MD WG Meeting 2013-07-18

Metadata Distribution WG 2013-07-18

Attending: IJ Kim, Ian Young, Scott Cantor, Tom Scavo, Harry Nicholos, Max Miller

In John's absence, Scott Cantor led today's discussion.

Minutes

- Last time we discussed offline signing; this time the goal is to understand the impact of an online signing process. Specifically, what are the use cases that require online signing?
- For reference, today InC Ops batch-produces a signed metadata aggregate every business day at approximately 3:00 pm Eastern. So in general there is a limited signing window (although emergency signings are possible).
- We could move the publishing process to the endpoints (like OpenID), or more generally, to a metadata service (like DNS).
- In some sense, DNS is a natural approach ("if it quacks like DNS, it must be DNS") but perhaps this is not a viable option. It seems many DNS administrators would rather keep DNS "pristine" although there might be cooperation at some sites if we were to go down this path.
- DNS SRV records are not sophisticated enough anyway. We would have to invent something new (using NAPTR records perhaps).
- In any case, we're looking at queries and lookups instead of batch downloads.
- In the near term, a feasible solution would be to move from a batch-oriented approach to a more dynamic approach involving individual entities. In that case, would we want to query a central service?
- From some (many?) IdP administrators' PoV, there would be concern if endpoints were responsible for publishing metadata. It would be safer to obtain metadata from the federation operator, not the endpoint.
- If we did move to per-entity metadata, the number of authoritative signatures would grow by orders of magnitude. (We have over 1500 entities in InC metadata today.) Alternatively, we could impose transport security.
- Whatever we do, important issues are trustworthiness of metadata and operational availability. Clearly metadata needs to be highly available.
- From an implementation PoV, Shibboleth has the best support for metadata. SimpleSAMLphp can consume large metadata aggregates but the process depends on cron. (We know from experience that external processes are less reliable and tend not to be deployed.) Ping has expressed interest in providing better metadata support. It seems questionable to expect much uptake of a more complex consumption model for metadata when the simple batch (/etc/hosts) model has so little uptake. So in large part this is about Shibboleth.
- A central registration and distribution model, where individual entity descriptors are signed by a trusted authority (i.e., the federation operator), would permit metadata to be aggregated and hosted from anywhere. Very flexible.
- (At this point in the discussion, still no use case for PKI.)
- Note that per-entity metadata breaks discovery processes that rely on a comprehensive metadata aggregate. These processes will need to be redesigned, perhaps using JSON metadata-like information retrieved just-in-time. This information tends to be less security-relevant than the full metadata is.
- As noted earlier, under normal circumstances there is a minimum 24-hour window for signing and distributing InCommon metadata. Are there use cases that require more frequent signing? Of course problems and corrections sometimes require more frequent signing but this is the exception rather than the rule.
- As a thought experiment, what would happen if the window were one hour (instead of one day)? In particular, what are the required interventions (necessitated by the RA process)?
- Today, **all** metadata is reviewed by the RA, regardless of the request. Can certain innocuous changes be automated? Can we identify what changes those are?
- A huge risk with the current metadata distribution model stems from the brittleness of metadata aggregates. In fact, this discourages experimentation and evolutionary change. This is reason alone to consider a different model.
- There is a lot of value in keeping the signing key offline. Just look at the SWITCH docs to see how complicated using a PKI can be for deployers.
- Thought experiment: If we had to go online, what would the requirements be?
- *Use Case*: Local metadata at OSU. Online signing key. Short validity window (1 or 2 days). Initially considered a PKI, but it didn't take long to realize that it was going to be mostly a waste of time. The cost/benefit ratio of the PKI model is high. Today the signing is still online, but no PKI.
- Offline CRLs (produced every week) actually decrease overall security by extending the window of vulnerability to a compromised key to weeks from potentially days (if a signing process is online and automated).
- Conclusion: An online signing key doesn't necessitate a PKI. We'd have to examine the individual use case.
- *Use Case*: InCommon/Comodo Certificate Service. Multiple online signing keys, each protected by an HSM. CRLs are refreshed daily, with 4-day validity window. OCSP.
- If we did move to a PKI model, think about how much work it would be for participants to make the transition. And they would *have* to make the transition, otherwise there would be a security vulnerability.
- Tentative conclusion: Maybe we don't have to rekey.
- One approach is consider types of risk. For example, an online signing key such that the key is not protected by an HSM.
- Suppose the key were protected by an HSM. The question is how does the HSM work? If an attacker needs physical access to the HSM, then why bother with a PKI?