“Generally, JavaScript codes are obfuscated and dense by software designers who are apprehensive about guarding against intellectual property or even sustaining data bandwidth.”

Hi Albert,

Having done a lot of Perl programming at the workplace, I’ve seen a lot of obfuscated code. I think though, that most of it was written without that intent, at least where I work. Purposefully obfuscating one’s code for the purpose of trying to keep it proprietary is an interesting concept, and one I haven’t heard about until you brought it up. However, if one studied the code, can they merely just reverse-engineer it, obfuscate it in another way, and claim it as their own?

Regards,

Emanuel

Hi Rony,

Thanks for your comments. I think that the pictures you provided aid in the description of these concepts. I’m finding myself applying the same habits from “regular” programming to web programming. Start with the bare minimum and simplest unit first, and make sure that it compiles and does what it was intended to do. Then proceed by adding another unit or functionality, and recompile. I find this preferable to debugging large units of code. Since I’m still kind of getting the hang of web programming, I’m applying a similar style, and using the W3C validator in place of the compiler. I think as I get better at this, I can probably write out bigger portions of code confident that there are no bugs before submitting to the validator.

Regards,

Emanuel

Hi Albert,

Thank you for the additional information on code obfuscation. So it looks like code obfuscation methods such as the one presented by Schrittwieser and Katzenbeisser can help thwart efforts to reverse engineer using automated deobfuscators. Though, I supposed that if one were to be totally committed to the goal, then most likely it will eventually be deciphered. However, if I were really serious about protecting my code, I would probably just put it on the server side. Still, from a risk mitigation perspective, code obfuscation does definitely have a role.

Regards,

Emanuel

“Flash is on the decline even for video sharing sites and full flash sites are a dime and a dozen now because once you implement it your viewers will need the plugin and a lot of persons won’t know how to install it, are prevented from installing it (On some local networks) and Apple devices don’t support it (C, 2013).”

Hi Adrian,

Thank you for bringing this topic up. I’ve always heard that Apple used the popularity of their mobile devices to kill Adobe Flash. However, I never knew the real story, so I decided to research it a bit. In 2010, Apple published an essay written by Steve Jobs titled “Thoughts on Flash” (Jobs, 2010). In it Steve Jobs cited the following reasons:

* Adobe Flash, being entirely owned and controlled by Adobe, is a closed system, and therefore should not be a web “standard”. Web standards should be open (E.g., HTML5, CSS, JavaScript).
* Most existing Flash video on the web is available in other formats.
* According to Symantec, Flash is one of the least secure software platforms.
* Flash does not perform well on mobile devices. Playback of Flash video requires decoding in software, which requires more battery power. The preferred way to decode in hardware.
* Because web pages using Flash rely on mice hovering over specific spots, not suitable for touchscreens.

Regards,

Emanuel

Reference:

Jobs, S. (2010) *Thoughts on Flash* [Online]. Available from: <http://www.apple.com/hotnews/thoughts-on-flash/> (Accessed: 10 June 2014)

Hi Tung,

Not only in JavaScript, but other languages as well. Where I work, Perl seems to dominate, followed by Java. Perl’s motto is “There’s more than one way to do it” (Wikipedia, 2009), which I think greatly contributes to a lot of Perl programs being naturally obfuscated. Before Albert’s post, I always assumed from experience that all the obfuscated code was like that unintentionally or the result of programmers just trying to show off their ability to write clever code. There were actually obfuscated Perl contests, which were judged according the following:

* Achieve the most with the least code possible.
* Creativity

(Gallo, 1998)

They listed their objective was “to determine who can write the most devious, inhuman, disgusting, amusing, amazing, and bizarre Perl code.” (Gallo, 1998)

Here’s one of their winners:

package S2z8N3;{

$zyp=S2z8N3;use Socket;

(S2z8N3+w1HC$zyp)&

open SZzBN3,"<$0"

;while(<SZzBN3>){/\s\((.\*p\))&/

&&(@S2zBN3=unpack$age,$1)}foreach

$zyp(@S2zBN3)

while($S2z8M3++!=$zyp-

30){$\_=<SZz8N3>}/^(.)/|print $1

;$S2z8M3=0}s/.\*//|print}sub w1HC{$age=c17

;socket(SZz8N3,PF\_INET,SOCK\_STREAM,getprotobyname('tcp'))&&

connect(SZz8N3,sockaddr\_in(023,"\022\x17\x\cv"))

;S2zBN3|pack$age}

These contests are no longer around, but there is a yearly obfuscated C contest (IOCCC, 2014). I was doing a bit of C before I went on to Perl. Knowing how structured most C programmers are, I can’t imagine the C code being as bizarre as the Perl code in these contests. I think it would be interesting if there ever were a Java contest.

References:

Gallo, F. (1998) ‘The 3rd Annual Obfuscated Perl Contest’, *The Perl Journal* [Online]. Available from: <http://www.foo.be/docs/tpj/issues/vol3_2/tpj0302-0012.html> (Accessed: 11 June 2014)

IOCCC (2014) The International Obfuscated C Code Contest [Online]. Available from: <http://www.ioccc.org> (Accessed: 11 June 2014).

Wikipedia (2009) *There’s More That One Way To Do It* [Online]. Available from: <http://en.wikipedia.org/wiki/There%27s_more_than_one_way_to_do_it> (Accessed: 11 June 2014)

Hi Frank, Jeremy, and Albert,

This is an interesting topic. Please allow me to join in. I too have noticed that some websites have different versions for desktop, mobile, and tablet. I think a good example is ESPN. Google recommends a technique they refer to as “responsive web design” (Google Developers, 2014). The same HTML code is sent to all devices when websites are designed using this method. The way in which a page is rendered is then determined by CSS “media queries” (Google Developers, 2014). Google sites the following advantages of this method:

* Only one URL is required to access content, and index the web page.
* Reduced loading time because no redirection is required.

Regards,

Emanuel

References:

Google Developers (2014) Building Smartphone-Optimized Websites [Online]. Available from: <https://developers.google.com/webmasters/smartphone-sites/details.html#rwd> (Accessed: 11 June 2014)

“Javascript in my eyes is not such a very good language because of a lot of its pitfalls especially how it makes some stuff so tricky and short hands code make code hard to read and figure out.”

Hi Adrian,

I couldn’t agree more with this. Even though I’m just a beginner at JavaScript, it reminds me a lot of Perl. Its advantages are that compared of more structured languages such as Java/C/C++, it has a relatively low learning curve and programs can be written fast. However, as I’ve learned in previous modules, as well as my own personal experience, maintenance of software takes up most of its life cycle. One study estimated maintenance at 75% of totals costs (Galorath, 2008). With that being said, it is in the best interest of the organization for the programmers to write neat code, which is easy to understand. However, just because a language allows one to be sloppy is probably not a good excuse for not adhering to some sort of coding standards. I remember that my coding style had slipped in regards to understandability from writing so much Perl code. Only did it improve when I started programming in Java.

Regards,

Emanuel

References:

Galorath, D. (2008) ‘Software Total Ownership Costs: Development Is Only Job One’, *Software Tech.News*, vol. 11, no. 3.