Why the end-to-end principle matters for privacy

Richard L. Barnes, BBN Technologies
Cullen Jennings, Cisco Systems

Introduction

It is a basic challenge in the design of location-based services to balance the needs of applications for information about users against the needs of users to maintain control of their private information. The standard APIs that have been developed in this area have endeavored to provide tools to help users and applications manage this balance – indeed, the name of the IETF group that works on geolocation is GEOPRIV, for its dual focus on geolocation and privacy.

Recently, however, some early implementations of commercial systems for enabling location-based services have taken an approach that poses severe risks to user privacy. For example, a company called Feeva has created a product that injects user location information into users' HTTP requests as they transit an IP network, enabling the recipients of these HTTP requests to access detailed information about the user without any interaction from that user. This development is particularly alarming since at least two major router vendors have announced partnerships with Feeva to embed this technology in their routers.

In this paper, we give a brief overview of the Feeva system as we understand it, and discuss its benefits and drawbacks from a functional perspective and a privacy perspective. We then describe an alternative solution that achieves similar functionality with a much better privacy model using open APIs developed by the IETF and W3C.

The Feeva System

Feeva does not publish the details of how its system works, but based on private conversations, we understand that the logical elements and data flow are essentially as shown in Figure 1.

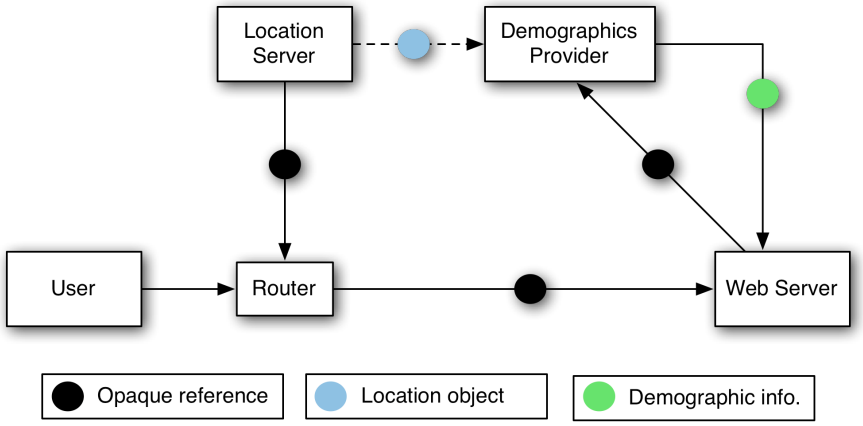

Figure 1: The Feeva system for location-based advertising

In this system, a deep packet inspection (DPI) device is installed on a router in the access network serving a given user. When that device observes a user making an HTTP request, it queries a location server for information about that user's location. The location server provides an opaque reference to the user's location, which the DPI device attaches to the HTTP request in an HTTP header. When the destination web server receives the request, it can use the reference to request information (e.g., demographic information) based on that location from Feeva (who has some way of turning the opaque reference into real location information). The destination web server can then use that information to customize content or provide target advertising. At an economic level, web site operators that use the Feeva headers can pay Feeva for access to demographic data, which allows Feeva to pay location providers (i.e., ISPs) for their subscribers' location information.

As a system to support location-based web services, this model has some appealing characteristics. It does the basic job of getting location-based information to content providers, it provides a model for location providers to be paid for location information, and it requires no modification to client devices or browsers. It also has one feature that seems like a privacy benefit. A single entity never has access to both the user's IP address and the user's location – Feeva only has access to geolocation information, and the web server only has an IP address and demographics.

Looking more deeply at the Feeva model, however, reveals serious privacy risks. While Feeva itself can't associate geolocation-IP bindings, web servers that use Feeva almost certainly can – it is well known that many types of demographic information can be used to "re-identify" other aspects of a user persona (see, e.g., [1][2]), such as geolocation, especially in combination with the myriad other tracking data that modern websites maintain about users. Even if Feeva does not currently provide sufficient data to facilitate re-identification, it is not inconceivable that they would provide more detail if requested by their customers (i.e., web sites).

Furthermore, the Feeva model distributes this private information much more broadly than is necessary. User location information is provided in every single HTTP request that the user makes, regardless of whether or not the server receiving the request has any desire for it. Since the Feeva approach applies only to unencrypted HTTP (as opposed to HTTPS), this means that location information is exposed to any party that has access to the HTTP request in transit, including, for example, transparent proxies introduced by network operators or governments. Any such intermediary can extract the opaque token from the request, use it to obtain demographic information from Feeva, and thus derive further information about the user, including the user's location.

Perhaps the most fundamental flaw in this model, however, is that the user is entirely disintermediated from this privacy-sensitive process. The user has no choice in whether his location information is embedded in every HTTP request he sends; indeed, it is difficult for the user even to *observe* that his information is being exposed in this way.

A Standards-Based Alternative

Modern Internet and web standards for dealing with geolocation can provide a model for location-based web applications that offers the same functionality, with a significantly better privacy model. Consider a system of the form outlined in Figure 2.

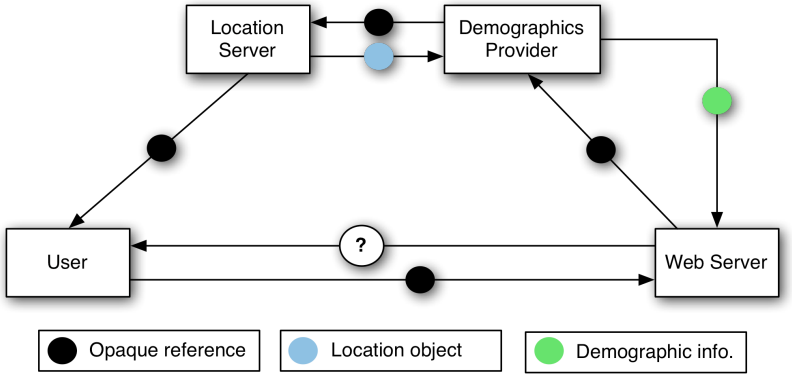

Figure 2: A standards-based model for location-based web applications

In this model, the user's browser receives configuration information that directs it to a local location server, which provides it with an opaque URI referring to the user's location [**Error! Reference source not found.**][**Error! Reference source not found.**][**Error! Reference source not found.**]. Through a standard javascript API (e.g., a simple extension to the W3C geolocation API [6]), the web page requests access to the user's location, allowing the browser to prompt the user for permission to use the location information. If permission is granted, the API provides the web page with the location URI, which it can then send back to the server. The server can then send the URI to the demographics provider, who uses it to obtain location information and provide demographic information.

This model offers several privacy advantages relative to the Feeva model. First and most importantly, it provides the user an explicit role in the provision of location information, making the user aware that location information is being shared and enabling him to allow or deny that sharing. Further, only sites that request location information are even provided with a URI, and that URI can be transmitted back to the server in an encrypted channel – namely, over HTTPS – to protect it from other observers. The ability to use HTTPS means that geolocation information will be protected to the same standard as other secure information on the web. Unlike the Feeva model, this system explicitly provides web sites with geolocation information, but only after the user has consented to this disclosure.

Economically, this system supports the same payment model as the Feeva system. The provider of demographic data is still paid for that data, and the location provider can still limit access to the referenced location to entities that are willing to pay for it (e.g., the demographic provider). The only case where this model would reduce the revenues seen by any party is when the user denies access to location information, but this can hardly be seen as a failure of the model. (There is even a possibility that ads served through this system would be *more* valuable than in the Feeva system, since the user has intentionally expressed interest in receiving location-customized content.)

In addition, the use of open APIs and reference formats creates a more open model, in which, for example, web servers can choose to interact directly with location providers, and demographic providers can dynamically discover new location providers. This openness also allows the model to scale better to include multiple actors at all levels – many location servers, many web sites, and many different types of value-added services based on location.

Practically speaking, only a small amount of development is needed to achieve this model. Browsers would need to be upgraded to discover their local location servers and obtain URIs, and the W3C Geolocation API would need to be extended to provide a location URI. But specification changes cost nothing, and the code to enable location access in browsers is relatively minor; the patch to Firefox that will enable support for the relevant protocol in the next version (without server discovery) involved fewer than 100 lines of code.

Conclusions

It is possible to enable location-based services on the web without compromising the privacy of users' geolocation information, but technologies such as Feeva's are inimical to that goal. A model in which location is injected by the network into HTTP transactions creates an environment in which sensitive information is exposed unnecessarily, to a dangerously broad set of possible recipients. In contrast, existing technology standards and existing browsers, with a few minor modifications, already provide most of the tools for a robust ecosystem of location providers and location-based web applications.

References

1. Malin, B. (2005) "Betrayed By My Shadow: Learning Data Identity via Trail Matching". Journal of Privacy Technology. Avalable at: <http://www.jopt.org/publications/20050609001_malin.pdf>
2. Panopticlick <http://panopticlick.eff.org/>
3. Thomson, M. & Winterbottom, J. (2010) "Discovering the Local Location Information Server (LIS)". Internet-draft. Available at: <http://tools.ietf.org/html/draft-ietf-geopriv-lis-discovery>
4. Barnes, M. et al. (2010) "HTTP-Enabled Location Delivery (HELD)". Internet-draft. Available at: <http://tools.ietf.org/html/draft-ietf-geopriv-http-location-delivery>
5. Marshall, R. (2010) "Requirements for a Location-by-Reference Mechanism". RFC 5808. Available at: <http://tools.ietf.org/html/rfc5808>
6. W3C Geolocation API <http://www.w3.org/TR/2009/WD-geolocation-API-20090707/>

Addendum: Feedback from Feeva and Clarifications

Shortly after this paper was first published, Feeva got in touch with the authors to address a few inaccuracies in our descriptions of their architecture. We would like to correct the following points:

1. The device installed in the provider network is more properly called a "transparent HTTP proxy" than a "DPI device". This proxy receives location information from the ISP's network management infrastructure in the form of mappings between an IP address and an "anonymized token", effectively a random value that Feeva can map back to a location value using information provided off-line. It is this anonymized token that is inserted into HTTP requests. In Figure 1, then, the "Location Server" represents the network management infrastructure, and the "Router" represents the transparent proxy.
2. Feeva does not inject location information into every HTTP request sent by a user. Only those that are destined for web servers belonging to Feeva advertising partners (as determined by the destination IP address and the HTTP Host header field) are tagged with a reference to the user's location information.
3. Feeva is working with their ISP partners to ensure that users have the ability to opt out of Feeva's services, requiring an ISP to have an opt-out mechanism as a condition of their relationship with Feeva. Their current technical approach to managing opt-out is for the ISP to interact with the user (e.g., directing them to a web interface via a post card or interstitial ad), and then inform the Feeva proxy within their network about which users have opted out.

These clarifications do address some of the concerns we raised. The limitation of Feeva interactions to a "white list" of Feeva partners is a significant reduction in the number of HTTP transactions that are affected. Feeva's opt-out program does as good a job at protecting user privacy as is possible within the technical framework of their solution.

Nonetheless, we believe that the broad conclusions of our paper still stand. Even with the restriction to partner sites, Feeva-tagged HTTP requests are still accessible to intermediaries on the wire, exposing the requests to collection by a variety of entities that process HTTP requests, including firewalls and filters, content distribution networks, government monitors, and others. This risk could be mitigated if the information embedded by Feeva encoded the identity of the destination site (in a way that could not be changed); then Feeva would at least be able to tell when an improper recipient was asking for user information.

This last note, however, points to the basic flaw of the Feeva solution, namely that users have to entrust the privacy of their information to Feeva, since they have no way to either control the distribution of their information or even to observe that their information is being distributed (beyond the opt-out notice, which may not be available to some users, e.g., visitors to a residential network). In this light, the restriction of Feeva tagging to partner advertisers makes the situation even worse, since it makes it effectively impossible for users to detect whether they are in a Feeva-enabled network. So we still find that a solution that explicitly intermediates the user would provide better privacy properties, especially given that the basic standards to support such a model already exist.