Semantic Modelling of Digital Forensic Evidence

Damir Kahvedžić* and Tahar Kechadi

Center for Cybercrime Investigations, University College Dublin, Dublin, Ireland
{damir.kahvedzic,tahar.kechadi}@ucd.ie

Abstract. The reporting of digital investigation results are traditionally carried out in prose and in a large investigation may require successive communication of findings between different parties. Popular forensic suites aid in the reporting process by storing provenance and positional data but do not automatically encode why the evidence is considered important. In this paper we introduce an evidence management methodology to encode the semantic information of evidence. A structured vocabulary of terms, ontology, is used to model the results in a logical and predefined manner. The descriptions are application independent and automatically organised. The encoded descriptions aim to help the investigation in the task of report writing and evidence communication and can be used in addition to existing evidence management techniques.

Keywords: Ontology Investigation Results Modelling Reporting.

1 Introduction

The Digital Investigation is becoming ever more complex with even small scale cybercrime investigations involving the analysis of multiple computers, flash disks, internet social networks, online accounts and mobile phones. Typically many tools may be used with results passed from one person to another during an investigation. The findings are typically reported in a prose document, detailing the steps taken, tools used and the interpretation of the results. As the size of the case grows, so does the complexity of its investigation and the description of the evidence. Once the case is concluded, the information is pooled together and presented to the relevant parties. Communication of the findings at each step in the process, in a clear, logical and reproducible manner is crucial for no evidence to be lost or miscommunicated. Current tools provide little context to the results they extract. The explanation of what the results imply, their relevance to the overall case, and where and how they relate to case events is all carried out in prose and may be prone to error.

In this paper we present a methodology to annotate results of digital investigative tools using a hierarchical vocabulary of computer forensics. The vocabulary provides context to the findings and allows the semantics and meaning of the results to be explicitly encoded. We use terms from DIALOG [9], a digital forensic ontology, developed to encompass knowledge associated with digital

* Financed by Irish Research Council for Science, Engineering and Technology, 2006.

I. Baggili (Ed.): ICDF2C 2010, LNICST 53, pp. 149–156, 2011.

forensics investigations. Describing evidence with the ontology involves instantiating relevant concepts and creating relationships between the individuals. The descriptions are textual, application independent and are used to supplement prose documents and existing evidence management techniques. The descriptions are easily merged and aggregated to form a complete view of the results. Further more, ontology specific processes, such as classifiers and inference engines, are employed to categorise the individuals and extract new knowledge. The ongoing development aims to supplement reporting procedures and relieve the investigator from manual descriptions of results with prose.

2 Previous Work

Managing, communicating and reporting of evidence is a crucial part of any investigation. Large forensic suites such as EnCase [4] highlight potential evidence and inserts them into report templates for quick dissemination of information. EnCase, for example, creates a web style report where the user browses through the findings in a similar manner to a web site. Bookmarks and hyperlinks are used to highlight significant information and lead the reader directly to where the evidence was found. A small number of bookmark types are available and allow both entire volumes as well as contents of files to be highlighted [1]. During the course of the investigation, users create hierarchical bookmark categories and place relevant evidence within them. The organisation is arbitrary and may result in logical inconsistencies where evidence should be in more than one category or are more applicable to one category rather than the other.

Bookmark information includes verbose structural, provenance and positional information only. Context, why the evidence is important and its meaning, is only specified in prose in the comment of the bookmark. Exchange of information between related parties and forensic suites is therefore limited to prose descriptions of evidence. Forensic tools cannot interpret prose without human intervention and therefore cannot automatically process the results further.

Forensic file formats have been developed to create such an exchangeable form of information across forensic applications. They provide compression, authentication and provenance to the data that was imaged. The formats often bundle images with a separate metadata component describing relevant case information [5,3,14,12]. The Sealed Digital Evidence Bag [12], used a set of limited ontologies to annotate the contents of individual evidence bags. The system first images the data and secondly records the details of the process and the evidence source by asserting properties from a predefined ontology.

Ontologies have been used to model arbitrary data in a number of domains [10]. In this work we are concerned with applying the logical structure of ontology [6] to evidence found during the course of the investigations. We use a hierarchical vocabulary of terms to allow the user to add semantic descriptions to arbitrary evidence. The ontology models all the results, including contents of files and events and automatically categorises them to relevant types. The evidence is categorised similarly to the EnCase bookmark categories but employs logical tools, such classifiers to verify the descriptions. The descriptions are stored in a separate file from the evidence image and provide an application independent and structured description of the case. Ontology descriptions are easily merged between tools and together create a rich description of the case.

In the rest of the paper we show how we utilise a digital forensics ontology, DIALOG [9], to describe the specific contents of the data. We use individuals and properties to annotate data and axioms and inference engines to classify them to relevant concepts. The hierarchical, logical and descriptive properties of ontology are utilised to organise the information in an application independent manner. The knowledge base can be explored and queried separately and supplements the creation of the final report.

3 Digital Investigation Concept

We use, DIALOG [9], a digital forensic ontology to describe forensic investigation results. It is a hierarchical metadata model describing concepts and the properties between them. The ontology can be regarded as a taxonomy of information with progressive restrictions (axioms) defining concepts down the hierarchy. DIALOG is defined using the ontology web language (OWL) and defines a single top level concept, the "*DigitalInvestigationConcept*". All other concepts are sub-classed from this parent, relate to associated concepts using defined properties and explicitly encoded a model of the forensics field. DIALOG has five main branches, the "*CrimeCase*", "*Information*" type, the "*Location*", the "*ForensicResource*" and the "*InvestigationActor*", each containing a hierarchy of concepts. The hierarchy is similar to the bookmark categories in EnCase but is predefined and more expressive since categories maintain links to related categories through predefined properties. Ontology can also leverage tools to process the information held within it.

Pellet [2], for instance, is an ontology reasoner that reclassifies individuals to the concepts whose axioms they satisfy. Based on the individual's properties, Pellet employs logical rules to find the concepts that the individual belongs to and may move the individual to more than one concept category. Therefore, to create any instance in the ontology, the user needs only to create an instance of a *DigitalInvestigationConcept* and annotate it with relevant properties. The Pellet inference engine would reclassify these individuals to the relevant concept. Alternatively, a sub-concept can be chosen, such as a *File*, and Pellet can specialise the individual to the specific *File* type *and* the generalise to the type of *Evidence* it is.

The definitions of concepts and properties are continually being expanded to increase the expressivity of the ontology. Our approach differs from [12] since we use inference mechanisms to automatically group and reclassify individuals.

All annotations are stored in a separate ontology file in OWL (Web Ontology Language). OWL has evolved from and is compatible with XML. Each instance occupies its own `<concept></concept>` block and connects to other instances with properties, also defined separately in concept blocks. Therefore, the annotations are supplementary to other investigation processes and referenced as an aid in creating the report. The following section describes how the evidence findings are encoded and how context is provided to the results.

4 Encoding of Results

In this section we illustrate how results from forensic tools are encoded in an ontology. The results instantiate relevant concepts, asserts properties and gives them a richer context than simple textual reporting. Each instance is named with a URI (Unique Resource Identifier) in the following format:

```
Format:  <namespace>:<asserted_type>-<Unique_Universal_ID>
Example: di:File-5038b43-511d-4381-9afe-c15d556b52c4
```

The <*namespace*>, locates the owl file that defines this individual. The default location is `damir.ucd.ie\\DigitalInvestigation.owl`. It is a blank ontology which simply imports all needed concepts and properties required for annotation. A copy is used locally for each result set and is modified to reflect the new location by editing the `xml:base` of the copy. The default namespace prefix is "*di*", but can also be modified in a similar manner. The "<*asserted_type*>" is a place holder for the individuals type. Additional types may be inferred by Pellet if the individual fulfil some concept properties. The "*UUID*" is a unique number. The instance name is verbose but designed to be unique. Typically, users would not see the underlying ID of an instance but some other more readable identifying information. For the rest of the paper, we use a shorthand notation, "*di:File-5038b43*", to refer to instances.

In the following we illustrate how we provide semantics to metadata, annotate file content and encode forensically relevant events. We model evidence found with tools built by our team [7,8] but results from other forensic tool can also be encoded. We describe a simple file, a document showing two people at some relevant location, that was found to be of evidentiary value.

4.1 Encoding Metadata

Metadata is loosely defined as data about data and used extensively by operating systems to provide rudimentary descriptions of specific information, such as files and folders. The metadata are simple tags and lack any structure, semantics or relationships. We encode these relationships using the concepts and properties from DIALOG and create a more richer description of the information.

For example, it is common to find files with names that relate to other concepts in the investigation. An image file, for instance, may be named after the person's portrait, the location that it was taken in, the event that it captures etc. Figure 1 shows how a simple scenario is encoded, where a filename is also a name of a person. Similar scenarios can occur if a filename is a name of an event or if the metadata reveals some other information.

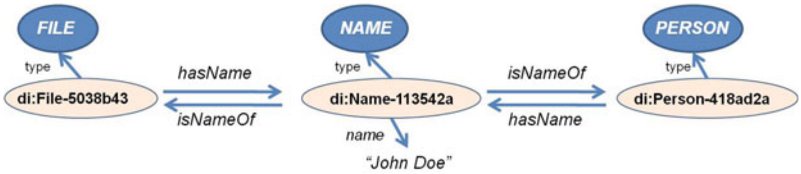

Fig. 1. Encoding Metadata Example

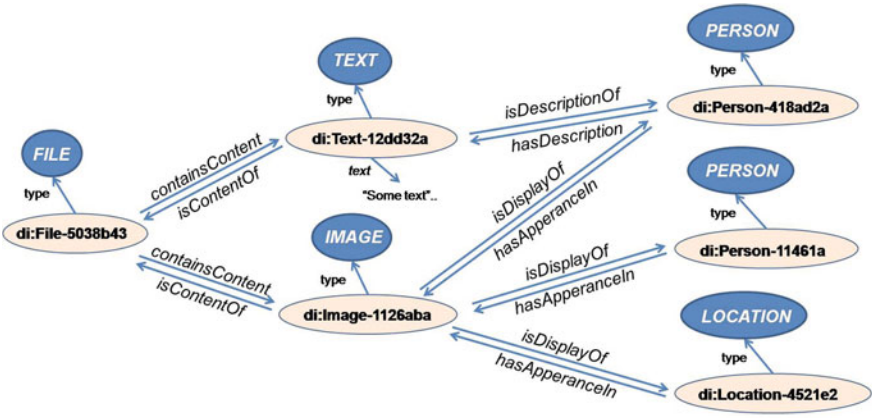

Fig. 2. Encoding Content Example

4.2 Encoding Content

Encoding the content of information concerns annotating the data container with the appropriate concepts that it holds information on. Data containers are defined as any object that contain other data. Here, we take the *File* as an illustrating example but we use the same method to describe other containers, such as the *Folder* or *RegistryValue*. Semantically, files cannot contain physical objects, rather they contain *evidence of* physical objects stored in some format. There is no restriction on the number of different information a file can hold. An email file, may contain a number of emails stored in a cascading format and a word processing file may contain both text, images and emails. Therefore, to describe the content of a file in DIALOG, the user first specifies the contents of the file, an image for example, then describes what the content is evidence of.

Figure 2 shows an encoding of the content of our example file. It contains an image and some text. The image display two people at a location, the person the file was named after and the person that created it. The various concepts are further annotated to describe them in more detail but are not present in the figure. These include the address of the location for example, the specific text that was found to be relevant and the location (page number) of the elements within the file.

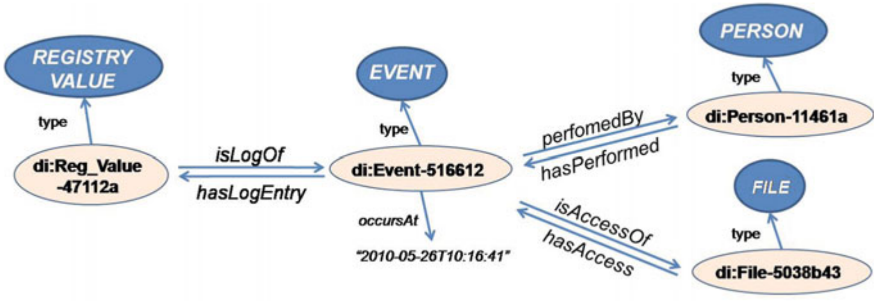

Fig. 3. Encoding Events Example

4.3 Encoding Events

Metadata and content are extracted by the majority of forensic tools and can be encoded as above. Event reconstruction is carried out implicitly based on this information. DIALOG also encodes event information. RPCompare [9] is a tool developed to extract event information from the Windows registry. Amongst other things, it can analyse MRU lists to extract the order of which files were accessed across time. The file accesses represent Events, an occurrence of an action, which are performed by some person and occur at a specific point in time. Time ontologies are already specified [15] and are integrated into DIALOG to encode when the event occurs as well as the order of events.

Figure 3 shows an example of how a *FileAccess* event is encoded. The *isLoggedBy* property links the source of the evidence to the actual event. In this case it is an MRU registry value. The figure omits any relationships, such as *hasName* and *isInRegistryKey* that the instance also asserts. Access events of external devices, operating systems or other data is similarly encoded. The event ontology also supports the description of deletion events, creation events and can be expanded incrementally for other event types. The creation of the example file by the user ("*di:Person-11461a*") is encoded in that manner.

5 Data Retrieval and Implementation

Once data is annotated within the ontology it can be readily manipulated and explored with ontology specific operations. Pellet, for example, manages the information and infers the category of every instance by testing them against concept axioms. For example, although file "*di:File-5038b43*" was asserted to be only an instance of a file, due to its properties it was inferred that it is also an instance of a *ImageFile*, *UserFile*, *MultimediaFile*, *MultimediaEvidence* amongst other concepts. Similarly, the various events are inferred to be *FileAccessedEvent*, *FileCreationEvent* etc. Additionally, the hierarchy of DIALOG groups instances to parent concepts and allows the individuals to be accessed through parent nodes. The grouping of similar evidence types simplifies the exploration of the knowledge. The knowledge can subsequently be explored either manually or by querying the information.

The ontology query language, SQWRL, is used to query for specific evidence. The query below, for example, queries for any elements that contain evidence of both the people used in the previous examples. It retrieves evidence that was found to prove that person *A* had some connection with person *B*. The query uses the DIALOG property *isComponentOf* which is a transitive super property of *isLogOf*, *isNameOf*, *isDisplayOf* used in the examples above. The query applies to all of them and only returns instances that satisfy the conditions. In the example above, the query returns the instance of image, "*di:Image-1126aba*", that was found to contain pictures of both the persons, it can be similarly used to extract conversations, events etc. that both agents took part in.

```
inf:isComponentOf(?di:x, di:Person-11461a)   ^
inf:isComponentOf(?di:x, di:Person-418ad2a)  ->
sqwrl:select(?di:x)
```

As well as querying the knowledge, the information can be explored manually. Since ontology descriptions are written in an extension of XML, existing XML editors can be used to explore the underlying data. Some ontology specific editors exist to interpret and present OWL information [13].

We have implemented a new ontology editing environment that is used to describe results found by forensic tool in the manner illustrated above. The environment can aggregate results from multiple sources to a central location and allow the exploration of the data in a variety of ways. It allows the user to query the information as above or explore it manually through either a grid or "linked-graph" manner. A number of operations have been implemented to allow the user to concentrate on specific evidence or properties of evidence. The graph view in particular allows the information to be viewed in a similar way to how ontology is visualised in the figures of this paper.

Our ongoing work will concentrate on simplifying the user interface to reduce the amount of effort required for data input and an exporting feature to report findings. Reporting entails parsing the instances and following their properties. The extracted text will be similar to prose and can be used to supplement existing reporting procedures.

6 Conclusion

In this paper we illustrated the ongoing development of an evidence management and reporting methodology for digital forensics. We use a digital investigation ontology to model metadata, file content and event evidence in an application independent and semantic manner. The descriptions provide context to the data and allows the evidence to be explored in an intuitive way. The methodology is similar to the bookmarking system of many forensic suites, in that evidence is progressively tagged during the course of the investigation, but differs in the fact that it is structured, application independent and annotates the meaning of evidence rather than its structure and position.

Ontology mechanisms, such as inference, classification and catagorisation, are used to manage the data and group instances together for easy understanding, exploring and querying. Future directions of the research include development of a reporting tool that parses asserted information and creates prose like text. Addition of general vocabularies such as WordNet [11] will also be carried out to add more expressivity to the descriptions.

References

1. Bunting, S.: EnCase Computer Forensics: EnCe The Official EnCase Certified Examiner Study Guide, 2nd edn., Sybex (2008)
2. Pellet, `http://clarkparsia.com/pellet/` (visited: May 2010)
3. Cohen, M., Garfinkel, S., Schatz, B.: Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow. Digital Investigation 6, 57–68 (2009)
4. Encase, `http://www.guidancesoftware.com/` (visited: May 2010)
5. Garfinkel, S.L., Malan, D.J., Dubec, K.A., Stevens, C.C., Pham, C.: Disk imaging with the advanced forensic format, library and tools. In: Research Advances in Digital Forensics (2nd Ann. IFIP WG 11.9 Int. Conf. on Digital Forensics). Springer, Heidelberg (2006)
6. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. Int. Jrnl. of Human-Computer Studies 43, 907–928 (1995)
7. Kahvedžić, D., Kechadi, T.: Extraction and Categorisation of User Activity from Windows Restore Points. Jrnl. of Digital Forensics, Security and Law 4 (2008)
8. Kahvedžić, D., Kechadi, T.: Correlating Orphaned Windows Registry Data Structures. In: ADFSL 2009, Proc. of the Conf. on Digital Forensics, Security and Law, pp. 67–81 (2009)
9. Kahvedžić, D., Kechadi, T.: DIALOG: A Framework for Modelling, Analysis and Reuse of Digital Forensic Knowledge. Digital Investigation 6, 23–33 (2009)
10. Semantic Web Case Studies and Use Cases, `http://www.w3.org/2001/sw/sweo/public/UseCases/` (visited: May 2010)
11. Miller, G.A.: WordNet: A Lexical Database for English. Comm. of the ACM 38, 39–41 (1995)
12. Schatz, B., Clark, A.: An open architecture for digital evidence integration. In: Proc. of the 2006 AusCERT Asia Pacific Information Technology Security Conference R&D Stream, pp. 15–29 (2006)
13. Protégé Ontology Editor and Knowledge Acquisition System, `http://protege.stanford.edu/` (visited: May 2010)
14. Turner, P.: Applying a forensic approach to incident response, network investigation and system administration using digital evidence bags. Digital Investigation 4, 30–35 (2007)
15. Time Ontology in OWL, `http://www.w3.org/TR/2006/WD-owl-time-20060927/` (visited: May 2010)