Augmenting Provenance Records with Trust in Enterprise Applications

Oshani Seneviratne and Ken Beckett

Oracle Corporation

Abstract

As more and more data get migrated to the cloud and various enterprise applications are accessing such data in increasing volumes, the consumers of these applications are demanding authoritative information as to where the data is coming from, and how it has been modified. While the Provenance Recommendation [1] has provided a very good mechanism to represent provenance information of a given artifact, it lacks mechanisms to guarantee those provenance records have not been tampered with, and are trustworthy. This position paper elicits some of the missing features in the current Provenance recommendation to enable trust, and illustrates how by having those features, Enterprise software vendors like Oracle can build applications that use provenance in a more meaningful way to assert trustworthiness of the provenance records by providing strong integrity and trust assurances.

Motivation

In the 1980s, a British conman named John Drewe gained access to the Letter Archives of the Institute of the Contemporary Arts in London, and used the Institute's official stationary to engineer false provenance records to forged artwork that he then sold for millions of British pounds. These false provenance records wreaked havoc, as it took years for the Institute to purge the false information off of their archival records, as well as to locate the historically accurate provenance records that Drewe had replaced in the archival records [2]. Since digital records of provenance are far easier to modify than the provenance records of artwork that Drewe was able to modify, we can expect to see similar or worse situations with respect to digital artifacts.

As a motivating example in the enterprise, consider the regulatory and legal considerations such as the US Sarbanes Oxley Act, which imposes strict punitive action to Financial Officers who sign their names on incorrect corporate financial statements. As a result, the Financial Officers are interested in provenance records that verify the path that a document took during its development, the data that contributed to the report, and the people who worked on the financial report. Such provenance records can be used to prove the accuracy and authenticity of the report, but only if it can be guaranteed that the provenance record itself has not been tampered with. Therefore, having an immutable provenance record for digital artifacts in the enterprise is crucial in many such scenarios.

Our Requirements

Oracle Human Capital Management (HCM) software [3] connects to different data sources for many Work Life product offerings. These data sources include, for example, SFTP endpoints (work) and Facebook (life). The software also lets the user define "flows" that involve transformations to the data. Similar to the use case mentioned earlier, our motivations for augmenting provenance records with trust are two-fold: (1) the input data to HCM Connect may have already have some provenance records, and we want to verify that those have not been modified out of band, and (2) the provenance of the output data from our HCM Connect flows should be protected against any subsequent provenance forgery in order to preserve the integrity of the data.

Our Proposal

The provenance recommendation identifies three main classes for provenance concepts in describing provenance, namely Entity, Activity and Agent. In many Enterprise software application scenarios, those are sufficient in identifying the necessary provenance objects. However, the recommendation lacks the formulation of the minimal set of provenance concepts that would identify a valid provenance description. For example, given a prov:Entity, what properties must be included as predicates for it to be sufficiently described as an entity with provenance? Even though OWL, and thus the Provenance Ontology, follows the open world assumption, in many enterprise applications, it is very beneficial to close the world, and identify the minimal set of properties required in describing provenance objects. With such bounded provenance in place, we can then generate a SHA512 hash for a given provenance record as proof of authenticity of the provenance record. This can then be published at the source of the data for verification by the consumers of the data. The source server may provide an API to query for such hashes, or it may simply publish them on a webpage. Since the data objects represented in the provenance records have a unique identifier, i.e. URI, the lookup for the corresponding hash will be straightforward. Similarly, when enterprise processes produce data using some initial input data that already has an immutable provenance record, a new provenance record will be generated on the newly minted data item. The SHA512 hash generated for this new data item will be based on a non-empty time-ordered sequence of previous provenance hashes as well as the minimal set of properties as stipulated by the recommendation attached to the new data item. This creates a Provenance Proof Chain that will preserve the lineage of the provenance record and thus assert the trustworthiness of the data. Although there can be multiple derivative paths for a given digital artifact, this methodology will prevent out of order actors from colluding in tampering the provenance records.

Related Work

Blockchains seem to be a very attractive approach in maintaining immutable records for secure traceability and certifications [4]. However, there are many practical challenges when applying a blockchain to enterprise software, such as ownership of the data, security of the data, and costs of maintaining a public ledger to support the blockchain infrastructure. However, our proposed method guarantees that the ownership of the data is retained with the data provider, costs of maintaining the data and the provenance records are clear cut, does not require any such infrastructure changes, no need for all the stakeholders of the data to be involved in running the shared blockchain infrastructure, and more importantly, it can be implemented with the technologies available today with minimum disruption to existing enterprise applications.

Conclusion

While significant effort has been expended on how to represent, collect, store and query provenance information, issues surrounding how to trust the provenance data was not explored in the Provenance recommendation to a satisfactory level. While we understand making provenance records trustworthy can be challenging, we believe the first step is to guarantee completeness of the provenance record. Thus, our first position is to introduce the minimal set for a provenance representation in the Provenance recommendation. Then, in order verify that a provenance record has not been tampered with to assert trustworthiness of the data, the data providers can utilize the low hanging fruit of publishing the SHA512 hashes of the provenance record. This would enable the consumers of the data to compare the provenance record that includes this minimal set to assert trustworthiness of the data they are consuming. Thus, our second position is to introduce a protocol for generating Provenance Proof Chains that is non-repudiable and also recommend that data providers make them available along with the data.

References

[1] “Prov Overview”, W3C, 30 April 2013, https://www.w3.org/TR/prov-overview/
[2]"Businessman John Drewe jailed for eight years following fraud trial at Norwich crown court", EDP 24 (UK), 12 March 2012. http://www.edp24.co.uk/news/crime/businessman_john_drewe_jailed_for_eight_years_following_fraud_trial_at_norwich_crown_court_1_1234548
[3] Oracle Human Capital Management, https://www.oracle.com/applications/human-capital-management/index.html
[4] “Blockchain: the solution for transparency in product supply chains”, Project Provenance Ltd. https://www.provenance.org/whitepaper