RADIUS Extensions Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: September 22, 2016

S. Winter
RESTENA

March 21, 2016

Considerations regarding the correct use of EAP-Response/Identity draft-ietf-radext-populating-eapidentity-00

Abstract

There are some subtle considerations for an EAP peer regarding the content of the EAP-Response/Identity packet when authenticating with EAP to an EAP server. This document describes two such considerations and suggests workarounds to the associated problems.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 22, 2016.

Table of Contents

1. Introduction

1.1. Problem Statement

An Extensible Authentication Protocol (EAP, [RFC3748]) conversation between an EAP peer and an EAP server starts with an (optional) request for identity information by the EAP server (EAP-Request/ Identity) followed by the peer's response with identity information (EAP-Response/Identity). Only after this identity exchange are EAP types negotiated.

EAP-Response/Identity is sent before EAP type negotiation takes place, but it is not independent of the later-negotiated EAP type. Two entanglements between EAP-Response/Identity and EAP methods' notions of a user identifier are described in this document.

1. The choice of identity to send in EAP-Response/Identity may have detrimental effects on the subsequent EAP type negotiation.

2. Using identity information from the preferred EAP type without thoughtful conversion of character encoding may have detrimental effects on the outcome of the authentication.

The following two chapters describe each of these issues in detail. The last chapter contains recommendations for implementers of EAP peers to avoid these issues.

1.2. Taxonomy of identities in EAP

The notion of identity occurs numerous times in the EAP protocol stack (EAP-Response/Identity, Outer identity, method-specific

identity, tunneled identity). This document uses the following terminology when discussing EAP identities.

- Method-specific Identity: Each EAP method has a means to identify the user or machine that tries to authenticate. There are no restrictions on the format or encoding of this method-specific identity. If an EAP methods distinguishes between this actual identity and a outer identity (see next bullet), then the Method-specific Identity is also often called the Inner Identity.

- Method-specific Outer Identity: Some EAP methods allow privacy-preserving enhancements where a string is sent as "identity" which is actually not necessarily related to the user or machine that tries to authenticate. There is often a relationship between the Method-specific Outer Identity and the Inner Identity (e.g. they often share the same NAI realm suffix); but this is not a requirement. There are no restrictions on the format or encoding of this method-specific identity. Method-specific outer identities are either

  * explicitly configured (e.g. string input UI: "Outer Identity")

  * implicitly configured by copying the actual Method-specific (Inner) Identity

  * implicitly configured by copying the NAI realm of the Method-specific (Inner) Identity and prefixing it non-configurably with a fixed privacy-preserving local username part like "anonymous" or the empty string (see [RFC7542])

  * configured in a mixed way, e.g. using a explicit string input UI for the local part of the outer identity and combining it implicitly with a copy of the NAI realm part of the Method-specific (Inner) Identity

- EAP-Response/Identity: a string representing the user or machine that tries to authenticate, used outside the EAP method-specific context for the entire EAP session. There can be only one EAP-Response/Identity per EAP session, even if that session is configured with more than one EAP method to authenticate with. As per [RFC3748] there is no encoding requirement on EAP-Response/Identity. In AAA protocol routing contexts, the content of EAP-Response/Identity is often used for request routing purposes. EAP-Response/Identity is chosen from the set:

  * all method-specific outer identities from all configured EAP types supporting the notion of an outer identity union

* all method-specific identities from all configured EAP types without the notion of an outer identity

One of the two problems addressed in this document stems from this fact: the set of identities may contain more than one element. The resulting EAP-Response/Identity always routes all configured EAP types to only one destination, even if different EAP types would need routing to different destinations.

- User-Name: when using EAP in AAA protocol contexts (e.g. RADIUS [RFC2865], Diameter [RFC6733]), this additional identity is created outside the EAP peer (typically in a pass-through authenticator) by copying EAP-Response/Identity content to the AAA protocol's User-Name attribute. There is no format requirement on User-Name, but there is an encoding requirement: the string MUST be UTF-8 encoded. One of the two problems addressed in this document stems from this fact: EAP-Response/Identity does not have an encoding requirement, nor does it carry meta-information about the encoding used - and yet, it needs to be coerced into a UTF-8 encoding.

- Further identities: Some EAP methods establish an EAP session inside EAP (e.g. PEAP first establishes a TLS tunnel using a method-specific outer identity, and then starts an EAP exchange inside the tunnel). This being a new, independent EAP session, it contains its own EAP-Response/Identity, can invoke EAP method negotiation with different (inner) EAP types (this happens e.g. with EAP-FAST and its configurable choice of EAP-GTC or EAP-MSCHAPv2 inside the inner EAP session), and those inner EAP methods then have their own (inner) method-specific identities. Where the inner EAP method itself supports the notion of method-specific outer identities, another identity could be configured. For the purposes of this document, none of those details are considered and the process by which the (outer) EAP method selects its method-specific identity is left entirely to that EAP type. This document does not consider the (inner) EAP-Response/Identity in scope; the recommendations in this document to not apply to such (inner) occurences of EAP-Response/Identity.

1.3. Requirements Language

In this document, several words are used to signify the requirements of the specification. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. [RFC2119]

2. EAP-Response/Identity: Effects on EAP type negotiation

Assuming the EAP peer's EAP type selection is not the trivial case (i.e. it has more than one configured EAP type for a given network or application, and needs to make a decision which one to use), an issue arises when the configured EAP types are not all configured with the same method-specific outer identity (or method-specific identity for EAP types not supporting the notion of an outer identity).

Issue: if the identities in the set of configured EAP types differ (e.g. have a different [RFC7542] "realm" portion), and the authenticator does not send identity selection hints as per [RFC7542], then EAP type negotiation may be limited to those EAP types which are terminated in the same EAP server. The reason for that is because the information in the EAP-Response/Identity is used for request routing decisions and thus determines the EAP server - a given user identifier may be routed to a server which exclusively serves the matching EAP type. Negotiating another EAP type from the set of configured EAP types during the running EAP conversation is then not possible.

Example:

Assume an EAP peer is configured to support two EAP types:

- EAP-AKA' [RFC5448] with user identifier imsi@mnc123.mcc123.3gpp-network.org
- EAP-TTLS [RFC5281] with user identifier john@realm.example

The user connects to hotspot of a roaming consortium which could authenticate him with EAP-TTLS and his john@realm.example identity. The hotspot operator has no business relationship at all with the 3GPP consortium; incoming authentication requests for realms ending in 3gppnetwork.org will be immediately rejected. Identity selection hints are not sent.

Consequence: If the EAP peer consistently chooses the imsi@mnc123.mcc123.3gpp-network.org user identifier as choice for its initial EAP-Response/Identity, the user will be consistently and perpetually rejected, even though in possession of a valid credential for the hotspot.

An EAP peer should always try all options to authenticate. As the example above shows, it may not be sufficient to rely on EAP method negotiation alone to iterate through all configured EAP types and come to a conclusive outcome of the authentication attempt. Multiple new EAP authentications, each using an EAP-Response/Identity from a

different element of the set of method-specific outer identities, may be required to fully iterate through the list of usable identities.

3. Character (re-)encoding may be required

The method-specific identities as configured in the EAP method configuration are not always suited as identities to choose as EAP-Response/Identity: EAP methods define the encoding of their method-specific outer identities at their leisure; in particular, the chosen encoding may or may not be UTF-8.

It is not the intention of EAP, as a mere method-agnostic container which simply carries EAP types, to restrict an EAP method's choice of encoding of method-specific identities. However, there are restrictions in what should be contained in the EAP-Response/Identity: EAP is very often carried over a AAA protocol (e.g over RADIUS as per [RFC3579]). The typical use for the contents of EAP-Response/Identity inside AAA protocols like RADIUS [RFC2865] and Diameter [RFC6733] is to copy the content of EAP-Response/Identity into a "User-Name" attribute; the encoding of the User-Name attribute is required to be UTF-8. EAP-Response/Identity does not carry encoding information itself, so a conversion between a non-UTF-8 encoding and UTF-8 is not possible for the AAA entity doing the EAP-Response/Identity to User-Name copying.

Consequence: If an EAP method's method-specific identity is not encoded in UTF-8, and the EAP peer verbatimly uses that method-specific identity for its EAP-Response/Identity field, then the AAA entity is forced to violate its own specification because it has to, but can not use UTF-8 for its own User-Name attribute. If the EAP method supports a method-specific outer identity in a non UTF-8 character set, and the EAP peer verbatimly uses that outer identity for its EAP-Response/Identity field, then the same violation occurs.

This jeopardizes the subsequent EAP authentication as a whole; request routing may fail, lead to a wrong destination or introduce routing loops due to differing interpretations of the User-Name in EAP pass-through authenticators and AAA proxies.

4. Recommendations for EAP peer implementations

Where method-specific identities or method-specific outer identities in configured EAP types in an EAP peer differ, the EAP peer can not rely on the EAP type negotiation mechanism alone to provide useful results. If an EAP authentication gets rejected, the EAP peer SHOULD re-try the authentication using a different EAP-Response/Identity than before. The EAP peer SHOULD try all possible EAP-Response/

Identity contents from the entire set of configured EAP types before declaring final authentication failure.

EAP peers need to maintain state on the encoding of the method-specific identities and outer identities which are used in their locally configured EAP types. When constructing an EAP-Response/Identity from the set of identities, they MUST (re-)encode the corresponding identity as UTF-8 and use the resulting value for the EAP-Response/Identity.

5. Privacy Considerations

Because the EAP-Response/Identity content is not encrypted, the backtracking to a new EAP-Response/Identity will systematically reveal all configured identities to intermediate passive listeners on the path between the EAP peer and the EAP server (until one authentication round succeeds).

This additional leakage of identity information is not very significant though because where privacy is considered important, the additional option for identity privacy which is present in most modern EAP methods can be used.

If the EAP peer implementation is certain that all EAP types will be terminated at the same EAP server (e.g. with a corresponding configuration option) then the iteration over all identities can be avoided, because the EAP type negotiation is then sufficient.

If a choice of which identity information to disclose needs to be made by the EAP peer, when iterating through the list of identities the EAP peer SHOULD

- in first priority honour a manually configured order of preference of EAP types, if any
- in second priority try EAP types in order of less leakage first; that is, EAP types with a method-specific outer identity that differs from the method-specific identity should be tried before other EAP types which would reveal actual user identities.

6. Security Considerations

The security of an EAP conversation is determined by the EAP method which is used to authenticate. This document does not change the actual authentication with an EAP method, and all the security properties of the chosen EAP method remain. The format requirements (character encoding) and operational considerations (re-try EAP with

a different EAP-Response/Identity) do not lead to new or different security properties.

7. IANA Considerations

There are no IANA actions in this document.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. Informative References

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC3579] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", RFC 3579, September 2003.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[RFC5281] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, August 2008.

[RFC5448] Arkko, J., Lehtovirta, V., and P. Eronen, "Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA')", RFC 5448, May 2009.

[RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <http://www.rfc-editor.org/info/rfc7542>.

Author's Address

Stefan Winter
Fondation RESTENA
6, rue Richard Coudenhove-Kalergi
Luxembourg 1359
LUXEMBOURG

Phone: +352 424409 1
Fax: +352 422473
EMail: stefan.winter@restena.lu
URI: http://www.restena.lu.