EAP Usability

ALAN DEKOK IETF 111

THE PROBLEM

- EAP is hard to configure
  - Many methods, many options
- Vendors randomly change UIs, APIs, work flows, etc. for EAP configuration
  - There is a pain point which is not being addressed!
- MDM vendors sell “add ons” for simplification and ease of use
  - Which don’t work as well as they could
THE REQUIREMENTS

- A device has:
  1) Network connection (untrusted is fine, slow is fine)
  2) root CAs for web PKI
  3) user name to authenticate with: bob@example.com
  4) Password* to authenticate with: superSecret

* Entry of the password can be delayed until much later
THE PROPOSAL

- Get NAI from username: `bob@example.com` ➔ `example.com`
- Look up DNS CERT RR: `_server._cert._eap.example.com`
  - get URI: `https://example.com/.well-known/eap/server.pem`
- Verify Web cert via web root CAs, download certs
- Similar method for CA cert / server cert / client cert
- Certs can include network identification information (SSID, RCOI, etc)
- Client can now authenticate to network, verify server cert, use name/password
THAT’S IT

› Lots of details in the draft about variations of the above

› To show how it works in a variety of situations

› Lots of details about non-workable solutions

› Ideally only needs DNS and WWW configured on the server side

› Only new code is a user space utility on the client side

› Initially no changes required to supplicant code
LIMITATIONS

- Works only for TLS-based EAP types*
- Requires some network access to bootstrap
- Getting more benefit means moving some checks to supplicants
- Likely needs new EKU fields (TBD)
- Document is long and covers a lot of issues

* Sorry, Dan
BENEFITS

- Works in captive portals, can use LTE to bootstrap WiFi
- Minimal server-side changes required
- Configuration can be refreshed with minimal user intervention
- Can follow a process similar to web UIs, but for network access:
  - if the lock icon is green for example.com,
  - then it’s safe to enter your name and password
RUNNING CODE

- https://github.com/NetworkRADIUS/automatic-eap/
- Host defines domain name and certificates (generation scripts included)
- Brings up docker images for client and servers (RADIUS, DNS, WWW)
- Client does lookups, downloads certs
- Generates configuration, and runs eapol_test against RADIUS server
- Trust on First Use End to end trust verified at every step
CONCLUSIONS

‣ Seems simple enough

‣ Can be deployed today with per-device utilities (standards-based, not MDM)

‣ Questions?