Proposals for Gathering Consumer Demographics

See also:

- Dave Kristols' ideas in this area

I. The Request-ID: header field:

As originally proposed in 3 Proposals: session ID, business-card auth, customer auth.

Each HTTP request should include a header field of the form:

```
        Request-ID: $session $request++
```

e.g.

```
        Request-ID: 342%33a4d443 12
```

that is, at the beginning of each session, the HTTP client chooses a random number, and each request in that session is identified by a number that increases monotincally with time. A "session" is not formally defined (other than "a set of requests with the same $session id"), though I suggest that browsers begin a session when they are invoked and when they have been idle for 30 mintues or more, and allow some user interface to say "start a new session" (i.e. "choose a new random session ID").

Each user agent must provide a mechanism to turn this feature off, especially for site security administrators that prohibit its use.

Some details: the $request is incremented for each HTTP request, including inline images; it is not reset when requests go to a different server or anything like that. On the other hand, HTTP clients which are not traditional user agents (e.g. multi-threaded robots) may in fact use several sessions in parallel.

A proxy must pass the Request-ID: header through unmodified. One might consider some sort of Proxy-Request-ID, though I doubt it would be valuable.

The response to an HTTP request must _not_ vary as a funciton of the Request-ID. That is, an HTTP proxy need not include the Request-ID in its "cache key." The request ID must _not_ be used to tailer the content of pages on a per-user basis. Use the anon athentication mechanism below.

II. Automated HTML forms through Profiles

In stead of business card auth

An HTML user agent should maintain a profile on behalf of a user. A profile is a set of name-value fields. It should be long-lived. For example, the user agent may allow editing of the profile in a dialog box, and store the profile in the local filesystem. The name of each profile field begins with "profile-". Profile fields should include:

```
        profile-full-name
        profile-first-name
        profile-last-name
        profile-email-address
        profile-home-url
        profile-affiliation
        profile-affiliation-url
        profile-postal-street
        profile-postal-street-2
        profile-postal-city
        profile-postal-state
        profile-postal-zip
        profile-business-phone
```

what else do information providers routinely ask for?

In processing a form, if a form field name matches a profile field name, the initial value of the field should be taken from the value of the profile field.

For example:

```
        <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
        title>Welcome to Guitar-Pick Land!</title>

        <h1>Welcome to Guitar-Pick Land!</h1>

        <p>Howdy partner! Welcome to Guitar-Pick land, the home
        of everthing there is to know about guitar picks.
```

```
<form method=POST action="/cgi-bin/new-user.pl">
<p>Name: <input name=profile-full-name>
<p>E-mail: <input name=profile-email-address>
<p>Would you like to be added to our mailing list? <input type=checkbox name=subscribe>

<p>Would you like to get our catalog? <input type=checkbox name=catalog>
<p>If so, please give us your shipping address:
<pre>
Street:<input name=profile-postal-street>
       <input name=profile-postal-street-2>
City: <input name=profile-postal-city> State: <input name=profile-postal-state> Zip: <input name=profile-postal-zip>
</pre>

<input type=submit> <input type=reset>
</form>
```

Security considerations

If an information provider collects data via HTML forms and keeps it for longer than the lifetime of the HTTP transaction, they should make their policy for treatment of the data available.

User agents must _not_ use profile field values in HIDDEN fields. Any data taken from the profile _must_ be visible to the user at the time of submission.

III. Anonymous Authentication

This is much like the Netscape cookie mechanism, except that we remove the need for clients to maintain a database by weakening it such that the server does not specify the cookie.

To facilitate applications involving per-user state, a new authentication scheme is defined: 'anon'.

To support this authentication scheme, a user agent chooses a 128 bit true-random number, called the client identity, on behalf of the user.

A user agent may support multiple identities per user. On the other hand, a user may want to use the same identity from a desktop and a laptop computer, so there should be some facility to, for example, copy and paste, or perhaps manually transcribe, user identities.

Each participating information provider chooses suitably unique a realm name.

To authenticate a client, an HTTP server responds to a request with, for example:

```
HTTP/1.0 401 Unauthorized
WWW-Authenticate: anon www.guitar-picks.com
```

To authenticate itself, the client sends an Authentication: header in the request; for example:

```
Authorization: anon <credentials>
```

where <credentials> = base64(MD5(realm . ":" . client-identity))

The server can then use the credentials, for example, as a key to its per-user database.

Discussion

Issue I: The desire to model the behaviour of an information provider's consumers, balanced with the privacy rights of those consumers

I believe the Request-ID mechanism addresses this issue. It does assume certain "norms" of behaviour in the public, global information space:

- that service providers have a right to collect information about who is using their service, in order to be able to do security audits, debugging, etc.
- that service providers have an obligation to keep information about individuals using their service confidential. They may release aggregate information, e.g. "12000 folks hit our site today" or "40% of our consumers are using Netscape."
- that consumers have the right to know what information is being gathered about them, especially if this information is kept longer than the lifetime of the transation.

  Consumers should be aware that each request they make is associated with their IP address, and hence possibly the domain they're affiliated with.
- that consumers have the right to refuse to give out identifying information (i.e. enter a bogus email address as an anon-ftp password, or using a firewall to aggregate individual IP addresses into one site proxy)
- that information providers have the right to refuse service to folks that won't identify themselves (e.g. ftp servers that do reverse name lookups on your IP address and check it against your email address, or require the use of IDENT, or ...)
- that producers and consumers should use as little public resources (e.g. internet bandwidth) as possible to achieve their goals. (i.e. don't do stuff that breaks caching unless you have to)

Perhaps these norms should be written up as an RFC in the user services area of the IETF?

Issue II: The desire to build applications with persistent state

A number of interesting applications can be delivered via the web using HTTP and HTMl forms, but they require state that persists between HTTP transactions, associated with a certain user, or associated with a short dialog with a certain user.

Popular implementation techniques include:

- Hidden form fields. This works well if all navigation is done with form submit buttons. But as soon as the user crosses a normal hypertext link, the state may be lost
- Munging URLs. A per-user identifer can be inserted in the URLs. This works correctly with caching proxies, but it has the unfortunate side effect that if the URL is placed in a hotlist or passed between users, it becomes invalid.
- The Netscape Cookie mechanism. This has the unfortunate requirement that the client maintain a database of realm->cookie mappings. The consumer must have access to this database whenever they use the service. It's also not clear how this mechanism interacts with caching proxies.

The anonymous authentication mechanism addresses this need without any of the unfortunate side-effects above. The credentials can be used as an anonymous user ID, or as a dialog ID.

It's possible that the anon authentication scheme is a special case of the simple-MD5 mechanism. For exmaple, the client can use the identity as both the username and password. The server need never know either the username nor password: they just use the credentials as the dialog ID.

Issue III: The desire to unobtrusively gather demographical information about consumers and to build user profiles, balanced with the privacy rights of consumers

The automated forms should reduce the burden of consumers to fill out basic name and address forms. Yet no information is given out without explicit approval of the user (clicking the "submit" button with the information being given away visible.)

The anon authentication should provide an "identity" in any requests that need it. This identity is not transferrable between realms (i.e. Wired and Pathfinder can't collude), and it is anonymous: the server never learns the client identity, let alone name, address, etc., unless the user explictly submits this information.

Issue IV: The effect of caching proxies on the demograhpics gathered at a site

To complete the picture, we really need some way for caching proxies to relay these demographics to origin servers. One possible mechanism is for original servers to include in their response:

```
        Log-To: mailto:logs@here.com
```

A caching proxy would keep track of cach hits on that resource, and report them in buld via email in this case, though HTTP posting is another possible mechanism.

The privacy considerations are tricky. A considerable amount of sensitive data could be gathered very quicly by a malicious third party.

Another compilcating factor is that a simple proxy may not even log the data that the original service provider is interested in.

Ideally, we could agree on some aggregate statistics that all proxies could gather and relay in batch. Unfortunately, I expect that the set of data that's "interesting" varies widely between providers.

Related Work

Net tracking standards slow to emerge

from c|net:

> Net tracking standards slow to emerge, experts say SAN FRANCISCO--At the Internet Tracking Roundtable discussion sponsored by c|net: the computer network here Friday, an industry group composed of Internet content providers, media planners, advertising agency representatives, and researchers agreed on one point: no one Internet tracking standard is likely to emerge in the near future.
>
> The discussion centered on which tracking criteria are needed to make the Net more hospitable to advertisers. Panel members included representatives from Neilsen, I/Pro, J. Walter Thompson, HotWired, NetCount, and c|net.
>
> In general, content providers argued that the uniqueness of the Web demands custom tracking standards that take into consideration current technology constraints.
>
> One hurdle to accurate tracking is the lack of a common vernacular to describe Web tracking criteria, said c|net CEO Halsey Minor. For example, "hits," "visits," and "pages accessed" all mean different things to different people, he said.
>
> Some representatives from the advertising side of the fence said the key is to use the same kinds of tracking devices already used in broadcast and print.
>
> ...

Daniel W. Connolly
$Id: Proposals.html,v 1.1 1998/10/22 20:05:00 renaudb Exp $
Webmaster
Created October 1995
Last updated 06 Nov 1995