Internet Engineering Task Force (IETF)
Request for Comments: 7199
Category: Standards Track
ISSN: 2070-1721

R. Barnes
M. Thomson
Mozilla
J. Winterbottom
Unaffiliated
H. Tschofenig

April 2014

Location Configuration Extensions for Policy Management

Abstract

Current location configuration protocols are capable of provisioning an Internet host with a location URI that refers to the host's location. These protocols lack a mechanism for the target host to inspect or set the privacy rules that are applied to the URIs they distribute. This document extends the current location configuration protocols to provide hosts with a reference to the rules that are applied to a URI so that the host can view or set these rules.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7199.

Table of Contents

1. Introduction

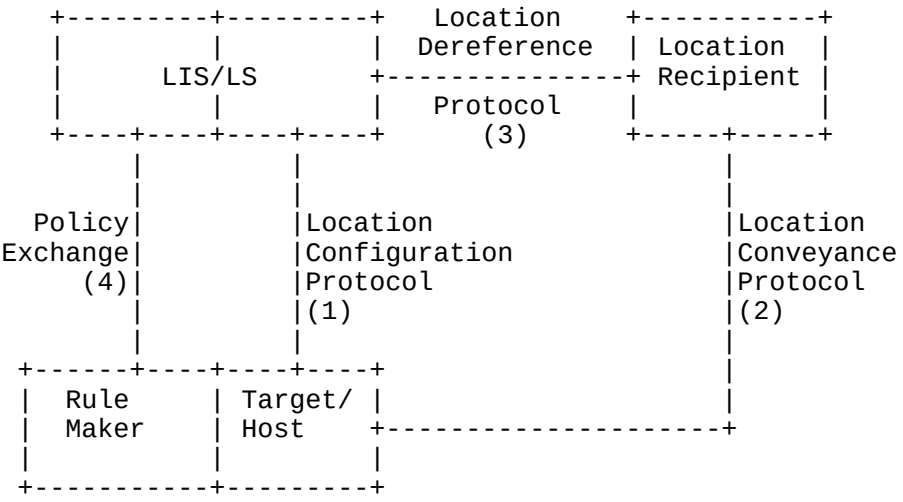
A critical step in enabling Internet hosts to access location-based services is to provision those hosts with information about their own location. This is accomplished via a Location Configuration Protocol (LCP) [RFC5687], which allows a location provider (e.g., a local access network) to inform a host about its location.

There are two basic patterns for location configuration, namely configuration "by value" and "by reference" [RFC5808]. Configuration by value provisions a host directly with its location, by providing it location information that is directly usable (e.g., coordinates or a civic address). Configuration by reference provides a host with a URI that references the host's location, i.e., one that can be dereferenced to obtain the location (by value) of the host.

In some cases, location by reference offers a few benefits over location by value. From a privacy perspective, the required dereference transaction provides a policy enforcement point so that if suitable privacy policies have been provisioned, the opaque location URI can be safely conveyed over untrusted media. (If the location URI is not subject to privacy rules, then conveying the location URI may pose even greater risk than sending location by value [RFC5606].) If the target host is mobile, an application provider can use a single reference to obtain the location of the host multiple times, saving bandwidth to the host. For some configuration protocols, the location object referenced by a location URI provides a much more expressive syntax for location values than the configuration protocol itself (e.g., DHCP geodetic location [RFC6225] versus Geography Markup Language (GML) in a Presence Information Data Format Location Object (PIDF-LO) [RFC4119]).

From a privacy perspective, however, current LCPs are limited in their flexibility, in that they do not provide hosts (the clients in an LCP) with a way to inform the Location Server with policy for how his location information should be handled. This document addresses this gap by defining a simple mechanism for referring to and manipulating policy and by extending current LCPs to carry policy references. Using the mechanisms defined in this document, an LCP server (acting for the Location Server (LS) or Location Information Server (LIS)) can inform a host as to which policy document controls a given location resource, and the host (in its Rule Maker role) can inspect this document and modify it as necessary.

In the following figure, adapted from RFC 5808, this document extends the Location Configuration Protocols (1) and defines a simple protocol for policy exchange (4).



The remainder of this document is structured as follows:

After introducing a few relevant terms, we define policy URIs as a channel for referencing, inspecting, and updating policy documents. We then define an extension to the HELD protocol to allow it to carry policy URIs. Examples are given that demonstrate how policy URIs are carried in this protocol and how it can be used by clients.

2. Definitions

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 RFC 2119 [RFC2119].

3. Policy URIs

A policy URI is an HTTP [RFC2616] or HTTPS [RFC2818] URI that identifies a policy resource that contains the authorization policy for a linked location resource. Access to the location resource is governed by the contents of the authorization policy.

A policy URI identifies an HTTP resource that a Rule Maker can use to inspect and install policy documents that tell a Location Server how it should protect the associated location resource. A policy URI always identifies a resource that can be represented as a common-policy document [RFC4745] (possibly including some extensions; e.g., for geolocation policy [RFC6772]).

Note: RFC 3693 [RFC3693] identified the Rule Holder role as the one that stores policy information. In this document, the Location Server is also a Rule Holder.

3.1. Policy URI Usage

A Location Server that is the authority for policy URIs MUST support GET, PUT, and DELETE requests to these URIs, in order to allow clients to inspect, replace, and delete policy documents. Clients support the three request methods as they desire to perform these operations.

Knowledge of the policy URI can be considered adequate evidence of authorization; a policy URI functions as a shared secret between the client and the server (see Section 7). A Location Server SHOULD allow all requests, but it MAY deny certain requests based on local policy. For instance, a Location Server might allow clients to inspect policy (GET), but not to update it (PUT). Or, a Location Server might require clients to authenticate using HTTP or Transport Layer Security (TLS) client authentication. Clients implementing this specification SHOULD support HTTP client authentication [RFC2617] and MAY support TLS client certificates.

A GET request to a policy URI is a request for the referenced policy information. If the request is authorized, then the Location Server sends an HTTP 200 response containing the complete policy identified by the URI.

A PUT request to a policy URI is a request to replace the current policy. The entity-body of a PUT request includes a complete policy document. When a Location Server receives a PUT request, it MUST validate the policy document included in the body of the request. If the request is valid and authorized, then the Location Server MUST replace the current policy with the policy provided in the request.

A DELETE request to a policy URI is a request to delete the referenced policy document. If the request is authorized, then the Location Server MUST delete the policy referenced by the URI and disallow access to the location URIs it governs until a new policy document has been put in place via a PUT request.

A policy URI is only valid while the corresponding location URI set is valid. A Location Server MUST NOT respond to any requests to a policy URI once the corresponding location URI set has expired. This expiry time is specified by the ’expires’ attribute in the HELD locationResponse.

A location URI can thus become invalid in three ways: By the expiration of a validity interval in policy, by the removal of a policy document with a DELETE request, or by the expiry of the LCP-specified validity interval. The former two are temporary, since the policy URI can be used to update the policy. The latter one is permanent, since the expiry causes the policy URI to be invalidated as well.

The Location Server MUST support policy documents in the common-policy format [RFC4745], as identified by the MIME media type of "application/auth-policy+xml". The common-policy format MUST be provided as the default format in response to GET requests that do not include specific "Accept" headers, but content negotiation MAY be used to allow for other formats.

This usage of HTTP is generally compatible with the use of Extensible Markup Language (XML) Configuration Access Protocol (XCAP) [RFC4825] or Web Distributed Authoring and Versioning (WebDAV) [RFC4918] to manage policy documents, but this document does not define or require the use of these protocols.

3.2. Policy URI Allocation

A Location Server creates a policy URI for a specific location resource at the time that the location resource is created; that is, a policy URI is created at the same time as the location URI that it controls. The URI of the policy resource MUST be different from the location URI.

A policy URI is provided in response to location configuration requests. A policy URI MUST NOT be provided to an entity that is not authorized to view or set policy. This document does not describe how policy might be provided to entities other than for location configuration, for example, in responses to dereferencing requests [RFC6753] or requests from third parties [RFC6155].

Each location URI has either one policy URI or no policy URI. The initial policy that is referenced by a policy URI MUST be identical to the policy that would be applied in the absence of a policy URI. A client that does not support policy URIs can continue to use the location URI as they would have if no policy URI were provided.

For HELD, the client assumes that the default policy grants any requester access to location information, as long as the request possesses the location URI. To ensure that the authorization policy is less permissive, a client updates the policy prior to distributing the location URI.

A Location Server chooses whether or not to provide a policy URI based on local policy. A HELD-specific extension also allows a requester to specifically ask for a policy URI.

A policy URI is effectively a shared secret between the Location Server and its clients. Knowledge of a policy URI is all that is required to perform any operations allowed on the policy. Thus, a policy URI should be constructed so that it is hard to predict and confidentiality protected when transmitted (see Section 7). To avoid reusing these shared secrets, the Location Server MUST generate a new policy URI whenever it generates a new location URI set.

3.3. Policy Defaults

Client implementors should keep in mind that setting no policy (never performing an HTTP request to a policy URI) is very different from setting an empty policy (performing a PUT with the empty policy). By "the empty policy", we mean a policy containing no rules, which would be represented by the following policy document:

```
<?xml version="1.0" encoding="UTF-8"?>
   <ruleset xmlns="urn:ietf:params:xml:ns:common-policy">
   </ruleset>
```

Figure 1: The Empty Policy

If no policy is set, then the client tacitly accepts whatever policy the server applies to location URIs, including a policy that provides location to anyone that makes a dereference request. If the empty policy is set, then the opposite is true; the client directs the server to never provide access to location. (Since there are no rules to allow access and the policy language is default-deny.)

Thus, implementors should consider carefully how to handle the case where the user provides no privacy policy input. On the one hand, an implementation might treat this case as if the user had no privacy preferences and, thus, set no policy. On the other hand, another implementation might decide that if a user provides no positive authorization, then the empty policy should be installed.

The same reasoning could also be applied to servers, with the caveat that servers do not know whether a given HELD client supports the use of policy URIs. A client that does not understand policy URIs will not be able to set its own policy, so the server must choose a default that is open enough that clients will find it useful. On the other hand, once a client indicates that it understands policy URIs (by including a "requestPolicyUri" element in its HELD request), the

server may change its default policy to something more restrictive -- even the empty, default-deny policy -- since the client can specify something more permissive if desired.

4. Location Configuration Extensions

Location configuration protocols can provision hosts with location URIs that refer to the host's location. If the target host is to control policy on these URIs, it needs a way to access the policy that the Location Server uses to guide how it serves location URIs. This section defines extensions to LCPs to carry policy URIs that the target can use to control access to location resources.

4.1. HELD

The HELD protocol [RFC5985] defines a "locationUriSet" element, which contains a set of one or more location URIs that reference the same resource and share a common access control policy. The schema in Figure 2 defines two extension elements for HELD: an empty "requestPolicyUri" element that is added to a location request to indicate that a Device desires that a policy URI be allocated and a "policyUri" element that is included in the location response.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
     targetNamespace="urn:ietf:params:xml:ns:geopriv:held:policy"
     xmlns:xs="http://www.w3.org/2001/XMLSchema"
     xmlns:hp="urn:ietf:params:xml:ns:geopriv:held:policy"
     elementFormDefault="qualified"
     attributeFormDefault="unqualified">

  <xs:element name="requestPolicyUri">
    <xs:complexType name="empty"/>
  </xs:element>

  <xs:element name="policyUri" type="xs:anyURI"/>

</xs:schema>
```

Figure 2: XML Schema for the Policy URI Extension

The URI carried in a "policyUri" element refers to the common access control policy for location URIs in the location response. The URI MUST be a policy URI as described in Section 3. A policy URI MUST use the "http:" or "https:" scheme, and the Location Server MUST support the specified operations on the URI.

A HELD request MAY contain an explicit request for a policy URI. The presence of the "requestPolicyUri" element in a location request indicates that a policy URI is desired.

4.2. Client Processing

It is possible that this document will be updated to allow the use of policy URIs that use protocols other than the HTTP-based protocol described above. To ensure that they fail safely when presented with such a URI, clients implementing this specification MUST verify that a policy URI received from HELD uses either the "http:" or "https:" scheme. If the URI does not match those schemes, then the client MUST discard the URI and behave as if no policy URI was provided.

5. Examples

In this section, we provide some brief illustrations of how policy URIs are delivered to target hosts and used by those hosts to manage policy.

A HELD request that explicitly requests the creation of a policy URI has the following form:

```
<locationRequest xmlns="urn:ietf:params:xml:ns:geopriv:held">
  <locationType exact="true">locationURI</locationType>
  <requestPolicyUri
    xmlns="urn:ietf:params:xml:ns:geopriv:held:policy"/>
</locationRequest>
```

A HELD response providing a single "locationUriSet", containing two URIs under a common policy, would have the following form:

```
<locationResponse xmlns="urn:ietf:params:xml:ns:geopriv:held">
  <locationUriSet expires="2011-01-01T13:00:00.0Z">
    <locationURI>
      https://ls.example.com:9768/357yc6s64ceyoiuy5ax3o
    </locationURI>
    <locationURI>
      sip:9769+357yc6s64ceyoiuy5ax3o@ls.example.com:
    </locationURI>
  </locationUriSet>
  <policyUri xmlns="urn:ietf:params:xml:ns:geopriv:held:policy">
    https://ls.example.com:9768/policy/357lp6f64prlbvhl5nk3b
  </policyUri>
</locationResponse>
```

5.1. Basic Access Control Policy

Consider a client that gets the policy URI <https://ls.example.com:9768/policy/357lp6f64prlbvhl5nk3b>, as in the above LCP example. The first thing this allows the client to do is inspect the default policy that the LS has assigned to this URI:

```
GET /policy/357lp6f64prlbvhl5nk3b HTTP/1.1
Host: ls.example.com:9768


HTTP/1.1 200 OK
Content-type: application/auth-policy+xml
Content-length: 388

<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy"
         xmlns:gp="urn:ietf:params:xml:ns:geolocation-policy">
  <rule id="AA56ia9">
    <conditions>
      <validity>
        <until>2011-01-01T13:00:00.0Z</until>
      </validity>
    </conditions>
    <actions/>
    <transformations>
      <gp:provide-location/>
      <gp:set-retransmission-allowed>
        false
      </gp:set-retransmission-allowed>
      <gp:set-retention-expiry>0</gp:set-retention-expiry>
    </transformations>
  </rule>
</ruleset>
```

This policy allows any requester to obtain location information, as long as they know the location URI. If the user disagrees with this policy, and prefers for example, to only provide location to one friend, at a city level of granularity, then the client can install this policy on the Location Server:

```
PUT /policy/357lp6f64prlbvhl5nk3b HTTP/1.1
Host: ls.example.com:9768
Content-type: application/auth-policy+xml
Content-length: 462

<?xml version="1.0" encoding="UTF-8"?>
 <ruleset xmlns="urn:ietf:params:xml:ns:common-policy"
   xmlns:gp="urn:ietf:params:xml:ns:geolocation-policy"
   xmlns:lp="urn:ietf:params:xml:ns:basic-location-profiles">
  <rule id="f3g44r1">
    <conditions>
      <identity>
        <one id="sip:friend@example.com"/>
      </identity>
      <validity>
        <until>2011-01-01T13:00:00.0Z</until>
      </validity>
    </conditions>
    <actions/>
    <transformations>
      <gp:provide-location
          profile="civic-transformation">
        <lp:provide-civic>city</lp:provide-civic>
      </gp:provide-location>
    </transformations>
  </rule>
</ruleset>


HTTP/1.1 200 OK
```

Finally, after using the URI for a period, the user wishes to permanently invalidate the URI.

```
DELETE /policy/357lp6f64prlbvhl5nk3b HTTP/1.1
Host: ls.example.com:9768


HTTP/1.1 200 OK
```

6. IANA Considerations

This document requires several IANA registrations, detailed below.

6.1. URN Sub-Namespace Registration for urn:ietf:params:xml:ns:geopriv:held:policy

This section registers a new XML namespace, "urn:ietf:params:xml:ns:geopriv:held:policy", per the guidelines in [RFC3688].

URI: urn:ietf:params:xml:ns:geopriv:held:policy

Registrant Contact: IETF, GEOPRIV working group, (geopriv@ietf.org), Richard Barnes (rlb@ipv.sx).

XML:

```
 BEGIN
   <?xml version="1.0"?>
   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
     <head>
       <title>HELD Policy URI Extension</title>
     </head>
     <body>
       <h1>Namespace for HELD Policy URI Extension</h1>
       <h2>urn:ietf:params:xml:ns:geopriv:held:policy</h2>
       <p>See <a href="http://www.rfc-editor.org/rfc/rfc7199.txt">
          RFC 7199</a>.</p>
     </body>
   </html>
 END
```

6.2. XML Schema Registration

This section registers an XML schema as per the guidelines in [RFC3688].

URI: urn:ietf:params:xml:schema:geopriv:held:policy

Registrant Contact: IETF, GEOPRIV working group (geopriv@ietf.org), Richard Barnes (rlb@ipv.sx)

Schema: The XML for this schema can be found in Section 4.1.

7. Security Considerations

There are two main classes of risks associated with access control policy management: The risk of unauthorized grants or denial of access to the protected resource via manipulation of the policy management process, and the risk of disclosure of policy information itself.

Protecting the policy management process from manipulation entails two primary requirements. First, the policy URI has to be faithfully and confidentially transmitted to the client; second, the policy document has to be faithfully and confidentially transmitted to the Location Server. The mechanism also needs to ensure that only authorized entities are able to acquire or alter policy.

7.1. Integrity and Confidentiality for Authorization Policy Data

Each LCP ensures integrity and confidentiality through different means (see [RFC5985]). These measures ensure that a policy URI is conveyed to the client without modification or interception.

In general, the requirements for TLS on policy transactions are the same as for the dereference transactions they set policy for [RFC6753]. To protect the integrity and confidentiality of policy data during management, the Location Server SHOULD provide policy URIs with the "https:" scheme and require the use of HTTP over TLS [RFC2818]. The cipher suites required by TLS [RFC5246] provide both integrity protection and confidentiality. If other means of protection are available, an "http:" URI MAY be used, but location servers SHOULD reject PUT and DELETE requests for policy URIs that use the "http:" URI scheme.

7.2. Access Control for Authorization Policy

Access control for the policy resource is based on knowledge of its URI. The URI of a policy resource operates under the same constraints as a possession model location URI [RFC5808] and is subject to the same constraints:

- Knowledge of a policy URI MUST be restricted to authorized Rule Makers. Confidentiality and integrity protections SHOULD be used when policy URIs are conveyed in a location configuration protocol and in the requests that are used to inspect, change, or delete the policy resource. Note that in some protocols (such as DHCP), these protections may arise from limiting the use of the protocol to the local network thus relying on lower-layer security

mechanisms. When neither application-layer nor network-layer security is provided, location servers MUST reject requests using the PUT and DELETE methods.

- The Location Server MUST ensure that it is not practical for an attacker to guess a policy URI value, even if the attacker has requested many policy URIs from the Location Server over time. The policy URI MUST NOT be derived solely from information that might be public, including the Target identity or any location URI. The addition of 128 bits or more of random entropy is RECOMMENDED to make it infeasible for a third party to guess a policy URI.

- Servers SHOULD apply rate limits in order to make brute-force guessing infeasible. If a server allocates location URIs that include N bits of entropy with a lifetime of T seconds, then the server should limit clients to (2^(N/2))/T queries per second. (The lifetime T of a location URI set is specified by the "expires" attribute in HELD.)

One possible algorithm for generating appropriately unpredictable policy URIs for a location URI set is described in Appendix A.

The goal of the above recommendation on rate limiting is to bound the probability that an attacker can guess a policy URI during its lifetime. If an attacker is limited to (2^(N/2))/T queries per second, then he will be able to make at most 2^(N/2) guesses over the lifetime of the URI. Assuming these guesses are distinct, the probability of the attacker guessing any given URI is (2^(N/2))/(2^N), so the probability of compromise over the T-second lifetime of the URI is at most 2^(-N/2). (Of course, if the attacker guesses the URI after the policy URI has expired, then there is no risk.) With N=128, the probability of compromise is 5.4e-20 under this rate-limiting scheme. Operators should choose values for N so that the corresponding risk of compromise presents an acceptable level of risk.

If M distinct URIs are issued within the same namespace, then the probability of any of the M URIs being compromised is M*2^(N/2). The example algorithm for generating policy URIs (see Appendix A) places them in independent namespaces (i.e., below the corresponding location URIs), so this compounding does not occur.

Note that the chosen entropy level will also affect how quickly legitimate clients can query a given URI, especially for very long-lived URIs. If the default lifetime T is greater than 2^(N/2), then clients will have to wait multiple seconds between queries. Operators should choose entropy and lifetime values that result in

acceptable high maximum query rates and acceptably low probability of compromise. For example, with 32 bits of entropy (much less than recommended above), the one-query-per-second policy URI lifetime is around 18 hours.

7.3. Location URI Allocation

A policy URI enables the authorization by access control lists model [RFC5808] for associated location URIs. Under this model, it might be possible to more widely distribute a location URI, relying on the authorization policy to constrain access to location information.

To allow for wider distribution, authorization by access control lists places additional constraints on the construction of location URIs.

If multiple Targets share a location URI, an unauthorized location recipient that acquires location URIs for the Targets can determine that the Targets are at the same location by comparing location URIs. With shared policy URIs, Targets are able to see and modify authorization policy for other Targets.

To allow for the creation of Target-specific authorization policies that are adequately privacy protected, each location URI and policy URI that is issued to a different Target MUST be different from other location URIs and policy URIs. That is, two clients MUST NOT receive the same location URI or the same policy URI.

In some deployments, it is not always apparent to an LCP server that two clients are different. In particular, where a middlebox [RFC3234] exists, two or more clients might appear as a single client. An example of a deployment scenario of this nature is described in [RFC5687]. An LCP server MUST create a different location URI and policy URI for every request, unless the requests can be reliably identified as being from the same client.

7.4. Policy URI Handling

Although servers may choose to implement access controls on policy URIs, by default, any holder of a policy URI is authorized to access and modify the referenced policy document and, thus, to control access to the associated location resources. Because policy URIs function as shared secrets, clients SHOULD protect them as they would passwords. For example, policy URIs SHOULD NOT be transmitted to other hosts or stored in plaintext.

It should be noted that one of the benefits of the policy URI construct is that in most cases, there is not a policy URI to leave the client device to which it is provided. Without policy URIs, location URIs are subject to a default policy set unilaterally by the server, and location URIs must be conveyed to another entity in order to be useful. With policy URIs, location URIs can have more nuanced access controls, and the shared secret used to authenticate the client (i.e., the policy URI) can simply be stored on the client and used to set the access control policy on the location URI. So while policy URIs do use a default model of authorization by possession, they reduce the overall risk to location privacy posed by leakage of shared secret URIs.

8. Acknowledgements

Thanks to Mary Barnes and Alissa Cooper for providing critical commentary and input on the ideas described in this document. Also, thanks to Ted Hardie and Adam Roach for helping clarify the relationships between policy URIs, policy documents, and location resources. Thanks to Stephen Farrell for a helpful discussion on security and privacy challenges.

9. References

9.1. Normative References

[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.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC4745] Schulzrinne, H., Tschofenig, H., Morris, J., Cuellar, J., Polk, J., and J. Rosenberg, "Common Policy: A Document Format for Expressing Privacy Preferences", RFC 4745, February 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5985] Barnes, M., "HTTP-Enabled Location Delivery (HELD)", RFC 5985, September 2010.

9.2. Informative References

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[RFC3693] Cuellar, J., Morris, J., Mulligan, D., Peterson, J., and J. Polk, "Geopriv Requirements", RFC 3693, February 2004.

[RFC4119] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, December 2005.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[RFC4825] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", RFC 4825, May 2007.

[RFC4918] Dusseault, L., "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", RFC 4918, June 2007.

[RFC5606] Peterson, J., Hardie, T., and J. Morris, "Implications of 'retransmission-allowed' for SIP Location Conveyance", RFC 5606, August 2009.

[RFC5687] Tschofenig, H. and H. Schulzrinne, "GEOPRIV Layer 7 Location Configuration Protocol: Problem Statement and Requirements", RFC 5687, March 2010.

[RFC5808] Marshall, R., "Requirements for a Location-by-Reference Mechanism", RFC 5808, May 2010.

[RFC6155] Winterbottom, J., Thomson, M., Tschofenig, H., and R. Barnes, "Use of Device Identity in HTTP-Enabled Location Delivery (HELD)", RFC 6155, March 2011.

[RFC6225] Polk, J., Linsner, M., Thomson, M., and B. Aboba, "Dynamic Host Configuration Protocol Options for Coordinate-Based Location Configuration Information", RFC 6225, July 2011.

[RFC6753] Winterbottom, J., Tschofenig, H., Schulzrinne, H., and M. Thomson, "A Location Dereference Protocol Using HTTP-Enabled Location Delivery (HELD)", RFC 6753, October 2012.

[RFC6772] Schulzrinne, H., Tschofenig, H., Cuellar, J., Polk, J., Morris, J., and M. Thomson, "Geolocation Policy: A Document Format for Expressing Privacy Preferences for Location Information", RFC 6772, January 2013.

Appendix A. Example Policy URI Generation Algorithm

One possible algorithm for generating appropriately unpredictable policy URIs for a location URI set is as follows:

1. Choose parameters:

   * A cryptographic hash function H, e.g., SHA256

   * A number N of bits of entropy to add, such that N is no more than the length of the output of the hash function

2. On allocation of a location URI, generate a policy URI in the following way:

   1. Generate a random value NONCE at least N/8 bytes long

   2. Compute hash = H(Location-URI-Set || NONCE) using some cryptographic hash function H and some serialization of the location URI set (e.g., the XML from a HELD response)

   3. Form the policy URI by appending the base64url-encoded form of the hash [RFC4648] to one of the location URIs, e.g., as a query parameter: "http://example.com/loc/foo?policy=j3WTGUb3smxcZA6eKIqmqdV3ALE"

Authors' Addresses

Richard Barnes
Mozilla
331 E. Evelyn Ave.
Mountain View, CA 94041
US

EMail: rlb@ipv.sx

Martin Thomson
Mozilla
Suite 300
331 E Evelyn Street
Mountain View, CA 94041
US

EMail: martin.thomson@gmail.com

James Winterbottom
Unaffiliated
AU

EMail: a.james.winterbottom@gmail.com

Hannes Tschofenig
Hall in Tirol 6060
Austria

EMail: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.priv.at