



How To Search Extracted Data

By

Javier Collado

Presented At

The Digital Forensic Research Conference

DFRWS 2015 EU Dublin, Ireland (Mar 23rd- 26th)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment. As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

<http://dfrws.org>

How to search extracted data

Javier Collado

Data extraction in mobile devices

- It's hard to decode data for each application with limited resources
 - There are a lot of applications
 - Each application version might change:
 - format (file type, database schema)
 - content (new and interesting data)
- Many applications store data in SQLite databases

Index and search

- **Libraries**
 - Low level interface
 - Examples: lucene, xapian, whoosh
- **Servers**
 - High level interface
 - Examples: solr, elasticsearch, sphinx

SQLite

- **Very flexible and permissive: each value has its own type**
- **Storage class: group of related datatypes (different lengths, encodings, ...)**
- **Type affinity: preferred storage class for a column based on column type**
- **Not all the content should be indexed:**
 - **sqlite_master, sqlite_sequence**
 - **FTS tables**
 - **BLOBs**

SQLite

```
sqlite> CREATE TABLE names (id INTEGER, name TEXT);  
sqlite> INSERT INTO names VALUES (1, "Alice");  
sqlite> INSERT INTO names VALUES ("Bob", 2);  
sqlite> SELECT typeof(id), id, typeof(name), name FROM names;  
integer|1|text|Alice  
text|Bob|text|2  
sqlite>
```

SQLite

```
sqlite> CREATE TABLE names (id INTEGER name TEXT);
```

```
sqlite> .schema names
```

```
CREATE TABLE names (id INTEGER name TEXT);
```

```
sqlite> INSERT INTO names VALUES (1, "Alice");
```

```
Error: table names has 1 columns but 2 values were supplied
```

ElasticSearch

- Search server
- Document oriented (json)
- RESTful API
- Schema (mapping) not required, but needed to avoid errors due to SQLite flexibility

ElasticSearch

```
$ curl -XPOST 'http://localhost:9200/dfrws/names' -d '{id: 1, name: "Alice"}'  
{"_index":"dfrws","_type":"names","_id":"AUxNeQ7-7Nsk22Tyod1W","_version":1,"created":true}
```

```
$ curl -XPOST 'http://localhost:9200/dfrws/names' -d '{id: "Bob", name: 2}'  
{"error":"MapperParsingException[failed to parse [id]]; nested: NumberFormatException[For input string: \"  
Bob\"]"; ","status":400}
```

```
$ curl -XGET 'http://localhost:9200/dfrws/_mapping/names'  
{"dfrws":{"mappings":{"names":{"properties":{"id":{"type":"long"},"name":{"type":"string"}}}}}}
```

ElasticSearch

```
$ curl -XPOST 'http://localhost:9200/dfrws/_names' -d '{id: 1, name: "Alice"}'  
{"error":"InvalidTypeNameException[mapping type name [_names] can't start with '_']","status":400}
```

```
$ curl -XGET 'http://localhost:9200/dfrws/names/_search' -d '{query: {match: {name: "Alice"}}}'  
{"took":27,"timed_out":false,"_shards":{"total":5,"successful":5,"failed":0},"hits":{"total":1,"max_score":  
0.30685282,"hits":[{"_index":"dfrws","_type":"names","_id":"AUxNeQ7-7Nsk22Tyod1W","_score":0.30685282,"  
_source":{"id: 1, name: "Alice"}}]}}
```

Example tool

- <https://github.com/jcollado/esis>
- Command line tool written in python
 - Ability to index every row in every table in every database file found under a given directory
 - Ability to search using simple queries

Conclusions

- **SQLite content can be indexed in elasticsearch but...**
 - **Types need to be consistent**
 - **Not relevant information needs to be discarded**

Future work

- Index text information from other file types (Apache Tika)
- Regular expressions
- Highlight search results
- Search suggestions
- Language detection and custom analyzers
- Proximity matching (match vs. match_phrase)



Thanks