EAP-DIE

EMU Working Group, IETF 115

Josh Howlett (josh@federated-solutions.com)

Presentation goals

1. Raise the profile of an issue that is prevalent in BYOD-dominated environments to advance progress towards a solution

2. Propose a strawman solution and obtain your feedback

3. Establish if there is appetite for work on this solution/problem

EAP-DIE problem statement

- Supplicant configuration is generally managed either
  - administratively, using device management tools (InTune, etc)
    - Enterprise scenarios
  - manually, by the end user
    - Small/medium enterprise
    - BYOD (e.g., eduroam, public access WiFi)
- If configuration is managed manually, it typically persists until the device is replaced, even if the user's credentials have expired
- For example, 25-30% of attempted associations on a metropolitan university's network can be caused by expired eduroam configurations
  - Wasteful consumption of resource (radio spectrum, infrastructure, and battery)
  - Distorts utilisation/connection metrics
  - Pollutes logging and reporting
- EAP-DIE is intended to *reduce the persistence of expired, manually-configured supplicant configurations*

EAP-DIE design requirements

- Minimise impact on EAP actors (supplicants and RADIUS servers)
  → Low complexity solution

- Compatible with deployed infrastructure, to maximise use
  → Do not require changes to wireless controllers, APs, etc.

- Independent of EAP authentication method
  → Do not define method-specific extensions

- Secure
  → Avoid creating a tool to DoS supplicants, and other potential attacks

EAP-DIE messaging

- What is EAP-DIE?
  - The EAP server sends a Notification-Request to the supplicant to signal the expiration date of network access
- An exception in “Support for Sequences” (RFC3748) permits a Notification-Request/Response exchange before an authentication method has concluded:
  - “Once a peer has sent a Response of the same Type as the initial Request, an authenticator MUST NOT send a Request of a different Type prior to completion of the final round of a given method (with the exception of a Notification-Request)”
  - “An authenticator MAY send a Notification Request to the peer at any time when there is no outstanding Request, prior to completion of an EAP authentication method”
- EAP-DIE uses Notification-Request to send data to the supplicant **after** EAP keys have been derived but **before** transmission of EAP-Success

EAP-DIE data

- Default maximum length of Notification-Request is 1020 octets, leaving 1015 octets for a “human readable message”
- The data should be parseable by both humans and supplicants
- Proposed data fields
  - Version
    - Indicates the version of EAP-DIE
  - Message
    - Gives a human-readable message (e.g., “Your access to this network will expire on”) that can be specified by the operator of the AAA/EAP server
  - Expiration date
    - A data in some standard format (TBD)
  - Message authenticator
    - Enables the supplicant to authenticate the values of the version, message and expiration fields using its knowledge of the derived EAP keys

EAP-DIE operation

1. The operator of the AAA/EAP server configures an expiration policy(ies) for their users
2. Before connecting to a network, the supplicant checks for cached expiration data; if the network configuration has expired, it is disabled and authentication is not attempted
3. If there is no expiration data, or if it has not expired, the supplicant and EAP server begin EAP authentication
4. When the EAP keys have been derived, the EAP server sends a Notification-Request containing EAP-DIE data to the supplicant
5. Supplicant validates and caches the EAP-DIE data, and returns a Notification-Response
6. The authenticator concludes authentication as usual

Issues

1. RFC3748 talks of the authenticator sending the Notification-Request, not the EAP server – is this material?
2. Does EAP-DIE overload the Notification-Request semantics?

   "The peer SHOULD display this message to the user or log it if it cannot be displayed. The Notification Type is intended to provide an acknowledged notification of some imperative nature, but it is not an error indication, and therefore does not change the state of the peer. Examples include a password with an expiration time that is about to expire, an OTP sequence integer which is nearing 0, an authentication failure warning, etc. In most circumstances, Notification should not be required."
3. Is the message sequencing consistent with the process of EAP key derivation on the EAP server and supplicant?
4. EAP-DIE requires a previous successful authentication to cache the expiration date; does this limit its utility?
5. Is EAP the right architectural layer to solve this problem?