Network Working Group
Request for Comments: 5421
Category: Informational

N. Cam-Winget
H. Zhou
Cisco Systems

March 2009

Basic Password Exchange within the Flexible Authentication via Secure Tunneling Extensible Authentication Protocol (EAP-FAST)

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

IESG Note

EAP-FAST has been implemented by many vendors and it is used in the Internet. Publication of this specification is intended to promote interoperability by documenting current use of existing EAP methods within EAP-FAST.

The EAP method EAP-FAST-GTC reuses the EAP type code assigned to EAP-GTC (6). The reuse of previously assigned EAP Type Codes is incompatible with EAP method negotiation as defined in RFC 3748.

Since EAP-GTC does not support method-specific version negotiation, the use of EAP-FAST-GTC is implied when used inside the EAP-FAST tunnel during authentication. This behavior may cause problems in implementations where the use of another vendor's EAP-GTC is required. Since such support requires special case execution of a method within a tunnel, it also complicates implementations that use the same method code both within and outside of the tunnel method. If EAP-FAST were to be designed today, these difficulties could be avoided by utilization of unique EAP Type codes. Given these issues, assigned method types must not be re-used with different meaning inside tunneled methods in the future.

Abstract

The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol (EAP-FAST) method enables secure communication between a peer and a server by using Transport Layer Security (TLS) to establish a mutually authenticated tunnel. Within this tunnel, a basic password exchange, based on the Generic Token Card method (EAP-GTC), may be executed to authenticate the peer.

Table of Contents

1. Introduction

EAP-FAST [RFC4851] is an EAP method that can be used to mutually authenticate a peer and server. This document describes the EAP-FAST inner EAP method, EAP-FAST-GTC, which is used to authenticate the peer through a basic password exchange. EAP-FAST-GTC was developed to support using cleartext passwords to authenticate to legacy user databases, to facilitate password change, and to support one time password features such as new pin mode. Message exchanges, including user credentials, are cleartext strings transferred within the encrypted TLS tunnel and thus are considered secure. For historical reasons, EAP-FAST-GTC uses EAP Type 6, originally allocated to EAP-GTC [RFC3748]. Note that EAP-FAST-GTC payloads used in EAP-FAST require specific formatting and therefore will not necessarily be compatible with EAP-GTC mechanisms used outside of EAP-FAST. To avoid interference between these two methods, EAP-FAST-GTC MUST NOT be used outside an EAP-FAST tunnel, and EAP-GTC MUST NOT be used inside an EAP-FAST tunnel. All EAP-FAST-GTC packets sent within the TLS tunnel must be encapsulated in EAP Payload TLVs, described in [RFC4851].

It is assumed that a reader of this document is familiar with EAP-FAST [RFC4851].

1.1. Specification Requirements

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].

2. EAP-FAST GTC Authentication

All EAP-FAST-GTC packets inside EAP-FAST other than the empty acknowledgment packet MUST follow the "LABEL=Value" format. All Labels are in ASCII text and SHALL NOT contain the space character. Currently, three Labels are defined:

- "CHALLENGE", the server request packet MUST be in the form of "CHALLENGE=Value", where Value is the server challenge, such as "please enter your password".

- "RESPONSE", the peer response packet MUST be in the form of "RESPONSE=Value", where Value is the peer response.

- "E", the server failure packet MUST be in the form of "E=Value", where Value is the error message generated by the server.

If the peer or the server receives an EAP-FAST-GTC request or response that is not in the format specified above, it SHOULD fail the authentication by sending a Result TLV with a failure.

After the TLS encryption tunnel is established and EAP-FAST Authentication phase 2 starts, the EAP server sends an EAP-FAST-GTC Request, which contains a server challenge. The server challenge is a displayable message for use by the peer to prompt the user.

A peer MAY prompt the user for the user credentials, or decide to use the user credentials gained through some other means without prompting the user. The peer sends the user credentials back in the EAP-FAST-GTC Response using the following format:

```
"RESPONSE=user@example.com\0secret"
```

where "user@example.com" is the actual username and "secret" is the actual password. The NULL character "\0" is used to separate the username and password.

The username and password are included in a single message in the first response packet as an optimization by eliminating the inner method EAP-Identity exchange to save an extra round trip.

Once the EAP-FAST server receives the user credentials, it SHOULD first validate the user identity with the Initiator ID (I-ID) [RFC5422] in the PAC-Opaque (Protected Access Credential) and if it matches, it will continue to authenticate the user with internal or external user databases.

Additional exchanges MAY occur between the EAP-FAST server and peer to facilitate various user authentications. The EAP-FAST server might send additional challenges to prompt the peer for additional information, such as a request for the next token or a new pin in the one time password case, or a server failure packet to indicate an error. The peer displays the prompt to the user again and sends back the needed information in an EAP-FAST-GTC Response. The exchange ends when a Result TLV is received.

An EAP-FAST-GTC server implementation within EAP-FAST uses the following format to indicate an error if an authentication fails:

```
"E=eeeeeeeeee R=r M=<msg>"
```

where:

The "eeeeeeeeee" is the ASCII representation of a decimal error code corresponding to one of those listed below, though peer implementations SHOULD deal with codes not on this list gracefully.

The error code need not be 10 digits long.

Below is a complete list of predefined error codes:

- 646 ERROR_RESTRICTED_LOGON_HOURS

  Indicates that access is attempted outside the allowed hours. Peer implementations SHOULD display the error message to the user and ask the user to try at a later time.

- 647 ERROR_ACCT_DISABLED

  Indicates that the requested account is disabled. Peer implementations SHOULD display the error message to the user, which helps the user to resolve the issue with the administrator.

- 648 ERROR_PASSWD_EXPIRED

  Indicates that the password has expired and a password change is required. Peer implementations SHOULD prompt the user for a new password and send back the new password in the peer response packet.

- 649 ERROR_NO_DIALIN_PERMISSION

  Indicates that access has been denied due to lack of dial-in permission. Peer implementations SHOULD display the error message to the user, which helps the user to resolve the issue with the administrator.

- 691 ERROR_AUTHENTICATION_FAILURE

  Indicates that there was authentication failure due to an incorrect username or password. Based on the retry flag described below, peer implementations MAY prompt the user again for a new set of username and password or simply send back an empty acknowledgment packet to acknowledge the failure and go into the termination phase of the authentication session.

- 709 ERROR_CHANGING_PASSWORD

  Indicates that the password change failed, most likely because the new password fails to meet the password complexity policy. Peer implementations SHOULD display the error message and prompt the user again for the new password.

- 755 ERROR_PAC_I-ID_NO_MATCH

  Indicates that the PAC used to establish the EAP-FAST session cannot be used to authenticate to this user account. Based on the retry flag described below, peer implementations MAY prompt the user again for a new set of username and password or simply send back an empty acknowledgment packet to acknowledge the failure and go into the termination phase of the authentication session.

The "r" is a single character ASCII flag set to ’1’ if a retry is allowed, and ’0’ if not. When the server sets this flag to ’1’, it disables short timeouts, expecting the peer to prompt the user for new credentials and to resubmit the response. When the server sets this flag to '0', the peer SHOULD NOT prompt the user for new credentials to try again without restarting the EAP-FAST authentication from the beginning.

The <msg> is human-readable ASCII text. Current implementations only support ASCII text.

The server failure packet can be broken into Label/Value pairs using the space character as the separator. The only value that may contain the space character is the <msg> value, which is always the last value pair in the failure packet. The peer SHOULD ignore any unknown label/value pair in the failure packet.

The error format described above is similar to what is defined in the Microsoft Challenge Handshake Authentication Protocol version 2 (MSCHAPv2) [RFC2759], except for the omission of a server challenge. So if the EAP-FAST server is distributing MSCHAPv2 exchanges to the backend inner method server, it can simply return what the backend inner method server returns less the server challenge. In the case of connecting to a one time password or Lightweight Directory Access Protocol (LDAP) [RFC4511] server, the EAP-FAST server can translate the error message into this format. With the addition of the retry count, the peer can potentially prompt the user for new credentials to try again without restarting the EAP-FAST authentication from the beginning. The peer will respond to the error code with another EAP-FAST-GTC Response packet with both the new username and password, or in case of other unrecoverable failures, an empty EAP-FAST-GTC packet for acknowledgement. The peer uses empty EAP-FAST-GTC payload as an acknowledgment of the unrecoverable failure.

If the EAP-FAST server finishes authentication for the EAP-FAST-GTC inner method, it will proceed to Protected Termination as described in [RFC4851]. In the case of an unrecoverable EAP-FAST-GTC authentication failure, the EAP server can send an EAP-FAST-GTC error code as described above, along with the Result TLV for protected termination. This way, no extra round trips will occur. The peer can acknowledge the EAP-FAST-GTC failure as well as the Result TLV within the same EAP-FAST packet. Once the server receives the acknowledgement, the TLS tunnel will be torn down and a clear text EAP-Failure will be sent.

The username and password, as well as server challenges, MAY support non-ASCII characters. In this case, international username, password, and messages are based on the use of Unicode characters, encoded as UTF-8 [RFC3629] and processed with a certain algorithm to ensure a canonical representation. The username and password input SHOULD be processed according to Section 2.4 of [RFC4282], and the server challenges SHOULD be processed according to [RFC5198].

Since EAP-FAST-GTC does not generate session keys, the MSKi (Master Session Key) used for crypto-binding for EAP-FAST will be filled with all zeros.

3. Security Considerations

The EAP-FAST-GTC method sends password information in the clear and MUST NOT be used outside of a protected tunnel providing strong protection, such as the one provided by EAP-FAST. Weak encryption, such as 40-bit encryption or NULL cipher, MUST NOT be used. In addition, the peer MUST authenticate the server before disclosing its credentials. Since EAP-FAST Server-Unauthenticated Provisioning Mode does not authenticate the server, EAP-FAST-GTC MUST NOT be used as the inner method in this mode. EAP-FAST-GTC MAY be used in EAP-FAST authentication and Server-Authenticated Provisioning Mode [RFC5422], where the server is authenticated. Since EAP-FAST-GTC requires the server to have access to the actual authentication secret, it is RECOMMENDED to vary the stored authentication validation data by domain so that a compromise of a server at one location does not compromise others.

3.1. Security Claims

This section provides the needed security claim requirement for EAP [RFC3748].

Auth. mechanism:	Password based.
Ciphersuite negotiation:	No. However, such negotiation is provided by EAP-FAST for the outer authentication.
Mutual authentication:	No. However, EAP-FAST provides server-side authentication.
Integrity protection:	No. However, any method executed within the EAP-FAST tunnel is protected.
Replay protection:	See above.
Confidentiality:	See above.
Key derivation:	Keys are not generated, see Section 2. However, when used inside EAP-FAST, the outer method will provide keys. See [RFC4851] for the properties of those keys.
Key strength:	See above.
Dictionary attack prot.:	No. However, when used inside the EAP-FAST tunnel, the protection provided by the TLS tunnel prevents an off-line dictionary attack.

Fast reconnect: No. However, EAP-FAST provides a fast reconnect capability that allows the reuse of an earlier session authenticated by EAP-FAST-GTC.

Cryptographic binding: No. Given that no keys are generated, EAP-FAST-GTC or its use within EAP-FAST cannot provide a cryptographic assurance that no binding attack has occurred. EAP-FAST-GTC is required only to run within a protected tunnel, but even the use of the same credentials in some other, unprotected context might lead to a vulnerability. As a result, credentials used in EAP-FAST-GTC SHOULD NOT be used in other unprotected authentication mechanisms.

Session independence: No. However, EAP-FAST provides session independence.

Fragmentation: No. However, EAP-FAST provides support for this.

Key Hierarchy: Not applicable.

Channel binding: No, though EAP-FAST can be extended for this.

4. IANA Considerations

EAP-FAST-GTC uses the assigned value of 6 (EAP-GTC) for the EAP Type in [RFC3748].

This document defines a registry for EAP-FAST-GTC error codes when running inside EAP-FAST, named "EAP-FAST GTC Error Codes". It may be assigned by Specification Required as defined in [RFC5226]. A summary of the error codes defined so far is given below:

- 646 ERROR_RESTRICTED_LOGON_HOURS

- 647 ERROR_ACCT_DISABLED

- 648 ERROR_PASSWD_EXPIRED

- 649 ERROR_NO_DIALIN_PERMISSION

- 691 ERROR_AUTHENTICATION_FAILURE

- 709 ERROR_CHANGING_PASSWORD

- 755 ERROR_PAC_I-ID_NO_MATCH

No IANA registry will be created for Labels, as current implementations only support the Labels defined in this document and new Labels are not expected; if necessary, new Labels can be defined in documents updating this document.

5. Acknowledgments

The authors would like thank Joe Salowey and Amir Naftali for their contributions of the problem space, and Jouni Malinen, Pasi Eronen, Jari Arkko, and Chris Newman for reviewing this document.

6. References

6.1. Normative References

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

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

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

[RFC4282] Aboba, B., Beadles, M., Arkko, J., and P. Eronen, "The Network Access Identifier", RFC 4282, December 2005.

[RFC4851] Cam-Winget, N., McGrew, D., Salowey, J., and H. Zhou, "The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST)", RFC 4851, May 2007.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

6.2. Informative References

[RFC2759] Zorn, G., "Microsoft PPP CHAP Extensions, Version 2", RFC 2759, January 2000.

[RFC4511] Sermersheim, J., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, June 2006.

[RFC5422] Cam-Winget, N., McGrew, D., Salowey, J., and H. Zhou, "Dynamic Provisioning Using Flexible Authentication via Secure Tunneling Extensible Authentication Protocol (EAP-FAST)", RFC 5422, March 2009.

Authors’ Addresses

Nancy Cam-Winget
Cisco Systems
3625 Cisco Way
San Jose, CA 95134
US

EMail: ncamwing@cisco.com

Hao Zhou
Cisco Systems
4125 Highlander Parkway
Richfield, OH 44286
US

EMail: hzhou@cisco.com