```objc
// AFSecurityPolicy.h
// Copyright (c) 2011–2016 Alamofire Software Foundation ( http://alamofire.org/ )
//
// Permission is hereby granted, free of charge, to any person obtaining a copy
// of this software and associated documentation files (the "Software"), to deal
// in the Software without restriction, including without limitation the rights
// to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and to permit persons to whom the Software is
// furnished to do so, subject to the following conditions:
//
// The above copyright notice and this permission notice shall be included in
// all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
// OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
// THE SOFTWARE.

#import <Foundation/Foundation.h>
#import <Security/Security.h>

typedef NS_ENUM(NSUInteger, AFSSLPinningMode) {
    AFSSLPinningModeNone,
    AFSSLPinningModePublicKey,
    AFSSLPinningModeCertificate,
};

/**
 `AFSecurityPolicy` evaluates server trust
against pinned X.509 certificates and public keys
over secure connections.

 Adding pinned SSL certificates to your app helps
prevent man-in-the-middle attacks and other
vulnerabilities. Applications dealing with
sensitive customer data or financial information
are strongly encouraged to route all
communication over an HTTPS connection with SSL
pinning configured and enabled.
 */

NS_ASSUME_NONNULL_BEGIN

@interface AFSecurityPolicy : NSObject
<NSSecureCoding, NSCopying>

/**
 The criteria by which server trust should be
evaluated against the pinned SSL certificates.
Defaults to `AFSSLPinningModeNone`.
 */
@property (readonly, nonatomic, assign)
AFSSLPinningMode SSLPinningMode;

/**
 The certificates used to evaluate server trust
according to the SSL pinning mode.

  By default, this property is set to any
(`.cer`) certificates included in the target
compiling AFNetworking. Note that if you are
using AFNetworking as embedded framework, no
certificates will be pinned by default. Use
```

```
`certificatesInBundle` to load certificates from
your target, and then create a new policy by
calling
`policyWithPinningMode:withPinnedCertificates`.

 Note that if pinning is enabled,
`evaluateServerTrust:forDomain:` will return true
if any pinned certificate matches.
 */
@property (nonatomic, strong, nullable) NSSet
<NSData *> *pinnedCertificates;

/**
 Whether or not to trust servers with an invalid
or expired SSL certificates. Defaults to `NO`.
 */
@property (nonatomic, assign) BOOL
allowInvalidCertificates;

/**
 Whether or not to validate the domain name in
the certificate's CN field. Defaults to `YES`.
 */
@property (nonatomic, assign) BOOL
validatesDomainName;

///-----------------------------------------
/// @name Getting Certificates from the Bundle
///-----------------------------------------

/**
 Returns any certificates included in the bundle.
If you are using AFNetworking as an embedded
framework, you must use this method to find the
certificates you have included in your app
bundle, and use them when creating your security
policy by calling
`policyWithPinningMode:withPinnedCertificates`.

 @return The certificates included in the given
```

```
bundle.
 */
+ (NSSet <NSData *> *)certificatesInBundle:
(NSBundle *)bundle;

///-----------------------------------------
/// @name Getting Specific Security Policies
///-----------------------------------------

/**
 Returns the shared default security policy,
which does not allow invalid certificates,
validates domain name, and does not validate
against pinned certificates or public keys.

 @return The default security policy.
 */
+ (instancetype)defaultPolicy;

///---------------------
/// @name Initialization
///---------------------

/**
 Creates and returns a security policy with the
specified pinning mode.

 @param pinningMode The SSL pinning mode.

 @return A new security policy.
 */
+ (instancetype)policyWithPinningMode:
(AFSSLPinningMode)pinningMode;

/**
 Creates and returns a security policy with the
specified pinning mode.

 @param pinningMode The SSL pinning mode.
 @param pinnedCertificates The certificates to
pin against.
```

```
 @return A new security policy.
 */
+ (instancetype)policyWithPinningMode:
(AFSSLPinningMode)pinningMode
withPinnedCertificates:(NSSet <NSData *>
*)pinnedCertificates;

///-----------------------------
/// @name Evaluating Server Trust
///-----------------------------

/**
 Whether or not the specified server trust should
be accepted, based on the security policy.

 This method should be used when responding to an
authentication challenge from a server.

 @param serverTrust The X.509 certificate trust
of the server.
 @param domain The domain of serverTrust. If
`nil`, the domain will not be validated.

 @return Whether or not to trust the server.
 */
- (BOOL)evaluateServerTrust:
(SecTrustRef)serverTrust
                  forDomain:(nullable NSString
*)domain;

@end

NS_ASSUME_NONNULL_END

///----------------
/// @name Constants
///----------------

/**
 ## SSL Pinning Modes
 The following constants are provided by
`AFSSLPinningMode` as possible SSL pinning modes.

 enum {
 AFSSLPinningModeNone,
 AFSSLPinningModePublicKey,
 AFSSLPinningModeCertificate,
 }

 `AFSSLPinningModeNone`
 Do not used pinned certificates to validate
servers.

 `AFSSLPinningModePublicKey`
 Validate host certificates against public keys
of pinned certificates.

 `AFSSLPinningModeCertificate`
 Validate host certificates against pinned
certificates.
*/
```