This document defines bindings for existing EAP methods to transport Diameter AVPs, called "AAA payloads". The primary application is to support EAP channel bindings, but this could be used for other applications as well.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 16, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

Table of Contents

1. Introduction ................................................. 3
2. Terminology ................................................. 3
3. Overview and Requirements ................................... 3
4. AAA Payload Format .......................................... 4
5. Bindings Requirements for EAP Methods ...................... 5
6. Bindings for Existing EAP Methods ............................ 6
   6.1. Generalized Pre-Shared Key (GPSK) ....................... 6
   6.2. Pre-Shared Key (PSK) ................................... 6
   6.3. Password Authenticated Exchange (PAX) ................ 6
   6.4. Tunneled Transport Layer Security (TTLS) ............ 7
   6.5. Flexible Authentication via Secure Tunneling (FAST) 7
7. Security Considerations ...................................... 7
8. IANA Considerations ........................................... 8
9. References .................................................... 8
   9.1. Normative References .................................... 8
   9.2. Informative References .................................. 8
Authors’ Addresses .............................................. 9
1. Introduction

This document defines a payload which can be securely transported by an Extensible Authentication Method (EAP) method [RFC3748] that carries arbitrary Diameter Attribute-Value Pairs (AVPs) [RFC3588]. While it may seem strange for EAP to encapsulate Authorization, Authentication, and Accounting (AAA) messages, since AAA typically encapsulates EAP, the security properties are different. In particular, AAA data transported by EAP between the client and server will be protected by an end-to-end security relationship. This provides a secure channel for doing things like channel bindings [RFC5056].

2. Terminology

In this document, several words are used to signify the requirements of the specification. These words are often capitalized. 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 [RFC2119].

3. Overview and Requirements

Many EAP [RFC3748] methods have extensible properties that allow you to embed arbitrary data within a secure channel. This channel is secured using keys derived during the EAP authentication. These channels vary in the properties that they provide, typically either providing integrity protection or both confidentiality and integrity protection.

In this document we define a payload format for encapsulating Diameter AVPs [RFC3588], and via backwards compatability [RFC4005], RADIUS TLVs [RFC2865]. We provide bindings for a variety of existing EAP methods that would allow them to transport this data. One specific application of this is to support EAP channel bindings [RFC5056][I-D.ietf-emu-chbind].

The main goal is to provide the peer and server to exchange AAA messages protected by an end-to-end security association. As such, any EAP method transporting the AAA payloads defined in this document MUST support integrity protection. To accomplish this, a method supporting AAA payloads MUST perform mutual authentication and derive session keys (i.e. MSK, etc), and during this derivation process MUST derive a unique, cryptographically independent, fresh key for protecting AAA payloads. Protocols SHOULD also support confidentiality in addition to integrity protection. Confidentiality
is important for identity protection, as a variety of identities can be passed over this channel.

4. AAA Payload Format

This section describes the formatting for the AAA Payloads. Each payload consists of the following fields:

- Version, 1 octet
- Flags, 1 octet
- length(Session-ID), 2 octets
- Session-ID [RFC5247], arbitrary length
- One or more Diameter AVPs [RFC3588], arbitrary length

The version field is 1 octet. This document defines version 0x01.

The flags field is 1 octet, and is the logical AND of the following applicable values:

- 0x01: Validation Failed
- 0x02: Validation Inconclusive
- 0x04: Validation Successful

The validation flag fields are used by the EAP server to convey the success of the consistency check to the EAP peer. These flags SHOULD NOT be used in messages from the peer to server.

The Session-ID field is arbitrary length and is proceeded by its 2-octet length specified in network-byte order.

Following the Session-ID is an arbitrary number of Diameter AVPs. AVPs already contain a length field internally, so they can be parsed without additional information.

The payload can be graphically depicted as:
5. Bindings Requirements for EAP Methods

This section describes a set of general requirements for EAP methods implementing the protected exchange of arbitrary data using the techniques described in this draft.

An EAP method requiring the protected exchange of payloads during its execution in order to enable certain features MUST:

- support at least one secure cryptographic algorithm that can be used for integrity protection, such as an HMAC;
- derive a session key that can be used in the integrity protection algorithm, as soon as fresh EAP session keys are available; and
- define a container enabling the exchange of arbitrary payloads; and
- provide an integrity-protected channel for at least 3 protocols flows (i.e. 1.5 roundtrips) in which containers with payloads can be securely exchanged.

Optionally an EAP method MAY also support an encryption algorithm that is used with a freshly derived encryption key to provide confidentiality of all data that is exchanged in the protected channel.
Only existing EAP methods already satisfying these requirements can support features that require the protected exchange of AAA payloads.

New EAP methods SHOULD be designed to meet all requirements.

For features demanding the protected exchange of potentially large chunks of information, the support of fragmentation is RECOMMENDED.

6. Bindings for Existing EAP Methods

This section describes how binding tokens can be included in some existing EAP methods that already meet the requirements for secure transport of arbitrary data as specified in Section 5.

6.1. Generalized Pre-Shared Key (GPSK)

EAP-GPSK [RFC5433] defines protected data payloads. The protected channel is available in three of its protocol flows, namely EAP-GPSK-2, EAP-GPSK-3, and EAP-GPSK-4, and provides integrity-protection. If a ciphersuite with encryption is selected (e.g. Ciphersuite 1 in [RFC5433]) the channel also provides confidentiality.

Use by GPSK simply requires instantiation of a new protected data specifier (PData/Specifier):

- 0x0000002 (IANA-TBD): AAA Payload

6.2. Pre-Shared Key (PSK)

EAP-PSK [RFC4764] defines a protected channel. In its standard mode, EAP-PSK provides a protected channel for payloads in two of its flows, namely EAP-PSK-2 and EAP-PSK-3. Features requiring additional flows can be supported in EAP-PSK extended authentication mode. The protected channel is integrity-protected and encrypted.

Use by PSK simply requires instantiation of a new EXT_Type value:

- 0x01 (IANA-TBD): AAA Payload

6.3. Password Authenticated Exchange (PAX)

EAP-PAX [RFC4746] provides an authenticated data exchange (ADE) in three of its protocol flows (EAP-PAX-2, EAP-PAX-3, and EAP-PAX-4). The channel provides integrity-protection but no encryption. Channel binding is explicitly supported in EAP-PAX, in which case the ADE flag needs to be set, and the ADE TYPE needs to be set to 0x02 for
client channel binding data and to 0x03 for server channel binding data.

Additional features could be supported by instantiation of a new ADE type:

- 0x04 (IANA-TBD): AAA Payload

6.4. Tunneled Transport Layer Security (TTLS)

EAP-TTLS [RFC5281] uses Diameter Attribute-Value-Pairs (AVPs) for its messaging. As such it natively supports transporting AAA payloads. Once the TLS tunnel is established, a variable number of flows can be exchanged in the second phase, e.g. to exchange payloads in an integrity-protected and encrypted channel. No protocol changes are necessary.

6.5. Flexible Authentication via Secure Tunneling (FAST)

EAP-FAST [RFC4851] uses Type-Length-Values (TLVs) for its messaging. Once the TLS tunnel is established a variable number of flows can be exchanged in the second phase, e.g. to exchange payloads in an integrity-protected and encrypted channel. To embed binding tokens, a new TLV must be defined:

- 0x15 (IANA-TBD): AAA Payload

7. Security Considerations

Section 3 documented a variety of requirements for EAP methods to transport these AAA payloads. They MUST support identity protection of arbitrary payloads, and SHOULD support confidentiality. Each method’s security considerations section would detail how they achieve those requirements.

The payload includes the EAP Session-ID field. This is to prevent replay attacks. In particular, depending on how an EAP method implements their secured channel, it may or may not be cryptographically bound to the rest of the session. By explicitly including the EAP Session-ID, we prevent replay attacks. Implementations MUST verify the consistency of the Session-ID received in the AAA payloads.

Certainly there are a whole host of issues surrounding the security of what may be contained within the AAA payload format. If the information being transported requires confidentiality, then the method SHOULD support that. Otherwise sensitive data could be
disclosed.

8. IANA Considerations

We require registry from a variety of IANA repositories.

From "EAP-GPSK PData/Specifier":

- 0x0000002 (IANA-TBD): AAA Payload

From "EAP EXT_Type Numbers":

- 0x01 (IANA-TBD): AAA Payload

From "EAP-PAX ADE Type Namespace":

- 0x04 (IANA-TBD): AAA Payload

From "EAP-FAST TLV Types":

- 0x15 (IANA-TBD): AAA Payload

9. References

9.1. Normative References


9.2. Informative References


Authors’ Addresses

T. Charles Clancy
Laboratory for Telecommunications Sciences
US Department of Defense
College Park, MD
USA

Email: clancy@LTSnet.net
Avi Lior (editor)
Bridgewater Systems Corporation
303 Terry Fox Drive
Suite 500
Ottawa, Ontario  K2K 3J1
Canada
Phone: +1 (613) 591-6655
Email: avi@bridgewatersystems.com
URI:  http://www.bridgewatersystems.com/

Glen Zorn
Network Zen
1310 East Thomas Street
Seattle, Washington  98102
US
Phone: +1 (206) 377-9035
Email: gwz@net-zen.net

Katrin Hoeper
Motorola, Inc.
1301 E. Algonquin Road
Schaumburg, Illinois  60196
USA
Email: khoeper@motorola.com