Faceted Data

Read Schmitz et al (2016) article about faceted data.

- Do you think this is a good approach to protect systems from data leakage? What are the pros and cons?

The authors implemented "faceted values technique" as a library rather than as a language extension. They looked into "previous work to provide a library consisting primarily of two monads, which track both explicit and implicit information flows. This implementation demonstrates how faceted values look in a typed context, as well as how they might be implemented as a library rather than a language feature. It also illustrates some of the subtle interactions between two monads. [Their] interpreter shows that this library can serve as a basis for other faceted value languages or as a template for further Haskell work."

Cons: to find out more

- Create a basic outline design of how you would create such a system in Python.

To check: https://www.programcreek.com/python/?CodeExample=get+facets

To read more on the subject – not clear for me.

Notes from: Schmitz et al (2016), "Faceted Dynamic Information Flow via Control and Data Monads"

"

Definition of faceted data: "Austin and Flanagan[2] introduce faceted values – values that present different behavior accord ing to the privilege of the observer – as a dynamic approach to enforce information flow policies for an untyped, imperative λ-calculus."

prevent misuse of sensitive data

property noninterference; that is, public outputs do not depend on private in puts3 . Secure multi-execution [9, 16, 23] is a relatively recent and popular informa tion flow enforcement technique. A program execution is split into two versions: the "high" execution has access to sensitive information, but may only write to private channels; the "low" execution may write to public channels, but cannot access any sensitive information. This elegant approach ensures noninterference. Faceted evaluation is a technique for simulating secure multi-execution with a single process, using special faceted values that contain both a public view and a private view of the data. With this approach, a single execution can provide many of the same guarantees that secure multi-execution provides, while achieving better performance.

Most information flow mechanisms fall into one of three categories: run-time monitors that prevent a program execution from misbehaving; static analysis techniques that analyze the whole program and reject programs that might leak sensitive information; and finally secure multi-execution, which protects sensitive information by evaluating the same program multiple times.

Faceted evaluation [2] simulates secure multi-execution by the use of spe cial faceted values, which track different views for data based on the security principals involved6 . While faceted evaluation cannot be parallelized as easily, it avoids many redundant calculations, thereby improving efficiency [2]. It also allows declassification, where private data is released to public channels. Austin et al. [3] exploit this benefit to incorporate policy-agnostic programming tech niques, allowing for the specification of more flexible policies than traditionally permitted in information flow systems.

“

What is faceting in data science?

Facets is **a useful tool for peering into your dataset** and seeing the relationships between different features as well as ensuring that there aren't missing or unexpected values in your dataset.21 Mar 2018

Visualize your data with Facets