Lafayette College Grouper Page

Grouper Pilot Use Case: VPN access group population

Lafayette College began the process (Oct. 2014) of a Grouper pilot using a test use case of VPN group management.

Employees at Lafayette are able to access the College's network remotely via VPN. Some contractors and students are also granted ad-hoc access based on work requirements, faculty sponsorship, etc.

Pre-Grouper Access Management for VPN Authorization

VPN access was controlled via LDAP group membership. Employee membership in this group was handled automatically by custom provisioning and deprovisioning processes. Temporary employees, contractors, and students were not covered by these processes, and those requests were routed through the College's IAM team within ITS.

Leveraging Grouper and Its Benefits for VPN Access Management

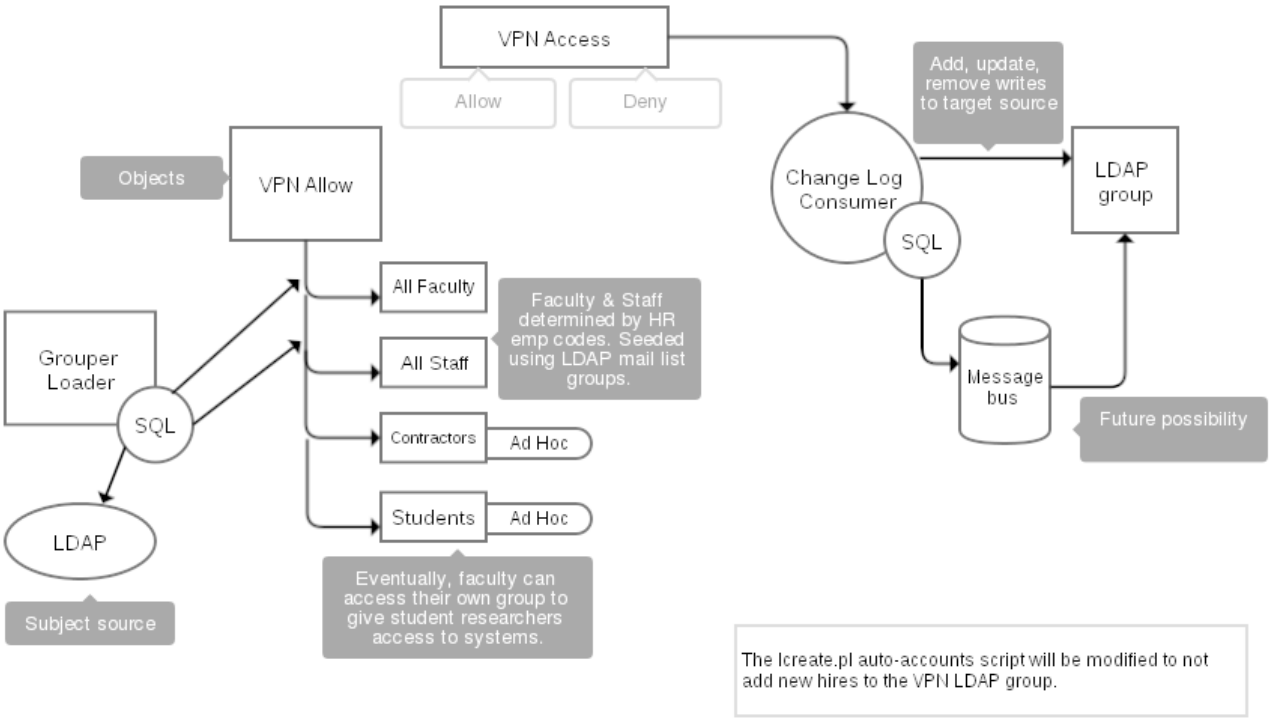
Lafayette College's IAM team created data-driven groups in Grouper. Those groups are populated based on employee class codes that are maintained in Banner. Those reference groups are included in a composite group which in turn is used to provision the LDAP group used to control VPN access.

Two additional ad-hoc groups for contractors and students factored into the Grouper pilot use case. Those groups will eventually be managed by end users that are authorized to grant VPN access to those cohorts.

Nothing changed with respect to the way in which authorization to the VPN happened at a technical level. Grouper writes memberships to the VPN LDAP group which is still used for the control group. A major improvement that Grouper brought to the process is that decision makers now have direct control over VPN access rather than having to route requests through the IAM team.

Architecture

Grouper at Lafayette College is deployed as 2 components-- the Grouper UI and the Grouper API (aka Grouper Daemon). The Grouper UI is deployed in a manner consistent with other web-based deployments at Lafayette. The Grouper API components require elevated access to alter LDAP data, so they are deployed in a hardened network. Banner reference data is exported to LDAP on a nightly basis, and the Grouper Loader service is used to sync that data into Grouper on a nightly schedule. A separate instance of the Grouper Shell runs as a change log consumer. It monitors membership changes in Grouper and reports them to an LDAP provisioning process. The LDAP provisioning process accumulates membership changes and writes them in batches to the Lafayette College LDAP DIT at 30 second intervals.



Grouper UI

The Grouper UI at Lafayette is deployed behind an NGINX proxy on a separate host.

Authentication to the web UI is managed by an Apache v2.2 proxy with mod_auth_cas enabled running on the same host as the servlet container (Tomcat) that hosts the Grouper UI. When an unauthorized request is made for a Grouper resource, the CAS authenticating proxy redirects the request to the College's CAS SSO service. Once authenticated at this service, the request is redirected back to the authenticating proxy with a service ticket appended to the query parameters of the request. The mod_auth_cas module is able to validate this ticket with the College's CAS service over a back channel (server to server) and establish an authenticated session that includes the authenticated identity's username. This request and all subsequent requests for this session are now recognized as authenticated by the proxy and passed along to the servlet container with the REMOTE_USER environment variable set to the authenticated username. The Grouper UI retrieves the username from this environment variable and matches it to a subject ID. This subject is the currently logged on user from Grouper's perspective.

This process works well with other authenticating proxies as well. During initial research into Lafayette's Grouper pilot, the Grouper UI was successfully deployed behind a variety of authenticating proxies, including Apache + Basic Auth to a local password file, Apache + mod_auth_ldap, and the Twisted CAS proxy. Ultimately, we selected Apache + mod_auth_cas because it was a good fit with the College's current SSO deployment architecture.

Grouper Loader

The Grouper Loader runs continuously as a daemon process on College's Grouper API node. Several Grouper groups are linked to the College's LDAP DIT. The loader pulls memberships for these reference groups into Grouper nightly, as per the Quartz cron settings.

Change Log Consumer

The change log consumer was written using Bill Thompson's "Shell Wrappers for Grouper". If you are unfamiliar with the project, it leverages scripting languages that compile to JVM bytecode to wrap the Grouper Shell. For interactive sessions, this adds lots of extras found in modern REPLs like readline support and command history. It also allows non-Java experts to make good use of the Grouper API from more familiar programming environments (e.g. Groovy, Clojure, Jython). Lafayette's change log consumer runs as a daemon and tracks membership changes in Grouper. It sends these changes to a custom provisioning process that batches the changes. Batches are synced to the Lafayette LDAP DIT at 30 second intervals.

Architectural Update - July 2015

The message bus in the above diagram is no longer a future possibility-- it is now reality. The change log consumer has been updated into separate parts. One piece reads the change log from Grouper and writes the messages to an AMQP message exchange. The messages are tagged with routing keys that are based on the provisioner type (at this point there is exactly one provisioner type-- LDAP). The second piece is the provisioners that read messages from the queues and act on them.

Change Log Consumer

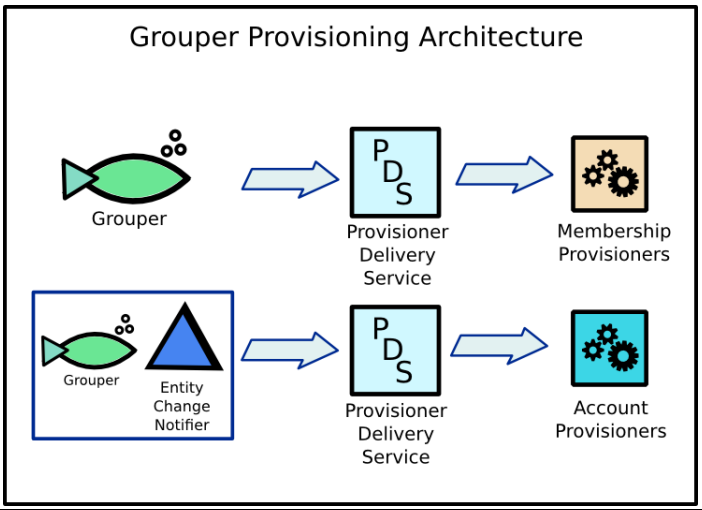
This piece is still implemented as a script using the GSH shell wrappers for Grouper (see above). The main modification is that the messages are delivered to an AMQP exchange (RabbitMQ) rather than directly to the provisioner via custom protocol as above.

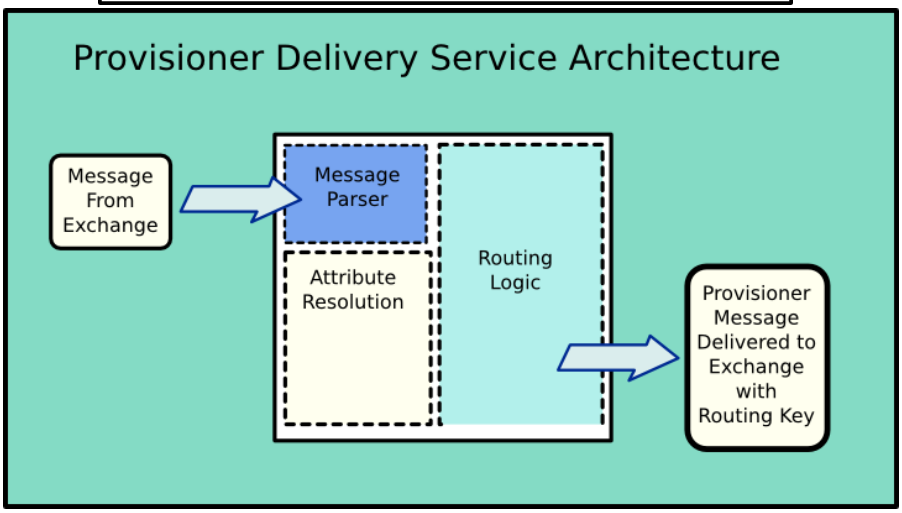
LDAP Provisioners

There are now 2 LDAP provisioners. One provisions LDAP groups with user entires, and the other provisions LDAP users with Group entries. Lafayette's LDAP DIT (OpenLDAP) is not configured to synchronize groups and users automatically, so parity is maintained via application code. In this case, the same message is delivered to **both** provisioners so that LDAP accounts and groups have the same information.

Architectural Update - May 2016

In Lafayette's development environment, there have been some interesting and exciting changes made to the Grouper provisioning architecture. The diagrams below tries to capture these changes pictorially. One significant change is that the message routing logic has been removed from the event sources (e.g. the Grouper change log consumer). Event sources send their messages to an exchange that delivers the message to a Provisioner Delivery Service (PSD). The PSD parses the messages it receives and determines routing keys to add to the output messages it delivers to a provisioner. The routing logic is based mostly on the groups related to the message. For example, a message about a member being added to the VPN group could be tagged with a "vpn" field in its routing key.





Additionally, this architecture recognizes a difference between *membership provisioners* and *account provisioners*. The College's LDAP provisioners are prime examples of membership provisioners. They only care about subject's relationships to groups. Viewed another way, membership provisioners apply "tags" to subjects in the systems they provision.

Account provisioners, on the other hand, are concerned with identity data associated with a subject. For example, Lafayette uses cloud based services that require users to have accounts within the vendor system. Typical provisioning strategies involve data extracts and nightly file transfers. However, some vendors provide REST APIs for manipulating accounts. Grouper can work in conjunction with an account provisioner to provision and deprovision accounts based on memberships/tags applied to a subject. Because the provisioning process may require identity data other than the subject identifier (e. g. given name, surname, email), the PSD has the ability to resolve subject attributes from external sources. The resolved attributes are included in the messages delivered to account provisioners.

This allows a new event source, an "Entity Change Notifier", to alert account provisioners when the attributes of a subject have changed. Interested account provisioners can update the identity data for a subject in the systems they target.