# Engineering Report for the Scalable Compositional Static Taint Analysis for Sensitive Data Tracing on Industrial Micro-Services Research


Zexin Zhong
*University of Technology Sydney*
Sydney, Australia
*Ant Group*
Hangzhou, China
zhongzexin.zzx@antgroup.com

Jiangchao Liu
*Ant Group*
Hangzhou, China
jiangchao.ljc@antgroup.com

Diyu Wu
*Ant Group*
Hangzhou, China
wudiyu.wdy@antgroup.com

Peng Di
*Ant Group*
Hangzhou, China
dipeng.dp@antgroup.com

Yulei Sui
*University of Technology Sydney*
Sydney, Australia
yulei.sui@uts.edu.au

Alex X. Liu
*Ant Group*
Hangzhou, China
alexliu@antgroup.com

John C.S. Lui
*Chinese University of Hong Kong*
HongKong, China
cslui@cse.cuhk.edu.hk



***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**



***Index Terms*—component, formatting, style, styling, insert**


## I. Adapting to enterprise-specific frameworks: a practice in SOFA-based FinTech micro-services

Different enterprises have different frameworks based on their business scenario and technology stack. In this section, we introduce how we resolve the framework problems in SOFA architecture. We first performed the modeling task on the target application before the compositional taint analysis. Then, the modeling result of the target applications will be processed by the taint analysis module for the analysis. This automated process includes preprocessing of the SOFA framework (the financial micro-services variant framework of Spring), RPC and database operation.

**Preprocessing and Modeling.** According to our experience, the AOP, Spring Bean injections, and SOFA Bean injections features that provided by the SpringBoot [1] and SOFA [2] framework usually lead to missing knowledge about the instance type of some object or the particular code that need to be executed at some function. AOP, Spring Beans, and SOFA Beans are usually configured in configuration XML files and are used in the form of annotation in the code at the beginning or end of the function. Thus, in our case, we will first automatically scan and extract the above information from the configuration XML files based on the semantics of the SOFA-RPC and Spring XML domain language. Then we model this information in a specified format by using the code transformation and modeling supporter. These models will be loaded into the memory for the further analysis task. In our compositional taint analysis process, when the specified annotation is found, the analyzer will query the model from memory based on the annotation details and use the loaded model for the taint analysis.

**Entry Point Determination.** Because our proposed approach, which used the compositional analysis approach, does not rely on the traditional top-down analysis mode, therefore, determining the specified entry points is not required for our analysis. Therefore, distinct from other approaches based on the traditional top-down analysis mode, we do not need to identify the specified entry point before the analysis. The analyzer will randomly select an entry analysis function and execute the analysis process on all functions in the program to cover the whole program.

**Sources and Sinks Determination.** According to the successful experience of ANTaint [3] in sources and sink determination, we also assume that all the incoming data in the program will be marked as sources. These incoming data include the input parameters of the entry points, the return value of the RPC function, the interested fields or parameters provided by the developers, and the fields or objects that query from the database by the 'Select' operation. Furthermore, we assume that all the out-coming (output) data or the output functions which will expose the data to other environments or namespace will be marked as sinks. These out-coming (output) data include all the return values from the entry points, parameters for the RPC functions, the log or system printing functions, and the object or fields that will be updated to the database by the 'update', 'insert', and 'delete' operation.

## REFERENCES


[1] R. Johnson, *Spring*, 2021. [Online]. Available: https://spring.io/

[2] SOFAStack, *SOFAStack*, 2021. [Online]. Available: https://www.sofastack.tech/en/

[3] J. Wang, Y. Wu, G. Zhou, Y. Yu, Z. Guo, and Y. Xiong, "Scaling static taint analysis to industrial soa applications: A case study at alibaba," in *ESEC/FSE'20*, 2020, p. 1477–1486.