RUM WG Meeting Minutes, IETF 109

IETF 109 - Virtual
Wednesday, November 18, 2020, 2:30 - 3:30 AM EST
Chairs: Brian Rosen, Paul Kyzivat
Minute Taker: Jim Malloy

Agenda:

- Introduction [0:10] : Chairs
    - Agenda bashing and WG status update
- Interoperability Profile for Relay User Equipment [0:45] : Brian Rosen (https://tools.ietf.org/html/draft-ietf-rum-rue-03)
    - - List remaining open issues
    - - Discuss and decide on direction for issues
    - Other open items
- AOB [0:05] : Chairs

RUE Profile Issues:

[Brian Rosen]

Two issues:

- **Provider control of authorized apps,**
- **How registration creds are stored and transmitted (Configuration file).**

Unauthorized apps:

Closed systems (e.g., Android with Google Play Store) can limit to certified apps. We don't have a closed system…

Alternative is "API Key" – secret key, send encrypted as part of app startup.

Could be xml/json or SIP header

Is this an acceptable mechanism? / which mechanism?

[Eugene Christensen]

Outside of SIP would be a good option – HTTPS transaction. Register is in SIP, but if it is per-SIP connection, HTTPS would be good. Not clear why putting it in the SIP header would be beneficial

[Brian]

Prefer HTTPS, but putting it in SIP is easier. Could be associated with configuration file

[Paul Kyzivat]

What is mechanism for certification? What testing is required? Do we need Provider-specific keys?

[Brian]

Can be out of scope. Mechanism can work either way – one key for all, one key per Provider. Given International scope, should be flexible.

[Paul]

Need to know cardinality – does the RUE need to keep one per Provider as part of configuration information?

[Brian]

Could maintain a per-Provider list. More complex if every app vendor needs to be approved by every VRS Provider separately.

[Paul]

Would mechanisms be country-specific?

[Brian]

No, If the environment only has one, or one per Provider. As long as we allow the notion of per-provider or global

Comments?

[Jim Malloy]

Should be flexible – while US may have a preference, other regions may not.

[Olle Johansson]

Have we looked at OAUTH or SAML?

[Brian]

Different Problem – not authorizing users, but code.

[Olle / Brian]

Would be more complicated. Single sign-on mechanism, requires centralized infrastructure. Get a key from one Provider to use with another Provider. Who would supply keys? Maybe you could use your Facebook or university login. (Ties to individual, not device) SAML has federated trust.

[Olle]

Is API a random string or a token containing useful data?

[Brian]

Typically, doesn't contain data, or is invisible – doesn't need to be in specification. Example, embed Provider ID in key

[Jim]

Facebook tied to Device/app

[Olle]

Looking to verify certain release of certain software

[James Hamlin]

RFC 8252 talks about types of keys for OAUTH

[Brian]

Summarizing:

- **Tentative decision (will be confirmed on list):**
- **API Key is an acceptable answer. May have alternative answer using OAUTH/SAML**
- **Brian will start writing text for API key, will insert after discussion/approval on list.**
- **Will incorporate whatever is agreed to in next version.**

Configuration File

[Brian]

Local file with sections per Provider. Contains storing password/userid in cleartext. May have more than one Provider in local storage.

Some ideas: Use SAML to authenticate users, local login so app has username/password, …

[Olle]

Implementations we have in SIP are embarrassing, should not be used as starting point. Certificate enrollment, bootstrap operations using encrypted public key of the device would be a good thing. What is described in slide is embarrassing.

[Brian]

There is a large base of existing code using these mechanisms. Could use OpenID instead of username/password

[Olle]

OPENID one possibility. Somehow setting up device certificates and renewal of those certificates is part of the puzzle – since we have HTTPS at the beginning of the process,…

[Brian]

Assume mutual auth – now what

[Olle]

Chicken and egg problem. Cert needs to be trusted by Providers. Is the device login the same as the user login?

<<longer discussion – check the video>>

Client authentication using SIP instead of username/passwords.

[Paul]

Missing some context. Need use cases/discussion of life cycle showing users, providers, devices and relationships. Including cases where keys are compromised/reset, including emergency situations.

[Brian]

Olle is expanding the issue to basic authentication for SIP.

[Jonathan Lennox]

Not clear what certs provide over OPENID

<<more discussion>>

Referencing existing Specs has an advantage

[Brian]

Proposal to widen this out and use certs for everything. Change to registration, SIP Authentication, more. Should we consider this?

[Paul]

One thing about client certs – they only make it as far as the front-end proxy for the server, not necessarily the registrar.

[Brian]

Doable, but not a lot of RFC coverage for it.

[Olle]

Start with agreeing that traditional username/password is something we don't want. Acknowledge lack of RFC support, proxy issues.

<<more discussion on general authentication>>

[Jim]

Should we be leading or following SIP authentication mechanisms?

[Olle]

Token-based authentication is the way to go.

[Brian]

Need to hear from Providers…

Providers?

<<crickets>>

[Paul]

What if config file supported both OAUTH and tokens? How is file populated?

Single local file, download single Provider sections from each Provider? How often do you download? Every time you turn your machine on or after timeout.

[Eugene]

Envisioned this would be downloaded once, but would only contain single Provider data, why not let the device store that data, if it needs to be updated, another mechanism would be needed.

[Brian]

The thing that says "once" could be triggered by manual action/timeout.

(Once a day/month/every time you login

[Paul]

Revocation/retry is triggered by bad password. Fetch a new configuration and try again.

[Brian]

How are authentication credential unlocked is important. Part we need to do right.

[Paul]

System needs to be able to connect without human interaction.

[Brian]

Incoming calls need to work after the device reboots, without user interaction.

[Eugene]

Much of the existing VRS infrastructure is based on user login

[Olle]

Need to separate running state from configuring state. Need to consider situations where we have to accept expired certs and other things.

[Brian]

Over time.

Close out will follow up on list.