Network Working Group
Request for Comments: 1994
Obsoletes: 1334
Category: Standards Track

W. Simpson
DayDreamer
August 1996

PPP Challenge Handshake Authentication Protocol (CHAP)

Status of this Memo

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 a method for Authentication using PPP, which uses a random Challenge, with a cryptographically hashed Response which depends upon the Challenge and a secret key.

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 a PPP authentication protocol. 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.

MUST	This word, or the adjective "required", means that the definition is an absolute requirement of the specification.
MUST NOT	This phrase means that the definition is an absolute prohibition of the specification.
SHOULD	This word, or the adjective "recommended", means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications must be understood and carefully weighed before choosing a different course.
MAY	This word, or the adjective "optional", means that this item is one of an allowed set of alternatives. An implementation which does not include this option MUST be prepared to interoperate with another implementation which does include the option.

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.

2. Challenge-Handshake Authentication Protocol

The Challenge-Handshake Authentication Protocol (CHAP) is used to periodically verify the identity of the peer using a 3-way handshake. This is done upon initial link establishment, and MAY be repeated anytime after the link has been established.

1. After the Link Establishment phase is complete, the authenticator sends a "challenge" message to the peer.

2. The peer responds with a value calculated using a "one-way hash" function.

3. The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authentication is acknowledged; otherwise the connection SHOULD be terminated.

4. At random intervals, the authenticator sends a new challenge to the peer, and repeats steps 1 to 3.

2.1. Advantages

CHAP provides protection against playback attack by the peer through the use of an incrementally changing identifier and a variable challenge value. The use of repeated challenges is intended to limit the time of exposure to any single attack. The authenticator is in control of the frequency and timing of the challenges.

This authentication method depends upon a "secret" known only to the authenticator and that peer. The secret is not sent over the link.

Although the authentication is only one-way, by negotiating CHAP in both directions the same secret set may easily be used for mutual authentication.

Since CHAP may be used to authenticate many different systems, name fields may be used as an index to locate the proper secret in a large table of secrets. This also makes it possible to support more than one name/secret pair per system, and to change the secret in use at any time during the session.

2.2. Disadvantages

CHAP requires that the secret be available in plaintext form. Irreversably encrypted password databases commonly available cannot be used.

It is not as useful for large installations, since every possible secret is maintained at both ends of the link.

> Implementation Note: To avoid sending the secret over other links in the network, it is recommended that the challenge and response values be examined at a central server, rather than each network access server. Otherwise, the secret SHOULD be sent to such servers in a reversably encrypted form. Either case requires a trusted relationship, which is outside the scope of this specification.

2.3. Design Requirements

The CHAP algorithm requires that the length of the secret MUST be at least 1 octet. The secret SHOULD be at least as large and unguessable as a well-chosen password. It is preferred that the secret be at least the length of the hash value for the hashing algorithm chosen (16 octets for MD5). This is to ensure a sufficiently large range for the secret to provide protection against exhaustive search attacks.

The one-way hash algorithm is chosen such that it is computationally infeasible to determine the secret from the known challenge and response values.

Each challenge value SHOULD be unique, since repetition of a challenge value in conjunction with the same secret would permit an attacker to reply with a previously intercepted response. Since it is expected that the same secret MAY be used to authenticate with servers in disparate geographic regions, the challenge SHOULD exhibit global and temporal uniqueness.

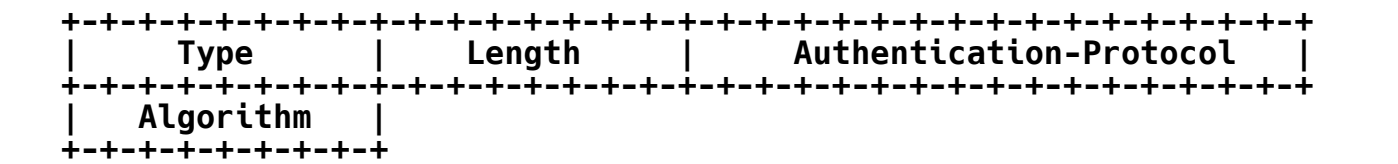
Each challenge value SHOULD also be unpredictable, least an attacker trick a peer into responding to a predicted future challenge, and then use the response to masquerade as that peer to an authenticator.

Although protocols such as CHAP are incapable of protecting against realtime active wiretapping attacks, generation of unique unpredictable challenges can protect against a wide range of active attacks.

A discussion of sources of uniqueness and probability of divergence is included in the Magic-Number Configuration Option [1].

3. Configuration Option Format

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



Type

3

Length

5

Authentication-Protocol

c223 (hex) for Challenge-Handshake Authentication Protocol.

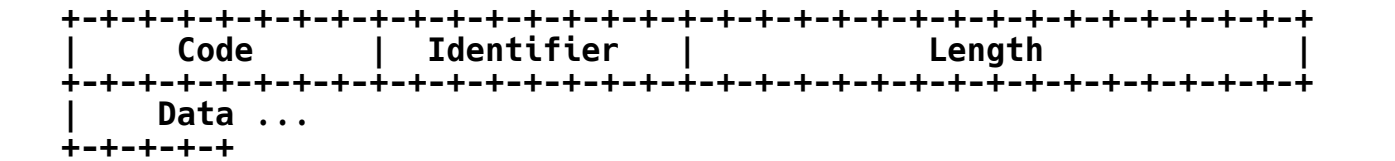
Algorithm

The Algorithm field is one octet and indicates the authentication method to be used. Up-to-date values are specified in the most recent "Assigned Numbers" [2]. One value is required to be implemented:

5 CHAP with MD5 [3]

4. Packet Format

Exactly one Challenge-Handshake Authentication Protocol packet is encapsulated in the Information field of a PPP Data Link Layer frame where the protocol field indicates type hex c223 (Challenge-Handshake Authentication Protocol). A summary of the CHAP 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 CHAP packet. CHAP Codes are assigned as follows:

1	Challenge
2	Response
3	Success
4	Failure

Identifier

The Identifier field is one octet and aids in matching challenges, responses and replies.

Length

The Length field is two octets and indicates the length of the CHAP 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.

4.1. Challenge and Response

Description

The Challenge packet is used to begin the Challenge-Handshake Authentication Protocol. The authenticator MUST transmit a CHAP packet with the Code field set to 1 (Challenge). Additional Challenge packets MUST be sent until a valid Response packet is received, or an optional retry counter expires.

A Challenge packet MAY also be transmitted at any time during the Network-Layer Protocol phase to ensure that the connection has not been altered.

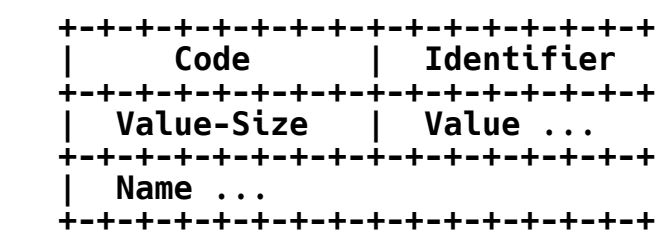
The peer SHOULD expect Challenge packets during the Authentication phase and the Network-Layer Protocol phase. Whenever a Challenge packet is received, the peer MUST transmit a CHAP packet with the Code field set to 2 (Response).

Whenever a Response packet is received, the authenticator compares the Response Value with its own calculation of the expected value. Based on this comparison, the authenticator MUST send a Success or Failure packet (described below).

Implementation Notes: Because the Success might be lost, the authenticator MUST allow repeated Response packets during the Network-Layer Protocol phase after completing the Authentication phase. To prevent discovery of alternative Names and Secrets, any Response packets received having the current Challenge Identifier MUST return the same reply Code previously returned for that specific Challenge (the message portion MAY be different). Any Response packets received during any other phase MUST be silently discarded.

When the Failure is lost, and the authenticator terminates the link, the LCP Terminate-Request and Terminate-Ack provide an alternative indication that authentication failed.

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



Code

1 for Challenge;

2 for Response.

Identifier

The Identifier field is one octet. The Identifier field MUST be changed each time a Challenge is sent.

The Response Identifier MUST be copied from the Identifier field of the Challenge which caused the Response.

Value-Size

This field is one octet and indicates the length of the Value field.

Value

The Value field is one or more octets. The most significant octet is transmitted first.

The Challenge Value is a variable stream of octets. The importance of the uniqueness of the Challenge Value and its relationship to the secret is described above. The Challenge Value MUST be changed each time a Challenge is sent. The length of the Challenge Value depends upon the method used to generate the octets, and is independent of the hash algorithm used.

The Response Value is the one-way hash calculated over a stream of octets consisting of the Identifier, followed by (concatenated with) the "secret", followed by (concatenated with) the Challenge Value. The length of the Response Value depends upon the hash algorithm used (16 octets for MD5).

Name

The Name field is one or more octets representing the identification of the system transmitting the packet. There are no limitations on the content of this field. For example, it MAY contain ASCII character strings or globally unique identifiers in ASN.1 syntax. The Name should not be NUL or CR/LF terminated. The size is determined from the Length field.

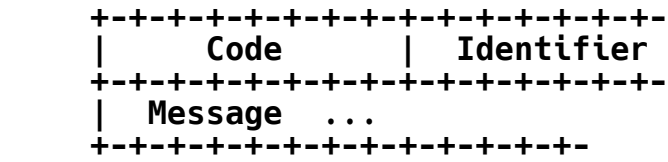
4.2. Success and Failure

Description

If the Value received in a Response is equal to the expected value, then the implementation MUST transmit a CHAP packet with the Code field set to 3 (Success).

If the Value received in a Response is not equal to the expected value, then the implementation MUST transmit a CHAP packet with the Code field set to 4 (Failure), and SHOULD take action to terminate the link.

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 requests and replies. The Identifier field MUST be copied from the Identifier field of the Response which caused this reply.

Message

The Message field is zero or more octets, and its contents are implementation dependent. It is intended to be human readable, and MUST NOT affect operation of the protocol. It is recommended that the message contain displayable ASCII characters 32 through 126 decimal. Mechanisms for extension to other character sets are the topic of future research. The size is determined from the Length field.

Security Considerations

Security issues are the primary topic of this RFC.

The interaction of the authentication protocols within PPP are highly implementation dependent. This is indicated by the use of SHOULD throughout the document.

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 re-tries 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.

The secret SHOULD NOT be the same in both directions. This allows an attacker to replay the peer's challenge, accept the computed response, and use that response to authenticate.

In practice, within or associated with each PPP server, there is a database which associates "user" names with authentication information ("secrets"). 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 CHAP). If the same

secret was used, PAP would reveal the secret to be used later with CHAP.

Instead, for each user name 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 user names SHOULD be employed, each of which identifies exactly one authentication method.

Passwords and other secrets should be stored at the respective ends such that access to them is as limited as possible. Ideally, the secrets should only be accessible to the process requiring access in order to perform the authentication.

The secrets should be distributed with a mechanism that limits the number of entities that handle (and thus gain knowledge of) the secret. Ideally, no unauthorized person should ever gain knowledge of the secrets. Such a mechanism is outside the scope of this specification.

Acknowledgements

David Kaufman, Frank Heinrich, and Karl Auerbach used a challenge handshake at SDC when designing one of the protocols for a "secure" network in the mid-1970s. Tom Bearson built a prototype Sytek product ("Poloneous"?) on the challenge-response notion in the 1982-83 timeframe. Another variant is documented in the various IBM SNA manuals. Yet another variant was implemented by Karl Auerbach in the Telebit NetBlazer circa 1991.

Kim Toms and Barney Wolff provided useful critiques of earlier versions of this document.

Special thanks to Dave Balenson, Steve Crocker, James Galvin, and Steve Kent, for their extensive explanations and suggestions. Now, if only we could get them to agree with each other.

References

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

[2] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, USC/Information Sciences Institute, October 1994.

[3] Rivest, R., and S. Dusse, "The MD5 Message-Digest Algorithm", MIT Laboratory for Computer Science and RSA Data Security, Inc., RFC 1321, April 1992.

Contacts

Comments should be submitted to the ietf-ppp@merit.edu mailing list.

This document was reviewed by the Point-to-Point Protocol Working Group of the Internet Engineering Task Force (IETF). The working group can be contacted via the current chair:

Karl Fox
Ascend Communications
3518 Riverside Drive, Suite 101
Columbus, Ohio 43221

karl@MorningStar.com
karl@Ascend.com

Questions about this memo can also be directed to:

William Allen Simpson
DayDreamer
Computer Systems Consulting Services
1384 Fontaine
Madison Heights, Michigan 48071

wsimpson@UMich.edu
wsimpson@GreenDragon.com (preferred)