CURRENT THREAT LEVEL

BROWSE ARCHIVES

-- Browse by Month --

LINKS

- About Frequency X
- Contact Us
- X-Force Web Site
- Vulnerability Disclosure Guidelines

Subscribe to Frequency X

« 637 million Users Vulnerable to Attack | Home

637 million Excuses

Posted by **Gunter Ollmann** on July 02, 2008 at 11:11 PM EDT.

It's been a couple of days since the release of the paper "Understanding the Web browser threat" and it's been interesting watching the public reception.

Along the way there have been a few questions raised and observations made, and I thought I'd take this time to offer my thoughts on some of them.

Patching Excuses

The most frequent point of dissent with the paper's findings appear to involve the fact that Microsoft still supports older versions of their Internet Explorer (IE) browser technology – that is, Microsoft continues to supply new vulnerability patches to IE 5.x and 6.x – so it was "unfair" to single out IE 7 as the most secure version.

As one of the authors of the paper, I can assure you that all the authors thought very hard about this point when writing the paper, and the key reasons for rejecting the older versions as being equal to the security of IE 7 were the following:

1. Microsoft has publicly stated that IE 7 is the most secure version of their Web browser technology. IE 7 is more secure than IE 5.x and IE 6.x because it also includes new security technologies and features.
2. The risks users are exposed to when surfing today's hostile Internet have changed over the years and, while Microsoft continues to patch vulnerable holes in their older browser technologies, they are not adding the new security features present in subsequent versions of the browser.
3. There is more to keeping and old Web browser technology secure than just keeping up to date with patches for publicly disclosed vulnerabilities.

Now, there are plenty of legitimate reasons for users (especially corporate users) not upgrading to the most current IE version, and many of those reasons revolve around software compatibility issues with internal Web applications and embedded browser objects. I've also seen the same problems result in organizations similarly not updating the Java runtime's. Irrespective of Microsoft's determination to support these older browser technologies – making new security patches available where necessary – let's not stray too far from the fact that these old browser technologies were not designed to protect against many of the threats we encounter today.

So, in that regard, while there may legitimate excuses for not be running the most current IE version, let's not fool anyone by pretending that users of those older Web browser technologies are just as secure as if they were running IE 7 (or any current-generation Web browser technology).

The Car Analogy

Perhaps it may be worthwhile thinking of Web browsers like cars. Version 2 browsers could be likened to a 1970's Ford Escort. At the time they were all the rage – at the cutting edge of technology (ok, perhaps not, but you get what I mean) – with all the driving features needed of that era. If you took care and maintained your Escort over the years –patching rust spots, replacing the tires and, well just about everything – you could still have an immaculate and working vehicle, just as "road safe" as it was when new.

However, things have changed. Road safety standards have changed, driving habits have changed, protection technologies have changed. Brakes that were good enough in the 1970's may now result in that very same Escort running up the rear of any modern ABS-fitted vehicle. Yes, you could upgrade the brakes with newer ones, but what about side impact bars, crumple zones, and air-bags? – all of these are newer features designed to keep the car occupants safe and secure.

Web browser security technologies have also advanced considerably over recent years, and what was good enough in older versions just doesn't cut it anymore. Patching vulnerabilities in old browser technologies is a bit like keeping the rust at bay – but it isn't likely to add the newest safety features found in the current generation of Web browser technologies.

Unpatched Threat

He's an interesting question – How responsible should the users of vulnerable and out-of-date Web browsers be for their actions? E.g. if you know your browser is woefully old and your host then becomes infected and a node of some criminal's botnet to propagate

and your host then becomes infected and a node of some criminal's botnet to propagate a crime, is it really your fault?

To be perfectly honest, that's a question best left to the lawyers, but the way cybercrime laws are developing and getting written in to legislation, I wouldn't be surprised if in a few years time it might be classed as some kind of "willful negligence" or perhaps "aiding and abetting".

Within the corporate world, what does it mean if your standard-build workstations (running IE 5.x because of compatibility issues with the corporate expense submission application) are subjected to a mass-attack, compromised and now join some former script-kiddies DDoS botnet empire to take down a popular website portal? I suspect that as long as your lawyers are better than theirs, you're probably not going to suffer financially, but I'm sure the media would have a field day.

Other Mitigation Technologies

OK, let's assume that for better or worse you are stuck with running an old and insecure Web browser technology (and you're stuck with the user permissions you've got and can't turn off scripting). What other technologies do you have at your disposal to help protect your corporate desktop users?

1. **Perimeter URL Filtering.** I'm still a fan of URL filtering technologies because of their high degree of efficiency for protecting against many of the mass-defacement, iframe injection and general badness Web sites known to be out there – and being able to do that for a large enterprise. Sure, the technology is hardly preemptive security, but it's an incredibly useful protection technology if kept up to date.
2. **Network IPS.** IPS technologies that implement protocol-based and content heuristic analysis engines are capable of identifying and stopping many of the exploit techniques used by the criminals to target vulnerable Web browsers. Good heuristics engines really can make a sizable dent in even obfuscated attack scripts (e.g. JavaScript obscured). Granted, there are many ways of making of these attacks near impossible to decode at the network layer, and use of HTTPS will certainly throw a sizable spanner in the works, but network IPS is still a critical defense against a sizable percentage of attacks against Web browsers – as well as offering protection against entire classes of deception attacks using cross-site scripting, cross-site request forgery, etc. I think the critical phrase with network IPS is "virtual patching".
3. **Patching Promptly.** Keep a careful eye on patch releases – not just for the Web browser, but also for all of the plug-in technologies accessible through it. There are a number of Managed Security Services, Threat Analysis Services and general patch notification subscription services around – most of which can help keep you informed about the availability of new patches and updates. Several of the better services provide customizable portals that will automatically notify you of new releases (and any new threats not yet covered by patches). Regardless, of where you get the information, download and install the patches as quickly as you can. You may also want to leverage IPS auto-blocking to provide "virtual patching" while patches are rolled out throughout an enterprise environment.

Sure, local host anti-virus technologies also have some value here – but, from previous experience, more as a clean-up technology rather than a protection perspective. Drive-by download malware installation is typically achieved through exploiting vulnerabilities within the Web browser (or plug-ins) and disabling anti-virus product in the shellcode payload before installing the actual malware.