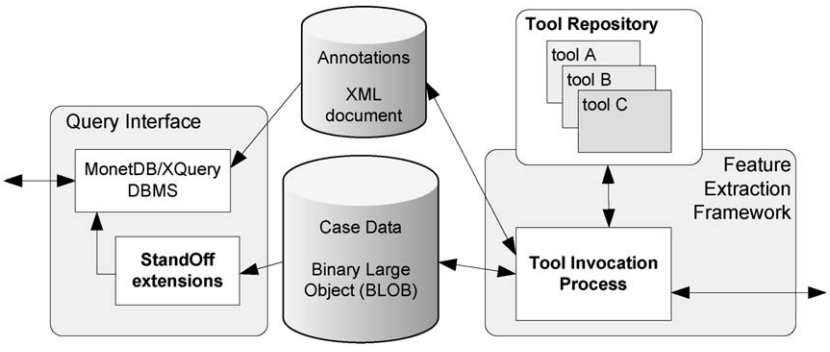

Fig. 1 – XIRAF framework architecture.