XMPP
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2013

M. Miller
P. Saint-Andre
Cisco Systems, Inc.

February 22, 2013

Using PKIX over Secure HTTP (POSH) as a Prooftype for XMPP Domain Name Associations
draft-miller-xmpp-posh-prooftype-03

Abstract

This document defines a prooftype involving PKIX over Secure HTTP (POSH) for associating a domain name with an XML stream in the Extensible Messaging and Presence Protocol (XMPP). It also defines a method involving HTTPS redirects (appropriate for use with the POSH prooftype) for securely delegating a source domain to a derived domain in XMPP.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 26, 2013.

Copyright Notice

Table of Contents

1. Introduction

The [XMPP-DNA] specification defines a framework for secure delegation and strong domain name associations (DNA) in the Extensible Messaging and Presence Protocol (XMPP). This document defines a DNA prooftype using PKIX certificates obtained over secure HTTP ("POSH"), as well as a secure delegation method, based on HTTPS redirects, that is appropriate for use with the POSH prooftype.

The rationale for POSH is driven by current operational realities. It is effectively impossible for a hosting service to provide and maintain PKIX certificates [RFC5280] that include the appropriate identifiers [RFC6125] for each hosted domain. It is true that DNS-based technologies are emerging for secure delegation, in the form of DNS Security ([RFC4033] and [RFC6698]); however, these technologies are not yet widely deployed and might not be deployed in the near future for domains outside the most common top-level domains (e.g., ".COM", ".NET", ".EDU"). Because the XMPP community wishes to deploy secure delegation and strong domain name associations as widely and as quickly as possible, this document specifies how to use secure HTTP ([RFC2616] and [RFC2818]) and PKIX certificates [RFC5280] to verify that a domain is delegated to a hosting provider and also establish a strong assocation between a domain name and an XML stream.

2. Terminology

This document inherits XMPP terminology from [RFC6120] and security terminology from [RFC5280]. The terms "source domain", "derived domain", "reference identifier", and "presented identifier" are used as defined in the "CertID" specification [RFC6125].

This document is applicable to connections made from an XMPP client to an XMPP server ("_xmpp-client._tcp") or between XMPP servers ("_xmpp-server._tcp"). In both cases, the XMPP initiating entity acts as a TLS client and the XMPP receiving entity acts as a TLS server. Therefore, to simplify discussion this document uses "_xmpp-client._tcp" to describe both cases, unless otherwise indicated.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Prooftype

POSH stands for PKIX Over Secure HTTP: the server’s proof consist of a PKIX certificate [RFC5280], the certificate is checked according to the rules from [RFC6120] and [RFC6125], the client obtains its verification material by retrieving the certificate over HTTPS ([RFC2616] and [RFC2818]) from a well-known URI [RFC5785], and secure DNS is not necessary since the HTTPS retrieval mechanism relies on the chain of trust based on the public key infrastructure.

The process for retrieving a PKIX certificate over secure HTTP is as follows.

1. The initiating entity performs an HTTPS GET at the source domain to the path "/.well-known/posh._<service>._tcp.json"; where "_<service>" MUST be either "_xmpp-client" for XMPP client-to-server connections or "_xmpp-server" for XMPP server-to-server connections. Here is an example:

```
HTTP GET /.well-known/posh._xmpp-server._tcp.json HTTP/1.1
Host: im.example.com
```

2. If the source domain HTTPS server has a certificate for the requested path, it MUST respond with a success status code, with the message body as a JSON Web Key Set (JWK Set) [JOSE-JWK], which itself contains at least one JWK of type "PKIX" [JOSE-PKIX-KEY] that the XMPP server at the source domain will

present during the TLS negotiation phase of XMPP stream setup (linebreaks and whitespace added for readability). Here is an example:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 806

{
  "keys":[
    {
      "kty":"PKIX",
      "x5c":[
        "MIICPTCCAaYCCQDDVeBaBmWC_jANBgkqhkiG9w0BAQUFADBjMQswCQY
         DVQQGEwJVUzERMA8GA1UECBMIQ29sb3JhZG8xDzANBgNVBAcTBkRlbn
         ZlcjEXMBUGA1UEChMOaW0uZXhhbXBsZS5jb20xFzAVBgNVBAMTDmltL
         mV4YW1wbGUuY29tMB4XDTEyMDYxMTIxNTQ0NFoXDTIyMDYwOTIxNTQ0
         NFowYzELMAkGA1UEBhMCVVMxETAPBgNVBAgTCENvbG9yYWRvMQ8wDQY
         DVQQHEwZEZW52ZXIxFzAVBgNVBAoTDmltLmV4YW1wbGUuY29tMRcwFQ
         YDVQQDEw5pbS5leGFtcGxlLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBj
         QAwgYkCgYEA4hoKhi_B07eQH-1NB9gWiNFDT__AbTHQOEC0AOr4Gh_o
         9PUp7kD0gklU4uv7rSAhAyCe4WaOiQ_HShzEryGfHiZmWht0BaYmj19
         iuPWRecZOXWqKZji9NtAxn9l3kdon_YLJcrPGyNTGK66-ggNaqy8LkQ
         QpI4rff60yHHZ_0XkCAwEAATANBgkqhkiG9w0BAQUFAAOBgQDcwiu30
         bSMlykWYz-tTDSlQ3wLSVB9RsR8jXmJvMo7y7icXwg54a9M3xipjZtr
         fAhYM5I5iqUTQPki6s26n9SQpRm5bonEFDA3WGwrwma35biP9-NSBWz
         SaDF8AztwFNKXXl6_U6hWwG05G_NdeS11gpww9NUDraJgVoDpRK04tg"
      ]
    }
  ]
}
```

4. Secure Delegation

When PKIX Over Secure HTTP (POSH) is the DNA prooftype, it is possible to use HTTPS redirects in determining if a domain is securely delegated, as follows:

1. The initiating entity performs an HTTPS GET at the source domain to the path "/.well-known/posh._<service>._tcp.json"; where "_<service>" MUST be either "_xmpp-client" for XMPP client-to-server connections or "_xmpp-server" for XMPP server-to-server connections. Here is an example:

```
GET /.well-known/posh._xmpp-server._tcp.json HTTP/1.1
Host: im.example.com
```

2. If the source domain HTTPS server has delegated to a derived domain, it MUST respond with one of the redirect mechanisms provided by HTTP (e.g., using the 302, 303, 307, or 308 response). The 'Location' header MUST specify an HTTPS URL, where the hostname and port is the derived domain HTTPS server, and the path MUST match the pattern "_<service>._tcp.json"; where "_<service>" MUST be identical to the "_<service>" portion of the original request (line breaks added for readability). Here is an example:

```
HTTP/1.1 302 Found
Location: https://hosting.example.net/.well-known
          /posh._xmpp-server._tcp.json
```

3. The initiating entity performs an HTTPS GET to the URL specified in the 'Location' header. Here is an example:

```
GET /.well-known/posh._xmpp-server._tcp.json HTTP/1.1
Host: hosting.example.net
```

4. If the derived domain HTTPS server has a certificate, it MUST respond with a success status code, with the message body as a JSON Web Key Set (JWK Set) [JOSE-JWK], which itself contains at least one JWK of type "PKIX" [JOSE-PKIX-KEY] that the XMPP server at the derived domain will present during the TLS negotiation phase of XMPP stream setup. Here is an example:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 806

{
  "keys":[
    {
      "kty":"PKIX",
      "x5c":[
        "MIICPTCCAaYCCQDDVeBaBmWC_jANBgkqhkiG9w0BAQUFADBjMQswCQY
         DVQQGEwJVUzERMA8GA1UECBMIQ29sb3JhZG8xDzANBgNVBAcTBkRlbn
         ZlcjEXMBUGA1UEChMOaW0uZXhhbXBsZS5jb20xFzAVBgNVBAMTDmltL
```

```
            mV4YW1wbGUuY29tMB4XDTEyMDYxMTIxNTQ0NFoXDTIyMDYwOTIxNTQ0
            NFowYzELMAkGA1UEBhMCVVMxETAPBgNVBAgTCENvbG9yYWRvMQ8wDQY
            DVQQHEwZEZW52ZXIxFzAVBgNVBAoTDmltLmV4YW1wbGUuY29tMRcwFQ
            YDVQQDEw5pbS5leGFtcGxlLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBj
            QAwgYkCgYEA4hoKhi_B07eQH-1NB9gWiNFDT__AbTHQOEC0AOr4Gh_o
            9PUp7kD0gklU4uv7rSAhAyCe4WaOiQ_HShzEryGfHiZmWht0BaYmj19
            iuPWRecZOXWqKZji9NtAxn9l3kdon_YLJcrPGyNTGK66-ggNaqy8LkQ
            QpI4rff60yHHZ_0XkCAwEAATANBgkqhkiG9w0BAQUFAAOBgQDcwiu30
            bSMlykWYz-tTDSlQ3wLSVB9RsR8jXmJvMo7y7icXwg54a9M3xipjZtr
            fAhYM5I5iqUTQPki6s26n9SQpRm5bonEFDA3WGwrwma35biP9-NSBWz
            SaDF8AztwFNKXXl6_U6hWwG05G_NdeS11gpww9NUDraJgVoDpRK04tg"
        ]
      }
    ]
  }
```

4.1. Permanent versus Temporary Redirects

Care needs to be taken with which redirect mechanism is used for delegation. Clients might remember the redirected location in place of the original, which can lead to verification mismatches when a source domain is migrated to a different delegated domain.

To mitigate this concern, source domains SHOULD use only temporary redirect mechanisms, such as HTTP status codes 302 (Found) and 307 (Temporary Redirect). Clients MAY treat any redirect as temporary, ignoring the specific semantics for 301 (Moved Permanently) or 308 (Permanent Redirect) [HTTP-STATUS-308].

5. Order of Operations

The processes for the POSH prooftype MUST be complete before the TLS handshake over the XMPP connection finishes, so that the client can perform verification of reference identities. Ideally a TLS client ought to perform the POSH processes in parallel with other XMPP session establishment processes; this is sometimes called the "happy eyeballs" approach, similar to [RFC6555] for IPv4 and IPv6. However, a TLS client might delay as much of the XMPP session establishment as it needs to in order to gather all of the POSH-based verification material. For instance, a TLS client might not open the socket connection until it retrieves the PKIX certificates.

6. Caching Results

Ideally, the initiating entity relies on the expiration time of the certificate obtained via POSH, and not on HTTP caching mechanisms.

To that end, the HTTPS servers for source and derived domains SHOULD specify a 'Cache-Control' header indicating a short duration (e.g., max-age=60) or "no-cache" to indicate the response (redirect or content) is not appropriate to cache at the HTTP level.

7. Alternates and Roll-over

To indicate alternate PKIX certificates, such as when an existing certificate will soon expire, the returned JWK Set can contain multiple "PKIX" JWK objects. The JWK Set SHOULD be ordered with the most relevant certificate first as determined by the XMPP server operator (e.g., the certificate soonest to expire), followed by the next most relevant certificate (e.g., the renewed certificate). Here is an example:


```
{
  "keys":[
    {
      "kty":"PKIX",
      "x5c":[
        "MIICYTCCAcqgAwIBAgIJAK_Lh7cXMZvdMA0GCSqGSIb3DQEBBQUAME
        8xCzAJBgNVBAYTAlVTMREwDwYDVQQIEwhDb2xvcmFkbzEPMA0GA1UEB
        xMGRGVudmVyMRwwGgYDVQQDExNob3N0aW5nLmV4YW1wbGUubmV0MB4X
        DTEzMDIwNzE4MjY0MFoXDTIzMDIwNTE4MjY0MFowTzELMAkGA1UEBhM
        CVVMxETAPBgNVBAgTCENvbG9yYWRvMQ8wDQYDVQQHEwZEZW52ZXIxHD
        AaBgNVBAMTE2hvc3RpbmcuZXhhbXBsZS5uZXQwgZ8wDQYJKoZIhvcNA
        QEBBQADgY0AMIGJAoGBAOLjqQxacJ-DQNOuVxNzoBBRyLku7V_ZEpFY
        8SHPyrK38I7Q3lWnEpAyUanpMClDMV0B_EJQDeueJgWkyrgd6bDZLvi
        _UtGha9E4q-IpHO6cM_cSE9d_oZuCcdGV8HHjK9m1xHUEyeTGAm1tMA
        m7j_BNFdhETkUqTfFPggFdMhAXAgMBAAGjRTBDMEEGA1UdEQQ6MDigI
        QYIKwYBBQUHCAWgFQwTaG9zdGluZy5leGFtcGxlLm5ldIITaG9zdGlu
        Zy5leGFtcGxlLm5ldDANBgkqhkiG9w0BAQUFAAOBgQAaz81gC5KqFQo
        WGf8mJz_mYx2pW6i-QeYw-BqpdAgdkrRvOHlJ4pYRhkajKfdiauvHcM
        ZDPWuuSm7jzIEOPqZdzYXkffgfr4br5UOAmYqpikpjlSsTLd5h_38p-
        3lz-l502wcs1xveBTYTIT13MAI844IBCZF-xDl-wpJG3kkttA"
      ]
    }
    {
      "kty":"PKIX",
      "x5c":[
        "MIIC-zCCAeOgAwIBAgIBAjANBgkqhkiG9w0BAQUFADBGMQswCQYDVQ
        QGEwJVUzERMA8GA1UECBMIQ29sb3JhZG8xDzANBgNVBAcTBkRlbnZlc
        jETMBEGA1UEAxMKRXhhbXBsZSBDQTAeFw0xMzAyMTIyMTI5MDBaFw0x
        NDAyMTIyMTI5MDBaME8xCzAJBgNVBAYTAlVTMREwDwYDVQQIEwhDb2x
        vcmFkbzEPMA0GA1UEBxMGRGVudmVyMRwwGgYDVQQDExNob3N0aW5nLm
        V4YW1wbGUubmV0MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDi4
        6kMWnCfg0DTrlcTc6AQUci5Lu1f2RKRWPEhz8qyt_CO0N5VpxKQMlGp
        6TApQzFdAfxCUA3rniYFpMq4Hemw2S74v1LRoWvROKviKRzunDP3EhP
```

```
            Xf6GbgnHRlfBx4yvZtcR1BMnkxgJtbTAJu4_wTRXYRE5FKk3xT4IBXT
            IQFwIDAQABo28wbTAMBgNVHRMBAf8EAjAAMB0GA1UdDgQWBBRgaaG6v
            5py2KwjtX-ToLKTEIqeVTALBgNVHQ8EBAMCBeAwEQYJYIZIAYb4QgEB
            BAQDAgZAMB4GCWCGSAGG-EIBDQQRFg94Y2EgY2VydGlmaWNhdGUwDQY
            JKoZIhvcNAQEFBQADggEBAE6Vhvd0OuMHJjyi8F8NoFSCRYOJXOry5B
            lmU6eVwEcUQSAkHaC4Q2isWCIES58Wm5P2VVQTYBUn58H7ZR9-7looj
            YVykwEIQmE_aaVsMM-8AwTMJ7qj7aGhXFlKT2xwiPMVq9JF_Gv43qSy
            V9GJ3Uw5Jz6AN4WawXmlIVD0eKhPoHSDO0wfnFc8KM8mHPu7JXqIriX
            18w4jfj3ySuHIkXeOjdbDWqZWJ7akBVf8McbB05tXP5T7sDTV-t8qH5
            6fdnSQC-qO-sQgmWlKLFtKybT6Fa6J7ChEd_sOJNqB9SoMar5sRYyfS
            foV0D7m_IF1MI6X95rL1YnKIGxDYWBq4ck",
            "MIIDeTCCAmGgAwIBAgIBATANBgkqhkiG9w0BAQUFADBGMQswCQYDVQ
            QGEwJVUzERMA8GA1UECBMIQ29sb3JhZG8xDzANBgNVBAcTBkRlbnZlc
            jETMBEGA1UEAxMKRXhhbXBsZSBDQTAeFw0xMzAyMTIyMTI4MDBaFw0y
            MzAyMTIyMTI4MDBaMEYxCzAJBgNVBAYTAlVTMREwDwYDVQQIEwhDb2x
            vcmFkbzEPMA0GA1UEBxMGRGVudmVyMRMwEQYDVQQDEwpFeGFtcGxlIE
            NBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzNQ30X7uX
            Tg-4jKadtRO5uQEMRMnkZvDnptbWAtx0d1PsufQ2kfvog0gDhigjPEZ
            DV9S-zm63Ia-eqJ3ROT9jDXjtF6s_IawITf5cPSNxn8qP8w-vbiy0rB
            4W4Nk1Dwji7KJ_wKNo0mwOx_qWNjSk3yoaU4sUEuIypizgLxKAr25vV
            vAJAxF6HAfdQoVAIdCZ_7qbBPI7aurdU_NdmbbKBK0lp8aV1MYLzz8D
            I0hWcBQa2-gOSUcd_yT1az7UpMjGllbnVlUDxyJeCzbBaHny5NlWWHs
            GnsbucbM-9yeAMbRes_z0KeHxcRtomd8bh7As12RIXKrk5GRoNVKAoi
            wLQIDAQABo3IwcDAPBgNVHRMBAf8EBTADAQH_MB0GA1UdDgQWBBSyie
            t77RfWpH3X8NMwGFVu2ldJPTALBgNVHQ8EBAMCAQYwEQYJYIZIAYb4Q
            gEBBAQDAgAHMB4GCWCGSAGG-EIBDQQRFg94Y2EgY2VydGlmaWNhdGUw
            DQYJKoZIhvcNAQEFBQADggEBAIE-gvYX-2MOAmL3qOraIYUb1eDeUyC
            rxroqrI1xX3jDapMPltCxuZr8VkLljHaNpe7sLJlFWSaQHkZe4snxWL
            SdINLrgFhxskclAlSLutPVTA4xPwo60t0hBJE0NJ8kC8gVvvlWXWAiI
            IVszG3vLBcfxZeuOS4JsVwGbTt5uKsVIJ2VkRiBG4ey5lsS5O8u0vRf
            ei7HFr1NzZ8y5BHoix9VLN2--n11SNicwDOo2V618B8GQnPqM2dsaDa
            A1wIrMZeEyoRtIN25jcW-as4sS9dPJ1ueNIzrSuzlXtKYGjflaTcEfD
            -_kImTw9tHzS57iBXHqgQTQo61pYzAZMlk9wA"
          ]
        }
      ]
    }
```

8. Security Considerations

This document supplements but does not supersede the security considerations provided in [RFC2616], [RFC2818], [RFC6120], and [RFC6125].

Specifically, communication via HTTPS depends on checking the identity of the HTTP server in accordance with [RFC2818].

9. IANA Considerations

9.1. The "posh._xmpp-client._tcp.json" Well-Known URI

This specification registers the "posh._xmpp-client._tcp.json" well-known URI in the Well-Known URI Registry as defined by [RFC5785].

URI suffix: posh._xmpp-client._tcp.json

Change controller: IETF

Specification document(s): [[this document]]

9.2. The "posh._xmpp-server._tcp.json" Well-Known URI

This specification registers the "posh._xmpp-server._tcp.json" well-known URI in the Well-Known URI Registry as defined by [RFC5785].

URI suffix: posh._xmpp-server._tcp.json

Change controller: IETF

Specification document(s): [[this document]]

10. References

10.1. Normative References

[JOSE-JWK]
Jones, M., "JSON Web Key (JWK)", draft-ietf-jose-json-web-key-08 (work in progress), December 2012.

[JOSE-PKIX-KEY]
Miller, M., "JSON Web Key (JWK) for PKIX Certificates", draft-miller-jose-pkix-key-01 (work in progress), February 2013.

[XMPP-DNA]
Saint-Andre, P. and M. Miller, "Domain Name Associations (DNA) in the Extensible Messaging and Presence Protocol (XMPP)", draft-saintandre-xmpp-dna-01 (work in progress), February 2013.

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

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

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, April 2010.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.

10.2. Informative References

[HTTP-STATUS-308] Reschke, J., "The Hypertext Transfer Protocol (HTTP) Status Code 308 (Permanent Redirect)", draft-reschke-http-status-308-07 (work in progress), March 2012.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, May 2005.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

10.3. Informative References

[RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.

Authors' Addresses

Matthew Miller
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Email: mamille2@cisco.com

Peter Saint-Andre
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Email: psaintan@cisco.com