Network Working Group
Request for Comments: 2284
Category: Standards Track

L. Blunk
J. Vollbrecht
Merit Network, Inc.

March 1998

PPP Extensible Authentication Protocol (EAP)

Status of this Memo

Copyright Notice

Abstract

The Point-to-Point Protocol (PPP) [1] provides a standard method for transporting multi-protocol datagrams over point-to-point links.

PPP also defines an extensible Link Control Protocol, which allows negotiation of an Authentication Protocol for authenticating its peer before allowing Network Layer protocols to transmit over the link.

This document defines the PPP Extensible Authentication Protocol.

Table of Contents

1. Introduction

In order to establish communications over a point-to-point link, each end of the PPP link must first send LCP packets to configure the data link during Link Establishment phase. After the link has been established, PPP provides for an optional Authentication phase before proceeding to the Network-Layer Protocol phase.

By default, authentication is not mandatory. If authentication of the link is desired, an implementation MUST specify the Authentication-Protocol Configuration Option during Link Establishment phase.

These authentication protocols are intended for use primarily by hosts and routers that connect to a PPP network server via switched circuits or dial-up lines, but might be applied to dedicated links as well. The server can use the identification of the connecting host or router in the selection of options for network layer negotiations.

This document defines the PPP Extensible Authentication Protocol (EAP). The Link Establishment and Authentication phases, and the Authentication-Protocol Configuration Option, are defined in The Point-to-Point Protocol (PPP) [1].

1.1. Specification of Requirements

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 RFC 2119 [6].

1.2. Terminology

This document frequently uses the following terms:

authenticator
: The end of the link requiring the authentication. The authenticator specifies the authentication protocol to be used in the Configure-Request during Link Establishment phase.

peer
: The other end of the point-to-point link; the end which is being authenticated by the authenticator.

silently discard
: This means the implementation discards the packet without further processing. The implementation SHOULD provide the capability of logging the error, including the contents of the silently discarded packet, and SHOULD record the event in a statistics counter.

displayable message
: This is interpreted to be a human readable string of characters, and MUST NOT affect operation of the protocol. The message encoding MUST follow the UTF-8 transformation format [5].

2. PPP Extensible Authentication Protocol (EAP)

The PPP Extensible Authentication Protocol (EAP) is a general protocol for PPP authentication which supports multiple authentication mechanisms. EAP does not select a specific authentication mechanism at Link Control Phase, but rather postpones this until the Authentication Phase. This allows the authenticator to request more information before determining the specific authentication mechanism. This also permits the use of a "back-end" server which actually implements the various mechanisms while the PPP authenticator merely passes through the authentication exchange.

1. After the Link Establishment phase is complete, the authenticator sends one or more Requests to authenticate the peer. The Request has a type field to indicate what is being requested. Examples of Request types include Identity, MD5-challenge, One-Time Passwords, Generic Token Card, etc. The MD5-challenge type corresponds closely to the CHAP authentication protocol. Typically, the authenticator will send an initial Identity Request followed by one or more Requests for authentication information. However, an initial Identity Request is not required, and MAY be bypassed in cases where the identity is presumed (leased lines, dedicated dial-ups, etc.).

2. The peer sends a Response packet in reply to each Request. As with the Request packet, the Response packet contains a type field which corresponds to the type field of the Request.

3. The authenticator ends the authentication phase with a Success or Failure packet.

Advantages

The EAP protocol can support multiple authentication mechanisms without having to pre-negotiate a particular one during LCP Phase.

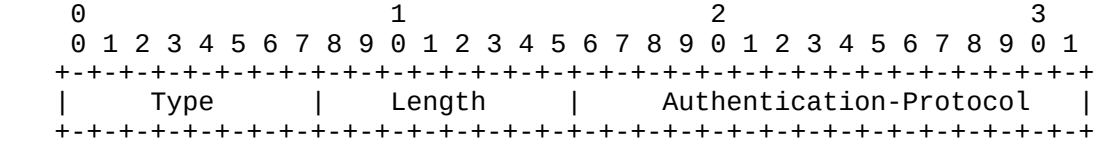
Certain devices (e.g. a NAS) do not necessarily have to understand each request type and may be able to simply act as a passthrough agent for a "back-end" server on a host. The device only need look for the success/failure code to terminate the authentication phase.

Disadvantages

EAP does require the addition of a new authentication type to LCP and thus PPP implementations will need to be modified to use it. It also strays from the previous PPP authentication model of negotiating a specific authentication mechanism during LCP.

2.1. Configuration Option Format

A summary of the Authentication-Protocol Configuration Option format to negotiate the EAP Authentication Protocol is shown below. The fields are transmitted from left to right.



Type

3

Length

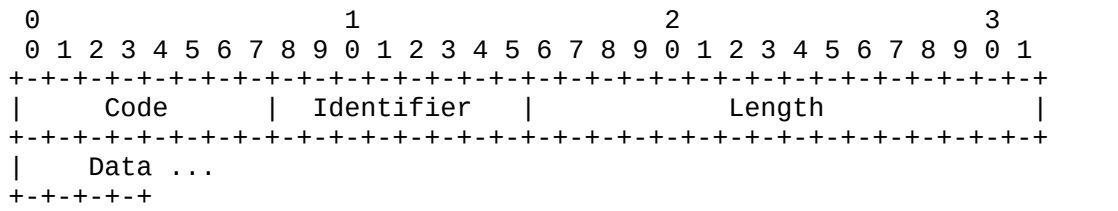
4

Authentication-Protocol

C227 (Hex) for PPP Extensible Authentication Protocol (EAP)

2.2. Packet Format

Exactly one PPP EAP packet is encapsulated in the Information field of a PPP Data Link Layer frame where the protocol field indicates type hex C227 (PPP EAP). A summary of the EAP packet format is shown below. The fields are transmitted from left to right.



Code

The Code field is one octet and identifies the type of EAP packet. EAP Codes are assigned as follows:

1	Request
2	Response
3	Success
4	Failure

Identifier

The Identifier field is one octet and aids in matching responses with requests.

Length

The Length field is two octets and indicates the length of the EAP packet including the Code, Identifier, Length and Data fields. Octets outside the range of the Length field should be treated as Data Link Layer padding and should be ignored on reception.

Data

The Data field is zero or more octets. The format of the Data field is determined by the Code field.

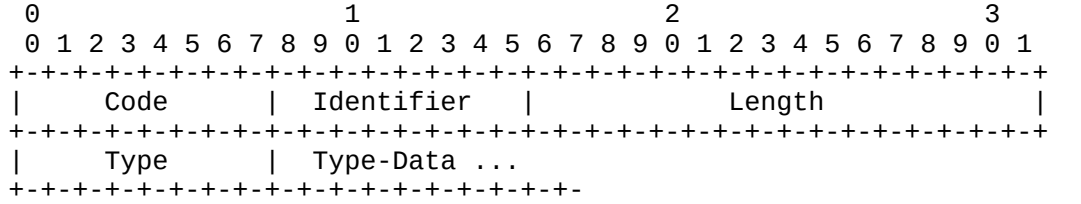
2.2.1. Request and Response

Description

The Request packet is sent by the authenticator to the peer. Each Request has a type field which serves to indicate what is being requested. The authenticator MUST transmit an EAP packet with the Code field set to 1 (Request). Additional Request packets MUST be sent until a valid Response packet is received, or an optional retry counter expires. Retransmitted Requests MUST be sent with the same Identifier value in order to distinguish them from new Requests. The contents of the data field is dependent on the Request type. The peer MUST send a Response packet in reply to a Request packet. Responses MUST only be sent in reply to a received Request and never retransmitted on a timer. The Identifier field of the Response MUST match that of the Request.

Implementation Note: Because the authentication process will often involve user input, some care must be taken when deciding upon retransmission strategies and authentication timeouts. It is suggested a retransmission timer of 6 seconds with a maximum of 10 retransmissions be used as default. One may wish to make these timeouts longer in certain cases (e.g. where Token Cards are involved). Additionally, the peer must be prepared to silently discard received retransmissions while waiting for user input.

A summary of the Request and Response packet format is shown below. The fields are transmitted from left to right.



Code

1 for Request;

2 for Response.

Identifier

The Identifier field is one octet. The Identifier field MUST be the same if a Request packet is retransmitted due to a timeout while waiting for a Response. Any new (non-retransmission) Requests MUST modify the Identifier field. If a peer recieves a duplicate Request for which it has already sent a Response, it MUST resend it's Response. If a peer receives a duplicate Request before it has sent a Response to the initial Request (i.e. it's waiting for user input), it MUST silently discard the duplicate Request.

Length

The Length field is two octets and indicates the length of the EAP packet including the Code, Identifier, Length, Type, and Type-Data fields. Octets outside the range of the Length field should be treated as Data Link Layer padding and should be ignored on reception.

Type

The Type field is one octet. This field indicates the Type of Request or Response. Only one Type MUST be specified per EAP Request or Response. Normally, the Type field of the Response will be the same as the Type of the Request. However, there is also a Nak Response Type for indicating that a Request type is unacceptable to the peer. When sending a Nak in response to a Request, the peer MAY indicate an alternative desired authentication Type which it supports. An initial specification of Types follows in a later section of this document.

Type-Data

The Type-Data field varies with the Type of Request and the associated Response.

2.2.2. Success and Failure

Description

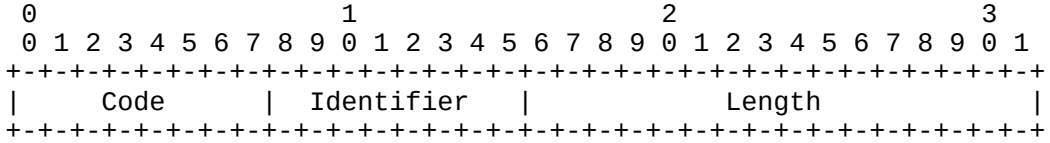
The Success packet is sent by the authenticator to the peer to acknowledge successful authentication. The authenticator MUST transmit an EAP packet with the Code field set to 3 (Success).

If the authenticator cannot authenticate the peer (unacceptable Responses to one or more Requests) then the implementation MUST transmit an EAP packet with the Code field set to 4 (Failure). An

authenticator MAY wish to issue multiple Requests before sending a Failure response in order to allow for human typing mistakes.

Implementation Note: Because the Success and Failure packets are not acknowledged, they may be potentially lost. A peer MUST allow for this circumstance. The peer can use a Network Protocol packet as an alternative indication of Success. Likewise, the receipt of a LCP Terminate-Request can be taken as a Failure.

A summary of the Success and Failure packet format is shown below. The fields are transmitted from left to right.



Code

3 for Success;

4 for Failure.

Identifier

The Identifier field is one octet and aids in matching replies to Responses. The Identifier field MUST match the Indentifier field of the Response packet that it is sent in response to.

Length

4

3. Initial EAP Request/Response Types

This section defines the initial set of EAP Types used in Request/Response exchanges. More Types may be defined in follow-on documents. The Type field is one octet and identifies the structure of an EAP Request or Response packet. The first 3 Types are considered special case Types. The remaining Types define authentication exchanges. The Nak Type is valid only for Response packets, it MUST NOT be sent in a Request. The Nak Type MUST only be

sent in repsonse to a Request which uses an authentication Type code. All EAP implementatins MUST support Types 1-4. These Types, as well as types 5 and 6, are defined in this document. Follow-on RFCs will define additional EAP Types.

1	Identity
2	Notification
3	Nak (Response only)
4	MD5-Challenge
5	One-Time Password (OTP) (RFC 1938)
6	Generic Token Card

3.1. Identity

Description

The Identity Type is used to query the identity of the peer. Generally, the authenticator will issue this as the initial Request. An optional displayable message MAY be included to prompt the peer in the case where there expectation of interaction with a user. A Response MUST be sent to this Request with a Type of 1 (Identity).

Implementation Note: The peer MAY obtain the Identity via user input. It is suggested that the authenticator retry the Indentity Request in the case of an invalid Identity or authentication failure to allow for potential typos on the part of the user. It is suggested that the Identity Request be retried a minimum of 3 times before terminating the authentication phase with a Failure reply. The Notification Request MAY be used to indicate an invalid authentication attempt prior to transmitting a new Identity Request (optionally, the failure MAY be indicated within the message of the new Identity Request itself).

Type

1

Type-Data

This field MAY contain a displayable message in the Request. The Response uses this field to return the Identity. If the Identity is unknown, this field should be zero bytes in length. The field MUST NOT be null terminated. The length of this field is derived from the Length field of the Request/Response packet and hence a null is not required.

3.2. Notification

Description

The Notification Type is optionally used to convey a displayable message from the authenticator to the peer. The peer SHOULD display this message to the user or log it if it cannot be displayed. It is intended to provide an acknowledged notification of some imperative nature. 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.

Type

2

Type-Data

The Type-Data field in the Request contains a displayable message greater than zero octets in length. The length of the message is determined by Length field of the Request packet. The message MUST not be null terminated. A Response MUST be sent in reply to the Request with a Type field of 2 (Notification). The Type-Data field of the Response is zero octets in length. The Response should be sent immediately (independent of how the message is displayed or logged).

3.3. Nak

Description

The Nak Type is valid only in Response messages. It is sent in reply to a Request where the desired authentication Type is unacceptable. Authentication Types are numbered 4 and above. The Response contains the authentication Type desired by the peer.

Type

3

Type-Data

This field MUST contain a single octet indicating the desired authentication type.

3.4. MD5-Challenge

Description

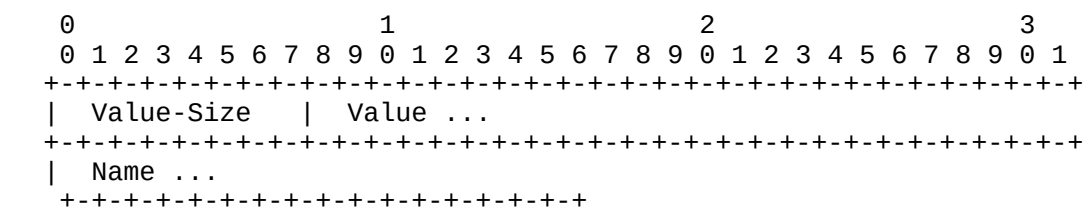
The MD5-Challenge Type is analagous to the PPP CHAP protocol [3] (with MD5 as the specified algorithm). The PPP Challenge Handshake Authentication Protocol RFC [3] should be referred to for further implementation specifics. The Request contains a "challenge" message to the peer. A Repsonse MUST be sent in reply to the Request. The Response MAY be either of Type 4 (MD5-Challenge) or Type 3 (Nak). The Nak reply indicates the peer's desired authentication mechanism Type. All EAP implementations MUST support the MD5-Challenge mechanism.

Type

4

Type-Data

The contents of the Type-Data field is summarized below. For reference on the use of this fields see the PPP Challenge Handshake Authentication Protocol [3].



3.5. One-Time Password (OTP)

Description

The One-Time Password system is defined in "A One-Time Password System" [4]. The Request contains a displayable message containing an OTP challenge. A Repsonse MUST be sent in reply to the Request. The Response MUST be of Type 5 (OTP) or Type 3 (Nak). The Nak reply indicates the peer's desired authentication mechanism Type.

Type

5

Type-Data

The Type-Data field contains the OTP "challenge" as a displayable message in the Request. In the Response, this field is used for the 6 words from the OTP dictionary [4]. The messages MUST not be null terminated. The length of the field is derived from the Length field of the Request/Reply packet.

3.6. Generic Token Card

Description

The Generic Token Card Type is defined for use with various Token Card implementations which require user input. The Request contains an ASCII text message and the Reply contains the Token Card information necessary for authentication. Typically, this would be information read by a user from the Token card device and entered as ASCII text.

Type

6

Type-Data

The Type-Data field in the Request contains a displayable message greater than zero octets in length. The length of the message is determined by Length field of the Request packet. The message MUST not be null terminated. A Response MUST be sent in reply to the Request with a Type field of 6 (Generic Token Card). The Response contains data from the Token Card required for authentication. The length is of the data is determined by the Length field of the Response packet.

Security Considerations

Security issues are the primary topic of this RFC.

The interaction of the authentication protocols within PPP are highly implementation dependent.

For example, upon failure of authentication, some implementations do not terminate the link. Instead, the implementation limits the kind of traffic in the Network-Layer Protocols to a filtered subset, which in turn allows the user opportunity to update secrets or send mail to the network administrator indicating a problem.

There is no provision for retries of failed authentication. However, the LCP state machine can renegotiate the authentication protocol at any time, thus allowing a new attempt. It is recommended that any counters used for authentication failure not be reset until after successful authentication, or subsequent termination of the failed link.

There is no requirement that authentication be full duplex or that the same protocol be used in both directions. It is perfectly acceptable for different protocols to be used in each direction. This will, of course, depend on the specific protocols negotiated.

In practice, within or associated with each PPP server, it is not anticipated that a particular named user would be authenticated by multiple methods. This would make the user vulnerable to attacks which negotiate the least secure method from among a set (such as PAP rather than EAP). Instead, for each named user there should be an indication of exactly one method used to authenticate that user name. If a user needs to make use of different authentication methods under different circumstances, then distinct identities SHOULD be employed, each of which identifies exactly one authentication method.

References

[1] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[2] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.

[3] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996.

[4] Haller, N. and C. Metz, "A One-Time Password System", RFC 1938, May 1996.

[5] Yergeau, F., "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, October 1996.

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

Acknowledgments

This protocol derives much of its inspiration from Dave Carrel's AHA draft as well as the PPP CHAP protocol [3]. Bill Simpson provided much of the boilerplate used throughout this document. Al Rubens (Merit) also provided valuable feedback.

Chair's Address

The working group can be contacted via the current chair:

Karl F. Fox
Ascend Communications, Inc.
655 Metro Place South, Suite 370
Dublin, Ohio 43017

EMail: karl@ascend.com
Phone: +1 614 760 4041
Fax: +1 614 760 4050

Authors' Addresses

Larry J. Blunk
Merit Network, Inc.
4251 Plymouth Rd., Suite C
Ann Arbor, MI 48105

EMail: ljb@merit.edu
Phone: 734-763-6056
FAX: 734-647-3185

John R. Vollbrecht
Merit Network, Inc.
4251 Plymouth Rd., Suite C
Ann Arbor, MI 48105

EMail: jrv@merit.edu
Phone: +1-313-763-1206
FAX: +1-734-647-3185

Full Copyright Statement