Network Working Group
Request for Comments: 5018
Category: Standards Track

G. Camarillo
Ericsson

September 2007

Connection Establishment in the Binary Floor Control Protocol (BFCP)

Status of This Memo

Abstract

This document specifies how a Binary Floor Control Protocol (BFCP) client establishes a connection to a BFCP floor control server outside the context of an offer/answer exchange. Client and server authentication are based on Transport Layer Security (TLS).

Table of Contents

1. Introduction

As discussed in the BFCP (Binary Floor Control Protocol) specification [RFC4582], a given BFCP client needs a set of data in order to establish a BFCP connection to a floor control server. These data include the transport address of the server, the conference identifier, and the user identifier.

Once a client obtains this information, it needs to establish a BFCP connection to the floor control server. The way this connection is established depends on the context of the client and the floor control server. How to establish such a connection in the context of an SDP (Session Description Protocol) [RFC4566] offer/answer [RFC3264] exchange between a client and a floor control server is specified in RFC 4583 [RFC4583]. This document specifies how a client establishes a connection to a floor control server outside the context of an SDP offer/answer exchange.

BFCP entities establishing a connection outside an SDP offer/answer exchange need different authentication mechanisms than entities using offer/answer exchanges. This is because offer/answer exchanges provide parties with an initial integrity-protected channel that clients and floor control servers can use to exchange the fingerprints of their self-signed certificates. Outside the offer/answer model, such a channel is not typically available. This document specifies how to authenticate clients using PSK (Pre-Shared Key)-TLS (Transport Layer Security) [RFC4279] and how to authenticate servers using server certificates.

2. Terminology

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. TCP Connection Establishment

As stated in Section 1, a given BFCP client needs a set of data in order to establish a BFCP connection to a floor control server. These data include the transport address of the server, the conference identifier, and the user identifier. It is outside the scope of this document to specify how a client obtains this information. This document assumes that the client obtains this information using an out-of-band method.

Once the client has the transport address (i.e., IP address and port) of the floor control server, it initiates a TCP connection towards it. That is, the client performs an active TCP open.

If the client is provided with the floor control server's host name instead of with its IP address, the client MUST perform a DNS lookup in order to resolve the host name into an IP address. Clients eventually perform an A or AAAA DNS lookup (or both) on the host name.

In order to translate the target to the corresponding set of IP addresses, IPv6-only or dual-stack clients MUST use name resolution functions that implement the Source and Destination Address Selection algorithms specified in [RFC3484]. (On many hosts that support IPv6, APIs like getaddrinfo() provide this functionality and subsume existing APIs like gethostbyname().)

The advantage of the additional complexity is that this technique will output an ordered list of IPv6/IPv4 destination addresses based on the relative merits of the corresponding source/destination pairs. This will result in the selection of a preferred destination address. However, the Source and Destination Selection algorithms of [RFC3484] are dependent on broad operating system support and uniform implementation of the application programming interfaces that implement this behavior.

> Developers should carefully consider the issues described by Roy et al. [RFC4943] with respect to address resolution delays and address selection rules. For example, implementations of getaddrinfo() may return address lists containing IPv6 global addresses at the top of the list and IPv4 addresses at the bottom, even when the host is only configured with an IPv6 local scope (e.g., link-local) and an IPv4 address. This will, of course, introduce a delay in completing the connection.

The BFCP specification [RFC4582] describes a number of situations when the TCP connection between a client and the floor control server needs to be reestablished. However, that specification does not describe the reestablishment process because this process depends on how the connection was established in the first place.

When the existing TCP connection is closed following the rules in [RFC4582], the client SHOULD reestablish the connection towards the floor control server. If a TCP connection cannot deliver a BFCP message from the client to the floor control server and times out, the client SHOULD reestablish the TCP connection.

4. TLS Usage

[RFC4582] requires that all BFCP entities implement TLS [RFC4346] and recommends that they use it in all their connections. TLS provides integrity and replay protection, and optional confidentiality. The floor control server MUST always act as the TLS server.

A floor control server that receives a BFCP message over TCP (no TLS) SHOULD request the use of TLS by generating an Error message with an Error code with a value of 9 (Use TLS).

5. Authentication

BFCP supports client authentication based on pre-shared secrets and server authentication based on server certificates.

5.1. Certificate-Based Server Authentication

At TLS connection establishment, the floor control server MUST present its certificate to the client. The certificate provided at the TLS level MUST either be directly signed by one of the other party's trust anchors or be validated using a certification path that terminates at one of the other party's trust anchors [RFC3280].

A client establishing a connection to a server knows the server's host name or IP address. If the client knows the server's host name, the client MUST check it against the server's identity as presented in the server's Certificate message, in order to prevent man-in-the-middle attacks.

If a subjectAltName extension of type dNSName is present, that MUST be used as the identity. Otherwise, the (most specific) Common Name field in the Subject field of the certificate MUST be used. Although the use of the Common Name is existing practice, it is deprecated and Certification Authorities are encouraged to use the subjectAltName instead.

Matching is performed using the matching rules specified by [RFC3280]. If more than one identity of a given type is present in the certificate (e.g., more than one dNSName name), a match in any one of the set is considered acceptable. Names in Common Name fields may contain the wildcard character *, which is considered to match any single domain name component or component fragment (e.g., *.a.com matches foo.a.com but not bar.foo.a.com. f*.com matches foo.com but not bar.com).

If the client does not know the server's host name and contacts the server directly using the server's IP address, the iPAddress subjectAltName must be present in the certificate and must exactly match the IP address known to the client.

If the host name or IP address known to the client does not match the identity in the certificate, user-oriented clients MUST either notify the user (clients MAY give the user the opportunity to continue with the connection in any case) or terminate the connection with a bad certificate error. Automated clients MUST log the error to an appropriate audit log (if available) and SHOULD terminate the connection (with a bad certificate error). Automated clients MAY provide a configuration setting that disables this check, but MUST provide a setting that enables it.

5.2. Client Authentication Based on a Pre-Shared Secret

Client authentication is based on a pre-shared secret between client and server. Authentication is performed using PSK-TLS [RFC4279].

The BFCP specification mandates support for the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite. Additionally, clients and servers supporting this specification MUST support the TLS_RSA_PSK_WITH_AES_128_CBC_SHA ciphersuite as well.

6. Security Considerations

Client and server authentication as specified in this document are based on the use of TLS. Therefore, it is strongly RECOMMENDED that TLS with non-null encryption is always used. Clients and floor control servers MAY use other security mechanisms as long as they provide similar security properties (i.e., replay and integrity protection, confidentiality, and client and server authentication).

TLS PSK simply relies on a pre-shared key without specifying the nature of the key. In practice, such keys have two sources: text passwords and randomly generated binary keys. When keys are derived from passwords, TLS PSK mode is subject to offline dictionary attacks. In DHE (Diffie-Hellman Exchange) and RSA modes, an attacker who can mount a single man-in-the-middle attack on a client/server pair can then mount a dictionary attack on the password. In modes without DHE or RSA, an attacker who can record communications between a client/server pair can mount a dictionary attack on the password. Accordingly, it is RECOMMENDED that, where possible, clients use certificate-based server authentication ciphersuites with password-derived PSKs in order to defend against dictionary attacks.

In addition, passwords SHOULD be chosen with enough entropy to provide some protection against dictionary attacks. Because the entropy of text varies dramatically and is generally far less than that of an equivalent random bitstring, no hard and fast rules about password length are possible. However, in general passwords SHOULD be chosen to be at least 8 characters and selected from a pool containing both upper and lower case, numbers, and special keyboard characters (note that an 8-character ASCII password has a maximum entropy of 56 bits and in general far lower). FIPS PUB 112 [PUB112] provides some guidance on the relevant issues. If possible, passphrases are preferable to passwords. It is RECOMMENDED that implementations support, at minimum, 16-character passwords or passphrases. In addition, a cooperating client and server pair MAY choose to derive the TLS PSK shared key from the passphrase via a password-based key derivation function such as PBKDF2 [RFC2898]. Because such key derivation functions may incorporate iteration functions for key strengthening, they provide some additional protection against dictionary attacks by increasing the amount of work that the attacker must perform.

When the keys are randomly generated and of sufficient length, dictionary attacks are not effective because such keys are highly unlikely to be in the attacker’s dictionary. Where possible, keys SHOULD be generated using a strong random number generator as specified in [RFC4086]. A minimum key length of 80 bits SHOULD be used.

The remainder of this section analyzes some of the threats against BFCP and how they are addressed.

An attacker may attempt to impersonate a client (a floor participant or a floor chair) in order to generate forged floor requests or to grant or deny existing floor requests. Client impersonation is avoided by using TLS. The floor control server assumes that attackers cannot hijack TLS connections from authenticated clients.

An attacker may attempt to impersonate a floor control server. A successful attacker would be able to make clients think that they hold a particular floor so that they would try to access a resource (e.g., sending media) without having legitimate rights to access it. Floor control server impersonation is avoided by having floor control servers present their server certificates at TLS connection establishment time.

Attackers may attempt to modify messages exchanged by a client and a floor control server. The integrity protection provided by TLS connections prevents this attack.

Attackers may attempt to pick messages from the network to get access to confidential information between the floor control server and a client (e.g., why a floor request was denied). TLS confidentiality prevents this attack. Therefore, it is RECOMMENDED that TLS is used with a non-null encryption algorithm.

7. Acknowledgments

Sam Hartman, David Black, Karim El Malki, and Vijay Gurbani provided useful comments on this document. Eric Rescorla performed a detailed security analysis of this document.

8. References

8.1. Normative References

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

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC4279] Eronen, P. and H. Tschofenig, "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4582] Camarillo, G., Ott, J., and K. Drage, "The Binary Floor Control Protocol (BFCP)", RFC 4582, November 2006.

[RFC4583] Camarillo, G., "Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams", RFC 4583, November 2006.

[PUB112] National Institute of Standards and Technology (NIST), "Password Usage", FIPS PUB 112, May 1985.

8.2. Informative References

[RFC2898] Kaliski, B., "PKCS #5: Password-Based Cryptography Specification Version 2.0", RFC 2898, September 2000.

[RFC4943] Roy, S., Durand, A., and J. Paugh, "IPv6 Neighbor Discovery On-Link Assumption Considered Harmful", RFC 4943, September 2007.

Author's Address

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Full Copyright Statement

Intellectual Property