Mesh Confirmation Protocol (Mesh/Confirm)

Mesh/Confirm

<series>draft-hallambaker-mesh-confirm

<status>informational

<stream>independent

<ipr>trust200902

<author>Phillip Hallam-Baker

<fullname>Phillip Hallam-Baker

<initials>P. M.

<organization>Comodo Group Inc.

<surname>Hallam-Baker

<email>philliph@comodo.com

<keyword>Cryptography

<keyword>PKI

<keyword>PKIX

<also>http://prismproof.org/Documents/draft-hallambaker-mesh-confirm.html

Mesh Confirmation Protocol (Mesh/Confirm) is a three-party Web Service that supports a transactional second factor confirmation mechanism that provides a superset of the capabilities of traditional second factor authentication schemes. The three parties in the protocol are Enquirer who posts a confirmation request, a Responder who may or may not respond to the request and the Broker which provides a repository to which requests and responses are posted.

Introduction

Authentication of end users is one of the biggest challenges for Internet and Web security today. Despite an abundance of technology that offers authentication mechanisms that are more robust, more secure and easier to use, the default mechanism for user authentication is the use of usernames and passwords.

Mesh/Confirm is a second factor authentication mechanism that binds the user's response to the decision asked of the user. If the user is attempting to log in to a network host, they receive a confirmation message on a device they habitually carry such as a watch or a smart phone asking if that is what they want to do and they respond by accepting or rejecting the request.

<figuresvg="../Images/ConfirmSummary.svg">Confirmation User Experience

Unlike traditional second factor authentication schemes, Mesh/Confirm does not require the user to carry a special purpose 'smart' token or to enter randomly changing PIN codes.

Mesh/Confirm is designed to make full use of the features afforded by a modern smartphone. In particular, a Mesh/Confirm client device MUST support a means of presenting text output to and accepting text input from the user and a network connection. While mobile devices offering this degree of functionality were rare in 2007, they have since become ubiquitous. In addition to smartphones, many users now carry smart watches and the class of wearable electronics is expected to expand further in years to come. It is thus now a practical proposition for a site requiring second factor authentication to support at least a part of its users using a technology that requires such affordances.

# Definitions

This section presents the related specifications and standards on which Mesh/Confirm is built, the terms that are used as terms of art within the documents and the terms used as requirements language.

## Requirements Language

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 <norm="RFC2119"/>.

## Defined Terms

The terms of art used in this document are described in the *Mesh Architecture Guide* <info="draft-hallambaker-mesh-architecture"/>.

## Related Specifications

The architecture of the Mathematical Mesh is described in the *Mesh Architecture Guide* <info="draft-hallambaker-mesh-architecture"/>. The Mesh documentation set and related specifications are described in this document.

## Implementation Status

The implementation status of the reference code base is described in the companion document <info="draft-hallambaker-mesh-developer"/>.

Overview

Second factor authentication mechanisms offer greater security over the use of passwords alone by combining a first factor (typically a password) with a second factor, typically a biometric or proof of possession of a physical token.

Traditional second factor authentication techniques have suffered from the need to distribute physical tokens and the difficulty of ensuring that a biometric authentication is presented to a trustworthy terminal.

The usability of traditional second factor authentication techniques has been poor or worse. Even the simplest scheme in which the user is required to read in a 'one time use' numeric code from the authentication token device and enter it into a password field. While such operations are relatively simple they require the user to engage in a sequence of operations that bears no necessary or natural relationship to the underlying task for which the authentication is required.

Nor does the act of engaging in a traditional second factor scheme offer proof of anything other than that the user was authenticated. Any correspondence between the act of authentication and the purpose for which the authentication was provided must be maintained separately.

Confirmation vs. Authentication

A confirmation service addresses by cryptographically binding responses to the request that they reply to.

A confirmation service allows the user experience to be precisely matched to the action that the user is attempted. This is simpler and more secure than a traditional second factor authentication scheme. Instead of being asked to read a random number from one device and enter it into another, the user is asked if they really want to perform the action for which authentication is requested.

A confirmation service offers better accountability for end users than a traditional second factor authentication scheme. An authentication service only provides an assertion that the user was present. A confirmation service provides an assertion that the user was present and that they confirmed (or refused) a specific request.

For example, Alice has been granted access to a machine storing classified data. If an authentication service is used for access control, the authentication service log will only record the dates and times that Alice accessed the system. to find out if Alice accessed a particular file on a particular day it is necessary to consult and correlate both the authentication log of the system and the activity log for the application.

If instead a confirmation service is used the confirmation log contains an authenticated record of both the authentication events and the transactions for which the authentication was requested.

Use Scenarios

A confirmation service complements rather than replaces a traditional authentication scheme. Providing a highly secure and convenient means of authenticating requests that carry a high degree of risk mitigates the risk of using convenient but intrinsically low security techniques for other actions.

Use in Financial Services

If an attacker is to profit from breaching an account with a financial service such as a bank or a brokerage they must find a way to move money out of the account. Thus, adding bill payment recipients, initiating wire transfers and trading in low volume 'penny stocks' represent high risk activities.

For example: Bank of Ethel might permit customers to use a simple username and password scheme to gain access to their account to check their balance or to send payments to existing recipients but require use of the second factor confirmation device for a high-risk transaction such as adding a new payee or making a substantially higher payment than normal.

Machine Binding

A second factor confirmation service may be combined with a machine level authentication scheme to permit a transparent form of authentication for low risk transactions.

For example: Alice stores her low risk authentication credentials (e.g. usernames and passwords) using a 'cloud' service. When she wishes to use those credentials an agent on her personal machine fetches credentials from the cloud service as necessary. When Alice wishes to access a site from a different machine she receives a confirmation request on her mobile device to grant access from that machine.

Use of such a mechanism is clearly more satisfactory when suitable cryptographic protocols such as SAML or Kerberos are employed to limit the disclosure and hence possible compromise of the credentials. The specification of such protocols is outside the scope of this document.

Tethered Use

Although Mesh/Confirm is designed for use in a three-party scenario, there are situations in which a two party mode may be preferred.

For example: Bob is a roadwarrior who requires access to confidential documents stored on his laptop device from anywhere in the world, including locations where Internet access is not possible. To permit access in such circumstances, Bob's Mesh/Confirm client supports use of a tethered mode in which the mobile device is connected via Bluetooth or plugged into his laptop via a USB port.

For example: Carol is a network manager of a large computing facility that uses Mesh/Confirm to authenticate and track all changes to critical resources. Since Mesh/Confirm is itself a network resource a bootstrap consideration arises: How can Carol confirm her network configuration requests using Mesh/Confirm when the network itself is down? Support for a tethered mode in which the Mesh/Confirm device communicates via USB or similar wired protocol allows this use case to be supported.

While availability of a tethered mode is clearly essential if Mesh/Confirm is to be used in certain applications, support for this feature outside the scope of this version of the specification.

Co-Browser

While Mesh/Confirm is designed for deployment on a secondary device, deployment on the same device as the one for which confirmation is being requested is also possible and can provide security benefits.

Modern Web browsers are large and complex with many features such as support for mobile code that are incompatible with a high security environment. Separating the confirmation protocol from the Web Browsing protocol permits implementation in a minimal client designed to permit detailed security analysis. Such a client might be embedded in or support means of secure interaction with a trustworthy operating system component.

While this means of deployment does not provide a true second factor confirmation, it is likely to provide a sufficient degree of authentication for many transactions.

Architecture

Mesh/Confirm is a Web Service that permits an Enquirer to request that a User confirm or reject a specified action. If the user responds, the response is signed with a digital signature under a key that is unique to the user account, the client and the device.

Parties

Each Mesh/Confirm protocol interaction takes place between a connection pair of the following parties:

Enquirer

A party that initiates a confirmation request.

User

The User is the person being asked to grant or refuse confirmation. A User MAY have multiple accounts with multiple Broker Services.

User Device

A device that the user has bound to their broker account.

Broker

A clearing house that stores and forwards requests from Initiators to Users Device and responses from Users to Initiators. The Broker is only trusted to perform routing filtering and recording of requests and responses. The Broker is not trusted with respect to the responses returned.

The communication between the parties is shown in Figure 1.

<figuresvg="../Images/confirm.svg">Mesh/Confirm Parties

Accounts

Users are identified by means of an account identifier. The display presentation of an account identifier is the form of an RFC2822 email address identifier without the enclosing angle braces, for example:

alice@example.com

The account identifier is used by the User when registering the use of the confirmation service with a Broker.

Open and Closed Services

A Mesh/Confirm service MAY be Open or Closed. An Open service provider provides Mesh/Confirm service to the general public. A Closed service provider only provides service to a specific community.

For example: An Internet Service Provider or DNS Registrar might provide an open Mesh/Confirm service as a part of their standard service offering to customers. An employer might operate a closed Mesh/Confirm service to be used for company business.

<include=..\Generated\ExamplesConfirm.md>

<include=..\Generated\SchemaConfirm.md>

Simple Request Markup Language (SRMLv1)

Confirmation requests are posted in SRML, a deliberately limited subset of HTML. SRML is limited to four elements and one attribute. These are:

srml

The top-level element for an SRML request

h1

Heading

p

Paragraph

button

Button specifying a value that the user can select.

While SRML is loosely based on the HTML forms markup, there are important differences. The HTML markup model supports multiple document types of which forms are only one and a single document may contain multiple forms with multiple different action values. In an SRML document is a single form and the form action to be performed is impicit in the presentation of the document to the user.

XML Schema and Content Type Identifier

The MIME Content Type and schema identifier for SRML are

Content-Type

text/xml

xmlns

http://hallambaker.com/Schemas/srml.xsd

The schema is:

include=Schemas\srml.md

Design considerations and future options

The capabilities of SRML are intentionally limited to the bare minimum. It should be possible to make use of SRML to display options to the user on a smartwatch or other device with a highly constrained display.

The function of the confirmation service is to provide confirmation of an action that was initiated elsewhere. It is therefore inappropriate for this or any future version of SRML to offer extensive data entry or validation capabilities. SRML applications MUST NOT support any form of scripting or active code extensions to SRML content.

It might prove advantageous in the future to extend the input types to include simple form elements such as checkboxes, numeric fields, text choices and possibly free form text.

Request Authentication and Authorization

The current version of the protocol does not address the question of how service requests are to be authorized or authenticated.

A triple lock security approach is anticipated in which cryptographic enhancements are applied at three separate levels to provide different security controls:

Transport Layer

Basic confidentiality and integrity controls are provided using TLS with a server-side certificate. It is necessary to provide encryption at this layer to protect confidentiality of meta-data.

Presentation Layer

Mutual authentication of the client and service is provided at the presentation layer. In the default JWB binding, this is provided within the HTTP content payload. The use of encryption at the presentation is optional.

Data Layer

Confirmation requests and responses are signed by the Enquirer and Responder respectively. This provides for non-repudiation of messages.

Service Authentication

Since the responder is identified by the responder’s account, Minimal Validation is sufficient but Domain Validation is preferred. These credentials MAY be bound using a strong DNS name.

Responder Authentication

The responder is authenticated by means of the user’s Mesh profile.

The ability to delegate access to a confirmation account might be useful in certain circumstances.

Enquirer Authentication

Authentication of the Enquirer presents very different challenges to authentication of the Service or the Responder as it is the only part of the service that is ‘open’. It is thus likely to be the target of abuse   
(i.e. spam). It is therefore important that the authentication mechanism enable appropriate authorization and accountability strategies.

For example, one strategy to control abuse might be to permit enquirers to post requests if they were signed with a key authenticated by an Extended Validation certificate or were sent by an enquirer approved by the responder to whom the request was directed. In the first case, abuse is mitigated by an accountability control, in the second by explicit authorization of the sender.

While it is possible to implement such a strategy in the responder application, this approach is clearly limiting. Filtering of messages in the service avoids the need to synchronize policy across the user’s confirmation devices and protects possibly limited wireless bandwidth by performing policy enforcement in the service rather than the responder’s device.

Mesh/Confirm does not provide a mechanism for specifying such a security policy. Leaving this requirement to a separate service allows for a protocol that can specify policy for multiple modes of communication. For instance, a customer of a bank might permit the bank to send confirmation messages and to deliver statements by email but not to make contact by voice or video calls.

Security Considerations

Consider spam control, how do users prevent unwanted requests? (EV accreditation, filtering at Broker)

People deploying Mesh/Confirm as a means of controlling access to networking infrastructure must consider the bootstrap issue. In particular since Mesh/Confirm requires Internet access the network administrator must ensure that it is possible to manage the network resources necessary to support an SXS service when that service is down.

Acknowledgements