August Update

We have continued to work on the new IdP plugin framework and are nearing feature completion on several of the first round of plugins we're building, which I mentioned in last month's update. The new Duo plugin in particular will require some time to polish and complete but the bulk of the new work is done.

The major remaining tasks center around completing the plugin framework itself and testing all of that, plus significant changes to some of the IdP's core configuration approaches, to address issues we've identified that make it harder for plugins to "slot themselves" into the system effectively without requiring lots of manual additions. We want to minimize "boilerplate" sorts of changes such that only changes to defaults will require any additions.

As an example, right now adding support for a new login flow via a plugin would require editing *general-authn.xml* to add a descriptor bean for the new flow. That's going to be eliminated so that default settings for the flow just get applied. The act of deploying the plugin will suffice to install the defaults.

This is part of a larger effort for V4.1 to refactor and improve the configuration. This is all in the vein of "we didn't have time to make it simpler" until now. In some ways, doing this as part of V4.1 is actually a better choice than V4.0 because it forces us to remain compatible as we make these changes, which in turn minimizes the pain of the V4.0 (and V4.1) upgrade process. Using major upgrades can be a crutch to make breaking changes that may be easier to implement but cause pain to others. I prefer we take the path of creating more work for us and less for everybody else.

The configuration improvements have been happening in a staged fashion:

1. First, we took on the task of eliminating the system/ folder. This is a long story but suffice to say it's hard to do, requires copying code out of Spring, and won't be 100% gone until V5, but it is essentially unneeded now.
2. Next we need to re-engineer some of the internals as I described above, which will allow plugins to be designed more effectively and reduce the need for new configuration of default settings.
3. Then we will be retrofitting some of these changes back into the original code and in the process hopefully removing some of the extraneous configuration files from the shipping defaults. This won't help much for upgrades, but new installs will hopefully include fewer files and the system's apparent, if not actual, complexity will be reduced. This may be too-little, too-late but we'll see.

One of the features we will need to add to make this a reality is the ability to "populate" a new default configuration for a feature so that it can be generated by the deployer using a command rather than having to pre-populate all of the possible settings the way we do now. This is also part of the plan for the plugin framework, modeled on many similar systems such as Jetty's.

Lastly we've added a new team member who will be starting to get familiar with the SP build process and eventually automating the packaging we do for it, which will help reduce the amount of time new releases will take. With OpenSSL 3.0 shipping at the end of this year, we hope to be able to exercise a new process to release a compatible version early next year.