Hal Berghel

ARE YOUR WEB BUGS NO-SEE-UMS?

Earlier in the week I was asked to comment on a complaint that a member of a non-for-profit professional society made. It seems that this society places web bugs in email that it sends to its members for tracking purposes. Email tracking takes its place alongside cookies and click-stream analysis in the collection of information about users. Email tracking is somewhat more alarming because it's more "invasive" - i.e. it is embedded in email and silently activates on opening. Although the courts have ruled that there is no expectation of privacy with respect to corporate email (vs. telephony) from your employer, this doesn't apply to third-party marketers and spammers so email tracking is on untested legal footing.

In any event, the incident with the professional society revealed a wide variety of opinions. As a privacy zealot, mine was near the conservative end of the spectrum (i.e., "thou shall not covet thy user's environment variables"). The other end was represented by the "everybody does it, so what's the problem" camp. The resulting dialog was heated enough that I thought that a discussion might be interesting to G&L readers.

Web Bugs

Web bugs are routinely used by marketers, political groups, investigative agencies, businesses, spammers, and phishers - virtually everyone who wants to snoop on computer users "in situ," as we say in the trade. Web bugs are euphemistically called "web beacons," "pixel tags," "invisible gif," "clear tags," and "pixilated anchors" to conceal their real purpose: to spy on users. Usually, justifications for their use run something like this: "In order to improve our quality of service so that we may best fit your needs, XYZ Corp may collect information about your use of its website, yadda, yadda." There is a reason why euphemisms are used to describe web bugs, tracking cookies, and other such snoopware: users don't want them, won't accept them, and resent companies who deploy them. So companies involved with digital snooping attempt concealment. The professional society mentioned above got "outed" by one of its members who objected to paying $250 in annual dues to have the opportunity to be spied on.

Web bugs are almost as old as the Web itself. To the source (marketer, spammer, phisher, e_crook), a Web bug is akin to an ICMP "ping" in networking - it identifies the target as alive. Web bugs are a result of some security shortcomings in HTML - the page language of the Web. Mention of HTML in this context is important, because all web bugs in the purest sense are embedded in HTML. The script fragment below is an example of a web bug that is embedded in a prominent phone company homepage.

```
<iframe
src="https://fls.doubleclick.net/activityi;src
=1475931;type=corpo676;cat=veriz532;ord
=0123456789?" width="1" height="1"
frameborder="0"></iframe>
```

In this case, prominent phone co. homepage is using a single pixilated anchor with a width and height of 1 pixel (the smallest dot that can be rendered on the display - read that as "invisible") to establish a secure connection with doubleclick.net to relay information about the user's use of the verizon.com website. Here's another example of a web bug - this time with documentation embedded in the page source:

```
<!-- "Network Pixel" c/o "Omniture", segment: 'Omniture Retargeting Segment' -
DO NOT MODIFY THIS PIXEL IN ANY
WAY -->
<img src="http://segment-pixel.invitemedia.com/pixel?pixelID=2710&partnerID=75&key=segment" width="1" height="1" />
            <!-- End of pixel tag -->
```

In both cases, the fact that the text "element" is a 1x1 pixel that links to an external URL betrays the tracking mechanism. It should be noted here that using a 1x1 pixel might be used for page alignment/formatting, so that alone does not signify a web bug. However, single pixels that link to an external URL are another matter altogether.

These examples illustrate how HTML may be used/abused (depending on one's point of view) to capture information about user behavior. Email web bugs use the same HTML tricks, by embedding HTML in the email itself. When you receive email with colored banners, links, and the like, you're receiving an email with HTML embedded. Modern mailers render HTML in the same way as web browsers like Firefox and Internet Explorer. So when you open the email, the tracking script is activated and the server begins to collect data just as if you had you connected to the URL in a web browser. Note in this regard, that the email doesn't have to look like a typical web page to contain a web bug - they can be embedded in plain text as well. If you have cookies enabled, the cookie contents are fair game as well. There's virtually no limit to the variety of information that can be collected on users, including IP addresses, time/date stamps, email address of user/victim, proxy servers in use, and so forth.

If you think that email web bugs aren't being abused, think again. As we now know, they were used in the HP pretexting scandal to spy on board members. They are also used to harvest account information from web mail users. As I write this column the BBC just announced that they discovered several lists of names and passwords for over 30,000 individuals that were derived from embedded HTML in web mail clients supported by Yahoo, AOL and Google posted on the Internet (see. news.bbc.co.uk/2/hi/technology/8292928.stm).

Defensive Measures

There are some things that we can do to protect ourselves from web bugs. But these defensive measures are not without cost in usability. As with all security, there is the inevitable trade-off. For simplicity, I'll break my analysis into two different but related themes: browsers and email clients. In both cases the goal for the privacy advocate is to limit the effect of the offending HTML. We do this in two ways. First, we download as little HTML as we can. Second, we make sure that we don't store any more information than necessary so that there's not much information available to the web bugs.

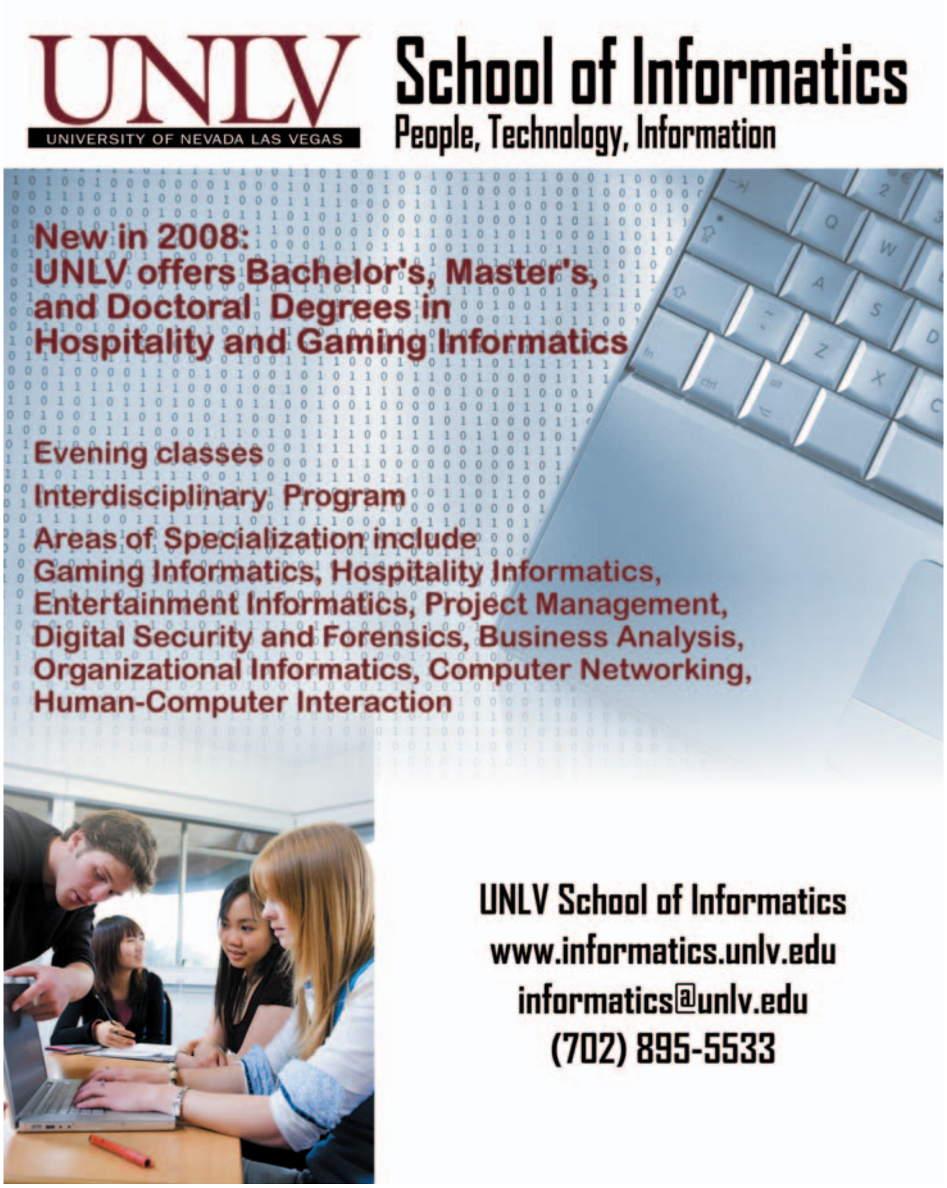
I begin with a caveat: proceed at your own risk. Neither I nor G&L makes any warranties, expressed or implied, about the suggestions to follow. All that I claim is that I have found these configurations useful in my office environment.

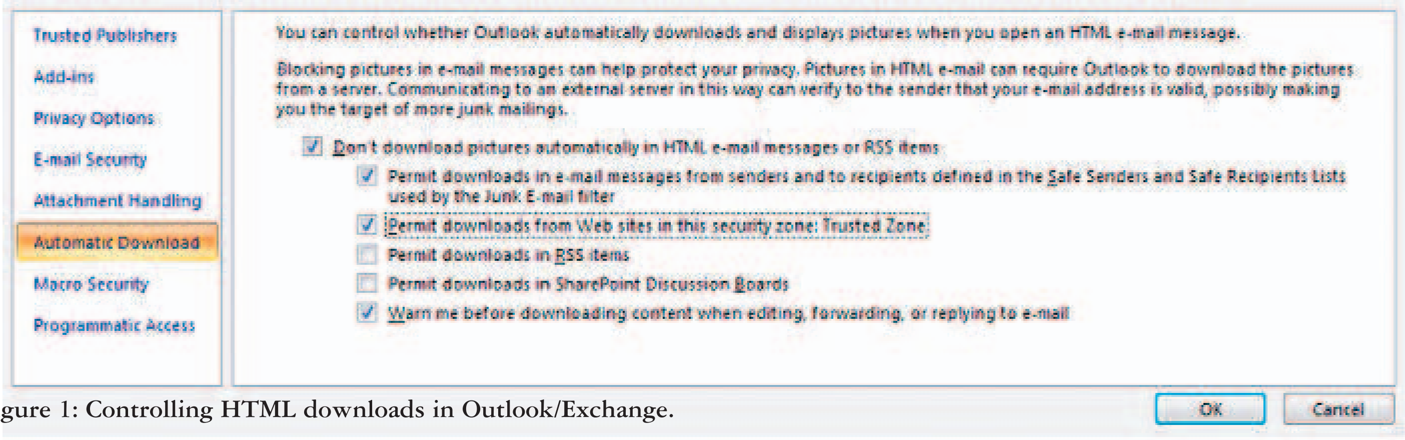
We start with browsers. With Firefox 3.5, the following menu path:

Tools>Options> click on privacy tab> click on "show cookies" provides a perspective on the grist for the web bugs' mill. If you are adventurous and want to reduce exposure, click on "remove all cookies" and watch them disappear. Preventing their reappearance, however, isn't quite as simple because blocking all cookies turns your browser into a screen saver. So here's a halfway measure that works for me. I check "accept cookies from sites" but do not click on "accept third-party cookies." I then select "Keep until ... I close Firefox". Also check "Clear History when Firefox closes." This will reduce the information deposited on your computer by websites and minimize the information available to web bugs and other cookie monsters. Next, I go to

Tools>Options> click on security tab and check "Warn me when sites try to install add-ons," check both of the "blocks", and uncheck the password retention commands. At this point, I'm flying as incognito as I can while still retaining a usable browser. If I experience problems with my web usage, I simply reverse the process to the point where I get the browser behavior I want.

Internet Explorer 8 is the first browser that Microsoft produced with the privacy and security of the user in mind. To tweak for protection, go to the menu bar and follow Tools>Internet Options>Internet. Under the security tab, I set the security level to medium-high at a minimum and click on "enable protected mode." You can tweak the custom level settings if you wish to spend some time researching the options. I set Local intranet to medium and also click the "enable protected mode" box. Trusted sites are just what the name implies, and if you are meticulous about what you trust, default settings should be acceptable. Restricted sites should have the highest





You can control whether Outlook automatically downloads and displays pictures when you open an HTML e-mail message.

Blocking pictures in e-mail messages can help protect your privacy. Pictures in HTML e-mail can require Outlook to download the pictures from a server. Communicating to an external server in this way can verify to the sender that your e-mail address is valid, possibly making you the target of more junk mailings.

- ☑ Don't download pictures automatically in HTML e-mail messages or RSS items
  - ☑ Permit downloads in e-mail messages from senders and to recipients defined in the Safe Senders and Safe Recipients Lists used by the Junk E-mail filter
  - ☑ Permit downloads from Web sites in this security zone: Trusted Zone
  - ☐ Permit downloads in RSS items
  - ☐ Permit downloads in SharePoint Discussion Boards
  - ☑ Warn me before downloading content when editing, forwarding, or replying to e-mail

Figure 1: Controlling HTML downloads in Outlook/Exchange.

security level setting and protected mode enabled at the very least.

I set the IE8 privacy setting to high, enable the pop-up blocker, and check both "InPrivate" settings. Note that IE8 introduced "InPrivate" browsing (Tools>InPrivate Browsing) which I strongly recommend. There is a similar setting in Firefox. These browsing environments radically reduce the amount of information about you that is cast out on the wire while web surfing.

Email is handled slightly differently and will require coordination with the local system administrator because any changes to the configuration may affect the integrity of your email environment. With Outlook/Exchange embedded HTML is controlled through the trust center (Tools>Trust>Automatic Download) as in Figure 1.

Figure 1: Controlling HTML downloads in Outlook/Exchange.

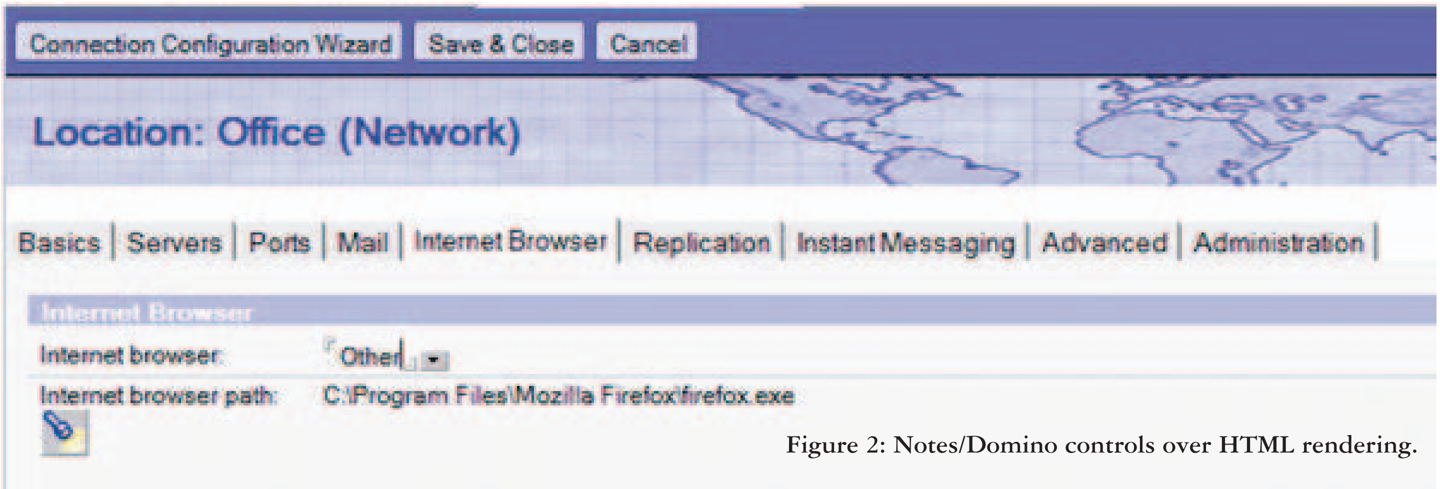
Note that I have chosen not do download HTML graphics in my email except when the source is trusted. Since the HTML isn't downloaded, neither are the web bugs in the HTML. The only downside is that I lose all of that delightful screen gumbo in my email!

Notes/Domino environments also add another option. As Figure 2 shows, Notes 7 allows the user to render HTML outside of the mailer where more control over HTML is available. In this case, we choose to set up our mailer so that it renders HTML in an already hardened version of Firefox by accessing File>Preferences>Location Preferences>Internet Browser Tab.

Figure 2: Notes/Domino controls over HTML rendering.
In either case, the effect of web bugs is minimized or eliminated.

So if you're a privacy zealot, the discussion above gives you the range of options available to prevent web bugs and other types of HTML snoopware from "bugging" you.

Hal Berghel is Associate Dean of the Howard R. Hughes College of Engineering at UNLV and Director of the new UNLV School of Informatics. He is also Director of the Identity Theft and Financial Fraud Research and Operations Center. His consultancy, Berghel.Net, provides security and management services to government and industry.



Internet browser: Other

Internet browser path: C:\Program Files\Mozilla Firefox\firefox.exe

Figure 2: Notes/Domino controls over HTML rendering.