Network Working Group
INTERNET-DRAFT
Category: Informational

D. Potter
J. Zamick
Cisco Systems

January 2002

PPP EAP MS-CHAP-V2 Authentication Protocol

<draft-dpotter-pppext-eap-mschap-01.txt>

Status of this Memo

1. Abstract

This document specifies an Extensible Authentication Protocol (EAP) mechanism for authentication using the Microsoft Challenge-Handshake Authentication Protocol (Version 2).

MS-CHAP-v2 provides authentication functionality consistent with LAN-based methods including password change sequences. Mutual authentication is provided for by the inclusion of an authenticator packet returned to the client after a successful server authentication.

2. Introduction

Prior to EAP [3] network access support for a specific authentication protocol had to be engineered in at least three places, the peer, the access device (AAA client) and the AAA server. EAP has significantly simplified this scheme by making the password protocol 'opaque' to the access device - support for EAP by an access device therefore infers support for all EAP types. The 802.1x protocol which

facilitates the deployment of user AAA on broadcast media networks such as wireless and Ethernet relies upon EAP as its password protocol vehicle as there is no point-to-point protocol negotiation as with conventional dial-in or VPN clients.

EAP prescribes mandatory support for EAP-MD5-CHAP and whilst this has been used widely in the past, the implementational requirement for the back-end server to hold the users password either in clear text or a reversible encryption is seen as a potential drawback.

EAP-TLS [2] is likely to achieve widespread adoption in the future, however there are currently issues over the cost and ease of deployment into existing network infrastructure.

Another very widely used authentication protocol that is not currently addressed by EAP is MS-CHAP [6]. The lack of support for MS-CHAP in EAP significantly reduces the utility of EAP since MS-CHAP provides an additional facility for password change and expiry notification (aging).

This document describes a method for encapsulating MS-CHAP v2 in EAP and extends MS-CHAP by allowing the peer to request a password change after a successful authentication.

2.1. Requirements language

In this document, the key words "MAY", "MUST, "MUST NOT", "optional", "recommended", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [4].

2.2 Definitions

AAA Server
Authentication, Authorisation and Accounting Server

Authenticator
Access device to which client desires connection

EAP
Extensible Authentication Protocol [3]

EAP Server
For the purpose of this document, see AAA Server

Peer/Client
Device (software or hardware) requiring access to/via the Authenticator.

3. Protocol overview

3.1. Overview of the EAP-MS-CHAP-V2 Authentication

As described in [3], the EAP-MS-CHAP-V2 conversation will typically begin with the authenticator and the peer negotiating EAP. The authenticator will then typically send an EAP-Request/Identity packet to the peer, and the peer will respond with an EAP-Response/Identity packet to the authenticator, containing the clients userId.

Unless otherwise stated, all EAP challenge/response messages MUST have an EAP-Type of EAP-MS-CHAP-V2. EAP Success and Failure messages have the EAP Message Code set to one of these values respectively.

Upon receipt of the users identity the EAP Server MUST respond with a EAP-MS-CHAP-V2 Challenge request. The MS-CHAP Challenge as defined in [6] should be cryptographically random. The EAP Server remembers the challenge for later authentication of the computed MS-CHAP Response.

The EAP Client MUST then reply with the MS-CHAP Response generated from the users credentials. The EAP Server re-computes the MS-CHAP Response, or devolves this operation to another back-end server. The EAP Server MUST then send an EAP Request with either MS-CHAP Success or MS-CHAP Failure packets depending on the result of the authentication.

The EAP Server SHOULD ensure the MS-CHAP Response is actually a version 2 formatted response and not version 1. In the version 2 packet the first 16 octets of the response contain a random challenge from the the client. The next 8 octets MUST be zero, otherwise the EAP Server SHOULD immediately send an EAP Failure message.

3.1.1 Successful Authentication

If the MS-CHAP Response was valid the EAP Server MAY send the MS-CHAP Success packet, however it may apply other local policy conditions resulting in a rejection (see 3.1.2)

To provide mutual authentication the MS-CHAP Success packet MUST be validated by the client as per 3.1.5. If the MS-CHAP Success packet is valid the client MUST send an EAP Response containing an Ack packet.

The EAP server MUST then send an EAP-Success message to the client/peer. The EAP authentication is now complete.

3.1.2 Failed Authentication (or Authorisation)

If the MS-CHAP Response was invalid the EAP Server MUST send the MS-CHAP Failure packet indicating the rejection (MS-CHAP error code 691 - ERROR_AUTHENTICATION_FAILURE)

If the MS-CHAP Response was actually valid but the user has failed some other authorisation policy then the EAP Server MAY indicate one of other MS-CHAP failure codes:

646 ERROR_RESTRICTED_LOGON_HOURS
647 ERROR_ACCT_DISABLED
649 ERROR_NO_DIALIN_PERMISSION

Upon receipt of the MS-CHAP Failure packet the client MUST send an EAP Response containing the MS-CHAP Ack packet. After receiving an MS-CHAP Ack to the MS-CHAP Failure packet, the EAP server MUST then send an EAP-Failure message to the client/peer.

The EAP authentication is now complete.

3.1.3 Password Expired

If the users password has expired, the EAP Server MAY send an MS-CHAP Failure packet indicating that the users password has expired (MS-CHAP error code 648 - ERROR_PASSWD_EXPIRED). If the EAP Server does not support password change then it MUST send a failed authentication result instead (MS-CHAP error code 691 - ERROR_AUTHENTICATION_FAILURE).

On receipt of the MS-CHAP Failure packet indicating that a password change is required, the client MUST send an EAP Response containing the MS-CHAP Ack packet.

The EAP Server MUST then generate a new random challenge and issue an EAP Request containing an MS-CHAP Challenge packet.

The client will then obtain a new password (in most cases directly from the user) and use the algorithms described in [6] to create a MS-CHAP Change Password packet. This is then sent in an EAP-Response message.

The EAP Server will then process the MS-CHAP Change Password packet and MUST issue an MS-CHAP Success or Failure packet. If the password change was unsuccessful, the MS-CHAP Failure packet MUST indicate this using MS-CHAP error code 709 (ERROR_CHANGING_PASSWORD). The EAP Server MAY start a new password change sequence by formatting the MS-CHAP Failure packet to indicate password expiry (MS-CHAP error

code 648 - ERROR_PASSWD_EXPIRED).

In response to the MS-CHAP Success or MS-CHAP Failure packets, the client MUST send an EAP Response containing the MS-CHAP Ack packet. Additionally, an MS-CHAP Success packet must be validated as per 3.1.5.

If the password change sequence was successful, the EAP Server MUST then send an EAP-Success message. If the password change sequence was unsuccessful the EAP Server MUST send an EAP-Failure message. If the EAP Server had decided to start a new password change sequence then it MUST generate a new random challenge and issue an EAP Request containing an MS-CHAP Challenge packet.

3.1.4 Successful Authentication - Client Wishes to Change Password

Upon receipt of the MS-CHAP Success packet (following a valid MS-CHAP authentication and validation of the Success packet as per 3.1.5) the client MAY optionally request that the users password is changed. In response to the MS-CHAP Success packet the client MAY send an EAP Response containing an MS-CHAP Failure packet indicating a password change is required (MS-CHAP error code 648 - ERROR_PASSWD_EXPIRED).

The EAP Server MAY choose to honour the request, in which case it starts a password change sequence by creating a random challenge and sending an EAP Request with a MS-CHAP Challenge packet as per 3.1.3. This ultimately ends with the EAP Server sending an EAP-Success or EAP-Failure message depending on the result of the password change sequence.

If the EAP Server does not support client requested password changes it MUST respond with an MS-CHAP Failure packet indicating the password change request has not been allowed (MS-CHAP error code 709 - ERROR_CHANGING_PASSWORD). The Client MUST then respond with an Ack packet. Finally the EAP Server SHOULD send an EAP-Success message.

3.1.5 Mutual Authentication

As described in [6] the MS-CHAP Success packet MUST be validated by the client in accordance with the algorithms described therein. If the Success packet is invalid the client MUST end the session.

3.2. Fragmentation

The maximum size of an EAP-MSCHAP-V2 record will be 586 bytes (during a password change conversation [6]) and therefore does not require a specific fragmentation scheme other than what is provided for in [8]

3.3. Session Encryption/Compression

PPP [1] encryption and compression are catered for using MPPE-based methods [5,7,9]. In general terms the AAA/EAP server will generate an MPPE session key during the MSCHAP-v2 authentication process [7]. The MPPE key information is then returned to the Authenticator [5].

3.4. Examples

In the case where the EAP-MS-CHAP-V2 authentication is successful, the conversation will appear as follows:

```
Authenticating Peer     Authenticator
-------------------     -------------
                        <- PPP EAP-Request/
                        Identity
PPP EAP-Response/
Identity (MyID) ->
                        <- PPP EAP-Request/
                        EAP-Type=EAP-MS-CHAP-V2
                        (Challenge)
PPP EAP-Response/
EAP-Type=EAP-MS-CHAP-V2
(Response)->
                        <- PPP EAP-Request/
                        EAP-Type=EAP-MS-CHAP-V2
                        (Success)

PPP EAP-Response/
EAP-Type=EAP-MS-CHAP-V2
(Ack) ->

                        <- PPP EAP-Success
```

In the case where the EAP-MS-CHAP-V2 authentication not successful, the conversation will appear as follows:

```
Authenticating Peer     Authenticator
-------------------     -------------
                        <- PPP EAP-Request/
                        Identity
PPP EAP-Response/
Identity (MyID) ->
                        <- PPP EAP-Request/
                        EAP-Type=EAP-MS-CHAP-V2
                        (Challenge)
```

```
   PPP EAP-Response/
   EAP-Type=EAP-MS-CHAP-V2
   (Response)->
                           <- PPP EAP-Request/
                           EAP-Type=EAP-MS-CHAP-V2
                           (Failure = failed)

   PPP EAP-Response/
   EAP-Type=EAP-MS-CHAP-V2
   (Ack) ->

                           <- PPP EAP-Failure
```

In the case where the users password has expired, the conversation will appear as follows:

```
   Authenticating Peer     Authenticator
   -------------------     -------------
                           <- PPP EAP-Request/
                           Identity
   PPP EAP-Response/
   Identity (MyID) ->
                           <- PPP EAP-Request/
                           EAP-Type=EAP-MS-CHAP-V2
                           (Challenge)
   PPP EAP-Response/
   EAP-Type=EAP-MS-CHAP-V2
   (Response)->

                           <- PPP EAP-Request/
                           EAP-Type=EAP-MS-CHAP-V2
                           (Failure = password expired)

   PPP EAP-Response/
   EAP-Type=EAP-MS-CHAP-V2
   (Ack) ->
                           <- PPP EAP-Request/
                           EAP-Type=EAP-MS-CHAP-V2
                           (Challenge)

   PPP EAP-Response/
   EAP-Type=EAP-MS-CHAP-V2
   (Change Password)->
                           <- PPP EAP-Request/
                           EAP-Type=EAP-MS-CHAP-V2
                           (Success)
```

```
   PPP EAP-Response/
   EAP-Type=EAP-MS-CHAP-V2
   (Ack) ->
                           <- PPP EAP-Success
```

Note that the AAA/EAP Server may choose to re-iterate around the password change cycle if required, for example if the new password did not meet some password validation policy.

In the case where the EAP-MS-CHAP-V2 authentication was successful, and the client wishes to change the users password, the conversation will appear as follows:

```
   Authenticating Peer     Authenticator
   -------------------     -------------
                           <- PPP EAP-Request/
                           Identity
   PPP EAP-Response/
   Identity (MyID) ->
                           <- PPP EAP-Request/
                           EAP-Type=EAP-MS-CHAP-V2
                           (Challenge)
   PPP EAP-Response/
   EAP-Type=EAP-MS-CHAP-V2
   (Response)->

                           <- PPP EAP-Request/
                           EAP-Type=EAP-MS-CHAP-V2
                           (Success)

   PPP EAP-Response/
   EAP-Type=EAP-MS-CHAP-V2
   (Failure = password expired) ->

                           <- PPP EAP-Request/
                           EAP-Type=EAP-MS-CHAP-V2
                           (Challenge)

   PPP EAP-Response/
   EAP-Type=EAP-MS-CHAP-V2
   (Change Password)->
                           <- PPP EAP-Request/
                           EAP-Type=EAP-MS-CHAP-V2
                           (Success)
   PPP EAP-Response/
   EAP-Type=EAP-MS-CHAP-V2
   (Ack) ->
                           <- PPP EAP-Success
```

4. Detailed description of the EAP-MS-CHAP-V2 protocol

4.1. PPP EAP MS-CHAP-V2 Packet Format

A summary of the PPP EAP MS-CHAP-V2 Request/Response packet format is shown below. The fields are transmitted from left to right.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Code      |   Identifier  |            Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |        Data...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

1 - Request
2 - Response

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, Type, 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.

Type

29 - EAP MS-CHAP V2

Data

The format of the Data field is determined by the Code field.

4.2. PPP EAP MS-CHAP-V2 Request Packet

A summary of the PPP EAP MS-CHAP-V2 Request packet format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |   Identifier  |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |  MS-CHAP Type |         MS-CHAP Data...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

1

Identifier

The Identifier field is one octet and aids in matching responses with requests. The Identifier field MUST be changed on each Request packet.

Length

The Length field is two octets and indicates the length of the EAP packet including the Code, Identifier, Length, Type, MS-CHAP Type and MS-CHAP Data fields.

Type

29 - EAP MS-CHAP V2

MS-CHAP Type

This value defines the content of the MS-CHAP Data defined in [6] with the exception of Ack which is added for the purposes of synchronisation between the peer and the AAA/EAP Server.

To aid clarity the RADIUS VSA names from [5] are given in parenthesis.

0 for Ack - length of MS-CHAP data is 0
1 for Challenge Packet (MS-CHAP-Challenge)
2 for Success Packet (MS-CHAP2-Success)
3 for Failure Packet (MS-CHAP-Error)

MS-CHAP Data

The MS-CHAP data consists of an encapsulated MS-CHAP-V2 packet as defined in [6]

4.3. PPP EAP MS-CHAP-V2 Response Packet

A summary of the PPP EAP MS-CHAP-V2 Response packet format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |   Identifier  |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      | MS-CHAP Type  |         MS-CHAP Data
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

2

Identifier

The Identifier field is one octet and MUST match the Identifier field from the corresponding request.

Length

The Length field is two octets and indicates the length of the EAP packet including the Code, Identifier, Length, Type, MS-CHAP Type and MS-CHAP Data fields.

Type

29 - EAP MS-CHAP V2

MS-CHAP Type

This value defines the content of the MS-CHAP Data defined in [6]. To aid clarity the RADIUS VSA names from [5] are given in parenthesis.

1 for Response Packet (MS-CHAP2-Response)
2 for Change Password Packet (MS-CHAP-CPW + MS-CHAP-NT-Enc-PW)
3 for Failure Packet (MS-CHAP-Error)

5. Security Considerations

Various cryptanalysis have been published on MS-CHAP versions 1 and 2 and most conclude that version 2 has overcome most of the weaknesses originally found in version 1. As noted in [6] a major issue is the use of weak passwords making the protocol more vulnerable to dictionary based attacks.

Version rollback (to MSCHAP v1) is avoided by the EAP/AAA Server ensuring the format of the MS-CHAP response matches that defined in [6].

Using a toolkit to generate cryptographically random challenges should also increase the overall security of the protocol.

The use of MPPE for session keys will not be as strong as those generated by some other EAP protocols such as EAP-TLS.

6. References

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

[2] Aboba, B., Simon, D., "PPP EAP TLS Authentication Protocol" RFC 2716, October 1999.

[3] Blunk, L. and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)", RFC 2284, March 1998.

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

[5] Zorn, G., "Microsoft Vendor-specific RADIUS Attributes", RFC 2548, March 1999.

[6] Zorn, G., "Microsoft PPP CHAP Extensions, Version 2", RFC 2759, January 2000.

[7] Zorn, G., "MPPE Key Derivation", draft-ietf-pppext-mppe-keys-03, October 2000.

[8] Rigney, C, et al, "RADIUS Extensions", RFC 2869, June 2000.

[9] Zorn, G., Pall G., "Microsoft Point-To-Point Encryption (MPPE) Protocol", draft-ietf-pppext-mppe-04, October 1999.

7. Acknowledgments

Thanks to Chris Murray, Ilan Frenkel, John Schnizlein and Glen Zorn for their comments and help.

8. Authors' Addresses

Darran Potter
Cisco Systems Ltd
New Square Park
Bedfont Lakes
Middlesex, TW14 8HA
UK

EMail: dpotter@cisco.com

John Zamick
Cisco Systems Ltd
New Square Park
Bedfont Lakes
Middlesex, TW14 8HA
UK

EMail: jzamick@cisco.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.