2012-01-16 Skype Chat re Authn

[1/16/2012 12:36:06 PM] Daniel W. Davis: The notes from the DfR meeting are posted on the meeting page. https://wiki.duraspace.org/display/DFR/2013-1-16+DfR+Technical+Meeting

[1/16/2012 1:00:03 PM] Andrew Woods: Chris, to be clear, the idea that you were raising in the meeting earlier was the possibility of the duracloud service impls in particular, and all three ECP clients depicted inside the duracloud-server in general being candidates for interacting with tomcat directly, versus those clients coming through Apache. In this way authN would be handled via basic-auth instead of shib? Each of the duracloud webapps has its own security context and requires its own authN/Z. Unless those depicted clients called durastore/duraservice/durareport via direct java calls (not even through tomcat) it would seem that they would still need to authenticate. Were you seeing/suggesting something different?

[1/16/2012 1:11:47 PM] Chris Wilper: If mod_shib is being used for AuthN at the Apache level, then the REMOTE_USER (user id) http header is all that's required to "authenticate" to Tomcat, right? So as you mentioned last week, it's important that Tomcat isn't exposed directly to the general public because spoofing would be trivial. But you can use that characteristic to your advantage by having your underlying impl code directly call Tomcat, passing in the already-known REMOTE_USER in an HTTP request header.

[1/16/2012 1:13:55 PM] Chris Wilper: I'm not positive this approach would work, but it looks like if it does, it could be a away to "cut to the chase" when making calls back to DuraCloud from the various services.

[1/16/2012 1:15:36 PM] Brad McLean: As an optimization, or as the only way for calls to come back from a service? Would there ever be a case, for example, where my duracloud service calls your duracloud instance (where the credentials would presumably be different)?

[1/16/2012 1:16:15 PM] Brad McLean: +1 for the optimization, btw, just trying to frame the cases.

[1/16/2012 1:18:46 PM | Edited 1:19:50 PM] Daniel W. Davis: Wouldn't any services that are not exposed to the public network be covered by a single-signon framework whether home grown or using someones elses, even if they are pure service with no UI.

[1/16/2012 1:20:26 PM] Chris Wilper: Yes, as an optimization. Certainly apps could choose to come in at the Apache level, and therefore be required to be more shib-savvy. Though I guess I assumed all callbacks that service impls do are done on behalf of the same user that the service was invoked as.

[1/16/2012 1:26:54 PM] Brad McLean: My thought experiment was along the lines of "What if CloudSync were a duracloud service, and I was using it to sync with another user's duracloud?" - and that led to me trying to think about what the code on the calling side would look like - would it have available the capability to generate a full shib call with a new set of credentials, or would it only be able to pass along it's REMOTE_USER value? It isn't clear to me that this is an important use case, but it would be precluded if we made the assumption that the optimization was the only available approach. That said, it might be a nice simplifying assumption.

[1/16/2012 1:30:07 PM] Bill Branan: I don't think we want to rule out the possibility of running services on an instance that is separate from that of one or more of the other duracloud apps. This sounds like a potentially valuable optimization strategy, but I wouldn't want to see it be the only option.

[1/16/2012 1:33:35 PM] Andrew Woods: Correct, Chris, putting the Shib infrastructure in place would loosen tomcat and the duracloud webapps to assume pre-authentication and therefore relax the need for another authN mechanism. Are there established patterns around securing a webapp on say 8080 without the help of a webserver while also assuming a pre-authenticated user? Possibly by limiting access to callers from localhost?

[1/16/2012 1:41:53 PM] Chris Wilper: Brad, that's a good example of a case where you can't just assume the current REMOTE_USER is the one you're using to make the request. It's also a case where presumably you'd have to obtain those different credentials from the user as part of the setup /configuration of the work you're about to do on their behalf. In the end I guess it boils down to nature of the callbacks that each individual service has to do. Establishing a pattern where a service could play nice as a real shib-aware client is good...I'm just not sure that's a necessary/desirable type of interaction for all services. But if it's easy and relatively performant, I don't think it hurts.

[1/16/2012 1:44:17 PM] Brad McLean: +1 That it isn't one size fits all; I'd expect the 90% to work well, and better with REMOTE_USER; the 10% would need the extra shib aware client capability.

[1/16/2012 1:48:31 PM | Edited 1:49:25 PM] Chris Wilper: Andrew, I have certainly seen taking advantage of direct Tomcat-to-Tomcat interaction (avoiding going through Apache) in the Atlassian products when they talk to each other -- not as a way of avoiding AuthN'ing twice, but as a way to avoid the SSL overhead when both services are on the same host. In such cases, the underlying Tomcat ports are not open to external access via firewall rules.

[1/16/2012 1:51:28 PM | Edited 1:51:40 PM] Chris Wilper: But I have not seen something exactly like what I'm suggesting, where AuthN is effectively bypassed by going Tomcat-to-Tomcat. I agree that it would be nice to know whether that pattern has been followed by other folks.

[1/16/2012 1:53:47 PM] Andrew Woods: How are the credentials being passed between tomcats (or, how is sso being handled) in the confluence case?

[1/16/2012 1:54:10 PM] Daniel W. Davis: I believe SSO solutions like Atlassians generally simplify AuthN on each call by caching credentials for a little while for additional efficiency. However, they don't avoid AuthZ.

[1/16/2012 1:56:13 PM] Chris Wilper: Atlassian SSO is handled via cookies from Crowd. Each individual Atlassian webapp has a crowd plugin that deals with redirecting to an AuthN page if not authenticated, or validating the cookie if the request includes one. So there's some Crowd-specific code running "at the top" of the call stack for each service running inside Tomcat. Whereas the road we're going down tries to push that all out to the Apache level.

[1/16/2012 2:09:02 PM] Chris Wilper: Also, when, say, JIRA needs to contact Bamboo to do some task on behalf of a user, it does so with a special "bamboo" account rather than a user account. I was just poking around a bit and found that it can use OAuth or HTTP Basic authentication to do this. But it varies among the products and versions.

[1/16/2012 2:09:50 PM] Andrew Woods: As you mentioned above, Chris, in the case of duracloud services that need to make calls back into durastore, it is a case-by-case question if they will be running locally and can therefore backdoor the Apache authN and if the current REMOTE_USER is the appropriate caller. It would potentially be a performance optimization as well as lower the barrier for new service creation.

[1/16/2012 2:11:56 PM] Daniel W. Davis: I guess CAS had an impl and JOSSO community edition too.