STIR Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 25, 2015

J. Peterson
NeuStar
S. Turner
IECA

March 24, 2015

Secure Telephone Identity Credentials: Certificates
draft-ietf-stir-certificates-01.txt

Abstract

In order to prove ownership of telephone numbers on the Internet, some kind of public infrastructure needs to exist that binds cryptographic keys to authority over telephone numbers. This document describes a certificate-based credential system for telephone numbers, which could be used as a part of a broader architecture for managing telephone numbers as identities in protocols like SIP.

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 September 22, 2015.

Table of Contents

Table of Contents

1. Introduction

As is discussed in the STIR problem statement [13], the primary enabler of robocalling, vishing, swatting and related attacks is the capability to impersonate a calling party number. The starkest examples of these attacks are cases where automated callees on the Public Switched Telephone Network (PSTN) rely on the calling number as a security measure, for example to access a voicemail system. Robocallers use impersonation as a means of obscuring identity; while robocallers can, in the ordinary PSTN, block (that is, withhold) their caller identity, callees are less likely to pick up calls from blocked identities, and therefore appearing to calling from some number, any number, is preferable. Robocallers however prefer not to call from a number that can trace back to the robocaller, and therefore they impersonate numbers that are not assigned to them.

One of the most important components of a system to prevent impersonation is an authority responsible for issuing credentials to parties who control telephone numbers. With these credentials, parties can prove that they are in fact authorized to use telephony numbers, and thus distinguish themselves from impersonators unable to present credentials. This document describes a credential system for telephone numbers based on X.509 version 3 certificates in accordance

with [7]. While telephone numbers have long been a part of the X.509 standard, the certificates described in this document may contain telephone number blocks or ranges, and accordingly it uses an alternate syntax.

In the STIR in-band architecture, two basic types of entities need access to these credentials: authentication services, and verification services (or verifiers); see [15]. An authentication service must be operated by an entity enrolled with the certification authority (see Section 3), whereas a verifier need only trust the root certificate of the authority, and have a means to acquire and validate certificates.

This document attempts to specify only the basic elements necessary for this architecture. Only through deployment experience will it be possible to decide directions for future work.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [1] and RFC 6919 [2].

3. Enrollment and Authorization

This document assumes a threefold model for certificate enrollment.

The first enrollment model is one where the certification authority (CA) acts in concert with national numbering authorities to issue credentials to those parties to whom numbers are assigned. In the United States, for example, telephone number blocks are assigned to Local Exchange Carriers (LECs) by the North American Numbering Plan Administrator (NANPA), who is in turn directed by the national regulator. LECs may also receive numbers in smaller allocations, through number pooling, or via an individual assignment through number portability. LECs assign numbers to customers, who may be private individuals or organizations - and organizations take responsibility for assigning numbers within their own enterprise.

The second enrollment model is one where a certification authority requires that an entity prove control by means of some sort of test. For example, an authority might send a text message to a telephone number containing a URL (which might be deferenced by the recipient) as a means of verifying that a user has control of terminal corresponding to that number. Checks of this form are frequently used in commercial systems today to validate telephone numbers provided by users. This is comparable to existing enrollment systems

used by some certificate authorities for issuing S/MIME credentials for email by verifying that the party applying for a credential receives mail at the email address in question.

The third enrollment model is delegation: that is, the holder of a certificate (assigned by either of the two methods above) may delegate some or all of their authority to another party. In some cases, multiple levels of delegation could occur: a LEC, for example, might delegate authority to customer organization for a block of 100 numbers, and the organization might in turn delegate authority for a particular number to an individual employee. This is analogous to delegation of organizational identities in traditional hierarchical Public Key Infrastructures (PKIs) who use the name constraints extension [3]; the root CA delegates names in sales to the sales department CA, names in development to the development CA, etc. As lengthy certificate delegation chains are brittle, however, and can cause delays in the verification process, this document considers optimizations to reduce the complexity of verification.

[TBD] Future versions of this specification may address adding a level of assurance indication to certificates to differentiate those enrolled from proof-of-possession versus delegation.

[TBD] Future versions of this specification may also discuss methods of partial delegation, where certificate holders delegate only part of their authority. For example, individual assignees may want to delegate to a service authority for text messages associated with their telephone number, but not for other functions.

3.1. Certificate Scope and Structure

The subjects of telephone number certificates are the administrative entities to whom numbers are assigned or delegated. For example, a LEC might hold a certificate for a range of telephone numbers. [TBD - what if the subject is considered a privacy leak?]

This specification places no limits on the number of telephone numbers that can be associated with any given certificate. Some service providers may be assigned millions of numbers, and may wish to have a single certificate that is capable of signing for any one of those numbers. Others may wish to compartmentalize authority over subsets of the numbers they control.

Moreover, service providers may wish to have multiple certificates with the same scope of authority. For example, a service provider with several regional gateway systems may want each system to be capable of signing for each of their numbers, but not want to have each system share the same private key.

The set of telephone numbers for which a particular certificate is valid is expressed in the certificate through a certificate extension; the certificate's extensibility mechanism is defined in [7] but the telephone number authorization extension is defined in this document.

3.2. Provisioning Private Keying Material

In order for authentication services to sign calls via the procedures described in [15], they must possess a private key corresponding to a certificate with authority over the calling number. This specification does not require that any particular entity sign requests, only that it be an entity with an appropriate private key; the authentication service role may be instantiated by any entity in a SIP network. For a certificate granting authority only over a particular number which has been issued to an end user, for example, an end user device might hold the private key and generate the signature. In the case of a service provider with authority over large blocks of numbers, an intermediary might hold the private key and sign calls.

The specification recommends distribution of private keys through PKCS#8 objects signed by a trusted entity, for example through the CMS package specified in [8].

4. Acquiring Credentials to Verify Signatures

This specification documents multiple ways that a verifier can gain access to the credentials needed to verify a request. As the validity of certificates does not depend on the circumstances of their acquistion, there is no need to standardize any single mechanism for this purpose. All entities that comply with [15] necessarily support SIP, and consequently SIP itself can serve as a way to acquire certificates. This specific does allow delivery through alternate means as well.

The simplest way for a verifier to acquire the certificate needed to verify a signature is for the certificate be conveyed along with the signature itself. In SIP, for example, a certificate could be carried in a multipart MIME body [9], and the URI in the Identity-Info header could specify that body with a CID URI [10]. However, in many environments this is not feasible due to message size restrictions or lack of necessary support for multipart MIME.

Alternatively, the Identity-Info header of a SIP request may contain a URI that the verifier dereferences with a network call. Implementations of this specification are required to support the use of SIP for this function (via the SUBSCRIBE/NOTIFY mechanism), as

well as HTTP, via the Enrollment over Secure Transport mechanisms described in RFC 7030 [11].

A verifier can however have access to a service that grants access to certificates for a particular telephone number. Note however that there may be multiple valid certificates that can sign a call setup request for a telephone number, and that as a consequence, there needs to be some discriminator that the signer uses to identify their credentials. The Identity-Info header itself can serve as such a discriminator.

4.1. Verifying Certificate Scope

The subjects of these certificates are the administrative entities to whom numbers are assigned or delegated. When a verifier is validating a caller's identity, local policy always determines the circumstances under which any particular subject may be trusted, but for the purpose of validating a caller's identity, this certificate extension establishes whether or not a signer is authorized to sign for a particular number.

The telephone number (TN) Authorization List certificate extension is identified by the following object identifier:

```
id-ce-TNAuthList OBJECT IDENTIFIER ::= { TBD }
```

The TN Authorization List certificate extension has the following syntax:

```
TNAuthorizationList ::= SEQUENCE SIZE (1..MAX) OF TNAuthorization

TNAuthorization ::= SEQUENCE SIZE (1..MAX) OF TNEntry

TNEntry ::= CHOICE {
   spid  ServiceProviderIdentifierList,
   range TelephoneNumberRange,
   one   E164Number }

ServiceProviderIdentifierList ::= SEQUENCE SIZE (1..3) OF
             OCTET STRING

  -- When all three are present: SPID, Alt SPID, and Last Alt SPID

TelephoneNumberRange ::= SEQUENCE {
   start E164Number,
   count INTEGER }

E164Number ::= IA5String (SIZE (1..15)) (FROM ("0123456789"))
```

[TBD] Do we really need to do IA5String? The alternative would be UTF8String, e.g.: UTF8String (SIZE (1..15)) (FROM ("0123456789"))

The TN Authorization List certificate extension indicates the authorized phone numbers for the call setup signer. It indicates one or more blocks of telephone number entries that have been authorized for use by the call setup signer. There are three ways to identify the block: 1) a Service Provider Identifier (SPID) can be used to indirectly name all of the telephone numbers associated with that service provider, 2) telephone numbers can be listed in a range, and 3) a single telephone number can be listed.

Note that because large-scale service providers may want to associate many numbers, possibly millions of numbers, with a particular certificate, optimizations are required for those cases to prevent certificate size from becoming unmanageable. In these cases, the TN Authorization List may be given by reference rather than by value, through the presence of a separate certificate extension that permits verifiers to either securely download the list of numbers associated with a certificate, or to verify that a single number is under the authority of this certificate. This optimization will be detailed in future version of this specification.

4.2. Certificate Freshness and Revocation

The problem of certificate freshness gains a new wrinkle in the telephone number context, because verifiers must establish not only that a certificate remains valid, but also that the certificate's scope contains the telephone number that the verifier is validating. Dynamic changes to number assignments can occur due to number portability, for example. So even if a verifier has a valid cached certificate for a telephone number (or a range containing the number), the verifier must determine that the entity that signed is still a proper authority for that number.

To verify the status of the certificate, the verifier needs the certificate, which is included with the call, and they need to:

- Rely on short-lived certificates and not check the certificate's status, or

- Rely on status information from the authority; there are three common mechanisms employed by CAs:

  * Certificate Revocation Lists (CRLs) [7],
  * Online Certificate Status Protocol (OCSP) [RFC6560], and
  * Server-based Certificate Validation Protocol (SCVP) [RFC5055].

The tradeoff between short lived certificates and using status information is the former's burden is on the front end (i.e., enrollment) and the latter's burden is on the back end (i.e., verification). Both impact call setup time, but it is assumed that performing enrollment for each call is more of an impact that using status information. This document therefore recommends relying on status information.

When relying on status information, the verifier needs to obtain the status information but before that can happen the verifier needs to know where to locate it. Placing the location of the status information in the certificate makes the certificate larger but it eases the client workload. The CRL Distribution Point certificate extension includes the location of the CRL and the Authority Information Access certificate extension includes the location of OCSP and/or SCVP servers; both of these extensions are defined in [7]. In all cases, the status information location is provided in the form of an URI.

CRLs are an obviously attractive solution because they are supported by every CA. CRLs have a reputation of being quite large (10s of MBytes) because CAs issue one with all of their revoked certificates but CRLs do support a variety of mechanisms to scope the size of the CRLs based on revocation reasons (e.g., key compromise vs CA compromise), user certificates only, and CA certificates only as well as just operationally deciding to keep the CRLs small. Scoping the CRL though introduces other issues (i.e., does the RP have all of the CRL partitions). CAs in this system will likely all create CRLs for audit purposes but it not recommended that they be relying upon for status information. Instead, one of the two "online" options is recommended. Between the two, OCSP is much more widely deployed and this document therefore recommends the use of OCSP in high-volume environments for validating the freshness of certificates, based on [12]. Note that OCSP responses have three possible values: good, revoked, or unknown.

[TBD] HVE OCSP requires SHA-1 be used as the hash algorithm, we're obviously going to change this to be SHA-256.

[TBD] What would happen in the unknown case?

The wrinkle here is that OCSP only provides status information it does not indicate whether the certificate's is authorized for the telephone number that the verifier is validating. There's two ways to ask the authorization question:

- For this certificate, is the following number currently in its scope of validity?

o What are the numbers associated with this certificate?

The former seems to lend itself to piggybacking on the status mechanism; since the verifier is already asking an authority about the certificate's status why not use that mechanism instead of creating a new service that requires additional round trips. Like most PKIX-developed protocols, OCSP is extensible; OCSP supports request extensions (OCSP supports sending multiple requests at once) and per-request extensions. It seems unlikely that the verifier will be requesting authorization checks on multiple callers in one request so a per-request extension is what is needed. But, support for any particular extension is optional and the HVE OCSP profile [12] prohibits the use of per-request extensions so there is some additional work required to modify existing OCSP responders.

The extension mechanism itself is fairly straightforward and it's based on the X.509 v3 certificate extensions: an OID, a criticality flag, and ASN.1 syntax as defined by the OID. The OID would be registered in the IANA PKIX arc, the criticality would likely be set to critical (i.e., if the OCSP responder doesn't understand the extension stop processing), and the syntax can be anything we desire. Applying the KISS principle, the syntax could simply be the TN being asserted by caller. The responder could then determine whether the TN asserted in the OCSP per-request extension is still authorized for the certificate referred to in the certificate request field; the reference is a tuple of hash algorithm, issuer name hash, issuer key hash, and serial number.

The second option seems more like a query response type of interaction and could be initiated through a URI included in the certificate. Luckily, the AIA extension supports such a mechanism; it's an OID to identify the "access method" and an "access location", which would most most likely be a URI. The verifier would then follow the URI to ascertain whether the list of TNs authorized for use by the caller. There are obviously some privacy considerations with this approach.

The need to check the authorizations in another round-trip is also something to consider because it will add to the call setup time. OCSP implementations commonly pre-generate responses and to speed up HTTPS connections the server provides OCSP responses for each certificate in their hierarchy. If possible, both of these OCSP concepts should be adopted.

Ideally, once a certificate has been acquired by a verifier, some sort of asynchronous mechanism could notify and update the verifier if the scope of the certificate changes. While not all possible categories of verifiers could implement such behavior, some sort of

event-driven notification of certificate status is another potential subject of future work.

5. Acknowledgments

Russ Housley, Brian Rosen, Cullen Jennings and Eric Rescorla provided key input to the discussions leading to this document.

6. IANA Considerations

This memo includes no request to IANA at this time. If we define an OCSP extension or AIA access method then we'll need an OID from the PKIX.

7. Security Considerations

This document is entirely about security. For further information on certificate security and practices, see RFC 3280 [5], in particular its Security Considerations.

8. Informative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Barnes, R., Kent, S., and E. Rescorla, "Further Key Words for Use in RFCs to Indicate Requirement Levels", RFC 6919, April 1 2013.

[3] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[4] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[5] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

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

[7] 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.

[8] Turner, S., "Asymmetric Key Packages", RFC 5958, August 2010.

[9] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

[10] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, August 1998.

[11] Pritikin, M., Yee, P., and D. Harkins, "Enrollment over Secure Transport", RFC 7030, October 2013.

[12] Deacon, A. and R. Hurst, "The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments", RFC 5019, September 2007.

[13] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement and Requirements", draft-ietf-stir-problem-statement-05 (work in progress), May 2014.

[14] Peterson, J., "Retargeting and Security in SIP: A Framework and Requirements", draft-peterson-sipping-retarget-00 (work in progress), February 2005.

[15] Peterson, J., Jennings, C., and E. Rescorla, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-stir-rfc4474bis-02 (work in progress), October 2014.

Authors' Addresses

Jon Peterson
Neustar, Inc.
1800 Sutter St Suite 570
Concord, CA 94520
US

Email: jon.peterson@neustar.biz

Sean Turner
IECA, Inc.
3057 Nutley Street, Suite 106
Farifax, VA 22031
US

Email: turners@ieca.com