The software talks to the MySQL database on COGITO via Php scripts which are accessed via a web server running on the COGITO head node. Those php scrips contain the login information into the MySQL database. This information is not visible to the user, however knoweldge of the URLs of these scripts gives any person unrestricted access to whatever these scripts do. All data that GenAMap uses is in database 'geneassoc', although it is theoretically possible to set up users which store their data in other databases (though this would necessitate a number of changes which I am not fully aware of ATM). In there, two tables store the login information of teams and users. These are called 'team' and 'user'. There are the following php scripts:

* genamapgetteams: Returns the list of team names in table 'team'
* genamapcheckteam: Needs parameters 'team' / 'passcode'. Checks whether these match up with a team registered in the 'team' table.
* Genamapgetusers: Needs parameter 'username'. Checks whether a given user name is in use
* genamapcreateteam: Needs parameters 'team' / 'passcode'. Creates a new team in the 'team' table with the given team names and passcode.
* genamapcreateuser: Needs user associated input parms (including user name / password). Creates a new user associated to some team in the 'user' table.
* genamapconn: Needs parameters 'username' / 'password' / 'query'. Advanced script for running arbitrary queries on user data files. Note this script requires a valid username and password to operate on that users files and is also used to verify whether a username / password is correct.
* Genamaptrans: ?

The software maintains two serialized files: A copy of GenAMapDist::realdata.Data1 and a copy of GenAMapData::datamodel.model. Data1 stores exactly the last username and password entered. When GenAMap is launched, it will try to load a copy of Data1. If it succeeds and it contains a username and password that can be verified via 'genamapconn.php', it will load its copy of 'model'. Otherwise, it will create an instance of 'GenAMapData::Control.DatabaseSetup', which is a GUI that allows account creation and login. Its three functions, (i) creating teams, (ii) creating usernames and (iii) logging in are ultimately executed through the php scripts. 'genamapgetteams' is used to initialize the drop down menu in the GUI. When a new team is created, 'genamapcreateteam' is called. When a new user is created, 'genamapgetusers' is used to verify the username is not in use and 'genamapcheckteam' is used to verify the user knows the password for the team he is trying to register for. Then 'genamapcreateuser' is used to create the user. Finally, when the 'login' button is pressed, the login GUI disappears and 'genamapconn' is used to verify the login data. If the verification fails, the process will end and the main GUI will not be launched (without error message). If the verification succeeds, things will proceed just as if the verification of the serialized contents of 'Data1' had succeeded. Also, the newly entered data is imprinted on 'Data1' and serialized for future reference. From now on, whenever a query is run on data, the login information will be pulled from 'Data1' just before 'genamapconn.php' is called in function 'GenAMapDist::realdata.DataManager.runquery'.

From a security standpoint, the php URLs can be observed in the clear in the network. Since all users must submit their user / team names / passwords to COGITO to begin work, these could be intercepted and give anyone the ability to resuse that data to obtain the full power of the php scripts for that user. This power even exceeds what is possible for the user from the GUI, because the java code underlying the GUI only submits certain queries to 'genamapconn.php'. However this script supports arbitrary queries on the entire 'geneassoc' database with ANY users name / passoword. Hence, an observer of network traffic has full control over the geneassoc database, including usernames and passwords.