LInk to 2018 ACAMP wiki

Advance CAMP Thursday , Oct. 18, 2018

2:40pm-3:30pm

Oceana 5

Improving IdP/SP Containerization/Automation by Various Means such as APIs for Config

CONVENER:

MAIN SCRIBE: ~~Jon Miner~~ Mike Zawacki (Rules!)

ADDITIONAL CONTRIBUTORS: (Hopefully many more)

# of ATTENDEES: 20

DISCUSSION:

Seed content for discussion
(One)Problem statement:
One aspect of automation is a consistent vocabulary of common settings such that they can be either ‘fetched’ or added and then consistently provisioned.

If a common dictionary exists that software could then implement as configuration API:

Candidate item	Candidate name	Shib IdP	Shib SP	SimpleSAMLPHP	ADFSToolkit	TIER Config tool
Signing Key URL	signing_url					

Aggregate URL (n elements)	aggregate _url					
Central DS URL						

REFEDS deployment list - deployment@refeds.org

Discussion started with thinking that if there was an API for setting up configuration of the IdP that could be re-used, but is current handed out and entered by hand in XML files, etc.

Configuration of various SPs (different vendors) could be done in an automated manner from the federation if there was a common way to tell them.

Confirguration API. When you generate new keys you rely on sed to populate, etc. API could be called to handle those functions.

- Jon: Is the barrier now is that Shib uses XML files? Is this something that could be solved with java properties files? Or something where federation pushes out to IdPs? Or to SPs? Other side for SP: Federation could tell the SPs about such changes. Question is how federations would do that - use their private keys?

Chris: Think we can come to convergence on would be a simple set of settings. Federation operator could then use those to..

- David: Is the question what the best way would be to inject run time changes into containerized ISP
    - And maybe SPs?
    - Paul: Possibly that, also pushing changes to container build & ensure uniformity of config files & build arch.
    - Dave: So this would be build time changes or runtime changes?
    - Paul: Build. You want to ensure keys in containers are up to date, etc. Useful also for those who are XML-aly challenged.
    - Chris: It's like SIM getputupate but for the settings of the component. I don't want to have to worry about versions - it needs the fingerprint, not the cert. Maybe it would be better stated as Federation trust settings
    - Greg: Would prefer layer to tell developers/deployers how to build base level machine, then push their own Google passwords, private key inserts, etc themselves. So if I could tell the IdP "let me plug in whatever my layer is" (Docker secrets, environment variables, etc). Then they could push interactive installers in as needed.
    - Dave: Push for 12factor.net (Yes!!! -- Jon) - injecting configs from environment, etc. The config should get pushed into container at runtime. Encourage us to use that as a common anchorpoint

    - Paul: Was discussion for a way to have containers go get appropriate secrets, etc.
    - Chris: Looking for a way to push federation operator specific bits out. Also need feature switches (e.g. InCommon's reference implementation). The "blessed set" of feature settings. Standardizing reference configs like this eases deployment angst across all Federation operators.
    - Mario: Janusz created API that takes basic HTTP commands to send configuration instructions from client to server. For example, passes keys, cert handler, etc. We can share it with you - could be useful.
        - Paul: How does it inject into the IdP config?
        - Mario: Generates JSON/YML file, API server gives that to the client.
        - **AI Mario: get code onto Github, share out**
        - **AI ALL: Review the code, convene to discuss**
        - API Schema: https://geantsrv15.ct1.garrservices.it/schema

- Chris: All of this good, still leaves SP underserved. Benefit of OIDC is that not as much is needed for them.
    - Can use ADFS in powershell, Docker containers that Chris has been working with for some of that.
    - So Jagger tool Mario talked about could create Federation specific configs for IdPs and SPs
    -

- Q: Gabor - Is it a barrier that SHib uses so many XML files?
    - Paul: Only for this use case. Problem is Shib chose XML files for its database, and you have to make changes manually. No API to automate that.
    - Gabor: Also problem is that it's only dynamically loaded at start up and under certain reload conditions. No way to easily/reliably trigger a data update followed by reload. If we relied on a flatfile for all that, the problem might be easier to solve.
    - Jon: Couple of different issues here. Building the containers, API needed to inject realtime changes and update XML files, and trying to do all of that in a way that isn't Shib specific. Doing so would prevent Federation to easily inform participants of updates/changes
    - Chris: Getting into Devops model makes this easier - updating attribute, key creation, etc is simpler. Shib's ability to automate these things is limited/
    - Roland: SO is the idea to have a shell-like interface?
        - Chris: Maybe. On some flavors there's API-like functionality.
    - Gabor: Capturing state changes important. You can sort of fix it with scripting Shib but that doesn't solve the problem of state collisions - lack of ability to grok versioning. Should be an effort made to decouple shib config files sitting on the disk from the state of shib's internals at runtime. It'd be nice to make transforms to the internal state (and that have that state persist to disk/config)

        - Chris: Kubernetes has automation infrastructure that could serve for that. Allows for base settings with option to inject more intricate, IdP/SP specific settings on top of that.
        - Chris: Best to develop that within the community. Devops method seems best. TIER packaging team could be a good fit for that development. Could also be worked out through Jagger mailing list. Or REFEDS mailing list (that could be noisy, though). I also had a mailing list that could be used, isn't currently getting heavy use. But lack of FedOp interoperability is a problem.
- Next steps?
    - Paul: Would like to figure out which use case would be best to pursue. Federation settings don't change often. Maybe keep those static and look at other uses.
        - Chris: Static code like that is an impediment though
        - Jim: If we had an API that could be used to bake FedOp settings into container builds. If we could put forward a proposal that came from community it could make work easier
    - Chris: Maybe start with a straw man (e.g. Janos' Jagger tool), look at what implementing would involve, engage Shib consortium with consolidated ask, also work with REFEDS.
        - Paul: I'd like to see work done on this. The lack of automation around all this is an impediment to existing work.
        - Chris: so….
            - **Get code from Janusz**
            - **Have Janusz give a demonstration/explanation of his work to date**
            - **Set up regular communication (Could use eduGAIN slack channel)**
            - **In next few weeks have Janos present on TIER Packaging call**
                - **AI JIM: Set up special call for Janusz to present**
            -

ACTIVITIES GOING FORWARD / NEXT STEPS:

Text

Tool for reloading stuff in Shibboleth :

https://github.com/lordal/idp-installer-SWAMID

idp-mgr.sh

=====

Note: please be sure to link to or attach any key resources from this breakout session