IETF86 - radext
Orlando, FL, U.S.A. - 12 March 2013

RESTENA

draft-ietf-radext-dynamic-discovery

-06

Document status

- **submitted -05 for WGLC**
  - **No comments during WGLC, but …**
  - **… lots of comments by Jim Schaad afterwards :-)**
- **submitted -06 addressing most of the small issues**
  - **Remaining ones in TRAC #148**
  - **Triggered new comments by Jim**
- **Will work on discussion and resolution of the remaining issues on the list ; a selection of some particularly interesting ones on later slides**

Issue : Certificate Validation

- **Started as Security Considerations discussion (TRAC #148.4)**
- **Zeroed in to the question : do we need a mandatory-to-implement certificate validation mechanism ?**
  - **e.g. SubjectAltName:otherName**
  - **or DNSSEC**
- **If yes, which should be the MTI mechanism ?**
  - **eduroam's policy OID « is an IdP » is not a good role model (well, works for us)**
  - **SNI only works with DNS names, not with NAIs**
  - **Sam : DNSSEC too complicated to implement**
  - **Remains SAN:otherName**

Resolution ? : Cert Validation

- **Q : Do we need an MTI mech ?**
- **How about : No ! ;-)**
- **If yes, I suggest SAN:otherName**
  - **not-so-great scalability but easy (easier than DNSSEC anyway)**
  - **scalability might be better with « wildcard » certificates**

Issue : Discovery of localhost

- NAI realm might be intended for local processing, but string representation of incoming request might not match config
- Triggers Dynamic Discovery
- DNS returns : localhost is *among the servers which should know*
- Q : If the result set contains « self »
  - should the entire discovery process be considered a failure ?
  - Or just remove that entry and use the rest

- I'd argue : server did discovery because didn't know how to handle request – but DNS says he's supposed to
  - Hints towards serious misconfig
  - Continuing to another server might create endless loops
  - And RADIUS has no loop detection
  - → better safe than sorry (or specify loop detection)

Issue : Discovery took too long
Now what ?

- RADIUS responses are time critical
  - >5s delay means « down ? » on previous hop
  - So can't wait that long, need to process packet after n seconds (n=3 in current draft)
  - If DNS takes longer
    - too bad, record failure and don't try until later (as in : configured negative reply timeout)
      - or -
    - Process packet, but keep trying the DNS lookup anyway ; might eventually result in a usable response ; store response for subsequent new Requests
- This makes for a nice DoS opportunity !
  - Create unresponsive DNS zones
  - « log in » with corresponding realm