Yasin YILDIZ

Design Document

The project consists of two main modules:

- Server script
  - Libraries/Packages/etc.
    - Python2.7
      - Development language
    - MySQL Database
      - Selected RDMS
    - python-lxml
      - xml construction/parsing
    - python-pycrypto
      - encryption/decryption
    - python-paramiko
      - SSH
      - SFTP
    - python-mysql-connector
      - database client
    - sqlalchemy/sqlalchemy_utils
      - ORM
  - Submodules
    - Database handler
    - Mail Server handler
    - Client handler
- Client script
  - Libraries/Packages/etc.
    - Python2.7
      - Development language
    - psutil
      - retrieval of statistics (cpu usage, memory usage, uptime)
      - platform independent
    - lxml
      - xml construction/parsing
    - pycrypto
      - encryption/decryption
    - pypiwin32 (Windows only)
      - retrieval of Windows Security Event logs

Yasin YILDIZ

Requirement Analysis

Communication

It is obvious that there is a need of a communication channel between client and server scripts to transfer statistics and log data from client machine to server machine.

Considering the requirement of connection to a remote machine via SSH and possible SFTP on top of SSH, a binary file created by the client and containing the client output can be downloaded/read by the server. SSH and SFTP protocols deployed via Python paramiko package.

Data

The structure of the data that will be transferred from client machine to server machine shall be designed in a way that is providing an extensibility for increasing number of statistics (possibly in the feature) and providing encryption for security.

XML is selected as the data container, and AES is selected for encryption. As a result, statistics and logs on the client shall be encapsulated in XML, and then shall be encrypted by AES algorithm with a key and initialization vector known by both server and client scripts.

Database

Server needs to keep data (statistics and logs) in a structured way, which addresses the need of a database and of course a database handler. The handler shall be responsible of only connection and insertion because consuming the data is out of the scope of the project.

Two tables are designed to store data from client. One is for statistics data such as cpu usage, memory usage and uptime. The other is for Windows Security logs. Both tables are designed in an extensible way. For example, a new type of statistics (for example, cpu idle time) can be stored in the same table. Similarly, Windows System and/or Application logs can be stored in the table that Security logs are stored.

SQLAlchemy is used to take the advantage of ORM.

Mail

It is required to send a mail with a proper content to the given receiver in case of exceeding the limit which are also provided to the server script. So, a mail server handler shall be considered within the design.

A simple wrapper is designed for mailing with connect, send and disconnect functions.

Configuration

There are many parameters that the server script needs to know

- list of client credentials
- database credentials
- mail server credentials

To provide all these information in a structured way, a configuration file shall be designed.

An extensible xml format is decided for server script configuration to provide required information.

Logging

To keep track of the execution, and error/success status of the client script runs as well as database and mail server connection status, a logging mechanism shall be designed.

Python logging module is integrated with a proper log format.

Server Script

It is designed by using Object-Oriented Programming paradigm such that:

- a Main Controller class
  - reads configuration
  - instantiates other classes
  - manage all objects
  - provides multi-threading for client handling: client scripts can be executed in parallel to improve performance
- Database handler
  - handles database operations
    - insert statistics and/or logs
- Mail server handler
  - handles database operations
    - connect
    - disconnect
    - send
- Client handler
  - handles operations on remote machine
    - connect (SSH)
    - upload file (SFTP via SSH)
    - download file (SFTP via SSH)
    - retrieve data from the output of the client script(stdout via SSH)
  - implements Observer Design Pattern to notify the observer when script execution completed.

Yasin YILDIZ

Client script

Client script is designed by using Functional Programming paradigm instead of OO.

- retrieval of statistics and/or logs
- encapsulating them in xml
- encryption

Architecture