

IETF GEOPRIV Authorization Policies

Hannes Tschofenig*, Alissa Cooper†, Richard Barnes‡, Martin Euchner§

*Nokia Siemens Networks, Email: Hannes.Tschofenig@nsn.com

†Center for Democracy and Technology, Email: acooper@cdt.org

‡BBN Technologies, Email: rbarnes@bbn.com

§Nokia Siemens Networks, Email: Martin.Euchner@nsn.com

Abstract—Location-based services (such as navigation applications, emergency services, and management of equipment in the field) need geographic location information about Internet hosts, their users, and other related entities. These applications need to securely gather and transfer location information from location services, and at the same time protect the privacy of the individuals and other entities being located.

To enable privacy-preserving information exchange on the Internet, underlying communication protocols are needed. The SIP presence architecture can be seen as a prototypical example for real-time communication to the Internet.

This document describes the design decisions on the access control policies that have been developed for the usage with location information that later got extended to other applications.

I. INTRODUCTION

The IETF GEOPRIV working group was set up to develop privacy-preserving mechanisms for the distribution of location information on the Internet. The requirements and the architecture described in RFC 3693 [1] illustrate a protocol independent framework for privacy protection. An update to RFC 3693 is currently being developed [2].

At a high level, the most important role defined in the architecture is the Location Server, which controls access to location information by entities that request access (called Location Recipients). RFC 4745 [3] defines an XML-based access control language that is not tied to a specific application, so that it can later be used in other contexts as well (beyond location distribution). New application usages, however, have their own semantics and often require enhancements to the basic authorization policy framework. Some of them are simple to define and others are more complex. Examples of such application specific enhancements can be found in the policy extensions for geolocation authorization [4] and presence authorization [5].

Most applications today make use of some form of authorization policies so one might infer that a standardized language for these policies would be useful, for a few different reasons. For the GEOPRIV architecture, the need to specify such a policy language was seen as necessary to allow authorization policies to be attached to data (in the form of sticky policies) when it leaves the Location Server towards the Location Recipient. Another reason is the ability to provide a framework for privacy interoperability among independent

software developments, by different vendors, of end device and Location Server implementations.

While the GEOPRIV working group has focused their work on location information, the general policy model also applies to other application environments. Location information is, however, particularly privacy-sensitive and thus serves as a good example to better understand the challenges.

II. THE GEOPRIV PRESENCE ARCHITECTURE

This section explains the history behind the GEOPRIV architecture, with a strong focus on how it uses the presence architecture as a foundation. There are numerous applications on the Internet today that require a subscription/notification architecture that also underlies presence, so presence is an representative example for many other applications.

A presence service is defined in the abstract in RFC 2778 [6]. It allows users of a communications service to monitor one another's availability and disposition in order to make decisions about communicating. Presence information is highly dynamic, and generally characterizes whether a not a user is online or offline, busy or idle, away from communications devices or nearby, and the like.

The Common Profile for Presence (CPP) [7] defines a set of logical operations for delivery of presence information. These primarily consist of subscription operations and notification operations. In a subscription operation, a 'watcher' (which corresponds to the Location Recipient in the Geopriv architecture) requests information about a 'presentity' (Geopriv: Target) from a 'presence server' (Geopriv: Location Server). A subscription typically leads to state being created at the presence server describing what information the watcher requires and how it should be communicated (for example, at what interval). After a watcher subscribes to a presentity, notifications of presence information will be sent to the watcher as the presence information changes. CPP also supports unsubscriptions (terminating the persistent subscription) and fetches (one-time requests for presence information that result in no persistent subscription).

CPP provides a number of attributes of these operations that flesh out the presence system. There is an option for subscriptions to automatically expire if the watcher does not refresh them at user-defined intervals (in order to eliminate stale subscriptions). There are transaction and subscription identifiers used to correlate messages, and a URI scheme ("pres:") is defined to identify watchers and presentities.

The idea of CPP is to have an abstract model that is applicable to multiple presence systems, for example, to use as a partial aid to interoperability between them. The SIMPLE presence system [8] uses CPP as its baseline architecture, and the presence operations in the Extensible Messaging and Presence Protocol (XMPP) have also been mapped to CPP [9]. (Instant messaging features of both have likewise been mapped to the Common Profile for Instant Messaging (CPIM) [10].)

At a high-level, then, the presence architecture is naturally applicable to the problem of delivering Geopriv information. However, the CPP framework is an abstract framework - it does not actually specify a protocol, it specifies a framework and a set of requirements to which presence protocols must conform. Also, CPP does not define how location information (as a form of presence information) to be published to a Location Server, an especially important challenge for location systems.

SIMPLE [8], the application of the Session Initiation Protocol (SIP) to instant messaging and presence, is one protocol that instantiates the CPP framework and extends it in a number of important ways. SIP has native support for subscriptions and notifications (in its events framework [11]) and has added an event package [12] for presence in order to satisfy the requirements of CPP. Other event packages were defined later to allow other, related information to be exchanged. Above and beyond CPP, SIMPLE has done work on a publication method [13] that allows presentities to install presence information on on a server, so that the server can apply access-control policies before sharing presence information with watchers (in the SIMPLE publication architecture, this server is known as a compositor). The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) [14] is the SIMPLE protocol for provisioning access control and authorization policies in to a presence or a location server.

The publish/subscribe architecture of presence servers places difficult requirements on protocol design. The HTTP environment is conceptually similar, with the constraint that a request (which would correspond to a subscription in CPP) is typically not persistent, in that it usually results in an immediate response.

III. AUTHORIZATION POLICIES

An important design decision for the presence architecture for location is that the overall presence communication framework being re-used. SIP defines a clear structure for identities (SIP or PRES URIs) and a distributed identity framework. Without these two mechanisms, an authorization policy design is less useful, particularly when considering identity based authorization. When a request for a presence resource (a presentity) arrives, the identity of the requestor needs to be determined in order to apply identity-based access controls. SIP authentication mechanisms effectively create a global roaming architecture, allowing the resource requestor to belong to a different administrative domain than the entity controlling access to the resource.

The work on authorization policies started with a minimal authorization policy baseline, called Common Policy.

The Common Policy specification, defined in RFC 4745 [3], defines a format for XML documents that contain access control rules. A single rule has three parts: conditions, actions, and transformations. RFC 4745 defines only three condition elements, namely an identity condition, a time-based validity condition, and an abstract "sphere" condition. The identity condition assumes that the requestor is identified by a URI (e.g., 'sip:' or 'tel:' URI) that the presence server can authenticate. The validity condition offers the possibility to limit the lifetime of a specific rule. The sphere element offers an easy way to switch named policies (e.g., 'home' to 'work'). A new conflict resolution mechanism is specified in the document which offers minimal disclosure in case that parts of the authorization policy, for example unknown extensions, are not understood. Individual rules can only lead to more information being disclosed but not less. This is a useful approach when the authorization policies travel with data and different recipient may have implemented different extensions.

The Common Policy mechanism alone, however, is not sufficient to provide a complete authorization policy mechanism for most applications; it needs to be extended to apply to a specific context. To deal with application-specific semantics, further specifications are necessary, such as described with the presence authorization policy or the geolocation authorization policy.

The presence authorization policies specifications provides a detailed description on how to use the identities in SIP with Common Policy rules. It also describes how the sphere element of Common Policy is utilized in the SIP presence context, since the user has to be able to control switching between the different contexts. The largest part of the specification is a set of transformations that allow access to various presence based information elements. A transformation may, for example, define that a certain watcher is allowed to receive a certain data element, such as 'mood', or 'device-info'.

The geolocation authorization policy defines authorization policies with location-based conditions ("If the entity who's location is being requested is within this geographical region then ..."), and transformations that allow the rule change the returned location object. These transformations include the ability to set certain usage policies for the data and to control the granularity of location information being exposed to third parties (e.g., "city level granularity").

To illustrate the standardization needed, consider a hypothetical usage of the Common Policy in the context of an identity management solution like OpenID. When a Relying Party (RP) asks for attributes about a user from the OpenID Provider (OP, identity provider), a user is typically asked for consent about sharing information to that specific Relying Party and the decision is typically stored at the OP to avoid repeating the same question again in a future exchange. Because the OpenID protocol exchange uses HTTP redirects between the OP and the RP, today users provide their authorization decisions via their browser and no further standardization need arrives. However, consider a model where OpenID is used for applications that are browser-based and these authorization policies are instead uploaded to the OP. In that case, the technical work that is necessary requires a

description of the type of identities being used in the HTTP context to allow the OP to make decisions about which RP they provide what information, and the type of information that may be exchanged. The latter case is particularly interesting as the current data model defined in OpenID (with the various attribute exchange related specifications) differs quite a bit to the data model used today in the presence environment (even though the information elements appear similar).

IV. CONCLUSION

The work in the IETF on the location privacy led to the development of a more generic authorization policy mechanism that can be extended by other application specific contexts. The main reason for the work on a standardized policy language was the perceived need for interoperability in this space. In order to accomplish that interoperability, it is necessary not only to define a basic authorization policy framework but also to define how the identities of a specific communication protocols fit into identities used by the authorization policy framework and how the data model used by the communication protocol maps to the conditions of the access control policies. Even more importantly, application-specific extensions need to define the semantics of potential actions and transformations. The downside of this standardization approach is the time it takes to complete the work.

The amount of standardization work required is a good reason for allowing communication service providers to actually develop their proprietary authorization policies as they have better means to innovate. Many of the providers in communication systems are leaning towards a deployment where they offer a specialist user interface, typically controlled via a web browser, to allow the users to control their privacy preferences and the access to their data. With the need to keep this functionality under full control of the provider offering the service for a better user experience the need for standardized solutions is lower.

The other main need for interoperability, namely to allow authorization policies to travel with the data, is still a very novel idea that has to find acceptance in the deployment community. The GEOPRIV architecture defined a lightweight version of these privacy policies that are more deployment friendly than the full-size authorization policies based on Common Policy. The need to share data in a "Web 2.0" environment, e.g., through mash-ups, will increase the need to let usage policies travel with the protected data itself, as already exercised for certain information items (such as documents, music, video, and pictures) in the form of Creative Commons license rules.

While the basic functionality of the authorization policy mechanism defined in the GEOPRIV working group can certainly be mapped to other authorization policy languages, such as XACML¹, the more interesting challenge is the need to define incentives for various parties to actually deploy

standardized solutions (in comparison to their preferred approach) given that new usage scenarios require a certain amount of standardization overhead in order to accomplish interoperability.

V. ACKNOWLEDGEMENTS

We would like to thank several IETF working groups, most notably SIP, SIMPLE, and GEOPRIV, for their work on the technical specifications.

REFERENCES

- [1] J. Cuellar, J. Morris, D. Mulligan, J. Peterson, and J. Polk, "Geopriv Requirements," Oct. 2003, RFC 3693, Request For Comments.
- [2] R. Barnes, M. Lepinski, A. Cooper, J. Morris, H. Tschofenig, and H. Schulzrinne, "An Architecture for Location and Location Privacy in Internet Applications," Oct. 2009, <http://tools.ietf.org/html/draft-ietf-geopriv-arch-01>.
- [3] H. Schulzrinne, H. Tschofenig, J. Morris, J. Cuellar, J. Polk, and J. Rosenberg, "Common Policy: A Document Format for Expressing Privacy Preferences," Feb. 2007, IETF 4745, Request For Comments.
- [4] H. Schulzrinne, H. Tschofenig, J. Morris, J. Cuellar, and J. Polk, "An Architecture for Location and Location Privacy in Internet Applications," Jul. 2009, <http://tools.ietf.org/html/draft-ietf-geopriv-policy-21>.
- [5] J. Rosenberg, "Presence Authorization Rules," Dec. 2007, RFC 5025, Request For Comments.
- [6] M. Day, J. Rosenberg, and H. Sugano, "A Model for Presence and Instant Messaging," Feb. 2000, RFC 2778, Request For Comments.
- [7] J. Peterson, "Common Profile for Presence (CPP)," Aug. 2004, RFC 3859, Request For Comments.
- [8] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," May 2002, RFC 3261, Request For Comments.
- [9] P. Saint-Andre, "Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM)," Oct. 2004, RFC 3922, Request For Comments.
- [10] J. Peterson, "Common Profile for Instant Messaging (CPIM)," Aug. 2004, RFC 3860, Request For Comments.
- [11] A. Roach, "Session Initiation Protocol(SIP)-Specific Event Notification," Jun. 2002, RFC 3265, Request For Comments.
- [12] J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)," Aug. 2004, RFC 3856, Request For Comments.
- [13] A. Niemi, "Session Initiation Protocol (SIP) Extension for Event State Publication," Oct. 2004, RFC 3903, Request For Comments.
- [14] J. Rosenberg, "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)," May 2007, RFC 4825, Request For Comments.
- [15] H. Tschofenig and J. Cuellar, "Geopriv Authorization Policies," Jun. 2003, IETF draft (expired), [draft-tschofenig-geopriv-Authz-policies-00.txt](#).

¹Standardization work would be necessary to map the functionality of the IETF developed authorization protocols to XACML. This includes the semantic of application specific information for conditions and obligations, the new conflict resolution mechanism, the identities used in the SIP or XMPP context, etc. An initial attempt for a XACML profile was done with [15]