Enabling SASL External and Client Certificates in OMF

We have modified OMF 5.2 code to enable OMF components securely communicate with each other using the GENI Messaging Service, which uses the modified Openfire XMPP server. We have also demonstrated this via a simple OMF experiment running on an ORCA slice where each OMF components runs on a different VM and communicate securely over geni-imf-dev.renci.org Openfire XMPP server.

OMF Components

OMF consists of three components: AM (Aggregate Manager), EC (Experiment Controller), RC (Resource Controller). Each of these components may run on different machines or they can run on the same machine. AM is the component where the information about resource nodes (such as IP address or testbed domains) are stored at a database. We refer to a “resource node” as a node that the experiment is going to run on. AM is also expected to create each resource node’s IP address as a specific PubSub node to XMPP server (such as “/Domain/System/10.0.2.5”) so that these resource nodes can be referred later. This registration is done before any experiment starts. As of OMF version 5.2, after the registration, AM is not involved in XMPP communication later, instead it responds to HTTP queries. The next component is EC, which sets up and manages the experiment. Users run their experiment definition script at EC component. At the beginning of experiment, EC queries AM (using HTTP) to gather resource nodes of the desired testbed domain. Having got a response, EC creates experiment specific PubSub nodes (such as “/Domain/Session/exp_testbed_datatime/”) on the XMPP server and notifies the resource nodes by publishing this information on resource node specific PubSub nodes (such as /Domain/System/10.0.2.5). Resource nodes now can subscribe to the experiment specific PubSub nodes and thereon all experiment related communication goes through this new PubSub nodes. RC is the other OMF component installed on resource nodes and runs experiment script commands on this node. It communicates to EC using XMPP server and each resource node is subscribed to PubSub node (which is created by AM earlier) and listens for an experiment message to come. OMF is entirely written in Ruby and uses XMPP4r library for XMPP communications.

OMF Security

All the components mentioned above communicate over XMPP server and the communication is encrypted through TLS. Components were authenticated via username and password using SASL Plain authentication. We added SASL External authentication functionality on XMPP4r library and open TLS socket with GENI Certified Client Certificates. Now the client does not need to present a username or password but client certificate will be enough to authenticate and authorize the component. Also, each component in OMF first tries to register an account on the fly and we have removed this, because these accounts must already have created in the XMPP server to initiate communication with GENI Messaging Service.

GEC13 Demo

We have created an ORCA slice using Flukes GUI with three VMs: one for each component. These components were communicating via geni-imf-dev.renci.org Openfire XMPP server. Our experiment definition was simply sending UDP packets (using OTG: Orbit Traffic Generator) for 5 seconds. The installation document shows the step about how to install OMF and modify the code to enable this experiment.