Paper Review

Database Forensic Analysis with DBCarver

Gurpreet Singh

February 12, 2018

CS4411

Research Summary

Background

Cyber crime has become more and more frequent every year and when large systems are hacked there is often a database involved in the process. The research paper "Database Forensic Analysis with DBCarver" exposes shortcomings in present day approaches used in the evidence collection process when dealing with databases. Wagner et al. [1]

The writers acknowledge the tools that are currently being used in the real world while stressing that they assume the databases to be in too much of an ideal state. Real world scenarios often involve databases with fragmented, damaged or deleted data and may not even have proper logging setup to capture a breach of security. They attempt to convince the reader that when analyzing a system after an attack preexisting detection mechanisms cannot be relied upon because the attacker could have tampered with them. Currently an incorrectly setup database would be able to greatly hinder an investigation.

Summary

This paper is written to provide a new and improved way of conducting forensic analysis on database management systems. A method that can be applied to any database, in any state, because it uses fundamental data processing at the smallest accessible unit of storage possible for a database. DBCarver relies upon each individual data page inside a database file instead of the pre-existing database management software installation. The researchers made use of file carving strategies established within the file system data recovery field and built on top of it to include the specific data structures present in data pages found within a database file. Since this is essentially a data recovery technique before a database recovery one, it is also able to recover deleted data pages that have not yet been overwritten by the file-system.

The fundamental concept of DBCarver is using all remaining data from a crime scene to reconstruct a new database of query-able and non-query-able data instead of try and manipulate the original database to make it work. This will allow investigators to conduct deeper analysis on the data and make connections faster. The writers covered three main aspects regarding the tool: explaining how the system actually works, showing how to solve real life problems with examples,

and exploring the time complexity of the solution. The system itself follows the national regulations set out by National Institute of Justice. This mainly ensures that all evidence is preserved during the investigation, therefore, assumptions are made about the running system because the data could dynamically change during runtime so it wouldn't be valid in an investigation.

Technical Explanation

DBCarver traverses pages within a desk image extracted from a relational database and tries to identify data within those pages to reconstruct evidence.

The actual DBCarver system itself works in two distinct steps. The first step of the tool is called the 'Parameter Detector'. The parameter detector sets up a fake database using the specified database management system and loads in some artificial data one entry at a time and records the system level characteristics. These characteristics include what a page in this database system looks like, where the row dictionary is located and other system configurations that help identify data in later stages. This data is extracted into a configuration file and only needs to be done once for each database management system. The writers state that this step requires a some user intervention.

The parameter detector sets up the second half of the tool, called the Carver, with it's extracted configuration as it's seed. Then, the carver uses this information on the real recovered data image to make sense of data pages.

The three major components that are present inside the pages of relational database management systems are: page headers, row dictionaries and row data entries. The page header provides general information about the page itself and how it relates to the database as a whole. For example, which table it belongs to, or the column names and types. The row directory helps index the data inside a single page. It can exist in different places within a page for each DBMS. The position of the row dictionary is extracted from the configuration generated earlier. This directory will have references to each of the separate row data entries. Each individual Row Data entry will have the raw data and the status in which the data was present. For example, It will indicate if the data was deleted or if it is still active.

The carver itself is responsible for taking the raw data page and the configuration and discovering the page header, row dictionary and row data, and then combining them back into a newly structured database for exploratory analysis. The carver is able to accept any type of data file of any size

because it is solely comparing against the criteria extracted by the parameter detector. Due to the fact that the carver is a binary command line utility, it is able to run concurrently with multiple databases and multiple configurations.

Depending on the type of data provided to the carver, it will produce a different format of output. If given a database image, a new database is created with the recovered data. If given a RAM image, the result is raw data present in the working memory in the form of cache pages.

Strengths and Weaknesses

Some strengths of this research paper and the DBCarver concept are: good use of experiments to prove the results the research claims is possible, system works on valid and corrupt databases, and works with many different database management system platforms.

I was very happy to see that specific experiments and examples were used to demonstrate how this tool would be used in the real world. I feel a paper that does this has a greater chance to be adopted into the industry because it's easier to imagine using a tool like this yourself.

The one major weakness I noticed in this research paper and it's presented solution is the unrefined DBMS bootstrapping utility. It seems that the parameter detector that is allowing the carver to dynamically adjust to a new database system requires too much user intervention from the writer's description of it. They state that the user may have to do things like write an entire wrapper class, run the SQL commands required to train the parameters, or even mess with schemas. I think the goal of a tool like this should be to allow easy pick up and usage in a larger collection of database analytics tools. In order for this to be possible either the parameter detector should have been a more automatic process or the parameters for the most popular databases be packaged with the carver itself. The paper states that you only need to extract the configuration once for each database, and there are a limited number of popular database systems, so it should be possible to ignore the detection step altogether.

Other weaknesses I discovered in the paper are: the lack of content regarding bad databases and a bit too much repetitive technical data. The paper talks about bad databases many times and then follows up with many examples of processing with good databases only. I was really interested in seeing how the system reacts to specific corrupted databases but I feel that was skipped over to talk

about RAM snapshots which are most likely less common to show any interesting data recovery. The writers also included many detailed step by step descriptions of the technical elements of the processing but I feel many steps were repeated making this paper longer than it had to be. More information is usually a good thing but not when it begins to lose the interest of the reader.

Opinions

I think the paper was very well structured and presented, it was easy to follow and understand the writer's point of view. The experiments and example usage shown at the end of the paper really help illustrate the usefulness of DBCarver. I feel the from a technical standpoint the system itself is not as impressive as they wrote it to be, but given the context of forensics and the large impact this research could have, it deserves praise. I feel the major aspect that sets DBCarver apart from other tools in the field is it's ability to operate on all file types and all databases due to it's low level processing. I would recommend this paper to my peers as It not only explains how typical file recovery systems operate but also how that theory can be applied to a database application. In my opinion, the system is ready for real life usage as there are already examples in the paper showing good performance and I would like to have a tool like this accessible to me if i ever need to recover a database.

References

[1] James Wagner et al. "Database Forensic Analysis with DBCarver". In: *8th Biennial Conference on Innovative Data Systems Research(CIDR '17)*. CIDR. 2017.