

Security Problems Caused by Aggregate Data of Client/Server Connections on the Internet

by:

Peter V. Radatti
CyberSoft, Inc
1508 Butler Pike
Conshohocken, PA 19428 USA

radatti@cyber.com URL: www.cyber.com

Copyright © September - 1998 by Peter V. Radatti, All Rights Reserved.

Introduction

This paper discusses security problems involving web server logs and Internet cookies, which are two of the primary ways that data is collected about Internet users. Its purpose is to explain (1) the kinds of data that are collected; (2) how that data can be abused; (3) what Internet users can do to protect themselves.

Web Server Logs

What kinds of data do web server logs collect?

Most people do not realize that when they are browsing on the Internet, they are creating complex bi-directional client server relationships between their individual workstation, their corporate identity, their Internet service provider (ISP), and the remote server that they are connecting to. In these relationships, the remote server identifies who it is operated by with its content, in addition to whatever information is contained in the InterNIC database about the domain and whatever information is in the Internet protocol (IP) registry. It is not required that the server owner provide accurate information about any of these things. This means that it is easy to create remote servers whose ownership is easily hidden. At the same time, most client software, such as Microsoft Internet Explorer and Netscape Communicator, reveal a good deal about the end user, even if the end user specifically configured the product to reveal incorrect or minimal information.

When users connect to a web server, they immediately reveal a good deal about themselves. Some of the information they may reveal is:

1. What uniform resource locator (URL) they were coming from.
2. The URL(s) at the site they visited and how long they visited it.
3. What operating system they are using and consequently what type of computer.
4. What web browser they are using.
5. The IP or domain address of their workstation.
6. What query they used in a search engine to locate the URL.
7. Internet cookies.

8. What day and time they accessed the **site**.
9. Configuration information about their web browser and **system**.

Most of these are standard log items from web servers. The careful use of some data reduction and sorting software will reveal a wealth of **information** from just these logs. Using some simple techniques, it is also possible to learn the user's email address, name, company name and other **information**.

Annotated Web Server Log Examples

Below is an actual example of some log entries from the `http://www.cyber.com` web server, which is running the Apache web server in a normal configuration. Information in the example has been changed to hide the identity of the **users**. **This** is from a live site with over 60,000 hits per week.

Example One:

```
customer.somedomain.net.il - - [18/Nov/1998:15:27:20 -0500] "GET /images2/new_title.gif HTTP/1.0" 200 10365 "http://www.cyber.com/" "Mozilla/4.05 [en] (WinNT; I) via NetCache version NetApp Release 3.2.1R1: Fri Aug 7 11:52:45 PDT 1998"
```

The host address of the user's computer: `customer.somedomain.net.il`

The date and time that the user visited: `[18/Nov/1998:15:27:20 -0500]`

The part of the site the user was looking at: `"GET /images2/new_title.gif HTTP/1.0" 200 10365 "http://www.cyber.com/"`

This user was running Netscape Navigator 4.05(en) on Windows NT and was running an http caching program called NetCache: `"Mozilla/4.05 [en] (WinNT; I) via NetCache version NetApp Release 3.2.1R1: Fri Aug 7 11:52:45 PDT 1998"`

Example Two:

```
cache1.otherdomain.net - - [18/Nov/1998:15:39:33 -0500] "GET /images2/new_title.gif HTTP/1.1" 200 10365 "http://www.cyber.com/" "Mozilla/4.0 (compatible; MSIE 4.01; Windows 95)"
```

The host address of the user's computer: `cache1.otherdomain.net`

The date and time that the user visited: `[18/Nov/1998:15:39:33 -0500]`

The part of the site the user was looking at: `"GET /images2/new_title.gif HTTP/1.1" 200 10365 "http://www.cyber.com/"`

This user was running Microsoft Internet Explorer 4.01 on Windows 95: `"Mozilla/4.0 (compatible; MSIE 4.01; Windows 95)"`

Example Three:

```
cache-dc03.proxy.aol.com - - [18/Nov/1998:15:48:40 -0500] "GET /personal/pete/specs.html HTTP/1.0" 200 1888 "http://netfind.aol.com/search.gw?search=www.cyber.com&lk=excite_netfind2_us&nrm=n&pri=on&xls=b&xll=40&test=Find%21" "Mozilla/4.0 (compatible; MSIE 4.01; AOL 4.0; Windows 98)"
```

Where the request came from: `cache-dc03.proxy.aol.com`

The date that this data was requested: [18/Nov/1998:15:48:40 -0500]

This user was reading **the** page from a search engine: "GET /personal/pete/specs.html HTTP/1.0" 200 1888 "http://netfind.aol.com/search.gw?search=www.cyber.com&lk=excite_netfind2_us&nrm=n&pri=on&xls=b&xll=40&test=Find%21"

This user has a Windows 98 machine and accesses the Internet via America Online version 4.0 (which comes with Microsoft Internet Explorer 4.01 as the default web browser): "Mozilla/4.0 (compatible; MSIE 4.01; AOL 4.0; Windows 98)"

How can the data from web server logs be abused?

There are several ways that data from web server logs can be collected for the specific purpose of being abused. One method of collecting intelligence on the Internet that is well focused and targeted is to create **honey pot web servers**. Just as honey attracts bees, **honey pot web servers** contain selected information that will attract the target audience. For example, if **someone were** interested in learning **which** companies are doing research on recombinant molecular DNA, **and they** expect that this type of information will generally be a closely held corporate secret, a very good method of accomplishing this task is to create a **honey pot web server** on that topic. Interestingly, not only will **they** be able to tell which companies are visiting **their** web site, but **they** should **also** be able to learn who within that company is doing the research. In addition, by careful analysis of the server logs **they** may be able to learn which specific subsets of information the **visitor** is interested in and thereby reveal specifically what area of research and development the visitor is working in. This is a valuable piece of information that may help the server owner steer their research into the same areas as their competitors.

A secondary benefit of controlling a **honey pot web server** on a specific topic is that **the server owner** may be able to solicit technical white papers on the focus topic. As the **system administrator** for the site, **they** could have up to **several** weeks in which to use the paper for **their** own purposes prior to the actual posting. This is in addition to controlling a valuable topic site on the Internet. This control allows **the server owner** to decide what to post on the site. If the site becomes trusted, then its value as a source of disinformation and the resulting wasted resources of competitors becomes priceless.

Finally, whoever controls the talent in a given field controls the field. **The people whom everyone most wants** to hire are **usually** already working, **so** locating them **can** be difficult. This is not true for the **honey pot web site** owner. They **can** gather information on many potential employees, including their email **addresses**. Once they have the email address, they may be able to use standard Internet search engines or even finger processes to locate the person's full name, phone number, and potentially their home address.

Of course, in order to facilitate this **the owner of a honey pot web server** would need to hide **their** identity. **There** are many Internet **service providers** who will host a web site and hide **the owner's** identity as a byproduct of the process.

What can Internet users do to protect their data from web server logs?

How can you fight **web server logs and honey pot web servers**? **There** are several ways. **Using** a proxy-based firewall that is configured to hide the actual user and reject specific types of connections (such as finger daemon requests) can go a long way alone. **Another thing** that can be done is to **ensure that** users do not configure their web browser in such a way as to reveal who they are. If you are only browsing, there is no reason to configure the email part of the web browser, or you can configure it in such a way as to provide false information. **The** standard Unix sendmail aliases file can also be used to create false email addresses that can still accept and forward email. This is valuable because the false email address will not be associated with the person or company using it. An example of how this may be done is if company A

controls the Internet domain names A.com and B.com. They can have their web server configured to use A.com while all of their browsers could be configured to reveal B.com. The email address xyzzzy@b.com will not reveal that the actual user is radatti@a.com and since the person xyzzzy does not exist, no amount of Internet or phone book searches will reveal any information.

Internet Cookies

What kinds of data do Internet cookies collect?

Internet cookies as a computer technology sound safe, slightly boring, and maybe even tasty. This paper will attempt to demonstrate that Internet cookies are actually mud pies. In old cowboy movies, there was always a scene where cows were branded. An Internet cookie does the same thing, but it is you who is being branded. If you are using Netscape, the browser arrives on your computer with a default of accepting cookies silently. You never even know that someone just smoked your hide. As a matter of good security policy, I turned silent acceptance of cookies off. There is no option to turn off acceptance completely in Netscape, so every time a cookie request is made to my system, a pop-up message window appears. The message window gives me the option of accepting the cookie and being branded or canceling the cookie. Since most people don't know what a cookie is, don't understand that there are any security issues in accepting them, and may, in fact, be afraid of breaking something by pressing the button labeled "cancel," I assume that most people accept cookies. In fact, they would never even know that they were being "cookied" unless they stumbled upon the button that disables automatic acceptance.

At this point in my paper, I feel safe in the given assumption that most people are accepting cookies. So what is the big danger? The military knows. As long as there has been warfare, militaries have been concerned by something called aggregate data. Aggregate data may be as simple as counting the number of cars that enter the gates at a military reserve. If someone counted the number of cars entering a few dozen reserves across the country over a period of time, then anyone who had access to the data from all of the reserves could, in fact, predict a major military engagement about to start. If the number of cars entering all of the reserves had a sudden jump across the country, and the people who entered didn't leave, they were about to go somewhere else en masse.

The same type of analysis can be done with your movements. There are now large networks of Internet cookie data collection companies. They keep track of where you are, where you came from, where you went, and the kind of computer, browser, and operating system you are using. In fact, they can also get your IP address, system name, and, if configured, your name, company name, and email address. That is a lot of information, but it is not the end. At some point, you will come across a form or you will order something over the Internet. When you do, your real name, home address, telephone number, credit card number, and anything else you tell them about yourself is now available to connect with your cookie. The interesting thing is that if the company kept all your old cookie information, then they could track your past, present, and future. This could be dangerous if you accidentally end up at an embarrassing web site.

Why do companies use Internet cookies to collect data?

So why does anyone try to brand you with cookies? The reason is simple: effective advertising. I feel that advertising is a useful thing since it helps me find things that I want to buy. The problem is that a billboard doesn't know who is looking at it, but a computer does. If I were a member of a vegetarian household and I suddenly started receiving email, banner advertisements, postal mail, and phone calls from meat producers, that could be a real problem. At sometime in the past, I might have bought a book from an on-line bookstore. I already had a cookie, so a relationship now exists between myself as a person and my cookie. The cookie is issued every time I enter one of the cookie networks and they target advertising to me based upon my movements.

How can the data from Internet cookies be abused?

I turned cookies on for a while and started looking for travel information at the Alta Vista search engine. They are part of a cookie-gathering network. The web site devoted to the Dilbert cartoon strip **and** many other sites **are also part of these cookie-gathering networks**. As soon as I did my first search on "airfare and Boston," I was presented with advertisements for travel agents. When I traveled to other cookie-affiliated sites, I received more travel-related advertisements. This is fine, but think about the implications.

If I browsed several financial oriented sites, I might start receiving unsolicited and unwelcome attention from sleazy **stockbrokers**. If I searched for medical information, I **wouldn't** want anyone to know what my problems are. It is none of their business. If my doctor or **stockbroker** shared that type of information about me, I would have them in front of their respective state boards for unsavory behavior. The fact of the matter is that a cookie tracker could learn my medical problems, hobbies, financial interests, and a whole lot more depending upon what I did on the Internet. This is an invasion of privacy, but legal.

Cookies can open the door to other security problems. Covert communications channels can continue to provide intelligence gathering long after a specific client/server connection has been broken. There now exists Internet web server based Trojan horses that can access private information stored in your computer. One of the most **well-known**, but still new, attacks is known as the Cache Cow Trojan. The Cache Cow program allowed web servers to extract a client's cookies. This could have dangerous results, since some financial institutions use cookies as part of the authentication process. Potentially, a **web server that** downloaded cookies for an individual's stock trading account could then attempt to pose as the legitimate account owner and move stock to their own accounts. The Cache Cow problem was corrected in Netscape 4.07.

The Son of Cache Cow is a set of Trojan horse programs that continues to work in all versions of Netscape, including version 4.07. These programs are called Cookie Monster, File List, and Cache Cow 4.07. The Cookie Monster program steals cookies from arbitrary locations. The File List program steals the contents of local directories. I tried **pointing** this program to my home directory and then pointing it to the root directory of my Unix workstation. It correctly showed me a complete file list! The **Cache Cow 4.07** program will steal the contents of the browser cache. It has the same effect as the original **Cache Cow** program.

What can Internet users do to protect their data from Internet cookies?

One possible solution is to shut off automatic silent acceptance of cookies and just press the cancel button. It appears that the cookie **networks** already thought of that. They have gotten pushy and rude. There are now many sites that enforce cookie branding by plastering you with dozens of cookie requests per page. Some of them plastered me with so many cookie requests per page that I lost count after 20. The message windows appear faster than I can cancel them, get in the way of what I wanted to do, and waste my time. How rude! Department stores don't keep me out just because I refuse their "free" credit card and gift at the door. I don't mind one cookie request because I have the option of saying no, but dozens of "requests" feels like getting mugged.

So how can you deal with cookies? Actually it's easy. Turn on silent acceptance of cookies. Enter the ".netscape" directory and delete the file named "COOKIE." There are all kinds of dire warnings not to edit or delete the file, but I do it anyway. Unfortunately, Netscape keeps recreating the cookie file, and I have to keep deleting it. On the UNIX computer that I use to browse the web, I could put the "rm/export/home/radatti /.netscape/COOKIE" in my ".login" and ".logout" files. **This command would tell the computer to delete the cookie file every time I log on or off**, but I found a better way. From your home directory, enter the ".netscape" directory. Remove the "COOKIE" file and put in a logical line to

“/dev/null” (In -s /dev/null COOKIE). This command instructs the operating system to delete new cookies as soon as they are received. As a result, each successive cookie request produces a response that the user has never before visited the site. I no longer get bothered with pop-up windows and I clog the cookie networks with hundreds of fake identities per day. In fact, the cookie trackers must think that 80 different people visit each page without finishing to download the page.

Cookies are only one way for people to gather aggregate data on you when using the Internet. In addition, cookies are not restricted to Netscape; Microsoft Explorer also processes cookies. Finally, your Internet service provider can gather all of this information and a great deal more about you. It's a dangerous world.

References:

“Cookie Monster, The Risks of Internet Cookies and Aggregate Data” by Peter V. Radatti, January 1998

Web Server Log extracted from <http://www.cyber.com>

MSNBC Web Site Information on Cache Cow

Dan Brumleve's Web Site on Son of Cache Cow

This document is Copyright© by Peter V. Radatti, February 2000. All Rights Reserved. CYBER.COM™, VFIND™ and AVATAR™ are registered trademarks of CyberSoft, Inc. CIT™, THD™, UAD™, MVFILTER™, JDIS™, ROBOTMODE™, NTI™, NTI-CRYPTO™ and RMI™ are trademarks of CyberSoft, Inc. Documentum™ is a registered trademark of Documentum, Inc. All other trademarks and copyrights are the property of their respective holders.

Source: <http://www.cyber.com/whitepapers/aggregate.html>