Facilitating Trust on Data through Provenance

Manolis Stamatogiannakis, Paul Groth, and Herbert Bos

VU University Amsterdam, The Netherlands,
{manolis.stamatogiannakis, p.t.groth, h.j.bos}@vu.nl

Abstract. Research on trusted computing focuses mainly on the security and integrity of the *execution environment*, from hardware components to software services. However, this is only one facet of the computation, the other being the *data*. If our goal is to produce *trusted results*, a trustworthy execution environment is not enough: we also need *trustworthy data*. Provenance of data plays a pivotal role in ascertaining trustworthiness of data. In our work, we explore how to use state-of-the-art systems techniques to capture and reconstruct provenance, thus enabling us to build trust on both newly generated and existing data.

1.1 Motivation

Provenance is a record that describes the sources and agents involved in producing a piece of data [6]. This record can be analyzed e.g. to understand if data conforms designated standards or to calculate a level of trust on the data in order to assist decision making. Thus, knowing the provenance of data can play a central role in the trust we put on them. On the other hand, not having any provenance information on our data could undermine the benefits of using a trustworthy execution environment: if we cannot trust the data we process, we will also be unable to trust the produced results.

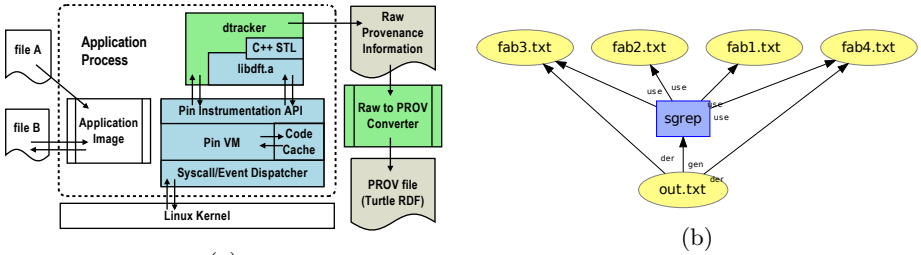
1.2 Capturing Provenance Through Dynamic Instrumentation

We have developed a new system called DataTracker[1] [7] which uses Dynamic Taint Analysis (DTA) to capture *high-fidelity provenance* from *unmodified programs*. DataTracker is based on Intel Pin[2] Dynamic Binary Instrumentation framework and a modified version of the libdft [4] library which provides a reusable framework for Dynamic Taint Analysis.

The architecture of DataTracker is depicted in Fig. 1a. Its main components are a Pin tool and a converter written in Python. The former generates provenance information in raw format which are converted to the W3C PROV format [6] by the latter. After converting to PROV, existing tools can be used to further process and visualize the provenance.

Fig. 1b shows the provenance graph produced for a simple grep-like utility. DataTracker attributes the output to only two of the four input files, which is

[1] Source code available on: http://github.com/m000/dtracker
[2] https://software.intel.com/articles/pintool



(a) (b)

Fig. 1: DataTracker architecture and output

an improvement over state of the art techniques [3,1,2]. These treat programs as *black-boxes*, tracking only program-OS interactions but not the actual use of data. Thus, they would have attributed the output to all four inputs. In addition to eliminating such cases of false-positives, DataTracker captures provenance with byte-level granularity vs. the file-level granularity offered by comparable systems.

1.3 Post-hoc Provenance Reconstruction

We plan to explore using DataTracker for the post-hoc reconstruction of provenance. Our ultimate goal is to be able to reconstruct provenance relations between files and programs stored in a disk image. We plan to achieve this by: *a*) Collecting high-fidelity provenance information from live systems. *b*) Abstracting this information to generate "provenance behavior signatures" that reflect provenance patterns generated by specific programs. *c*) Matching these signatures with the files in the disk image. It is understood that reconstructed provenance will be of less fidelity than provenance directly captured by DataTracker during execution. We can improve the quality of this provenance by later applying heuristic-based methods (e.g. [5]).

References

1. Frew, J., Metzger, D., Slaughter, P.: Automatic capture and reconstruction of computational provenance. Concurr. Comput.: Pract. & Exper. 20(5) (2008)
2. Gessiou, E., Pappas, V., Athanasopoulos, E., Keromytis, A., Ioannidis, S.: Towards a Universal Data Provenance Framework Using Dynamic Instrumentation. IFIP Advances in Information and Communication Technology, vol. 376 (2012)
3. Holland, D.A., Seltzer, M.I., Braun, U., Muniswamy-Reddy, K.K.: PASSing the provenance challenge. Concurr. Comput.: Pract. & Exper. 20(5) (2008)
4. Kemerlis, V.P., Portokalidis, G., Jee, K., Keromytis, A.D.: libdft: Practical Dynamic Data Flow Tracking for Commodity Systems. In: Proceedings of VEE'12 (2012)
5. Magliacane, S.: Reconstructing Provenance. In: Proceedings of ISWC'12. Boston, MA, USA (2012)
6. Moreau, L., Missier, P.: PROV-DM: The PROV Data Model. Recommendation REC-prov-dm-20130430, W3C (2013)
7. Stamatogiannakis, M., Groth, P., Bos, H.: Looking Inside the Black-Box: Capturing Data Provenance using Dynamic Instrumentation. In: Proceedings of IPAW'14. Cologne, Germany (2014)