EAP-NOOB : Nimble Out-of-Band Authentication for EAP
EMU WG, IETF 104
Prague, March 2019

Tuomas Aura, Aalto University
Mohit Sethi, Ericsson
various other contributors
EAP-NOOB: Nimble Out-of-Band Authentication for EAP

Bootstrapping security for smart appliances

draft-aura-eap-noob
Short EAP-NOOB overview
What problems EAP-NOOB solves?

- EAP method for bootstrapping devices out-of-the-box without professional administration
- Initial user-assisted out-of-band (OOB) authentication
  - E.g. scanning a dynamic QR code, dynamic NDEF tag
- Registration of new peer devices
  - Create persistent association between AAA and device and authorize network connectivity at the same time
  - Application-level bootstrapping: assign an owner to the device and redirect to application server
- Once registered, reauthentication without user interaction
EAP-NOOB architecture

- Remote AAA
- Local AAA
- AP
- New device
- User-assisted OOB channel
- EAP in-band
- UI or API
- OOB output (or input)
  - e.g. dynamic QR code

Trick: in-band communication over EAP between peer and server before device is registered.
Recent developments in EAP-NOOB
New in draft version -05

Minor changes based on feedback from implementation and verification

- Improved security considerations section
- Error message codes changed for better structure
- Easier implementation of rekeying (Reconnect Exchange)

Plan for -06:

- Add one roundtrip to each exchange. Deliver PeerId and peer state to server without updating NAI
  - Comply better with RFC 3748 section 5.1 guidance
  - Simpler peer implementation in wpa_supplicant
  - Better support for identifier randomization
Formal models and verification

Continued work on formal models:

• mCRL2 model
  • Modeling Protocol messages and state machines
  • Deadlock-freeness
  • DoS resistance for intentionally dropped messages

• ProVerif model
  • Cryptographic key-exchange properties
  • Authentication and confidentiality
  • Misbinding: correspondence between user intention and protocol completion
Analysis of misbinding

• **Generic attack against device-pairing protocols**
  where devices have no verifiable identifiers and authentication is based on physical access

• **Device with compromised UI can trick user to pair another device instead**

• Bluetooth and others are vulnerable

• **EAP-NOOB “pairs” devices with cloud. Device authenticated by user’s physical access**
  → misbinding possible

• **Mitigation:** channel binding, trusted path, device certificates, asset tracking

More at SAAG on Thursday or in **this report**
Temporal identifiers?

Output from hackathon

• Path to adding identity protection
  • PeerId is a persistent identifier for the peer
    → Can it be randomized in the future?
  • Recently-added Kz identifier also identifiers peer
    → Decided to remove in -06

• Why not randomize PeerId right now?
  • Not an easy task: Identifier update must be synchronized between peer and server. Must balance anonymity, reliability and server scalability
  • Should not be vulnerable to misuse of fall-back identifier (similar to IMSI catcher)
Other issues on our TODO list

• Thorough modeling and analysis of error message handling
• Timeouts in the protocol need modeling and user testing
• Hooks for bootstrapping application configuration, e.g. service URL (currently only creating shared key)
Recovering from dropped messages

• High-level goal: after initial bootstrapping, never repeat the user-assisted OOB step

• Problem:
  • Dropped last message in bootstrapping can cause failure
  • Dropped last message in cryptosuite upgrade could cause persistent failure

• Protocol (since -05) recovers from dropped messages in cryptosuite update i.e. Reconnect Exchange
  • Avoid persistent DoS that could break existing associations
  • Formal model and verification (mCRL2)
  • To minimize complexity, decided not to add similar recovery during initial bootstrapping
Changes to EAP spec (RFC 3748)?

Possibly controversial personal opinion

• Base EAP should be updated to provide features commonly required by methods:
  • Method payload in EAP-Response/Identity
    → avoid wasting a roundtrip
  • (Maybe) method payload in EAP-Success, delivered to peer
  • Fragmentation support (without wrapping in EAP-TLS)

• For channel binding required by RFC 7057, methods should have access to AAA AVPs:
  • Calling-Station-ID, Called-Station-ID
EAP-NOOB Summary

• EAP method with user-assisted OOB authentication for bootstrapping security of smart appliances

• Current version: draft-aura-eap-noob-05

There seems to be interest. If and when EMU WG is rechartered, this could be a work item