"What's Related?" Everything But Your Privacy

Matt Curtin* cmcurtin@interhack.net
Gary Ellison gfe@interhack.net
Doug Monroe monwel@interhack.net

Date: 1998/10/07 12:43:29
Revision: 1.5
http://www.interhack.net/pubs/whatsrelated/

Abstract

Netscape Communications Corporation's release of *Communicator* 4.06[1] contains a new feature, "Smart Browsing", controlled by a new icon labeled *What's Related*, a front-end to a service that will recommend sites that are related to the document the user is currently viewing. The implementation of this feature raises a number of potentially serious privacy concerns, which we have examined here.

Specifically, URLs that are visited while a user browses the web are reported back to a server at Netscape. The logs of this data, when used in conjunction with cookies, could be used to build extensive dossiers of individual web users, even including their names, addresses, and telephone numbers in some cases.

Keywords: Privacy, world-wide web (WWW), Netscape, Alexa, smart browsing, what's related.

1 Introduction

The Internet has often been called the world's largest library—with all of the books on the floor. While recent advances such as web-based directories like *Yahoo!* and smart search engines have helped make navigation of the Internet easier, it is clear that there is still a great deal of room for improvement.

Currently, a user searching for information about a specific product, service, or organization is likely to get a great deal of irrelevant information included with the relevant. This has a range of consequences, from mildly annoying the user to making the Internet nearly impossible to use for research on a specific item.

Enter the notion of "smart browsing". Netscape has teamed with Alexa Internet in order to offer users of Netscape's browser software the ability to use the Alexa service, as a built-in part of the browser. The Alexa service is intended to help users find information that is relevant to them by asking their browser *what's related?*

A user clicking the *What's related?* button in *Communicator* 4.06 will be presented with a number of sites that are intended to be related to the web document he's viewing.

(It is worth noting that Alexa has a client of its own that is similar in functionality, and problems. We're focusing on Netscape's implementation of the technology because of its inclusion with the standard browser, the fact that it is turned on by default, and that it wasn't until after the first publication of this report that we were able to find *any* documentation on this feature.)

*Author's address: The Ohio State University, Department of Computer and Information Science, 791 Dreese Laboratories, 2015 Neil Ave, Columbus, OH 43210.

[1]This document applies also to *Communicator* 4.5, which is in beta now.

2 Anatomy of "Smart Browsing"

Of course, *how* this sort of thing is implemented is of great interest to those involved with Internet architecture. Our findings indicate that a much more broad spectrum of users should be interested. What appears to be an interesting and useful feature comes at a significant price.

Communicator 4.06 offers a number of options for "Smart Browsing" configuration. These are to load *What's Related?* automatically:

- "Always";
- "After First Use";
- "Never".

When *What's Related?* loads, we found that in addition to the normal requests, an additional HTTP session was started with the host `www-rl4.netscape.com`[2], which we'll refer to as "our shadow" for the remainder of this document. This continues as the user bounces from site to site, leaving an electronic trail of our activity on the web with a centralized server. We examine the conversation between the browser and this host for the remainder of this session.

By running a network "sniffer" and examining HTTP proxy logs, we were able to capture all of the data between the browser and "our shadow".

2.1 Current URL

The URL of the page that the user is currently viewing is sent in the query string of an HTTP GET request. Specifically, when viewing `http://www.example.com/`[3], we find that the browser sends the following to "our shadow":

```
GET /wtgn?www.example.com/ HTTP/1.0
```

After performing a variety of requests, we have the following observations:

- URLs are reported back to "our shadow". This includes both "public" URLs and "private", i.e., those that are on an Intranet, unless that URL is part of a group that has been explicitly excluded by the user by browser configuration.
- HTTP query strings are *not* included on the URL that is sent to "our shadow". Specifically, the URL `http://www.example.com/search.cgi?secret` will be reported as `http://www.example.com/search.cgi?`.

2.2 Response

In answer to our query, "our shadow" returns a file of the MIME type `text/rdf`. This is a basic HTML/XML-style markup file containing a series of links that the server believes to be relevant to the URL sent in the request.

There isn't anything especially peculiar about this file, except that all of its links are in the form of

```
http://info.netscape.com/fwd/rl/http://www.example.com:80/
```

This means that rather than being linked directly to the recommended site, the user will be make the connection by first telling "our shadow" where we're going. This is the feedback mechanism which tells the server which, if any, of the recommended sites we've followed.

All of this business of watching everyone and deciding who like to visit what kinds of sites is especially interesting in the context of having software recommend various sites. Section A.3, "Choosing a Recommended Site", shows the actual site "our shadow" recommended to us as relevant to `http://www.example.com/`.

[2] Other hosts were also involved, but it appears that these are simply redundant servers, sharing what is no doubt a very heavy load.

[3] `example.com` is a special domain reserved by the Internet domain name registry, suitable for publication and use in documentation without fear of who might operate the domain in the future. Specifically, it's been reserved, and cannot be registered. We'll use this domain throughout this document, and actually *did* use this domain in some of our tests, with extremely interesting results.

2.3 The Cookie

Perhaps the most interesting, and the most alarming of the headers in the fetch to "our shadow" is this:

```
Cookie: NETSCAPE_ID=10010014,12f8fee8
```

After exiting the browser, we examined the `.netscape/cookies` file to determine if this cookie is persistent across sessions. Interestingly, the file had not been updated in several days. It was then that we discovered that the cookie the browser was sending is the *same* cookie that is sent when any Netscape site requests it. Netcenter, Netscape developers' site, downloads, etc.

2.4 Frequency of the Fetch

Communicator does appear to obey the user's configuration of the option. After testing, we were able to determine that the "our shadow" fetches only happen after the user pushes the button. Afterward, the "our shadow" fetch will happen for the next 1,000 request the user makes when "always" is selected, on the current page and next three pages when "After First Use" is selected, and only on the current page when "Never" is selected.

3 Musings

This feature raises some extremely serious privacy concerns, not only for individuals, but organizations that might have "sensitive" information leaked outside of the boundaries of their firewalls.

Here we'll consider some of the implications of our observations.

3.1 Leaking Intellectual Property Beyond the Firewall

Having an extremely descriptive URL like `http://products.example.com/secret/foobar` or `http://products.example.com/team/some_guy/`, the names of unannounced products, the people working on them, and potentially other information can be leaked. Something along these lines makes an excellent find before attempting a little social engineering to further compromise an organization's intellectual property.

We were, in fact, able to find a particular organization's *internal* sites included in the "our shadow" database. Not only did the "smart browsing" relate this organization's internal URLs, but also included information from the HTML header, specifically the title of the document.

In all fairness, this isn't the only case of URL-leaking on the web, and probably isn't the most problematic. The HTTP `Referer` header is more dangerous, as it leaks the *entire* URL, including any query string data. Poorly implemented systems that pass private data in the query string will expose their users to many sorts of privacy invasions and security risks. This is commonly used as an attack against web-based mail readers, sometimes allowing those running a web site linked to in a piece of email to read the entire mailbox of the user following the link.

The danger here is that rather than having a few "juicy bits" spread randomly throughout the Internet, there is now a single place that could be theoretically used to find more information about a site's internal hosts and URLs. Mining these databases for clues about a site's internals might very well prove to be an effective method of gathering information needed to break into a given site.

It is also noteworthy that, like HTTP `Referer` headers, URLs behind authentication schemes will be reported. However, their authentication credentials are not. Thus, to date, the only leak comes from the URL itself and its title.

The blurring line between "intranet" and "internet" is worthy of further consideration, but goes beyond the scope of this report.

3.2 Extremely Detailed Click-Trails

By collecting detailed browsing data, marketers can classify an individual user and direct advertising content explicitly for that user, based on the site currently being browsed, as well as historical data collected.

3.3 Building a Dossier

Part of the way that privacy concerns with cookies on the web were addressed was by their decentralized nature. Specifically, the domain for which cookies are active are limited. Those sites inside of three-letter top-level-domains (i.e., `com`) have to have at least two level-separators (i.e., dots), and those inside of two-letter top-level-domains (i.e., `us`) have to have at least three level-separators to be valid. This prevents, for example, a cookie from being valid within a domain like `com`, which would be accessible to a wide range of sites managed by different organizations.

By forcing the level of granularity on a cookie's domain, the user has the ability to give certain information to a vendor he might trust more without having to worry about that being stored in a cookie that could then be used by a different vendor, one that the user trusts less.

By sending a stream of URLs back to "our shadow", each of which is accompanied by the same persistent cookie, it now becomes possible for Netscape to completely circumvent the privacy designs of cookies, collecting a rather complete picture of an individual user's browsing habits across the web.

Remember that the cookie being passed for each of these requests is the *same* cookie used for visits to all Netscape web sites, including browser downloads. Now, not only is there now potential to associate all of these web-browsing patterns and sites with a specific user, but these can also be associated with all of the requests to any Netscape pages the user might make.

3.4 Adding Your Name to the Dossier

In order to download Netscape products whose security is limited to domestic US use, the user must provide his name, address, and telephone number, and there's now the potential for Netscape to associate a detailed browsing history with a specific individual.

This can certainly become the most complete database of web users and their browsing habits in very short order, and likely completely without the knowledge of the users involved.

Marketers and totalitarians must drool at this sort of potential.

4 Remedies

Problems that we've identified can be succinctly summarized as:

- Leaking proprietary information through overdescriptive URLs.
- Providing the means for a central repository of a huge number of users' browsing habits, on an extremely granular level.
- Allowing the aforementioned repository the ability to identify individual users with a relatively high degree of certainty.

There are a number of steps that can be taken in order to neutralize the privacy-invading effects of the "smart browsing" feature.

4.1 Filtering Excessively Descriptive URLs

This is most dangerous to organizations with an "intranet", that is, a private part of the web that might contain information that it deems proprietary.

It has been said before, but it's worth repeating: *URLs should not themselves include proprietary information.* Due to such things as the HTTP `Referer` header, and now "smart browsing", it's safe to assume that, at some point, your "private" or "internal" URLs will be seen by third parties.

This becomes a much more real threat as one considers the increasingly available option of corporate espionage.

Organizations with concerns about this can address this problem by having their gateways filter out the HTTP `Referer` header, either to eliminate sites that appear to be internal, or by eliminating the header altogether.[4]

Unlike HTTP `Referer` headers, the passing of the URL is not an optional part of the system in order to maintain functionality. The passing of the URL is necessary in order for the server to report what other URLs are related to the current one. We recognize the difficulty of doing this in a way that does not compromise user privacy, and suspect that this can only be handled by the use of third parties, such as those described in section 4.3, "Anonymizing Proxies".

4.2 Cookie Refusal

Refusing cookies can help prevent the accurate building of dossiers on visitors, but it cannot completely stop it. In the place of cookies can come secondary indications that a user visiting now is the same user who visited yesterday, such as the user's ISP or company, browser and operating system versions, etc. These mechanisms, though, are much less effective than use of cookies.

4.3 Anonymizing Proxies

Services and products such as Anonymizer[5], Lucent Personalized Web Assistant[6][1], and Crowds[7] seem the most effective defenses. Corporate firewalls and web proxies can also provide similar sorts of protection.

Features such as filtering cookies and hiding the request's origin aren't themselves effective against the potential privacy violations. However, used in combination, it appears that one could use the "smart browsing" features of *Communicator* without compromising his privacy.

5 A Word About Intent

We want to stress that we aren't accusing anyone of malice. Both Netscape, the implementor of the technology, and Alexa, the provider of the technology, have reasonable privacy statements on their web sites. And we have absolutely no indication that the data being sent to the "our shadow" server is recorded, or even logged, in any way. However, we do find it more than a little bit disturbing that we found no documentation about the "smart browsing" feature on the Netscape web site as of the first release of this document, and there's no mention of how it is implemented anywhere, even in the READMEs included in the product distribution.

(Since initially releasing this report, we've learned that a file of answers to Frequently Asked Questions now exists on the Netscape web site[3] at `http://home.netscape.com/escapes/related/faq.html`. However, the FAQ fails to paint a complete picture by making statements that are technically correct, but fail to address the real question. Specifically, the FAQ addresses privacy concerns thusly:

> No personal information about you is gathered when you use What's Related. Only the URL you are viewing and your current web address (it changes every time you connect) is sent to the Netscape system so that it can send you a list of related sites.

This conveniently does not mention the fact that the *What's Related?* request includes a cookie which would allow that user to be identified by name if he's ever downloaded a secure version of any of Netscape's software.)

The best-intended systems can sometimes have undesirable consequences. For example, if Netscape were to be purchased by a larger organization that does not respect its customers' privacy, the data that Netscape has collected would then be in "their" hands. Imagine detailed dossiers, including the names of the users, of web users around the world being sold to marketers. Or, perhaps significant changes in Netscape's fortunes

[4] Interestingly, the HTTP/1.1 protocol specification strongly recommends that clients have the ability to decide whether to send this header at all.[2]

[5] `http://www.anonymizer.com/`

[6] `http://lpwa.com/`

[7] `http://www.research.att.com/projects/crowds/`

will cause it to reconsider its stand on what information it will sell to third parties, if someone is offering enough money for the data, and will guarantee deniability.

However unlikely, either of these scenarios is within the realm of possibility. Legally, there would be no recourse for the people whose dossiers have been included, as the legalese of the Netscape site explicitly states that the terms of use (where the privacy statement can be found) are subject to change without notice.

A huge number of other possibilities also exist. One obvious possibility is to have a computer cracker break into the site where the personal data is stored, copy it, and offer it on a sort of "black market", all without the knowledge of Netscape. Perhaps another undesirable scenario is for an individual or group of dossiers to be subpoenaed by a court that deems the data relevant.

Rather than rhetoric about privacy, we would prefer to see new products and services that instead *build in privacy and security by design.* Once data has been given to someone, it cannot effectively be taken back. Rhetoric can change from day to day, but the infrastructure of a worldwide network, and applications running on millions of desktops cannot. Building applications that add functionality at the price of privacy—especially when this is done surreptitiously—is a bad idea at the very least, and potentially irresponsible or dangerous.

A Complete Log Data

Here we include a single transaction, in its entirety.

A.1 The Request

The request made to "our shadow":

```
GET /wtgn?www.example.com/ HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.06 [en] (X11; I; SunOS 5.6 sun4u)
Host: www-rl.netscape.com
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
Cookie: NETSCAPE_ID=10010014,12f8fee8
```

A.2 The Reply

"Our shadow" replied thusly:

```
HTTP/1.0 200 OK
Content-type: text/rdf; charset=utf-8
Connection: Keep-Alive
Content-length: 00459

<RDF:RDF>
<RelatedLinks>
<aboutPage href="http://info.netscape.com/fwd/rl/http://www.example.com:80/"/>
<child instanceOf="Separator1"/>
<child href="http://info.netscape.com/fwd/rl/http://www.a.com/"
  name="The Alternative Japan Web Page! For Adults Over Only Please!"/>
<child instanceOf="Separator1"/>
</RelatedLinks>
</RDF:RDF>
```

A.3 Choosing a Recommended Site

A user who has the `http://www.example.com/` site recommended will make the following request:

```
GET http://info.netscape.com/fwd/rl/http://www.example.com:80/ HTTP/1.0
```

And will receive the following answer:

```
HTTP/1.0 302 NSAPI REDIRECTOR: INVALID URL
Server: Netscape-Enterprise/2.01
Date: Wed, 26 Aug 1998 04:27:47 GMT
Location: http://www.example.com:80/

<HTML><HEAD><TITLE>NSAPI REDIRECTOR: INVALID URL</TITLE></HEAD>
  <BODY><H1>NSAPI REDIRECTOR: INVALID URL</H1>
This document has moved to a new <a href="URL UNKNOWN">location</a>.
Please update your documents and hotlists accordingly.</BODY></HTML>
```

References

[1] E. Gabber, P. Gibbons, Y. Matias, and A. Mayer. *How to Make Personalized Web Browsing Simple, Secure, and Anonymous.* Proceedings of Financial Cryptography 97, February, 1997, Springer-Verlag, LNCS 1318.

[2] R. Fielding, et al. 1997. *Hypertext Transfer Protocol – HTTP/1.1* [online]. Internet Engineering Task Force (IETF) RFC 2068. Available from World Wide Web: `http://www.cis.ohio-state.edu/htbin/rfc/rfc2068.html`.

[3] Netscape Communications Corporation. 1998. *What's Related FAQ* [online]. Available from World Wide Web: `http://home.netscape.com/escapes/related/faq.html`.