Event Server “Cheat Sheet”

Database: Event Server use Hibernate for persistent mapping to a relational db.
Logging: Logging server options can be configured in es/config/log4j-config.xml file.

Edge nodes: The Event Server running at Edge nodes have two basic configurations “Process or Forward”. This override the Group Policies

Script Level: The scripts are served to the gateway or server following Group Policies, you can make Policies just for scripts and include gateway nodes on them, one or many.
Scripts are stored in Active Directory schema and checked for modification by es server periodically.

Beyond Group Policies: In situations that Group Policies of Active Directory don’t fit your requirements, or you use OpenLDAP, you can use “Dynamic Attributes”, that are variable data that you store directly into PostgreSQL and access from LUA. Be careful with this, actually you can override all the secure data architecture proposed to customer.

Speed by Filter: In order to achieve efficiency you can filter events before processing them in LUA by using aigsgPolicyAllowEvent, aigsgPolicyDenyEvent attributes.

Meaningful data restriction: All objects contained in LDAP are transformed into UUIDs, generated by ES itself, so any content of an Edge Gateway or Server, lacks meaning without all the architecture running.

Sensordaemon: Event Server receive messages in protobuffers format sent by “any” sensordaemon that can run on a gateway, box or even smart sensor (Intel Quark Sensor’s), depending on the situation the sensordaemon can make an interpretation of data or pass it forward to be interpreted by LUA script, based on properties of object in AD. This make ES Future Compatible with any sensor or protocol on the market.

Structure

Internal Dependencies

Protobuf: ES use protobuf to serialize structured data to transport over network. Protobuf has two parts, core library and protoc, the version of protobuf core library and protoc ***must be matched*** in order to compile ES.

Basically, we define data structures in *.proto files then use protoc to parse *.proto and generate source code (java or C++).

Hibernate: ES use hibernate to interact with database, Hibernate is a well-know ORM framework for Java.

In database level, we see tables, rows, relationships, etc. Hibernate maps tables to classes, fields to class’ properties, rows to class instances, so in ORM level, we see data as classes, instances instead of tables, rows, relationships. Hibernate also take care of low level DB connection, transaction, etc, so that we just concern to application logic.

External Dependencies

LDAP

RabbitMQ
PosgreSQL

Objects Retrieval

An option to retrieve objects via libmclient and ObjGetReq
https://github.com/AIGSG/es-proto/blob/master/devapi/object.proto#L48
It returns all info about specific elements connected to GW on DevApi level. On user api level it returns all objects that user is able to access.

To retrive events via UserApi EventQueryReq message
https://github.com/AIGSG/es-proto/blob/master/userapi/event.proto#L57
using libmclient it is possible to set different operators for events compare.

Use libmclient build a listener, that will be triggered on events and save them to db that is needed