

The Wayback Machine - https://web.archive.org/web/20100824203358/http://en.wikipedia.org:80/wiki/Web_beacon

Web bug

From Wikipedia, the free encyclopedia
(Redirected from Web beacon)

A **web bug** is an object that is embedded in a web page or e-mail and is usually invisible to the user but allows checking that a user has viewed the page or e-mail.^[1] One common use is in e-mail tracking. Alternative names are **web beacon**, **tracking bug**, **tracking pixel**, **pixel tag**, **1×1 gif**, and **clear gif**.^[2]

Contents

- 1 Overview
- 2 Implementation
- 3 E-mail web bugs
- 4 See also
- 5 References
- 6 External links

Overview

A web bug is any one of a number of techniques used to track who is reading a web page or e-mail, when, and from what computer. They can also be used to see if an e-mail was read or forwarded to someone else, or if a web page was copied to another website. The first web bugs were small images.

Some e-mails and web pages are not wholly self-contained. They may refer to content on another server, rather than including the content directly. When an e-mail client or web browser prepares such an e-mail or web page for display, it ordinarily sends a request to the server to send the additional content.

These requests typically include the IP address of the requesting computer, the time the content was requested, the type of web browser that made the request, and the existence of cookies previously set by that server. The server can store all of this information, and associate it with a unique tracking token attached to the content request.

Web bugs are typically used by third parties to monitor the activity of customers at a site.^[3] Turning off a browser's cookies can prevent some web bugs from tracking a customer's specific activity. The web site logs will still record a page request from the customer's IP address, but unique information associated with a cookie cannot be recorded. However, web site server techniques that do not use cookies can be employed to help track a site's cookie-blocking users. For example, a web site can identify a request from a new visitor and send that visitor links that pass a unique ID as a GET parameter.

As an example of the way web bugs can make user logging easier, consider a company that owns a network of sites. This company may have a network that requires all images to be stored on one host computer while the pages themselves are stored elsewhere. They could use web bugs in order to count and recognize users traveling around the different servers on the network. Rather than gathering statistics and managing cookies on all their servers separately, they can use web bugs to keep them all together.

For e-mail, many web bugs can be avoided by turning off HTML display and displaying only the text. Turning off the display of images while still using HTML may still allow other techniques to be used.

Web bugs are frequently used in spamming (sending unsolicited commercial e-mail) as a way of "pinging" to find which spam recipients open (and presumably read) before deleting it.

Implementation

Originally, a web bug was a small (usually 1×1 pixel) transparent GIF or PNG image (or an image of the same color of the background) that was embedded in an HTML page, usually a page on the web or the content of an e-mail. Modern web bugs also use the HTML IFrame, style, script, input link, embed, object, and other tags to track usage.^[4] Whenever the user opens the page with a graphical browser or e-mail reader, the image or other information is downloaded. This download requires the browser to request the image from the server storing it, allowing the server to take notice of the download. As a result, the organization running the server is informed when the HTML page has been viewed.

While web bugs are used in the same way in web pages or e-mails, they have different purposes:

1. If the bug is embedded in an e-mail, the image is requested when the user reads the e-mail for the first time, and can also be requested every time that the user subsequently loads the e-mail;
2. Whenever a web page (with or without bugs) is downloaded, the server holding the page knows and can store the IP address of the computer requesting the page; this information can therefore be retrieved from the server log files without the need of using bugs. Bugs are used when monitoring has to be done by a server that is different from the one holding the web pages; this is necessary, for example, when the web pages are served by different servers, or when the monitoring has to be done by a third party.

As with all files transferred using the Hypertext Transfer Protocol, web bugs are requested by sending the server their URL, and possibly the URL of the page containing them. Both URLs contain information that can be useful for the server:

1. The URL of the page containing the bug allows the server to determine which particular web page the user has accessed;
2. The URL of the bug can be appended with an arbitrary string in various ways while still identifying the same object; this extra information can be used to better identify the conditions under which the bug has been loaded; this extra information can be added while sending the page or by JavaScript scripts after the download.

For example, an e-mail sent to the address `somebody@example.org` can contain the embedded image of URL `http://example.com/bug.gif?somebody@example.org`. Whenever the user reads the e-mail, the image at this URL is requested. The part of the URL after the question mark is ignored by the server for the purpose of determining which file to send, but the complete URL is stored in the server's log file. As a result, the file `bug.gif` is sent and shown in the e-mail reader; at the same time, the server stores the fact that the particular e-mail sent to `somebody@example.org` has been read. Using this system, a spammer or e-mail marketer can send similar e-mails to a large number of addresses to check which ones are valid and read by the users.

Web bugs can be used in combination with HTTP cookies like any other object transferred using the HTTP protocol.

E-mail web bugs

Web bugs embedded in e-mails have greater privacy implications than bugs embedded in web pages. Through the use of unique identifiers contained in the URL of the web bugs, the sender of an e-mail containing a web bug is able to record the exact time that a message was read, as well as the as well as the IP address of the computer used to read the mail or the proxy server that the user went through. In this way, the sender can gather detailed information about when and where each particular recipient reads e-mail. Every subsequent time the e-mail message is displayed can also send information back to the sender.

Web bugs are used by e-mail marketers, spammers, and phishers to verify that e-mail addresses are valid, that the content of e-mails has made it past the spam filters, and that the e-mail is actually viewed by users. When the user reads the e-mail, the e-mail client requests the image, letting the sender know that the e-mail address is valid and that e-mail was viewed. The e-mail need not contain an advertisement or anything else related to the commercial activity of the sender. This makes detection of such e-mails harder for mail filters and users.^[5]

Tracking via web bugs can be prevented by using e-mail clients that do not download images whose URLs are embedded in HTML e-mails. Many graphical e-mail clients can be configured to avoid accessing remote images. Examples include the Gmail, Yahoo!, and SpamCop/Horde webmail clients; Mozilla Thunderbird, Opera, Pegasus Mail, IncrediMail, later versions of Microsoft Outlook, and KMail mail readers. Other HTML techniques (such as IFrames) can still be used to track e-mail viewing.

Text-based mail readers (such as Pine or Mutt) and graphical e-mail clients with purely text-based HTML capabilities (such as Mulberry) do not interpret HTML or display images, so their users are not subject to tracking by e-mail web bugs. Plain-text e-mail messages cannot contain web bugs because their contents are interpreted as display characters instead of embedded HTML code, so opening messages does not initiate communication. Some e-mail clients offer the option to disable all HTML in every message (thus rendering all messages as plain text), which prevents any web bugs from loading.

Many modern e-mail readers and web-based e-mail services will not load images when opening an HTML e-mail from an unknown sender or that is suspected to be spam mail. The user must explicitly choose to load images. Web bugs can also be filtered out at the server level so that they never reach the end user. MailScanner is an example of gateway software that can disarm IFrames as well as web bugs. Momentarily disabling a computer's Internet connection before reading new emails and deleting those messages suspicious of containing web bugs may also eliminate the threat.

A hosts file can be used to specify that some servers are never to be contacted for any reason. This file must be continually updated to reflect the fact that new tracking servers are periodically brought online, and old ones repurposed to serve legitimate content.

As a result of these measures, web bugs are slowly losing their effectiveness and cannot be relied on to accurately count read rates for e-mail campaigns.

Disposition-Notification-To email headers may be seen as another form of web bug. See RFC 4021.

See also

- Facebook Beacon
- E-mail fraud
- Internet privacy

References

1. ^ <http://news.cnet.com/2100-1017-243077.html>

2. ^ http://w2.eff.org/Privacy/Marketing/web_bug.html
3. ^ <http://www.mailsbroadcast.com/email.bolts.nuts/about.web.bugs.htm>
4. ^ Freedom to Tinker » Blog Archive » Email Tracking: It Gets Worse (<https://web.archive.org/web/20100824203358/http://www.freedom-to-tinker.com/?p=610>)
5. ^ <http://www.zdnet.com/blog/btl/have-you-received-any-traceable-pattymail-recently/3671>

External links

- The Web Bug FAQ (https://web.archive.org/web/20100824203358/http://www.eff.org/Privacy/Marketing/web_bug.html) from EFF
- Did they read it? (<https://web.archive.org/web/20100824203358/http://lwn.net/Articles/86893/>) from the Linux Weekly News
- Trojan Marketing (<https://web.archive.org/web/20100824203358/http://bcn.boulder.co.us/~neal/privacy/trojanmarketing.html>)
- Slashdot on Web Bugs (<https://web.archive.org/web/20100824203358/http://yro.slashdot.org/yro/06/10/13/1623217.shtml>) - Slashdot.org Forum on Blocking Web Bugs
- "Have you received any “traceable” PattyMail recently?" (<https://web.archive.org/web/20100824203358/http://blogs.zdnet.com/BTL/?p=3671>) - David Berlind, ZDNET.

Retrieved from "http://en.wikipedia.org/wiki/Web_bug"

Categories: Internet privacy | E-mail | Spamming

- This page was last modified on 7 July 2010 at 08:53.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.
Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.