G. Monroe
C. Howells
JanRain
December 6, 2006

OpenID Service Key Discovery 1.0 - Draft 01

Abstract

This document describes a standard way for Services to advertise and discover Public Keys.

Table of Contents

1. Requirements Notation

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] (Bradner, B., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

2. Terminology

Identifier:
An Identifier is a URL or XRI.

Service Endpoint:
A web application described in an XRDS document.

Public Key:
A document containing the public material of a key pair used in asymmetric cryptography e.g., a PEM formatted RSA public key or an X.509 certificate.

3. Introduction

It is common for two Service Endpoints to wish to communicate securely. Public key cryptography is one mechanism for securing a channel between two Service Endpoints without requiring a previously established shared secret. This document describes a standard way for Service Endpoints to advertise and discover Public Keys.

4. XML Namespaces

The default namespace in this document for XML fragments without a namespace prefix is xmlns="http://openid.net/keydisc/1.0". Other namespace prefixes used include xmlns:xrd="xri://$xrd*($v*2.0)".

5. Public Key Advertising

Public Keys are advertised by adding a <PublicKey> element to an endpoints <xrd:Service> element. The new element. The <PublicKey> element contains a single child text node that contains the URL where the Public Key can be retrieved.

Here is an example XRDS service element,

```
<xrd:Service priority="0">
  <xrd:Type
    >http://www.example.com/secure_messaging/1.0
  </xrd:Type>
  <xrd:URI>
    http://www.example.com/server
  </xrd:URI>
  <PublicKey>
    https://www.example.com/X509.crt
  </PublicKey>
</xrd:Service>
```

6. Public Key Discovery

The Public Key for a Service Endpoint is discovered by performing XRDS resolution on the Endpoint's Identifier. Once the the Identifier is resolved to and XRDS, the Service element is looked up by Type, and the URL of the Public Key can be parsed out of the <PublicKey> element.

The Public Key is then retrieved by making an HTTP [RFC2616] (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” June 1999.) GET request for the parsed URL which SHOULD be over TLS [RFC2818] (Rescorla, E., “HTTP Over TLS,” May 2000.). The type of key material is application dependant, and can be identified by the Content-Type header when retrieving the key.

6.1. Security Considerations

There are two ways that Public Key discovery can be secure. The first way requires that every URL requested has a scheme of "https" and SSL certificates are verified during each request. The second way requires that the Public Key is a certificate containing the Identifier used to initiate discovery and is signed by a trusted third party.

If these requirements are not met, clients discovering Public Keys may be vulnerable to attack. The types of vulnerabilities include DNS cache poisoning and man in the middle attacks.

7. References

7.1. Normative References

[RFC2119] Bradner, B., “Key words for use in RFCs to Indicate Requirement Levels,” RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” RFC 2616, June 1999.

7.2. Informative References

[RFC2818] Rescorla, E., “HTTP Over TLS,” RFC 2818, May 2000.

Authors' Addresses

Grant Monroe
JanRain, Inc.
5331 SW Macadam Avenue
Suite #375
Portland, OR 97239
USA
Email: grant@janrain.com

Carl Howells
JanRain, Inc.
5331 SW Macadam Avenue
Suite #375
Portland, OR 97239
USA
Email: chowells@janrain.com