Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 17, 2016

P. Hallam-Baker
Comodo Group Inc.
January 14, 2016

Mathematical Mesh: Developer?s Guide
draft-hallambaker-mesh-developer-00

Abstract

The Mathematical Mesh ?The Mesh? is an end-to-end secure infrastructure that facilitates the exchange of configuration and credential data between multiple user devices.

This document describes how to install and run the Mesh reference code and make use of the reference code in applications. It does not form a part of the Mesh specifications and is not normative.

Status of This Memo

Copyright Notice

1. Getting the Reference Code and Build Tools

The Mesh Reference library was developed using Visual Studio 2015 Community Edition using PHB?s Build Tools extensions. The reference code itself is currently limited to C# libraries.

The code should in theory run under other operating systems but this is not currently tested.

Development under different development environments is also possible but would require re-engineering to make use of the line mode versions of the build tools.

1.1. Obtaining the Development Environment

Visual Studio 2015 Community Edition is currently available at no cost for a wide range of non-commercial development including personal use and development of Open Source software. For full details, please consult the license published by Microsoft.

https://www.visualstudio.com/

1.2. Obtaining the Build Tools

Over half the code in the reference code library is generated using code generators. These are used to ensure that the specification, examples and reference code are always kept in synchronization.

The build tools are published under an MIT License and are available in two forms:

As stand-alone tools to be run from the command line.

As a VSIX package that integrates into the Visual Studio environment.

The source distribution is configured to use the tools integrated into the Visual Studio environment. If development on other platforms is desired, the simplest approach is likely to be to write a tool that reads the Visual Studio configuration files and generates the corresponding files for use with make.

The VSIX package is available from the Visual Studio extensions gallery:

PHB Code Generation Tools

The source code for the build tools is available from:

https://sourceforge.net/projects/phb-build-tools/

1.3. Obtaining the Mesh Source Libraries

The Mesh reference library source code is published under an MIT license and is available from:

https://sourceforge.net/projects/mathematicalmesh/

2. Running the Reference Code Examples

The reference code examples are designed to illustrate how the Mesh might be used in an application rather than be standalone tools in their own right. The Mesh is designed to make it each for developers to add security to their own applications rather than providing the applications themselves.

2.1. Starting the Server

On the Windows platform, the server runs in the context of the platform Web server and must be granted permission to bind to the range of server addresses used using the netsh command.

From a command prompt with administrator privileges, run the following command:

```
netsh http add urlacl http:///.well-known/mmm/ \user=\
```

Where is the DNS domain name under which the service is run, is the Windows domain name of the machine and the account name.

To start the service from the command line type:

```
servermesh
```

The server does not require administration privileges.

2.2. The Profile Manager Wizard

The profile manager wizard demonstrates functions that are performed on an administration device. These include creating a completely new profile and initial configuration of applications, connecting a device to the profile and recovery of the profile from escrow data.

To run the client from the command line, place the executable image in a location that it will be found in the PATH variable and type:

```
meshclient
```

2.3. The Profile Connection Wizard

The Profile connection wizard demonstrates the much more restricted functionality that would be required in a Mesh connected application and/or a profile manager for a non-administration device.

To run the client from the command line, place the executable image in a location that it will be found in the PATH variable and type:

```
meshconnect
```

3. Platform specific configuration data

3.1. Windows

3.1.1. Private Key Data

All private key data is stored using the Windows public key store. At minimum, this ensures that private keys are obfuscated and encrypted under the account password to protect the data against casual extraction attacks. On a machine with cryptographic hardware support such as a TPM or HSM, extraction of the private key may be infeasible without physical access to the machine and possibly require sophisticated diagnostic equipment.

3.1.2. Registry settings

Separate settings are used for production and test code. Test Code should use the Registry Hive:

```
HKEY_CURRENT_USER\SOFTWARE\CryptoMesh
```

Production code should use the hive

```
HKEY_CURRENT_USER\SOFTWARE\MathematicalMesh
```

In either case the sub structure is:

Accounts Contains the set of Mesh Portal Accounts for the user. The default value is the account name of the default account. The Name of the each key is a portal account name and the value a REG_SZ entry containing the UDF of the profile master key.

PersonalProfiles Contains the set of Mesh Profiles for the user. The default value is the UDF of the default profile master key. The Name of each key is the UDF of the master key and the value a REG_SZ entry containing the file location of the cached copy of the personal profile.

ThisDevice Contains the set of Device profiles in the same format as the PersonalProfiles.

3.1.3. Profile data files

The profile data itself is stored in data files at the location specified in the registry. The files are standard XML files in UTF8 encoding.

3.2. OSX and Linux

[[Not yet implemented, subject to change.]

All configuration information is stored in the user directory ~/.mmm

Keys are stored in SSH key file format [RFC4716] using the customary name and extension conventions for that application.

4. Using the Mesh C#/.Net Libraries in an Application

The application ExampleGenerator shows the use of the Mesh in an application using the convenience API. It is the application program used to generate the examples in the reference document.

ExampleGenerator implements a client that connects to a remote WebService, creates new personal profile with an escrow entry with offline recovery codes, attaches applications and other devices, updates an application profile, deletes all the profile data from the local machine and then restores them using the recovery codes and escrow entry.

4.1. Creating a Portal Client Connection

The normal method of creating a Portal Client connection is?

Since the purpose of the ExampleGenerator is to create examples for the documentation, it is not necessary for the JSON Remote Procedure Calls to actually be ?Remote?. Instead the ?Local? Procedure Call mode is used in which the client and server both run in the same process with the client API invoking the server dispatch methods through an interface that performs JSON serialization and deserialization but does not invoke the network transport.

?.

For purposes of testing and initial development of a Web Service it is frequently desirable to further simplify the implementation by dispensing with the serialization layer and the client calling the server dispatch methods directly.

?.

4.2. Checking that a Portal Account name is acceptable

4.3. Creating a Personal Profile

4.4. Creating an Offline Escrow Entry

4.5. Publishing the Profile and Escrow Entries

4.6. Attaching a New Device

4.7. Attaching a new Application

4.8. Deleting Profile Data

4.9. Recovering Profile Data

5. Using other languages

If you are building Mesh applications in another language, the least effort approach may be to rewrite the PROTOGEN build tool to target your language.

Protogen does support generation of C header files that may be used to drive a parser. If however you are adding Mesh support for an application that already uses JSON based protocols, you might want to edit the generator scripting files to generate code for your existing libraries.

5.1. Using the C Binding

6. Reference Code Architecture

6.1. Protocol Definition

6.1.1. Serialization

6.1.2. Deserialization

6.1.3. Class library

6.2. Profile

6.2.1. Generation

6.2.2. Validation

6.2.3. Operations

6.3. Server

6.3.1. Management Class

6.3.2. Dispatch Class

6.3.3. Connection Modes

6.3.3.1. Direct

6.3.3.2. Local

6.3.3.3. Remote

6.4. Client

6.4.1. Stub Library

6.4.2. Convenience API

7. Security Considerations

Security Considerations are addressed in the companion document [draft-hallambaker-mesh-architecture]

8. IANA Considerations

IANA Considerations are addressed in the companion document [draft-hallambaker-mesh-architecture]

9. Acknowledgements

Comodo Group: Egemen Tas, Melhi Abdulhayo?lu, Rob Stradling, Robin Alden.

10. Normative References

[RFC4716] Galbraith, J. and R. Thayer, "The Secure Shell (SSH) Public Key File Format", RFC 4716, DOI 10.17487/RFC4716, November 2006.

[draft-hallambaker-mesh-architecture] "[Reference Not Found!]".

Author's Address

Phillip Hallam-Baker
Comodo Group Inc.

Email: philliph@comodo.com