The Need for Secure Logging

Logging is one of the more important administration tools of a complex IT system such as a cloud. The objective of such process is to track the events that happen in the system. Since the logs may be used to rebuild the past history of a system (e.g. *after-the-facts* analysis in forensics activity) the logging process is frequently victim of cyber-attacks. In cloud computing, users entrust their data and processes to the Cloud Service Providers (CSP). In this context, the logs could be used to attest the CSP activity, for instance, to identify possible abuses related to the CSP privileged position (e.g. the system administrators). In order to consider logs as valid event/action evidence, it is necessary to provide procedures to attest their security in terms of **integrity** and **authenticity**. The **LogService** is a cloud oriented logging service that has been designed in order to support different secure logging schemes. The base version of the LogService implements the secure logging scheme proposed by Schneier and Kelsey [1] that makes possible the generation of secure log entries whose integrity and authenticity can be **cryptographically verifiable** and which are characterized by a **per log entry access control**.

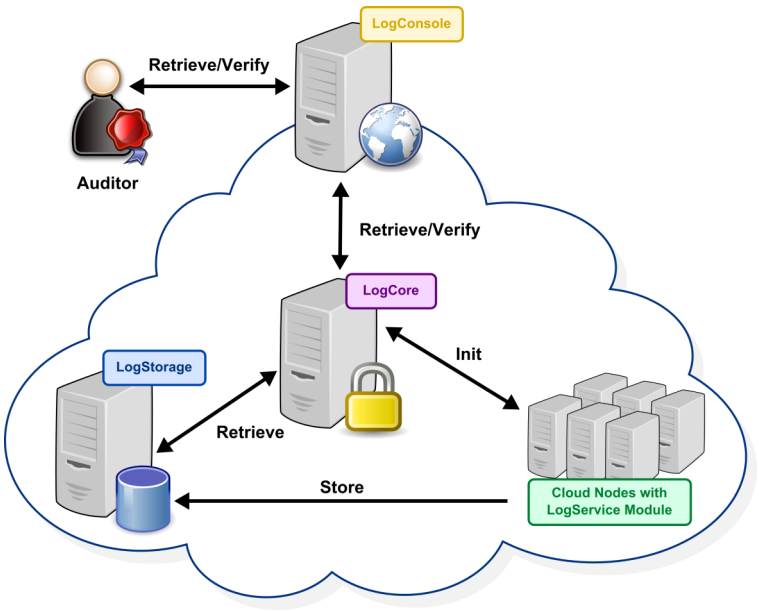

Figure 1: LogService Components

High-level Description of the LogService

The figure above depicts the relations among the LogService components whose definition is presented in the following.

The **LogCore** is the trusted entity (server in a secure location) that collects all the cryptographic material necessary to perform the logs verification.

The **LogStorage** is a storage entity which has good *resiliency* and *availability* features.

The **LogService Module** is a software module that must be included in the applications in order to enable secure logging features.

The **LogConsole** is a web based management console that could be used by the **Auditor** to access and verify the logs.

The generation of secure log entries must be preceded by the **initialization** of a new logging session. During this phase, the LogCore collects all the cryptographic material necessary to perform the verification. Afterwards, the log entries can be created and periodically **stored** on the LogStorage. The storage frequency may depends, for instance, by the number of created log entries or by by the elapsing of a certain period of time. At a certain point, the Auditor wishes to **retrieve** and **verify** a logging session. Such request is forwarded by the LogConsole to the LogCore which **retrieves** the logs from the LogStorage, verifies them and finally sends the result and the data to the LogConsole and hence, to the Auditor.

TClouds — Trustworthy Clouds

SECURE LOGGING

www.tclouds-project.eu

Details about LogService Internals

The LogService is based on the **libseclog** library. Such library, developed by TClouds, provides a set of APIs that allow the generation and the verification of secure log entries using different logging schemes. The current release of the library (v0.1) supports only the Schneier and Kelsey scheme [1] but the support for additional logging schemes can be easily extended thanks to the driver-like management of the logging scheme engines.

The library is implemented in pure C and provides the bindings for Python. While the LogService Module is implemented as Python Logging Handler [2], other LogService components as LogCore, LogStorage and LogConsole, are deployed as web services using the framework Tornado (www.tornadoweb.org) and their features are accessible via RESTful interface.

To accomplish the security requirements of a secure web application, all data and communication among the LogService components, are HTTPS based. Moreover, each component executes clients authentication and filtering based on X509 certificates.

OpenStack and TClouds Integration

The Python Logging Handler that represents the **LogService Module** has been included in the Python class that manages the logging process within OpenStack. Such module can be optionally enabled and configured through the main OpenStack configuration file (both a new configuration section and directives have been defined). The **LogConsole** has been merged in the OpenStack dashboard. The **LogStorage** relies on **Cheap-BFT** subsystem for the availability and the resiliency of the storage. Finally, the **LogCore** runs as independent web service in order to be accessible from all layers of the TClouds architecture.

References

[1] Schneier, Bruce, and John Kelsey. "Secure audit logs to support computer forensics." ACM Transactions on Information and System Security (TISSEC) 2.2 (1999): 159-176.

[2] Python Logging Handlers, http://docs.python.org/2/library/logging.handlers.html

Where To Find LogService?

http://security.polito.it/secure-logging/libseclog/

Further Information

Further information about LogService can be found under Deliverable „D2.1.2—Preliminary Description of Mechanisms and Components for Single Trusted Clouds".

Disclaimer

The TClouds project has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement number ICT-257243.

TClouds at a glance

Project number:
257243

TClouds mission:

- Develop an advanced cloud infrastructure that delivers computing and storage with a new level of security, privacy, and resilience.
- Change the perceptions of cloud computing by demonstrating the prototype infrastructure in socially significant application areas.

Project start:
01.10.2010

Project duration:
3 years

Total costs:
EUR 10.536.129

EC contribution:
EUR 7.500.000

Consortium:
14 partners from 7 different countries.

Project Coordinator:
Dr. Klaus-Michael Koch
coordination@tclouds-project.eu

Technical Leader:
Dr. Christian Cachin
cca@zurich.ibm.com

Project website:
www.tclouds-project.eu