DOI:10.1145/2398356.2398372

Article development led by acmqueue
queue.acm.org

A discussion with Jeremiah Grossman, Ben Livshits, Rebecca Bace, and George Neville-Neil.

Browser Security: Appearances Can Be Deceiving

IT SEEMS EVERY day we learn of some new security breach. It is all there for the taking on the Internet—more and more sensitive data every second. As for privacy, we Facebook, we Google, we bank online, we shop online, we invest online...we put it all out there. And just how well protected is all that personally identifiable information? Not very.

The browser is our most important connection to the Web, and our first line of defense. But have the browser vendors kept up their end of the bargain in protecting users? They claim to have done so in various ways, but many of those claims are thin. From SSL (Secure Sockets Layer) to the Do Not Track initiative to browser add-ons to HTML5, attempts to beef up security and privacy safeguards have fallen well short.

For example, many experts dismiss the notion that the most widely used protocol for providing security over the Internet, the SSL CA (certificate authority) model, actually provides adequate transport-layer security. But for all its faults, there is much resistance among vendors to changing the model.

HTML5 is waiting in the wings, viewed by many as the next step toward improving the Web experience, while retaining compatibility with existing browsers. It has been put forth with great promise, but so far it has not adequately addressed security shortcomings.

Vendors have attempted to achieve better browser security by supplying add-ons for protection, but users first must know where to find, and then download, install, and configure them. That is a lot to ask. It also means first being aware of the dangers—many businesses have never heard of cross-site request forgery or clickjacking and most users have no idea just how exposed their personal information really is. This is not an easy message to get across.

Likewise, users must be proactive to derive any protection from the Do Not Track initiative, a means of requesting Internet companies to stop following a user's every move. Though endorsed by the W3C and the Federal Trade Commission, it, too, falls short by putting the burden on generally uninformed users to opt in rather than making it a default setting.

For this case study on browser security ACM has assembled an experienced group to break down some of the mythical claims of security in today's browsers and argue the case for increased protection.

Jeremiah Grossman is founder and CTO at WhiteHat Security, a leading provider of Web application security services, including Sentinel, a website vulnerability management solution. A founding member of WASC (Web Application Security Consortium), he is

PHOTOGRAPH COURTESY OF JEREMIAH GROSSMAN. TREATMENT BY BRIAN GREENBERG/ANDRIJ BORYS ASSOCIATES

sought after for his expertise in Web application security. Prior to WhiteHat, he was an information security officer at Yahoo!.

Ben Livshits is a researcher at Microsoft Research and an affiliate professor at the University of Washington. He has been focusing on improving Web 2.0 application and browser reliability, performance, and security.

Security technology expert **Rebecca Gurley Bace** is president/CEO of Infidel, a network security consulting practice, and chief strategist for the Center for Forensics, Information Technology, and Security at the University of South Alabama. Her career has included a decade overseeing security investments, founding roles in several IT security communities, and advisory roles in a number of successful security ventures, both in the public and private sectors. Previously, Bace was a senior electronics engineer at National Security Agency (NSA) and served as a charter member of NSA's Information Security (Infosec) Research and Technology Group. She left NSA to become the deputy security officer for the computing, information, and communications division of the Los Alamos National Laboratory.

Facilitating the discussion is **George Neville-Neil**, a software engineer who builds high-speed, low-latency systems for customers in the financial-services sector. Previously, he was part of the Yahoo! Paranoids security team. From 2004 to 2008, Neville-Neil worked in Japan, where he developed a set of courses dubbed "The Paranoid University," teaching safe and secure programming to engineers at Yahoo!. For the past 10 years he has served on the *ACM Queue* editorial board and more recently he joined the ACM Practitioner Board.

GEORGE NEVILLE-NEIL: In talking about the current SSL CA model, Jeremiah, you have commented previously that no SSL feature ever gets turned off. What would it mean exactly to turn off something from the CA model?

JEREMIAH GROSSMAN: Many security experts, including myself, consider Convergence viable and believe it should replace the CA model as soon as possible, since what we currently have clearly isn't working. But there hasn't been general acceptance of that yet. Beyond acceptance, the bigger challenge would be to manage the migration to Convergence. Let's say we just add it alongside the CA model. At what point would we turn off the CA model? It's only by doing so, after all, that we would actually realize the security benefits of Convergence. Otherwise, as with the wait to cut over to IPv6, ev-

JEREMIAH GROSSMAN

Is there still anyone out there who seriously believes the CA model works? It's completely broken.

BEN LIVSHITS

There is a lot of fear that any real enforcement of Do Not Track might end up destroying the fundamental revenue model for the Web economy. I don't think we are likely to see enforcement in any form anytime soon.

eryone would just continue to be stuck with the same old mess as before.

GN-N: How would that actually work?

JG: CAs would be converted into notaries, and then the browser user would choose which notaries to trust. If any of those notaries were to become untrustworthy for any reason, the user could easily remove the trust indicated for that particular notary. That's very important because in the current CA model it's very difficult—if not impossible—to withdraw trust from any one CA without breaking the Web, which makes things very challenging.

One of the major criticisms [computer security researcher] Moxie Marlinspike (a pseudonym) has raised about the CA model has to do with this lack of trust agility. That is, whomever we decide to trust, we're then obliged to trust forever. Still, Moxie and the team responsible for introducing the Convergence plugin say they have taken the idea about as far as they can, and the browser vendors now need to take it the rest of the way, but the browser vendors seem pretty disinterested.

GN-N: The biggest problem with the Convergence model is that it trusts the user to do the right thing, but most users will just do whatever they're told.

JG: Maybe this is naive on my part, but I think users have a pretty good idea of whom they trust and whom they would like to trust and whom they know they're not about to trust.

Convergence also offers flexibility. I can trust five notaries today and then change to five different ones tomorrow. I would be able to do that without a whole lot of technical know-how.

Still, there are two major challenges facing Convergence. The first has to do with getting the browser vendors to implement it, and frankly, they just don't seem to have a lot of incentive for doing that. Just for the sake of argument, however, let's say they did. The next challenge would be to get people to run the notaries. That's a pretty big challenge since there is no obvious business model—which is to say there is no way for anybody to make any money. So, achieving a critical mass of notaries is going to be really difficult.

All that being said, is there still anyone out there who seriously believes the CA model works? It's completely broken.

REBECCA BACE: Even in the earliest days of the certificate model, there was a lot of criticism that it had been blindly adopted from an archaic paper-driven DoD model, without really thinking things through from a technology perspective.

JG: During a presentation on authentication, Moxie said he had located the

PHOTOGRAPH COURTESY OF MICROSOFT RESEARCH. TREATMENT BY BRIAN GREENBERG/ANDRIJ BORYS ASSOCIATES

person most directly responsible for the browser SSL CA model as we know it, and that guy told him, "Oh yeah, the CA model... we just threw that in at the end. We really had no idea."

So why do the browser vendors hold onto this obviously outmoded CA model, while making it obvious they don't want to help out with Convergence despite all the community support for that?

GN-N: It's probably because moving to Convergence would represent more work on their part. That's usually why people resist doing something.

Anyway, are the implementers going to have to worry about it, or are they just going to wait for the browser vendors to create it?

JG: As I understand the Convergence spec, the 1.8 million websites that currently have SSL enabled should not have to do anything, since the idea is for everything to work exactly as it currently does. Everything should happen over on the browser and the notary side. We should be able to carry forward the CA model through an interim period, but we would also need to have 20 or 100 notaries set up at different organizations, and the browsers would need to support that.

GN-N: So far, we have talked about protection. Let's look now at what is happening over on the attack side.

JG: I caused a bit of a furor at a conference a few years ago by talking about intranet hacking. What I meant is that you can go to a website and use it to force your browser to make basically any type of Web request of any location you want. We generally refer to that now as cross-site request forgery, but until 2006, no one had really thought about that. People knew, of course, that you could force your browser to make a request of any public website, but then Robert Hampton and I made the observation that you could force your browser to make a request of an RFC-1918 network, such as a 10.0.0.1, and then just start hacking the intranet.

We showed how you could go to a public website and force your browser to hack into your own DSL router from the inside and then move out to the Web interface and change the settings. Normally, devices on the intranet don't have very good Web security because of the understanding that you can't hack them from the outside, which is true. But at the same time, there is nothing to prevent the browser itself from being used as an attack platform by bad guys on the outside.

I've asked various browser vendors the following question: "If I'm on a public website, why do you allow that site to force my browser to make RFC-1918 requests?" They usually raise two points in response. One is that to do otherwise might mess up certain proxy configurations—I'm not sure what they mean by that. The other point is that sometimes there's actually a legitimate use case—that is, some corporate public websites have actually referenced various resources for the benefit of their employees on RFC-1918 networks. So basically, the argument is that because some big companies have adopted some really stupid practices, the rest of us have to live with compromised security on the Internet.

GN-N: Somehow I doubt they would frame it in quite that way, however.

JG: The browser vendors are just not willing to do anything that is going to disrupt the Web because of their concerns about market share. Any feature that might break some tiny portion of the Web and lose 1% of their market share is something they are just not going to consider. This is where it's useful to remember that we, the folks who use these browsers, are not really considered to be the customers. Instead, we are the product—or at least the data related to our online behavior is the product.

GN-N: On the privacy front, there seem to be some stirrings now to challenge the status quo. What are your thoughts about the Do Not Track initiative, which has been promoted by the U.S. Federal Trade Commission?

JG: Basically, that amounts to a header that browsers can pass along to websites saying, "Please do not track this user." It effectively puts websites on the honor system where tracking is concerned.

GN-N: Are you saying this is kind of like robots.txt, only in the opposite direction?

BEN LIVSHITS: Maybe something similar to that.

JG: There are no criminal sanctions to back it up, so any enforcement will have to come in the form of civil suits once the initiative really starts to get adopted. Google was the last holdout in terms of providing browser support for it, but didn't commit to any particular date.

The other challenge is that there's no clear definition of what it means to "not track" someone. Some have taken that to mean they can track you but not advertise to you.

BL: It's very easy for browser vendors to implement this as a feature. Some people will then choose to turn it on, but probably not a very high percentage if the feature isn't on by default. Even if they do decide to turn on Do Not Track and are able to figure out how to do that, they still have to make sense of what it even is. What about a site that authenticates the user? Is that site not also allowed to track the user? That would be kind of ridiculous—a contradiction in terms. What about online merchants? They have to track things just to make sure your order gets delivered, right?

What's really odd is that we have browser support for this thing that's likely to become available soon pretty much across the board, and yet there's still no consensus on what it even means.

JG: The only place where it actually makes sense to tell the user about Do Not Track is at the browser level, and the browser guys are completely disinclined to do anything of the sort. To Ben's point, if you look at all the implementations to date, you'll find that Do Not Track is turned off in every last one of them by default and buried three clicks deep where no one is ever going to find it. There is one notable and very controversial exception: Internet Explorer 10, which effectively installs with Do Not Track enabled.

BL: There's also a lot of fear that any real enforcement of Do Not Track might end up destroying the fundamental revenue model for the Web economy. I don't think we are likely to see enforcement in any form anytime soon.

JG: It's hard to imagine how they're going to be able to enforce this in any event. How would I, as a user, find out someone had been following me around in violation of Do Not Track? How would you ever discover that?

RB: My own curmudgeon's view is that this is a classic example of what happens all too often when policy-

makers decide to issue some dictum just because it seems like a good idea. Then, the technology solution providers readily agree, knowing full well that the new policy will be totally unenforceable. The policy becomes nothing more than window-dressing for the industry.

Data is money, and that goes to the core of the browser-security debate. Browser users do not fully appreciate the value of their own data, but the Facebooks and Googles of the world certainly do. Introducing measures to help users protect their data gets in the way of milking that data for all its worth. That is a strong disincentive for implementing strong browser privacy protection measures.

Adding stronger security also comes with a trade-off—more security usually means less functionality. With loss of functionality comes loss of market share, which vendors fear more than anything.

Only when users begin to see the value of their data and demand more protection for it will privacy measures get their due. If the market shifts in this direction and vendors see that adding better protection to their browsers could actually increase market share, then and only then will those measures become standard operating practice.

GN-N: We talked a little earlier about how it's the browser users, rather than the browsers themselves, that are the real products here. Anyone care to expand?

RB: Well, that is the case, and it's fundamental to this whole space. I would argue that every last conundrum in the area of browser security is rooted in the fact that we are not dealing with a classic commercial model. That is, at present users don't pay browser makers for software or, for that matter, the maintenance and upkeep of that software.

JG: The browser makers are monetizing your data, directly or indirectly, and therefore cannot see a way to protect that data without losing money. That makes for a really difficult situation.

BL: I'm not sure you can actually say it's the browser makers who are "monetizing your data." If anything, it's the sites that are monetizing your data.

JG: Actually, there is a clear interplay there. Just look at Google Chrome; it's pretty obviously monetizing your data. The Mozilla guys derive 98% of their revenue directly from Google. Then you've got Microsoft, which you could argue is also desperate now to get into the advertising business. So that raises the question: How can you work to institute healthier business incentives when those efforts are so obviously at odds with the foundation the whole business sits upon?

BL: I don't know. One of the problems with privacy is that it is difficult to put a value on it. It's difficult even to convince the users that their own privacy is actually worth all that much.

JG: Maybe users just aren't all that aware of what they're giving up with every single mouse click.

BL: Right, but there are a few companies such as Allow (http://i-allow.com) that will sign you up quite explicitly for $20 to $50 for each site you're willing to share your information with. There also are various experiments under way to establish the value of each Facebook "Like," for example. They are finding that, while some users' information is quite valuable, there are many others whose information is largely useless.

RB: I think this question rides a bigger value wave where the age dynamic comes into play. It's hard to find anybody under the age of, say, 25 who really cares about privacy. My young nieces happily tell me they have never felt like they had any privacy to begin with, so why should they start caring now?

GN-N: You have also got those people who lived through the 1960s and 1970s when the stories were rampant about people having their data exposed by the government. There are plenty of people that age who have just become inured to privacy violations. They might have cared at one point in their lives, but they're over that now.

JG: Another aspect of this is that security and privacy have become conflated. For example, if you have decided you can trust Google with your data, then the question is no longer about privacy; it's all about security. On the other hand, if you don't trust your provider, you can distinguish between security and privacy.

Once you cross that threshold and decide to trust someone with your data, you're in kind of the same situation we were talking about earlier with regard to the CA model. That is, you're essentially stuck with trusting them forever. It's not like you can take back your data from Facebook and say, "Hey, you're not allowed to have that anymore."

GN-N: Yeah, just try!

JG: You can get a copy of your data—and, according to [WikiLeaks'] Julian Assange, that can literally run to 1,000 pages. But, guess what, I don't think they are going to delete that information.

RB: Violations of our trust are already common occurrences even in the holy of holies—namely, the healthcare space, where you'd like to believe the protection of personal data would be considered sacrosanct. If people's trust isn't being honored in that domain, what hope can we hold out for more faithful protection anywhere else?

JG: That's why the fact that Do Not Track is off by default really bothers me. By the time users figure out what it is they've given up, there's no way to undo the damage or to take back any degree of control. As [computer security specialist] Bruce Schneier once pointed out, there's no delete button in the cloud, or at least there's no guarantee that, once you've pressed delete, things are actually going to be deleted.

GN-N: Is there any cause for hope?

JG: I have a pretty good strategy for protecting my own data—at least it's good enough to improve my level of comfort. I think it's an approach other people could use. The challenge is that it takes some behavioral discipline and a bit of know-how, both of which are lacking for most users. There has also been little motivation for people to work on cleaning up their acts since, for the most part, they're not even aware of the issues we've been talking about. Still, I'd say there is some reason for hope in that there are steps you can take to protect yourself.

GN-N: Do you see the browser vendors helping matters at all?

JG: No. To give you an example: since I really don't like the whole SSL model, I've put SSL VPNs (virtual private networks) on the Amazon cloud so that, no matter where I am, I can be encrypted over a hostile or untrusted network

> REBECCA BACE
> **It's hard to find anybody under the age of, say, 25 who really cares about privacy. My young nieces happily tell me they have never felt like they had any privacy to begin with, so why should they start caring now?**

while also making sure no one is able to sniff on me over the last mile. That is just one small thing you can do. It's not something my mom would be able to do, but any techie certainly could handle it.

RB: I've had a long-running debate with [risk management specialist] Dan Geer about when people might start offering the functional equivalent of gated communities on the Internet, where you would be able to buy into a managed security environment with a ready-made Internet safety barrier capable of protecting you from breaches of privacy or revelations of personal information.

JG: Geer says it's not so much about who's at fault for the current mess but instead who's going to take responsibility for it. If you say, "The user is the one who ought to take responsibility"—which is kind of where we are today—well, that just doesn't work all that well, does it?

So you might say, "OK, the ISP should take responsibility for all this bad traffic," but then you're going to have to let the ISP monitor, log, and analyze all your traffic down to a very detailed degree. You could ask the government to handle the mess, but it would need that same detailed level of access to the data and so would need to establish new powers and laws to provide for that. None of those options seems particularly attractive.

HTML5 may not be perfect, but it is inevitable and will soon be a part of all modern browsers. It adds features, particularly multimedia functions, and is meant to make the browser a richer environment. That's what Web developers want because it could lead to increased market share.

What HTML5 does not do particularly well is add security to the browser. It also leaves the door open to some Internet attacks. Many security experts thus have come to see HTML5 as an inexcusable missed opportunity. Any security work-arounds will have to be made separate from HTML5. So, yes, it's new, it's improved, but it's not going to save us.

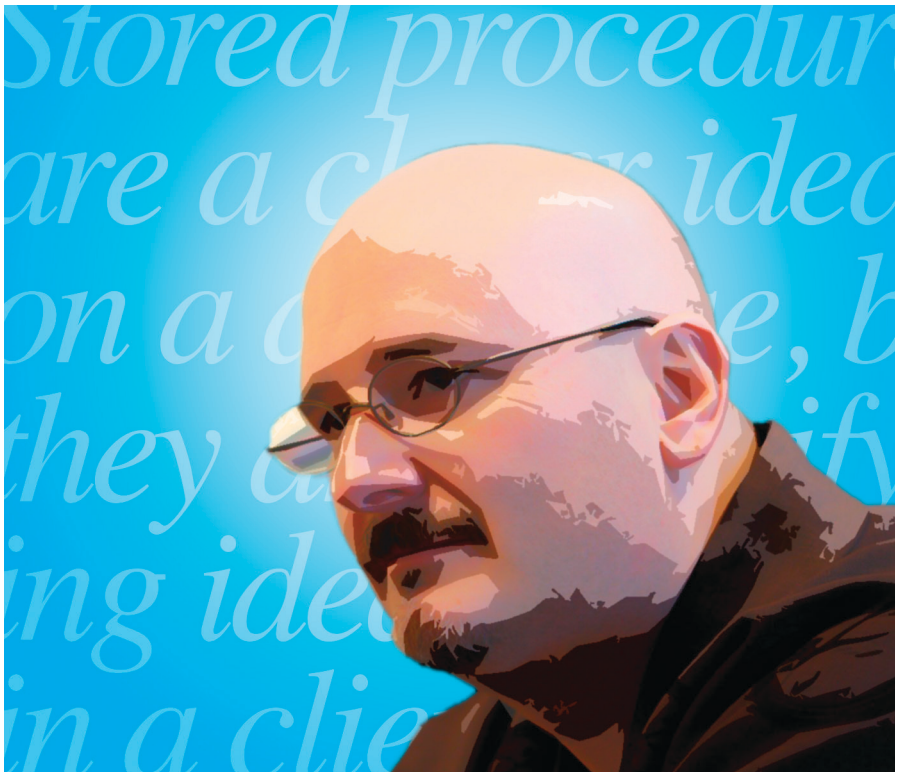
GN-N: HTML5 is now with us, and some people have probably been hoping that would bring some relief on the security front. Any comments?

JG: The whole idea of HTML5 was to bring richer media to the browser—all native through an open standard—so you wouldn't need to add plugins such as Flash, QuickTime, and all

PHOTOGRAPH COURTESY OF REBECCA BACE. TREATMENT BY BRIAN GREENBERG/ANDRIJ BORYS ASSOCIATES

GEORGE NEVILLE-NEIL

Stored procedures are a clever idea on a database, but they are a terrifying idea in a client.



kinds of other crazy stuff. That's huge because plugins have proved to be major sources of security vulnerabilities. The missed opportunity, though, is that HTML5 fails to address some long-standing Web security issues such as cross-site scripting, clickjacking, and cross-site request forgery. HTML5 developers just sort of punted on all that.

Then they added the sandbox tag as a kind of Band-Aid to be able to say they had done their bit to provide for Web security. I could go on. There are many examples of how I think HTML5 is going to make browser security much worse.

GN-N: My experience with cross-site scripting and cross-site referral forgery has been that the only real way to deal with it is to handle it on the server. This generally means drilling into the heads of the people who are using the server-side code that what they need to be doing is to make sure those exploits don't happen again.

Clickjacking is something else altogether. Right now it's probably the exploit most likely to pay off in a big way for the bad guys, whereas cross-site scripting and cross-site referral hijacking are more what you would expect from someone who is just trying to cause trouble.

Most of Facebook's security effort is expended on preventing clickjacking, and it's certainly not alone in that. In fact, I think that's really the new frontier, and I don't think HTML5 is going to address that.

JG: That could have been addressed, but as it stands, HTML5 has no security model for safely incorporating third-party data or code into your website. That model is supposed to come later with something called "cross-site security policy" or "cross-site content security policy." Even then, it will still be separate from HTML5.

As for Facebook, clickjacking is only an issue because Facebook is looking to track you around the Web. That aspect of clickjacking is going to remain unfixable since what Facebook really wants is to put Like buttons on everybody's pages. You can always clickjack something that's meant to be framed. On its own website, Facebook has already more or less fixed the clickjacking problem.

GN-N: Of course, it's not just Facebook that's looking to put some sort of button everywhere.

JG: That's right, and that's why one of the briefings at the most recent BlueHat conference described a new solution that involves putting anti-click-

PHOTOGRAPH COURTESY OF GEORGE NEVILLE-NEIL. TREATMENT BY BRIAN GREENBERG/ANDRIJ BORYS ASSOCIATES

jacking stuff in the browser. But, again, all that work is separate from HTML5.

GN-N: What else concerns you about HTML5?

JG: Are you familiar with the use of session storage as an alternative to cookies? Basically, some Web programmers are starting to put actual executable JavaScript code into local storage in addition to data. That way, when the page loads, they can just eval that code directly rather than having to make a network call, because that gets them a performance win.

Of course, the bad guys find this attractive. If they cross-site script the site that loaded that code, they'll be able to backdoor the application and thus enjoy permanent access to any client that thing happens to get loaded onto, since that backdoor code will always run.

GN-N: Stored procedures are a clever idea on a database, but they are a terrifying idea in a client.

JG: Even once you become aware of the exploit, backing out of it will be all but impossible. You certainly wouldn't be able to override it from the server. So while the HTML5 guys will say they haven't increased the attack surface, I don't think they actually know yet what all the implications are going to be.

GN-N: This would really simplify the distribution of something that looks an awful lot like a virus.

JG: It really does, but that isn't obvious yet since use of HTML5 in that way still isn't particularly widespread. Give it a few years, though, and it will be everywhere, because it really is a lot faster.

GN-N: This tells me that the browser vendors ought to include a feature that lets you flush an application's program space—perhaps not from the server, but the user ought to be able at least to flush a bad application. And now I'm suddenly picturing virus scanners that run in your browser.

JG: Oh, yeah, that's definitely going to be the case.

BL: Even then, ensuring data integrity is not going to be easy. If you have complex data structures, who's to say some of those haven't been affected in some subtle ways?

JG: I think what the browser vendors have done—knowingly or unknowingly—is to turn the browser into a new operating system.

GN-N: Well, Chrome isn't called Google Chrome OS for nothing, you know.

JG: That's right. Actually, within that sandbox there's not all that much security buffer between applications.

GN-N: We keep ripping on HTML5, but is there anything people might be able to do to provide for a better and safer user experience?

JG: Well, let's be clear: if you are using any modern browser, you are going to end up using HTML5. There's no way to turn it off in your browser since it's not a feature. It's HTML. You can't turn off HTML in the browser.

GN-N: I wasn't actually thinking in terms of turning off HTML5, although it's an interesting notion. In any event, I don't think the typical user ever turns off anything. It's up to the client and server application developers to build things in such a way that, even in the face of a wide-open browser, the user won't end up getting abused constantly.

JG: I can share how I try to protect myself and how I've instructed my mom to do it. Take two browsers—any modern browsers that have been updated will do. The important thing is to have two of them so you can compartmentalize risk. The first of these will be the primary browser, the one you use for all your promiscuous browsing—read the news, visit your favorite websites, click on the links in your Twitter feed, and whatever else you feel tempted to do. But don't ever use the primary browser to do anything with online accounts you consider sensitive or important.

If you're using Chrome or Firefox, you should also turn on ad blocking and tracker blocking as extensions in the browser. That's not just for sanity purposes, but also to prevent a whole lot of malware, which often ends up getting propagated over advertising networks. Bonus points if you run in incognito or private mode. That might save you a little bit of privacy as well. Another thing you should do is to block plugins from playing by default. You can run them whenever you want to with a right click, but don't let them automatically run. Generally, when you get infected with a virus or a piece of malware, it's because of some invisible plugin that runs automatically.

Your secondary browser is the one you want to fire up only when it's time to do online banking or online shopping or anything involving a credit card number, an account number, or anything else you want to protect. Once you have fired up that browser, get in and do what you need to do quickly, and then close that thing down.

If you can manage to keep those two worlds separate, when you are out surfing the Web with your primary browser, it won't even be possible to hack your bank with a cross-site request forgery request because it will be like you've never logged in at that bank. So clickjacking, cross-site request forgery, and cross-site scripting pose almost no threat, since there effectively is no cross site.

GN-N: What advice do you have for Web developers?

BL: I think CSP (content security policy) and the sandbox tag are among the best things for security-conscious Web developers to have come along in a long time.

JG: Also, of course, Web developers would be well advised to pay special attention to input validation, parameterized SQL statements, and output filtering. That covers about 90% of website vulnerabilities.

If you were to talk to the Facebook guys or even the Microsoft guys, you would find they usually have standard controls and libraries for printing the screen. By extension, that means removing all the nonstandard options—some of which might be unsafe—so people have no choice but to use the corporate standard version.

Then I guess the other thing is: don't ever try to roll your own crypto.

GN-N: That's solid advice. If you're not a cryptographer, don't try that at home.

Related articles on queue.acm.org

Building Secure Web Applications
George V. Neville-Neil
http://queue.acm.org/detail.cfm?id=1281889

CTO Roundtable: Malware Defense Overview
Mache Creeger
http://queue.acm.org/detail.cfm?id=1734092

Java Security Architecture Revisited
Li Gong
http://queue.acm.org/detail.cfm?id=2034639

© 2013 ACM 0001-0782/13/01