Brian Owen
SE4930-Information Security
Article Review #4
2/6/2012

Browser Extensions and Verified Security

As our software continues to grow, the desire for extending our software grows with it. As creativity and software grows, we continually wish to add on to our creations. A common trend is browser extensions and plug-ins, used daily by people browsing the internet. Software such as Adobe Readers, Twitter extensions, weather applications and password storage utilities are used by many. The issue is how is security controlled as we add onto our browsers in unexpected ways (to that of the browser developer) and how can we fix the security issues these browsers prevent while continue to allow growth of our software.

The article I read proposes a new framework for browser extensions, called IBEX, which could author, analyze, verifying and deploy secure browser extensions. Before discussing IBEX, however, it is important to see some of the exploits and current problems with browser extensions as they currently stand.

First, the article discusses the current big three browsers, Internet Explorer Mozilla Firefox and Google Chrome and their respectively own independent model for extension. Of the three, Internet Explorer's is the weakest, as it has several extension mechanisms with browser helper objects (BHO) being the most common. These BHOs have "virtually unrestricted access" to the browsers events, making it very easy to write password capturers and key loggers. Because of the close relationship with the browser, even benign extensions can cause exploits when they are implemented poorly (unintentionally opening up exploits with bad extension code). Firefox provides a bit more security in the form a community review. Only extensions that are considered safe are added to the 'curated extension gallery'; and since Firefox will not install extensions that do not originate from this gallery, some peer review has been done prior to user exposure. Some malicious extensions have found their way on to the extension gallery, however. This system is very similar to that of the Android Marketplace. The harm, however, is if a bad extension does get through it "can modify Firefox in fairly unrestricted ways". Google Chrome extensions are on separate 'extension' pages, but do have access to APIs not available to web pages. There separation runs different from the browser process, but has the ability to see and change the browser's UI. They can also see special events (window closing, etc.). Chrome runs into issues with over-privileging extensions, allowing them to see browser history and data on those websites. This over-privileging can cause Chrome's many protection mechanisms useless.

IBEX provides a 'finger-grained access control model for browser extension', by formally define a security property for extensions and develop a method that will allow security to be enforced. IBEX hopes to be used for verification across all browsers on HTML5 based extensions for security verification. The article contains all of the details and gives several examples of its use in practice, but I will not get into the details of those, as the point of the article has been made. The security of extending our browsers and internet access are not currently sufficient.

Brian Owen
SE4930-Information Security
Article Review #4
2/6/2012

What this article has taught me is in part a reinforcement of past knowledge, and a building on that said knowledge. One of the core reasons security is so difficult is because of unpredictable growth of software. It is always hard to guess as a developer just how your software is going to be used in the future. It is even more difficult to guess how people will try to extend your software and make it better and more personalized. Because not allowing extensions at all would cause your software to hardly be used, it is a good reflection to see how others (Microsoft, Google, Mozilla) have implemented their design and the short comings of those designs. With that in regard, I can learn from their mistakes and potentially not make the same ones in the future. In addition, it seems like standardization of extensions in this article is a particular good idea, when thought about in a box as the articles does. The only thing that concerns me is just how many standards currently exist and how many people don't follow as it is. This of course is just my opinion, but it seems IPEX would be great *if* everyone chose to use it; just like any other standard in existence.

My final thoughts on this article are how this will help me grow as a developer. I think the biggest information I gained was the different sort of things you need to be aware of when considering extensions to your current software. It may be easy to make a development decision based on how useful a resource could be to an extension, but it could be even easier for a malicious or poorly written extension give access to that same resource without the users consent. That is why I think when you give someone access to a resource, or let it cross a trust boundary outside of your scope as a developer, it must be assumed it will be access by all. Never assume a developer of an extension will maintain the integrity of your code, as more often than not it won't.