Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 6, 2013

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

June 4, 2013

PKIX over Secure HTTP (POSH)
draft-miller-posh-00

Abstract

This document defines two methods that make it easier to deploy certificates for proper server identity checking in application protocols. The first method enables a TLS client to obtain a TLS server's end-entity certificate over secure HTTP as an alternative to standard Public Key Infrastructure using X.509 (PKIX) and DNS-Based Authentication of Named Entities (DANE). The second method enables a source domain to securely delegate an application to a derived domain using HTTPS redirects.

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 December 6, 2013.

Table of Contents

1. Introduction

Channel encryption with TLS depends on proper checking of the server's identity, as specified in [RFC6125] or its application-specific equivalent for Public Key Infrastructure using X.509 (PKIX) [RFC5280] and in [RFC6698] for DNS-Based Authentication of Named Entities (DANE). However, in multi-tenanted environments it is effectively impossible for a hosting service to offer the correct PKIX certificates on behalf of a hosted domain, since neither party wants the hosting service to hold the hosted domain's private keys. As a result, typically the hosting service offers its own PKIX certificate (say, for hosting.example.net), which means that TLS clients need to "just know" that the hosted domain (say, foo.example.com) is offered at the hosting service rather than the hosted domain. Further background information on this problem can be found in [XMPP-DNA].

This situation is clearly insecure. It is true that DNS-based technologies are emerging for secure delegation, in the form of DNS SRV records [RFC2782] or their functional equivalent when DNS Security [RFC4033] is used, along with DNS-Based Authentication of Named Entities (DANE) [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. Hosting services and hosted domains need a method that can be deployed more quickly to overcome the lack of secure delegation and proper server identity checking on the Internet today.

POSH (PKIX Over Secure HTTP) provides two interconnected methods for solving the problem, at least with application protocols other than HTTP:

1. A TLS client retrieves the material to be used in checking the TLS server's identity by requesting it from a well-known HTTPS URI, where the response contains one or more certificates formatted as a JSON Web Key set [JOSE-JWK] defined within the JOSE WG.

2. If a hosted domain securely delegates an application to a hosting service, it redirects all requests for the well-known HTTPS URI to an HTTPS URI at the hosting service.

The discussion venue for this document is the posh@ietf.org mailing list; visit https://www.ietf.org/mailman/listinfo/posh for subscription information and discussion archives.

2. Terminology

This document inherits 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 uses the Extensible Messaging and Presence Protocol (XMPP) [RFC6120] in its examples. Whether connections are made from an XMPP client to an XMPP server (based on a DNS SRV record of "_xmpp-client._tcp") or between XMPP servers ("_xmpp-server._tcp"), 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. Obtaining Verification Materials

Server identity checking (see [RFC6125]) involves three different aspects:

1. A proof of the TLS server's identity (in PKIX, this takes the form of a PKIX certificate [RFC5280]).
2. Rules for checking the certificate (which vary by application protocol, although [RFC6125] attempts to harmonize those rules).
3. The materials that a TLS client uses to verify the TLS server's identity or check the TLS server's proof (in PKIX, this takes the form of chaining the end-entity certificate back to a trusted root and performing all validity checks as described in [RFC5280], [RFC6125], and the relevant application protocol specification).

When POSH is used, the first two aspects remain the same: the TLS server proves it identity by presenting a PKIX certificate [RFC5280] and the certificate is checked according to the rules defined in the appropriate application protocol specification (such as [RFC6120] for XMPP). However, the TLS client obtains the material it will use to verify the server's proof by retrieving a JSON Web Key (JWK) set [JOSE-JWK] over HTTPS ([RFC2616] and [RFC2818]) from a well-known URI [RFC5785]. (In this case, 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 TLS client performs an HTTPS GET at the source domain to the path "/.well-known/posh.{service}.{protocol}.json". For example, if the application protocol is XMPP then the "{service}" is either "_xmpp-client" for XMPP client-to-server connections, and the "{protocol}" is "_tcp"; thus if an XMPP client were to use POSH to verify an XMPP server for the domain "im.example.com", the HTTPS GET request would be as follows:

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

2. The source domain HTTPS server responds in one of three ways:

   * If it possesses a PKIX certificate for the requested path, it responds as detailed in Section 3.1.

   * If it has a reference to where the PKIX certificate can be obtained, it responds as detailed in Section 3.2.

   * If it does not have any PKIX certificate for the requested path, it responds with a client error status code (e.g., 404).

3.1. Source Domain Possesses PKIX Certificate

If the source domain HTTPS server possesses the certificate information, it responds to the HTTPS GET with a success status code and the message body set to a JSON Web Key (JWK) set [JOSE-JWK]. The JWK set MUST contain at least one JWK with the following information:

- The "kty" field set to the appropriate key type (e.g., "RSA" for a certificate using an RSA key).

- The required fields for the key type (e.g., "n" and "e" for a certificate using an RSA key).

- The "x5c" field set to the certificate chain.

Example Content Response

```
HTTP/1.1 200 OK
Content-Type: application/jwk-set+json
Content-Length: 2785
```

```
{
  "keys": [
    {
      "kty": "RSA",
      "kid": "im.example.com:2011-07-04",
      "n":   "ANxwssdcU3LbODErec3owrwUhlzjtuskAn8rAcBMRPImn5xA
              JRX-1T5g2D7MTozWWFk4TlpgzAR5slvM0tc35qAI9I0Cqk4Z
              LChQrYsWuY7alTrnNXdusHUYc6Eq89DZaH2knTcp57wAXzJP
              IG_tpBi5F7ck9LVRvRjybix0HJ7i4YrL-GeLuSgrjO4-GDcX
              Ip8oV0FMKZH-NoMfUITlWYl_JcX1D0WUAiuAnvWtD4Kh_qMJ
              U6FZuupZGHqPdc3vrXtp27LWgxzxjFa9qnOU6y53vCCJXLLI
              5sy2fCwEDzLJqh2T6UItIzjrSUZMIsK8r2pXkroI0uYuNn3W
              y-jAzK8",
      "e":   "AQAB",
      "x5c": [
        "MIIDgzCCAmugAwIBAgIBBjANBgkqhkiG9w0BAQUFADBGMQswCQYDV
         QQGEwJVUzERMA8GA1UECBMIQ29sb3JhZG8xDzANBgNVBAcTBkRlbn
         ZlcjETMBEGA1UEAxMKRXhhbXBsZSBDQTAeFw0xMTA3MDQwMDAwMDB
         aFw0xMzA3MDIyMzU5NTlaMEoxCzAJBgNVBAYTAlVTMREwDwYDVQQI
         EwhDb2xvcmFkbzEPMA0GA1UEBxMGRGVudmVyMRcwFQYDVQQDEw5pb
         S5leGFtcGxlLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQ
         oCggEBANxwssdcU3LbODErec3owrwUhlzjtuskAn8rAcBMRPImn5x
         AJRX+1T5g2D7MTozWWFk4TlpgzAR5slvM0tc35qAI9I0Cqk4ZLChQ
         rYsWuY7alTrnNXdusHUYc6Eq89DZaH2knTcp57wAXzJPIG/tpBi5F
         7ck9LVRvRjybix0HJ7i4YrL+GeLuSgrjO4+GDcXIp8oV0FMKZH+No
         MfUITlWYl/JcX1D0WUAiuAnvWtD4Kh/qMJU6FZuupZGHqPdc3vrXt
         p27LWgxzxjFa9qnOU6y53vCCJXLLI5sy2fCwEDzLJqh2T6UItIzjr
         SUZMIsK8r2pXkroI0uYuNn3Wy+jAzK8CAwEAAaN4MHYwDAYDVR0TA
         QH/BAIwADAdBgNVHQ4EFgQUTmRcur7xqaIUoU6wjVFPFxpf3UYwCw
         YDVR0PBAQDAgXgMCcGA1UdEQQgMB6gHAYIKwYBBQUHCAWgEAwOaW0
         uZXhhbXBsZS5jb20wEQYJYIZIAYb4QgEBBAQDAgZAMA0GCSqGSIb3
         DQEBBQUAA4IBAQBrtpz4USAT+gNWI8ccU9rFiP0Jr+76VCf8Leims
         qjINfKuUFxVUK5TBcCU8pyRUdXBk5THt+LUW+bPqE4SAuKjTJ1wwm
         e8kOqtsvrr6XDfPHyX6H7nQAaKD0VbvbHfTBKh6jNVVi+4gJACeSE
         JdiskoNYuJAxNDI8DmN9qAxu/8dlQHlIT3NkTxMWFUdmW8rj2xdia
         nfZEwuPXoI93jdpgvGhcSM92ahumFyEZ5ysK6KFsXyUmVuOQFaVsH
         tSAwrSGr70ASLzsCAi7JsvzO53QFW/KddkFLvEwCh/tgKK876poBo
         x1NI6YYuWqhcKWADOOJdSfiXeu23E25tlbDRo8",
        "MIIDWTCCAkGgAwIBAgIBATANBgkqhkiG9w0BAQUFADBGMQswCQYDV
         QQGEwJVUzERMA8GA1UECBMIQ29sb3JhZG8xDzANBgNVBAcTBkRlbn
         ZlcjETMBEGA1UEAxMKRXhhbXBsZSBDQTAeFw0xMTA1MDIwMDAwMDB
         aFw0yMzA1MTYyMzU5NTlaMEYxCzAJBgNVBAYTAlVTMREwDwYDVQQI
         EwhDb2xvcmFkbzEPMA0GA1UEBxMGRGVudmVyMRMwEQYDVQQDEwpFe
         GFtcGxlIENBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQ
         EAzNQ30X7uXTg+4jKadtRO5uQEMRMnkZvDnptbWAtx0d1PsufQ2kf
         vog0gDhigjPEZDV9S+zm63Ia+eqJ3ROT9jDXjtF6s/IawITf5cPSN
         xn8qP8w+vbiy0rB4W4Nk1Dwji7KJ/wKNo0mwOx/qWNjSk3yoaU4sU
         EuIypizgLxKAr25vVvAJAxF6HAfdQoVAIdCZ/7qbBPI7aurdU/Ndm
```

```
            bbKBK0lp8aV1MYLzz8DI0hWcBQa2+gOSUcd/yT1az7UpMjGllbnVl
            UDxyJeCzbBaHny5NlWWHsGnsbucbM+9yeAMbRes/z0KeHxcRtomd8
            bh7As12RIXKrk5GRoNVKAoiwLQIDAQABo1IwUDAPBgNVHRMBAf8EB
            TADAQH/MB0GA1UdDgQWBBSyiet77RfWpH3X8NMwGFVu2ldJPTALBg
            NVHQ8EBAMCAQYwEQYJYIZIAYb4QgEBBAQDAgAHMA0GCSqGSIb3DQE
            BBQUAA4IBAQBd1mMx4Wx9xFLqecbjWyy7tOE2+mrWhWxg82q7z3bB
            rHWjUGzolHe97Ch+6QI3+MPk9JQWYaMgYe11tyf0mgZ18NFQall4M
            ho2yT+E8ju11PW+RNqUdRG6rZfdeN5Geb1o1L2g5WNTdtPXoFYgHY
            VPQ1HmjloEic2eGnlBvOi49wAdwnASv53fgzkSJB2/GdBJ3wPIWp0
            49/1vS5rsF5SJg+3mj3ZAuPYt80TRKbA/cjxEny5RfK+VJs3f7RQ/
            Y3CTPxoJqskWs06/eUpjXKyzZ+MmkCs5cm1yers8goWhaI8JmLlBW
            LQE6v8MHdbUfb4M8la5cUd2BGtTlILOVnMv"
        ]
      }
    ]
  }
```

The TLS client uses the provided certificate to verify the TLS connection to the TLS server. In order for the TLS client to verify the identity of the TLS server, it MUST ensure that the PKIX certificate presented by the TLS server during the TLS negotiation matches the certificate that it obtained via POSH.

The TLS client MAY verify the certificate chain provided in the JWK, but it SHOULD consider the final issuer certificate to be a trust anchor for the purposes of this verification only. Once it has verified the identity of the TLS server, the TLS client MUST NOT continue to treat this final issuer certificate as a trust anchor.

3.2. Source Domain References PKIX Certificate

If the source domain HTTPS server has a reference to the certificate information, it responds to the HTTPS GET with a redirect status code (e.g., 302, 303, 307, or 308), and includes a 'Location' header, which MUST specify an HTTPS URL.

Example Redirect Response

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

The client follows the redirect, the HTTPS server for the URI at which the client has been redirected responds to the request, and the client performs actions appropriate to the new response (whether it is a possession, a reference, or another redirect).

3.2.1. Redirect Status Codes

Care needs to be taken regarding the 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) and 308 (Permanent Redirect) [HTTP-STATUS-308].

3.2.2. Redirect Depth

To protect against circular references, clients MUST NOT follow an infinite number of redirects. It is RECOMMENDED that clients follow no more than 10 redirects, although applications or implementations can require that fewer redirects be followed.

3.3. Additional Security Mechanisms

POSH can benefit from additional HTTPS security mechanisms, such as HTTP Strict Transport Security [RFC6797] and key pinning [KEYPIN], especially if the TLS client shares some information with a common HTTPS implementation (e.g., platform-default web browser).

4. Secure Delegation

The delegation from the source domain to the delegated domain can be considered secure if the certificate offered by the TLS server matches the POSH certificate, regardless of how the POSH certificates are obtained.

5. Order of Operations

POSH processes MUST be complete before the end of the TLS handshake for the application protocol, so that the TLS client can perform verification of reference identifiers. Ideally a TLS client ought to perform the POSH processes in parallel with other application-level negotiation; 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 application-level negotiation 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 via POSH.

6. Caching Results

Ideally, the TLS client 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 that 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 MAY contain multiple JWK objects. The JWK set SHOULD be ordered with the most relevant certificate first as determined by the application service operator (e.g., the renewed certificate), followed by the next most relevant certificate (e.g., the certificate soonest to expire). Here is an example:

```
{
  "keys":[
    {
      "kty": "RSA",
      "kid": "hosting.example.net:2011-07-04",
      "n":   "AM-ktWkQ8btj_HEdAA6kOpzJGgoHNZsJmxjh_PifpgAUfQeq
              MO_YBR100IdJZRzJfULyhRwn9bikCq87WToxgPWOnd3sH3qT
              YiAcIR5S6tBbsyp6WYmwM1yuC0vLCo6SoDzdK1SvkQKM3QWk
              0GFNU4l4qXYAMxaSw83i6yv5DBVbST7E92vS6Gq_4pgI26l1
              0JhybZuTEVPRUCG6pTKAXQpLxmjJ5oG9M91RP17nsuQeE7Ng
              0Ap4BBn5hocojkfthwgbX4lqBMecpBAnky5jn6slmzS_rL-L
              w-_8hUldaTPD9MHlHPrvcsRV5uw8wK5MB6QyfS6wF4b0Kj2T
              vYceNlE",
      "e":   "AQAB",
      "x5c": [
        "MIIDXzCCAkegAwIBAgIBAzANBgkqhkiG9w0BAQUFADBGMQswCQYDV
         QQGEwJVUzERMA8GA1UECBMIQ29sb3JhZG8xDzANBgNVBAcTBkRlbn
         ZlcjETMBEGA1UEAxMKRXhhbXBsZSBDQTAeFw0xMTA3MDQxOTUyMDB
         aFw0xMzA3MDMxOTUyMDBaME8xCzAJBgNVBAYTAlVTMREwDwYDVQQI
         EwhDb2xvcmFkbzEPMA0GA1UEBxMGRGVudmVyMRwwGgYDVQQDExNob
         3N0aW5nLmV4YW1wbGUubmV0MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ
         8AMIIBCgKCAQEAz6S1aRDxu2P8cR0ADqQ6nMkaCgc1mwmbGOH8+J+
         mABR9B6ow79gFHXTQh0llHMl9QvKFHCf1uKQKrztZOjGA9Y6d3ewf
         epNiIBwhHlLq0FuzKnpZibAzXK4LS8sKjpKgPN0rVK+RAozdBaTQY
         U1TiXipdgAzFpLDzeLrK/kMFVtJPsT3a9Loar/imAjbqXXQmHJtm5
         MRU9FQIbqlMoBdCkvGaMnmgb0z3VE/Xuey5B4Ts2DQCngEGfmGhyi
         OR+2HCBtfiWoEx5ykECeTLmOfqyWbNL+sv4vD7/yFSV1pM8P0weUc
         +u9yxFXm7DzArkwHpDJ9LrAXhvQqPZO9hx42UQIDAQABo08wTTAMB
```

```
        gNVHRMBAf8EAjAAMB0GA1UdDgQWBBQ/veMa6XwrIaUv8Y7PmW0RyA
        Um9jALBgNVHQ8EBAMCBeAwEQYJYIZIAYb4QgEBBAQDAgZAMA0GCSq
        GSIb3DQEBBQUAA4IBAQA7V50iyHg8+l2UBkFa8l6APKQ5zL2qN8d3
        sE3mDK5a6l/597xHDxzHKMmROvHD9+MHZtYxbB0dHz1lJY0zCUAgo
        nfYc9J3VB4kKPB9H7u8h70pRuPudFWvQ4ZRraRPm7eSP+7/kTlOKa
        MCpBCiA95GKAbsbY3vQ0Hkmu5sgbIwGNs5x5V4kZSN9AfFHcmaQ2K
        ZufaiLjLPj6UC5CObGXjsMCWMiS7kzw8GWxnQ9viCM0uopmraMOUH
        cPnmtlzXparpWbiFKXGwFo1qU9QntoO71kJwVm9+ABl+lMD22WKxj
        5DDutWSyxV7Nbbhni/j6HdWHNNcCNllbKzqJ54RhDoi",
       "MIIDWTCCAkGgAwIBAgIBATANBgkqhkiG9w0BAQUFADBGMQswCQYDV
        QQGEwJVUzERMA8GA1UECBMIQ29sb3JhZG8xDzANBgNVBAcTBkRlbn
        ZlcjETMBEGA1UEAxMKRXhhbXBsZSBDQTAeFw0xMTA1MDIwMDAwMDB
        aFw0yMzA1MTYyMzU5NTlaMEYxCzAJBgNVBAYTAlVTMREwDwYDVQQI
        EwhDb2xvcmFkbzEPMA0GA1UEBxMGRGVudmVyMRMwEQYDVQQDEwpFe
        GFtcGxlIENBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQ
        EAzNQ30X7uXTg+4jKadtRO5uQEMRMnkZvDnptbWAtx0d1PsufQ2kf
        vog0gDhigjPEZDV9S+zm63Ia+eqJ3ROT9jDXjtF6s/IawITf5cPSN
        xn8qP8w+vbiy0rB4W4Nk1Dwji7KJ/wKNo0mwOx/qWNjSk3yoaU4sU
        EuIypizgLxKAr25vVvAJAxF6HAfdQoVAIdCZ/7qbBPI7aurdU/Ndm
        bbKBK0lp8aV1MYLzz8DI0hWcBQa2+gOSUcd/yT1az7UpMjGllbnVl
        UDxyJeCzbBaHny5NlWWHsGnsbucbM+9yeAMbRes/z0KeHxcRtomd8
        bh7As12RIXKrk5GRoNVKAoiwLQIDAQABo1IwUDAPBgNVHRMBAf8EB
        TADAQH/MB0GA1UdDgQWBBSyiet77RfWpH3X8NMwGFVu2ldJPTALBg
        NVHQ8EBAMCAQYwEQYJYIZIAYb4QgEBBAQDAgAHMA0GCSqGSIb3DQE
        BBQUAA4IBAQBd1mMx4Wx9xFLqecbjWyy7tOE2+mrWhWxg82q7z3bB
        rHWjUGzolHe97Ch+6QI3+MPk9JQWYaMgYe11tyf0mgZ18NFQall4M
        ho2yT+E8ju11PW+RNqUdRG6rZfdeN5Geb1o1L2g5WNTdtPXoFYgHY
        VPQ1HmjloEic2eGnlBvOi49wAdwnASv53fgzkSJB2/GdBJ3wPIWp0
        49/1vS5rsF5SJg+3mj3ZAuPYt80TRKbA/cjxEny5RfK+VJs3f7RQ/
        Y3CTPxoJqskWs06/eUpjXKyzZ+MmkCs5cm1yers8goWhaI8JmLlBW
        LQE6v8MHdbUfb4M8la5cUd2BGtTlILOVnMv"
    ]
  },
  {
    "kty": "RSA",
    "kid": "hosting.example.net:2013-07-04",
    "n":   "AM-ktWkQ8btj_HEdAA6kOpzJGgoHNZsJmxjh_PifpgAUfQeq
            MO_YBR100IdJZRzJfULyhRwn9bikCq87WToxgPWOnd3sH3qT
            YiAcIR5S6tBbsyp6WYmwM1yuC0vLCo6SoDzdK1SvkQKM3QWk
            0GFNU4l4qXYAMxaSw83i6yv5DBVbST7E92vS6Gq_4pgI26l1
            0JhybZuTEVPRUCG6pTKAXQpLxmjJ5oG9M91RP17nsuQeE7Ng
            0Ap4BBn5hocojkfthwgbX4lqBMecpBAnky5jn6slmzS_rL-L
            w-_8hUldaTPD9MHlHPrvcsRV5uw8wK5MB6QyfS6wF4b0Kj2T
            vYceNlE",
    "e":   "AQAB",
    "x5c": [
      "MIIDjTCCAnWgAwIBAgIBBTANBgkqhkiG9w0BAQUFADBGMQswCQYDV
      QQGEwJVUzERMA8GA1UECBMIQ29sb3JhZG8xDzANBgNVBAcTBkRlbnZ
```

```
            lcjETMBEGA1UEAxMKRXhhbXBsZSBDQTAeFw0xMzA1MTcwMDAwMDBaF
            w0xNTA1MTYyMzU5NTlaME8xCzAJBgNVBAYTAlVTMREwDwYDVQQIEwh
            Db2xvcmFkbzEPMA0GA1UEBxMGRGVudmVyMRwwGgYDVQQDExNob3N0a
            W5nLmV4YW1wbGUubmV0MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMII
            BCgKCAQEAz6S1aRDxu2P8cR0ADqQ6nMkaCgc1mwmbGOH8+J+mABR9B
            6ow79gFHXTQh0llHMl9QvKFHCf1uKQKrztZOjGA9Y6d3ewfepNiIBw
            hHlLq0FuzKnpZibAzXK4LS8sKjpKgPN0rVK+RAozdBaTQYU1TiXipd
            gAzFpLDzeLrK/kMFVtJPsT3a9Loar/imAjbqXXQmHJtm5MRU9FQIbq
            lMoBdCkvGaMnmgb0z3VE/Xuey5B4Ts2DQCngEGfmGhyiOR+2HCBtfi
            WoEx5ykECeTLmOfqyWbNL+sv4vD7/yFSV1pM8P0weUc+u9yxFXm7Dz
            ArkwHpDJ9LrAXhvQqPZO9hx42UQIDAQABo30wezAMBgNVHRMBAf8EA
            jAAMB0GA1UdDgQWBBQ/veMa6XwrIaUv8Y7PmW0RyAUm9jALBgNVHQ8
            EBAMCBeAwLAYDVR0RBCUwI6AhBggrBgEFBQcIBaAVDBNob3N0aW5nL
            mV4YW1wbGUubmV0MBEGCWCGSAGG+EIBAQQEAwIGQDANBgkqhkiG9w0
            BAQUFAAOCAQEAfb3++qh/bxsVtcEWmNtj1QLTBiXC5T9U6NuWUKY3a
            RTIsthO5yLsToUG1bbxLceVG+xPQT9uRqvV02sumiIn3gv87NU5gMR
            b1Gy1P9N0YT5GiHCN5qCfegGH6jIokh0Js3v+ttR/rUns0/MEsr/ka
            YQ7KRKzOdlxT+gWN9Lp0fOYdYcasDWNMTg1NZwzXKcdkhb1lRBYuP4
            SeiJvy6ZbyUnoswthVTe0CK9+StDtJZLPSUu7TB2MwYMnv344NoTO2
            ufRawLU0+8cJZxrXq/H/YQyoAFB9opw0wEoTIB5aZqI2em4Wjgx9sl
            aonFlLEQysm5qJnhTGxEIDxQfqCcRfw==",
          "MIIDWTCCAkGgAwIBAgIBATANBgkqhkiG9w0BAQUFADBGMQswCQYDVQ
            QGEwJVUzERMA8GA1UECBMIQ29sb3JhZG8xDzANBgNVBAcTBkRlbnZl
            cjETMBEGA1UEAxMKRXhhbXBsZSBDQTAeFw0xMTA1MDIwMDAwMDBaFw
            0yMzA1MTYyMzU5NTlaMEYxCzAJBgNVBAYTAlVTMREwDwYDVQQIEwhD
            b2xvcmFkbzEPMA0GA1UEBxMGRGVudmVyMRMwEQYDVQQDEwpFeGFtcG
            xlIENBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzNQ3
            0X7uXTg+4jKadtRO5uQEMRMnkZvDnptbWAtx0d1PsufQ2kfvog0gDh
            igjPEZDV9S+zm63Ia+eqJ3ROT9jDXjtF6s/IawITf5cPSNxn8qP8w+
            vbiy0rB4W4Nk1Dwji7KJ/wKNo0mwOx/qWNjSk3yoaU4sUEuIypizgL
            xKAr25vVvAJAxF6HAfdQoVAIdCZ/7qbBPI7aurdU/NdmbbKBK0lp8a
            V1MYLzz8DI0hWcBQa2+gOSUcd/yT1az7UpMjGllbnVlUDxyJeCzbBa
            Hny5NlWWHsGnsbucbM+9yeAMbRes/z0KeHxcRtomd8bh7As12RIXKr
            k5GRoNVKAoiwLQIDAQABo1IwUDAPBgNVHRMBAf8EBTADAQH/MB0GA1
            UdDgQWBBSyiet77RfWpH3X8NMwGFVu2ldJPTALBgNVHQ8EBAMCAQYw
            EQYJYIZIAYb4QgEBBAQDAgAHMA0GCSqGSIb3DQEBBQUAA4IBAQBd1m
            Mx4Wx9xFLqecbjWyy7tOE2+mrWhWxg82q7z3bBrHWjUGzolHe97Ch+
            6QI3+MPk9JQWYaMgYe11tyf0mgZ18NFQall4Mho2yT+E8ju11PW+RN
            qUdRG6rZfdeN5Geb1o1L2g5WNTdtPXoFYgHYVPQ1HmjloEic2eGnlB
            vOi49wAdwnASv53fgzkSJB2/GdBJ3wPIWp049/1vS5rsF5SJg+3mj3
            ZAuPYt80TRKbA/cjxEny5RfK+VJs3f7RQ/Y3CTPxoJqskWs06/eUpj
            XKyzZ+MmkCs5cm1yers8goWhaI8JmLlBWLQE6v8MHdbUfb4M8la5cU
            d2BGtTlILOVnMv"
        ]
      }
    ]
  }
```

8. Security Considerations

This document supplements but does not supersede the security considerations provided in specifications for application protocols that decide to use POSH (e.g., [RFC6120] and [RFC6125] for XMPP). Specifically, communication via HTTPS depends on checking the identity of the HTTP server in accordance with [RFC2818].

Additionally, the security of POSH can benefit from other HTTP hardening protocols, such as HSTS [RFC6797] and key pinning [KEYPIN].

9. IANA Considerations

Protocols that use POSH MUST register an appropriate well-known URI or URIs [RFC5785] with the IANA. The IANA registration policy [RFC5226] is Specification Required.

The following sections register two such URIs for XMPP.

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-11 (work in progress), May 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.

[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.

[KEYPIN] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", draft-ietf-websec-key-pinning-04 (work in progress), December 2012.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S.

Rose, "DNS Security Introduction and Requirements", RFC 4033, May 2005.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

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

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

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

[RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTPS Strict Transport Security (HSTS)", RFC 6797, November 2012.

[XMPP-DNA] Saint-Andre, P. and M. Miller, "Domain Name Associations (DNA) in the Extensible Messaging and Presence Protocol (XMPP)", draft-ietf-xmpp-dna-02 (work in progress), April 2013.

Appendix A. Acknowledgements

Thanks to Dave Cridland, Max Pritikin, and Joe Salowey for their feedback.

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