ABLS

An Attribute-Based Logging System for the Cloud

User Manual

Christopher A. Wood
`caw4567@rit.edu`

January 11, 2013

Abstract

User-based non-repudiation is an increasingly important property of cloud-based applica-tions. It provides irrefutable evidence that ties system behavior to specific users, thus enabling strict enforcement of organizational security policies. System logs are typically used as the basis for this property. Thus, the effectiveness of system audits based on log files reduces to the problem of maintaining the integrity and confidentiality of log files. In this project, we study the problem of building secure log files. We investigate the benefits of ciphertext-policy attribute-based encryption (CP-ABE) to solve a variety of log design issues. In addition, we also present the architecture and a preliminary analysis for a proof-of-concept system that fulfills the confidentiality and integrity requirements for a secure log.

Contents

Chapter 1

Bootstrapping

ABLS comes packaged with a set of configuration scripts and SQL files that initialize the database to a clean state. These files are included in the DatabaseModule directory that comes packaged with ABLS, as shown below:

```
ABLS
 ├── main - main executable
 ├── LoggerModule
 ├── PolicyEngineModule
 ├── AuditModule
 ├── Common
 ├── TestModule
 └── DatabaseModule
      ├── bootstrap - bash script
      └── bootstrap SQL files
```

In order to configure a new ABLS instance to be run on a server in development mode, one must run the following commands from the root ABLS directory.

```
$> ./DatabaseModule/bootstrap
$> python main.py -c
```

The first bootstrap script will wipe the database files and configure them for use with an ABLS instance. This is the script that should be modified if the user wants to change the physical location of each database server. The second command will tell the main ABLS executable script to “configure” the database with some fake data for testing purposes. As such, this should only be used when configuring ABLS for development tasks.

Once complete, the user should then run the following command from the root ABLS directory to start an ABLS instance on the local host.

```
$> python main.py -s
```

If one wants to deploy an ABLS instance in production mode, they should only run the main executable with the "-s" flag, not the "-c" flag. Also, for convenience, these two flags can be combined during the bootstrapping process, as shown below.

```
$> ./DatabaseModule/bootstrap
$> python main.py −c −s
```